# FreeBSD Handbuch

## *Zusammenfassung*

Willkommen bei FreeBSD! Dieses Handbuch beschreibt die Installation und den täglichen Umgang mit *FreeBSD 12.1-RELEASE*, *FreeBSD 11.3-RELEASE*. Das Handbuch ist das Ergebnis einer fortlaufenden Arbeit vieler Einzelpersonen. Dies kann dazu führen, dass einige Abschnitte nicht aktuell sind. Bei Unklarheiten empfiehlt es sich daher stets, die englische Originalversion des Handbuchs zu lesen.

Wenn Sie bei der Übersetzung des Handbuchs mithelfen möchten, senden Sie bitte eine E-Mail an die Mailingliste FreeBSD German Documentation Project <de-bsd-translators@de.FreeBSD.org>.

Die aktuelle Version des Handbuchs ist immer auf dem FreeBSD-Webserver verfügbar und kann in verschiedenen Formaten und in komprimierter Form vom FreeBSD FTP-Server oder einem der vielen Spiegel herunter geladen werden (ältere Versionen finden Sie hingegen unter https://docs.FreeBSD.org/doc/). Gedruckte Kopien können bei FreeBSD Mall erworben werden. Vielleicht möchten Sie das Handbuch oder andere Dokumente auch durchsuchen.

# Inhaltsverzeichnis


Vorwort . . . 12
Über dieses Buch . . . 12
Änderungen gegenüber der dritten Auflage . . . 12
Änderungen gegenüber der zweiten Auflage (2004) . . . 12
Änderungen gegenüber der ersten Auflage (2001) . . . 13
Gliederung . . . 14
Konventionen in diesem Buch . . . 17
Danksagung . . . 18
I: Einleitung . . . 19
1. Überblick . . . 20
2. Willkommen zu FreeBSD! . . . 21
2.1. Was kann FreeBSD? . . . 22
2.2. Wer verwendet FreeBSD? . . . 23
3. Über das FreeBSD Projekt . . . 26
3.1. Kurzer geschichtlicher Abriss zu FreeBSD . . . 26
3.2. Ziele des FreeBSD-Projekts . . . 27
3.3. Das FreeBSD-Entwicklungsmodell . . . 27
3.4. Programme von Drittherstellern . . . 29
3.5. Zusätzliche Dokumentation . . . 29
4. FreeBSD installieren . . . 31
4.1. Übersicht . . . 31
4.2. Minimale Hardwareanforderungen . . . 32
4.3. Vor der Installation . . . 33
4.4. Die Installation starten . . . 37
4.5. Verwendung von bsdinstall . . . 40
4.6. Plattenplatz bereitstellen . . . 45
4.7. Abrufen der Distributionen . . . 61
4.8. Benutzerkonten, Zeitzone, Dienste und Sicherheitsoptionen . . . 63
4.9. Netzwerkschnittstellen . . . 76
4.10. Fehlerbehebung . . . 87
4.11. Verwendung der Live-CD . . . 88
5. Grundlagen des FreeBSD Betriebssystems . . . 89
5.1. Übersicht . . . 89
5.2. Virtuelle Konsolen und Terminals . . . 89
5.3. Benutzer und grundlegende Account-Verwaltung . . . 92
5.4. Zugriffsrechte . . . 101
5.5. Verzeichnis-Strukturen . . . 106
5.6. Festplatten, Slices und Partitionen . . . 108

5.7. Anhängen und Abhängen von Dateisystemen . . . 114
5.8. Prozesse und Dämonen . . . 117
5.9. Shells . . . 120
5.10. Text-Editoren . . . 124
5.11. Geräte und Gerätedateien . . . 124
5.12. Manualpages . . . 124
6. Installieren von Anwendungen: Pakete und Ports . . . 127
6.1. Übersicht . . . 127
6.2. Installation von Software . . . 127
6.3. Suchen einer Anwendung . . . 129
6.4. Benutzen von pkg zur Verwaltung von Binärpaketen . . . 131
6.5. Benutzen der Ports-Sammlung . . . 138
6.6. Pakete mit Poudriere bauen . . . 148
6.7. Nach der Installation . . . 151
6.8. Kaputte Ports . . . 151
7. Das X-Window-System . . . 153
7.1. Übersicht . . . 153
7.2. Terminologie . . . 153
7.3. Xorg installieren . . . 155
7.4. Xorg konfigurieren . . . 155
7.5. Schriftarten in Xorg benutzen . . . 165
7.6. Der X-Display-Manager . . . 169
7.7. Grafische Oberflächen . . . 171
7.8. Compiz Fusion installieren . . . 175
7.9. Fehlersuche . . . 178
II: Desktop-Anwendungen . . . 184
8. Übersicht . . . 185
9. Browser . . . 186
9.1. Firefox . . . 186
9.2. Konqueror . . . 187
9.3. Chromium . . . 187
10. Büroanwendungen . . . 188
10.1. Calligra . . . 188
10.2. AbiWord . . . 188
10.3. The GIMP . . . 189
10.4. Apache OpenOffice . . . 189
10.5. LibreOffice . . . 190
11. Anzeigen von Dokumenten . . . 192
11.1. Xpdf . . . 192
11.2. gv . . . 192
11.3. Geeqie . . . 193

11.4. ePDFView . . . 193
11.5. Okular . . . 194
12. Finanzsoftware . . . 195
12.1. GnuCash . . . 195
12.2. Gnumeric . . . 195
12.3. KMyMoney . . . 196
13. Multimedia . . . 197
13.1. Übersicht . . . 197
13.2. Soundkarten einrichten . . . 197
13.3. MP3-Audio . . . 203
13.4. Videos wiedergeben . . . 205
13.5. TV-Karten . . . 211
13.6. MythTV . . . 213
13.7. Scanner . . . 214
14. Konfiguration des FreeBSD-Kernels . . . 219
14.1. Übersicht . . . 219
14.2. Wieso einen eigenen Kernel bauen? . . . 219
14.3. Informationen über die vorhandene Hardware beschaffen . . . 220
14.4. Die Kernelkonfigurationsdatei . . . 221
14.5. Einen angepassten Kernel bauen und installieren . . . 223
14.6. Wenn etwas schiefgeht . . . 224
15. Drucken . . . 226
15.1. Schnellstart . . . 226
15.2. Druckerverbindungen . . . 227
15.3. Gebräuchliche Seitenbeschreibungssprachen . . . 228
15.4. Direktes Drucken . . . 230
15.5. LPD (Line Printer Daemon) . . . 230
15.6. Andere Drucksysteme . . . 239
16. Linux®-Binärkompatibilität . . . 241
16.1. Übersicht . . . 241
16.2. Konfiguration der Linux®-Binärkompatibilität . . . 241
16.3. Weiterführende Themen . . . 244
III: Konfiguration und Tuning . . . 247
17. Übersicht . . . 248
18. Start von Diensten . . . 249
18.1. Dienste über das rc.d-System starten . . . 249
18.2. Andere Arten, um Dienste zu starten . . . 250
19. cron(8) konfigurieren . . . 251
19.1. Eine Benutzer-crontab erstellen . . . 252
20. Dienste unter FreeBSD verwalten . . . 254
20.1. Systemspezifische Konfiguration . . . 255

21. Einrichten von Netzwerkkarten . . . 257
21.1. Bestimmen des richtigen Treibers . . . 257
21.2. Konfiguration von Netzwerkkarten . . . 259
21.3. Test und Fehlersuche . . . 261
22. Virtual Hosts . . . 264
23. Konfiguration der Systemprotokollierung . . . 265
23.1. Konfiguration der lokalen Protokollierung . . . 265
23.2. Management und Rotation von Logdateien . . . 267
23.3. Protokollierung von anderen Hosts . . . 268
24. Konfigurationsdateien . . . 273
24.1. /etc Layout . . . 273
24.2. Hostnamen . . . 273
25. Einstellungen mit sysctl(8) . . . 276
25.1. sysctl.conf . . . 276
25.2. Schreibgeschützte Variablen . . . 277
26. Tuning von Laufwerken . . . 278
26.1. Sysctl Variablen . . . 278
26.2. Soft Updates . . . 279
27. Einstellungen von Kernel Limits . . . 283
27.1. Datei und Prozeß Limits . . . 283
27.2. Netzwerk Limits . . . 284
27.3. Virtueller Speicher (Virtual Memory) . . . 286
28. Hinzufügen von Swap-Bereichen . . . 287
28.1. Swap auf einer neuen Festplatte oder einer existierenden Partition . . . 287
28.2. Swap-Dateien erstellen . . . 287
29. Energie- und Ressourcenverwaltung . . . 289
29.1. Konfiguration des ACPI . . . 289
29.2. Häufige Probleme . . . 290
29.3. Die voreingestellte ASL überschreiben . . . 293
29.4. Abrufen und Einreichen von Informationen zur Fehlersuche . . . 294
29.5. Referenzen . . . 295
30. FreeBSDs Bootvorgang . . . 297
30.1. Übersicht . . . 297
30.2. FreeBSDs Bootvorgang . . . 297
30.3. Willkommensbildschirme während des Bootvorgangs konfigurieren . . . 303
30.4. Konfiguration von Geräten . . . 305
30.5. Der Shutdown-Vorgang . . . 306
31. Sicherheit . . . 307
31.1. Übersicht . . . 307
31.2. Einführung . . . 307
31.3. Einmalpasswörter . . . 316

31.4. TCP Wrapper . . . 320
31.5. Kerberos . . . 322
31.6. OpenSSL . . . 330
31.7. VPN mit IPsec . . . 333
31.8. OpenSSH . . . 339
31.9. Zugriffskontrolllisten für Dateisysteme (ACL) . . . 346
31.10. Sicherheitsprobleme in Software von Drittanbietern überwachen . . . 348
31.11. FreeBSD Sicherheitshinweise . . . 349
31.12. Prozess-Überwachung . . . 353
31.13. Einschränkung von Ressourcen . . . 354
31.14. Gemeinsame Administration mit Sudo . . . 358
32. Jails . . . 362
32.1. Übersicht . . . 362
32.2. Jails - Definitionen . . . 363
32.3. Einrichtung und Verwaltung von Jails . . . 364
32.4. Feinabstimmung und Administration . . . 367
32.5. Mehrere Jails aktualisieren . . . 369
32.6. Verwaltung von Jails mit ezjail . . . 374
33. Verbindliche Zugriffskontrolle . . . 385
33.1. Übersicht . . . 385
33.2. Schlüsselbegriffe . . . 386
33.3. Erläuterung . . . 388
33.4. MAC Labels verstehen . . . 389
33.5. Planung eines Sicherheitsmodells . . . 394
33.6. Modulkonfiguration . . . 395
33.7. Das MAC Modul seeotheruids . . . 395
33.8. Das MAC Modul bsdextended . . . 396
33.9. Das MAC Modul ifoff . . . 397
33.10. Das MAC Modul portacl . . . 398
33.11. Das MAC Modul partition . . . 399
33.12. Das MAC Modul Multi-Level Security . . . 401
33.13. Das MAC Modul Biba . . . 403
33.14. Das MAC Modul LOMAC . . . 404
33.15. Beispiel 1: Nagios in einer MAC Jail . . . 405
33.16. Beispiel 2: User Lock Down . . . 409
33.17. Fehler im MAC beheben . . . 410
34. Security Event Auditing . . . 412
34.1. Einleitung . . . 412
34.2. Schlüsselbegriffe . . . 413
34.3. Audit Konfiguration . . . 413
34.4. Audit-Trails . . . 417

35. Speichermedien . . . 421
35.1. Übersicht . . . 421
35.2. Hinzufügen von Laufwerken . . . 421
35.3. Partitionen vergrößern . . . 422
35.4. USB Speichermedien . . . 425
35.5. Erstellen und Verwenden von CDs . . . 429
35.6. DVDs benutzen . . . 435
35.7. Disketten benutzen . . . 440
35.8. Datensicherung . . . 441
35.9. Speicherbasierte Laufwerke . . . 446
35.10. Schnappschüsse von Dateisystemen . . . 448
35.11. Disk Quotas . . . 449
35.12. Partitionen verschlüsseln . . . 453
35.13. Den Auslagerungsspeicher verschlüsseln . . . 459
35.14. Highly Available Storage (HAST) . . . 461
36. GEOM: Modulares Framework zur Plattentransformation . . . 470
36.1. Übersicht . . . 470
36.2. RAID0 - Striping . . . 470
36.3. RAID1 - Spiegelung . . . 472
36.4. RAID3 - Byte-Level Striping mit dedizierter Parität . . . 482
36.5. Software RAID . . . 484
36.6. GEOM Gate Netzwerk . . . 488
36.7. Das Labeln von Laufwerken . . . 489
36.8. UFS Journaling in GEOM . . . 492
37. Das Z-Dateisystem (ZFS) . . . 495
37.1. Was ZFS anders macht . . . 495
37.2. Schnellstartanleitung . . . 496
37.3. `zpool` Administration . . . 503
37.4. `zfs` Administration . . . 522
37.5. Delegierbare Administration . . . 545
37.6. Themen für Fortgeschrittene . . . 546
37.7. Zusätzliche Informationen . . . 549
37.8. ZFS-Eigenschaften und Terminologie . . . 550
38. Dateisystemunterstützung . . . 560
38.1. Übersicht . . . 560
38.2. Linux® Dateisysteme . . . 560
39. Virtualisierung . . . 562
39.1. Übersicht . . . 562
39.2. FreeBSD als Gast-Betriebssystem unter Parallels für Mac OS® X . . . 562
39.3. FreeBSD als Gast-Betriebssystem unter Virtual PC für Windows® . . . 572
39.4. FreeBSD als Gast-Betriebssystem unter VMware Fusion für Mac OS® . . . 581

39.5. FreeBSD als Gast mit VirtualBox™ . . . 592
39.6. FreeBSD als Host mit Virtualbox . . . 594
39.7. FreeBSD als Host mit bhyve . . . 597
39.8. FreeBSD als Xen™-Host . . . 603
40. Localization - i18n/L10n Usage and Setup . . . 610
40.1. Übersicht . . . 610
40.2. Lokale Anpassungen benutzen . . . 610
40.3. I18N-Programme . . . 617
40.4. Lokalisierung für einzelne Sprachen . . . 617
41. FreeBSD aktualisieren . . . 620
41.1. Übersicht . . . 620
41.2. FreeBSD-Update . . . 620
41.3. Aktualisieren der Dokumentationssammlung . . . 629
41.4. Einem Entwicklungszweig folgen . . . 632
41.5. FreeBSD aus den Quellen aktualisieren . . . 634
41.6. Installation mehrerer Maschinen . . . 640
42. DTrace . . . 642
42.1. Überblick . . . 642
42.2. Unterschiede in der Implementierung . . . 642
42.3. Die DTrace Unterstützung aktivieren . . . 643
42.4. DTrace verwenden . . . 644
43. USB Gerätemodus . . . 648
43.1. Übersicht . . . 648
43.2. Virtuelle serielle USB-Ports . . . 648
43.3. Netzwerkkarten im USB-Gerätemodus . . . 650
43.4. Virtuelle USB-Speichergeräte . . . 651
IV: Serielle Datenübertragung . . . 654
44. Übersicht . . . 655
45. Begriffe und Hardware . . . 656
45.1. Kabel und Schnittstellen . . . 656
45.2. Kernelkonfiguration . . . 658
45.3. Gerätedateien . . . 659
45.4. Konfiguration der seriellen Schnittstelle . . . 659
46. Terminals . . . 661
46.1. Konfiguration . . . 662
46.2. Fehlersuche . . . 664
47. Einwählverbindungen . . . 666
47.1. Schnittstellenbausteine . . . 666
47.2. Überblick . . . 667
47.3. Konfigurationsdateien . . . 667
47.4. Modemkonfiguration . . . 671

47.5. Fehlersuche . . . 672
48. Verbindungen nach Außen . . . 674
48.1. Ein Hayes Modem benutzen . . . 674
48.2. AT-Befehle benutzen . . . 674
48.3. Das @ Zeichen funktioniert nicht . . . 674
48.4. Wie kann ich von der Kommandozeile eine Telefonnummer wählen? . . . 675
48.5. Die bps-Rate angeben . . . 675
48.6. Über einen Terminal-Server auf verschiedene Rechner zugreifen . . . 675
48.7. Mehr als eine Verbindung mit tip benutzen . . . 676
48.8. Eine Übertragung erzwingen . . . 676
48.9. Großbuchstaben . . . 677
48.10. Dateien mit tip übertragen . . . 677
48.11. zmodem mit tip benutzen . . . 677
49. Einrichten der seriellen Konsole . . . 678
49.1. Schnelle Konfiguration der seriellen Konsole . . . 678
49.2. Konfiguration der seriellen Konsole . . . 678
49.3. Zusammenfassung . . . 682
49.4. Hinweise zur seriellen Konsole . . . 682
49.5. Die Konsole im Bootloader ändern . . . 685
49.6. Vorbehalte . . . 685
50. PPP . . . 687
50.1. Übersicht . . . 687
50.2. PPP konfigurieren . . . 687
50.3. Probleme bei PPP-Verbindungen . . . 696
50.4. PPP over Ethernet (PPPoE) . . . 699
50.5. PPP over ATM (PPPoA) . . . 701
51. Elektronische Post (E-Mail) . . . 705
51.1. Terminologie . . . 705
51.2. Übersicht . . . 705
51.3. E-Mail Komponenten . . . 706
51.4. Sendmail-Konfigurationsdateien . . . 707
51.5. Wechseln des Mailübertragungs-Agenten . . . 710
51.6. Fehlerbehebung . . . 712
51.7. Weiterführende Themen . . . 715
51.8. Ausgehende E-Mail über einen Relay versenden . . . 717
51.9. E-Mail über Einwahl-Verbindungen . . . 718
51.10. SMTP-Authentifizierung . . . 719
51.11. E-Mail-Programme . . . 721
51.12. E-Mails mit fetchmail abholen . . . 728
51.13. E-Mails mit procmail filtern . . . 729
52. Netzwerkserver . . . 731

52.1. Übersicht . . . 731
52.2. Der inetd"Super-Server" . . . 731
52.3. Network File System (NFS) . . . 735
52.4. Network Information System (NIS) . . . 740
52.5. Lightweight Access Directory Protocol (LDAP) . . . 755
52.6. Dynamic Host Configuration Protocol (DHCP) . . . 763
52.7. Domain Name System (DNS) . . . 767
52.8. Apache HTTP-Server . . . 770
52.9. File Transfer Protocol (FTP) . . . 776
52.10. Datei- und Druckserver für Microsoft® Windows®-Clients (Samba) . . . 777
52.11. Die Uhrzeit mit NTP synchronisieren . . . 780
52.12. iSCSI Initiator und Target Konfiguration . . . 784
53. Firewalls . . . 789
53.1. Einführung . . . 789
53.2. Firewallkonzepte . . . 790
53.3. PF . . . 791
53.4. IPFW . . . 809
53.5. IPFILTER (IPF) . . . 825
53.6. Blacklistd . . . 838
54. Weiterführende Netzwerkthemen . . . 843
54.1. Übersicht . . . 843
54.2. Gateways und Routen . . . 843
54.3. Drahtlose Netzwerke . . . 849
54.4. USB Tethering . . . 871
54.5. Bluetooth . . . 872
54.6. LAN-Kopplung mit einer Bridge . . . 881
54.7. Link-Aggregation und Failover . . . 888
54.8. Plattenloser Betrieb mit PXE . . . 894
54.9. IPv6 . . . 899
54.10. Common Address Redundancy Protocol (CARP) . . . 903
54.11. VLANs . . . 907
Anhang A: Bezugsquellen für FreeBSD . . . 909
55. CD and DVD Sets . . . 910
56. FTP-Server . . . 911
57. Benutzen von Subversion . . . 918
57.1. Einführung . . . 918
57.2. SSL Root-Zertifikate . . . 918
57.3. Svnlite . . . 918
57.4. Installation . . . 918
57.5. Subversion benutzen . . . 919
57.6. Subversion Mirror Sites . . . 920

57.7. Weiterführende Informationen . . . 920
58. Benutzen von rsync . . . 921
Anhang B: Bibliografie . . . 923
B.1. Bücher speziell für FreeBSD . . . 923
B.2. Handbücher . . . 924
B.3. Administrations-Anleitungen . . . 924
B.4. Programmierhandbücher . . . 924
B.5. Betriebssystem-Interna . . . 925
B.6. Sicherheits-Anleitung . . . 926
B.7. Hardware-Anleitung . . . 926
B.8. UNIX® Geschichte . . . 926
B.9. Zeitschriften, Magazine und Journale . . . 927
Anhang C: Ressourcen im Internet . . . 928
C.1. Webseiten . . . 928
C.2. Mailinglisten . . . 928
C.3. Usenet-News . . . 948
C.4. Offizielle Spiegel . . . 949
Anhang D: OpenPGP-Schlüssel . . . 952
D.1. Ansprechpartner . . . 952

# Vorwort

## Über dieses Buch

Der erste Teil dieses Buchs führt FreeBSD-Einsteiger durch den Installationsprozess und stellt leicht verständlich Konzepte und Konventionen vor, die UNIX® zu Grunde liegen. Sie müssen nur neugierig sein und bereitwillig neue Konzepte aufnehmen, wenn diese vorgestellt werden, um diesen Teil durchzuarbeiten.

Wenn Sie den ersten Teil bewältigt haben, bietet der umfangreichere zweite Teil eine verständliche Darstellung vieler Themen, die für FreeBSD-Administratoren relevant sind. Wenn Kapitel auf anderen Kapiteln aufbauen, wird das in der Übersicht am Anfang eines Kapitels erläutert.

Weitere Informationsquellen entnehmen Sie bitte Bibliografie.

## Änderungen gegenüber der dritten Auflage

Die aktuelle Auflage des Handbuchs ist das Ergebnis der engagierten Arbeit Hunderter Mitarbeiter des FreeBSD Documentation Projects in den vergangenen 10 Jahren. Die wichtigsten Änderungen dieser Auflage gegenüber der dritten Auflage von 2004 sind:

- DTrace informiert Sie über die mächtigen Funktionen zur Leistungsmessung, die dieses Werkzeug bietet.
- Dateisystemunterstützung enthält Informationen über die Unterstützung nicht-nativer Dateisysteme in FreeBSD, wie beispielsweise ZFS von Sun™.
- Security Event Auditing informiert über die neuen Auditing-Fähigkeitenvon FreeBSD.
- Virtualisierung enthält Informationen zur Installation von FreeBSD in verschiedenen Virtualisierungs-Programmen.
- FreeBSD installieren wurde hinzugefügt, um die Installation von FreeBSD mit dem neuen Installationswerkzeug, bsdinstall, zu dokumentieren.

## Änderungen gegenüber der zweiten Auflage (2004)

Die dritte Auflage des Handbuchs war das Ergebnis der über zwei Jahre dauernden engagierten Arbeit des FreeBSD Documentation Projects. Die gedruckte Ausgabe war derart umfangreich, dass es notwendig wurde, sie in zwei Bände aufzuteilen. Die wichtigsten Änderungen dieser Auflage waren:

- Konfiguration und Tuning enthält neue Abschnitte über ACPI, Energie- und Ressourcenverwaltung und das Werkzeug `cron`.
- Sicherheit erläutert nun Virtual Private Networks (VPNs), Zugriffskontrolllisten (ACLs) und Sicherheitshinweise.
- Verbindliche Zugriffskontrolle ist ein neues Kapitel, das vorgeschriebene Zugriffskontrollen vorstellt und erklärt, wie FreeBSD-Systeme mit MACs abgesichert werden können.

- Speichermedien enthält neue Informationen über USB-Speichergeräte, Dateisystem-Snapshots, Quotas, Datei- und Netzwerk-basierte Dateisysteme sowie verschlüsselte Partitionen.
- Zum PPP wurde ein Abschnitt über Fehlersuche hinzugefügt.
- Elektronische Post (E-Mail) wurde um Abschnitte über alternative Transport-Agenten (MTAs), SMTP-Authentifizierung, UUCP, fetchmail, procmail und weitere Themen erweitert.
- Netzwerkserver ist ein weiteres neues Kapitel dieser Auflage. Das Kapitel beschreibt, wie der Apache HTTP-Server, ftpd und ein Samba-Server für Microsoft® Windows®-Clients eingerichtet werden. Einige Abschnitte aus dem Weiterführende Netzwerkthemen befinden sich nun, wegen des thematischen Zusammenhangs, in diesem Kapitel.
- Das Weiterführende Netzwerkthemen beschreibt nun den Einsatz von Bluetooth®-Geräten unter FreeBSD und das Einrichten von drahtlosen Netzwerken sowie ATM-Netzwerken.
- Neu hinzugefügt wurde ein Glossar, das die im Buch verwendeten technischen Ausdrücke definiert.
- Das Erscheinungsbild der Tabellen und Abbildungen im Buch wurde verbessert.

# Änderungen gegenüber der ersten Auflage (2001)

Die zweite Auflage ist das Ergebnis der engagierten Arbeit der Mitglieder des FreeBSD Documentation Projects über zwei Jahre. Die wichtigsten Änderungen gegenüber der ersten Auflage sind:

- Ein Index wurde erstellt.
- Alle ASCII-Darstellungen wurden durch Grafiken ersetzt.
- Jedes Kapitel wird durch eine Übersicht eingeleitet, die den Inhalt des Kapitels zusammenfasst und die Voraussetzungen für ein erfolgreiches Durcharbeiten des Kapitels darstellt.
- Der Inhalt wurde in die logischen Abschnitte "Erste Schritte", "Systemadministration" und "Anhänge" unterteilt.
- Grundlagen des FreeBSD Betriebssystems wurde um den Abschnitt "Dämonen, Signale und Stoppen von Prozessen" erweitert.
- Das Installieren von Anwendungen: Pakete und Ports behandelt nun auch Pakete.
- Das X-Window-System wurde neu geschrieben. Der Schwerpunkt liegt auf modernen Benutzeroberflächen wie KDE und GNOME unter XFree86™.
- Das FreeBSDs Bootvorgang wurde erweitert.
- Speichermedien ist aus den beiden Kapiteln "Laufwerke" und "Sicherungen" entstanden. Die in den beiden Kapiteln diskutierten Themen sind so leichter zu verstehen. Hinzugekommen ist ein Abschnitt über Software- und Hardware-RAID.
- Das Serielle Datenübertragung wurde reorganisiert und auf FreeBSD 4.X/5.X angepasst.
- Das PPP wurde aktualisiert.
- Weiterführende Netzwerkthemen wurde um viele neue Abschnitte erweitert.
- Elektronische Post (E-Mail) wurde um einen Abschnitt über die Konfiguration von Sendmail erweitert.

- Linux®-Binärkompatibilität behandelt zusätzlich die Installation von Oracle® und SAP® R/3®.
- Neu hinzugekommen sind:
  - Konfiguration und Tuning.
  - Multimedia.

# Gliederung

Dieses Buch ist in fünf Abschnitte unterteilt. Der erste Abschnitt, *Erste Schritte*, behandelt die Installation und die Grundlagen von FreeBSD. Dieser Abschnitt sollte in der vorgegebenen Reihenfolge durchgearbeitet werden, schon Bekanntes darf aber übersprungen werden. Der zweite Abschnitt, *Oft benutzte Funktionen*, behandelt häufig benutzte Funktionen von FreeBSD. Dieser Abschnitt sowie alle nachfolgenden Abschnitte können in beliebiger Reihenfolge gelesen werden. Jeder Abschnitt beginnt mit einer kurzen Übersicht, die das Thema des Abschnitts und das nötige Vorwissen erläutert. Die Übersichten helfen dem Leser, interessante Kapitel zu finden und erleichtern das Stöbern im Handbuch. Der dritte Abschnitt, *Systemadministration*, behandelt die Administration eines FreeBSD-Systems. Der vierte Abschnitt, *Netzwerke*, bespricht Netzwerke und Netzwerkdienste. Der fünfte Abschnitt enthält Anhänge und Verweise auf weitere Informationen.

### *Einleitung*

Dieses Kapitel macht Einsteiger mit FreeBSD vertraut. Es behandelt die Geschichte, die Ziele und das Entwicklungsmodell des FreeBSD-Projekts.

### *FreeBSD installieren*

Beschreibt den Ablauf der Installation von FreeBSD 9.*x* und neuere mittels bsdinstall.

### *Grundlagen des FreeBSD Betriebssystems*

Erläutert die elementaren Kommandos und Funktionen von FreeBSD. Wenn Sie schon mit Linux® oder einem anderen UNIX® System vertraut sind, können Sie dieses Kapitel überspringen.

### *Installieren von Anwendungen: Pakete und Ports*

Zeigt wie mit der innovativen Ports-Sammlung oder mit Paketen Software von Fremdherstellern installiert wird.

### *Das X-Window-System*

Beschreibt allgemein das X Window System und geht speziell auf X11 unter FreeBSD ein. Weiterhin werden graphische Benutzeroberflächen wie KDE und GNOME behandelt.

### *Desktop-Anwendungen*

Enthält eine Aufstellung verbreiteter Anwendungen wie Browser, Büroanwendungen und Office-Pakete und beschreibt wie diese Anwendungen installiert werden.

### *Multimedia*

Erklärt, wie Sie auf Ihrem System Musik und Videos abspielen können. Beispielhaft werden auch Anwendungen aus dem Multimedia-Bereich beleuchtet.

### *Konfiguration des FreeBSD-Kernels*

Erklärt, warum Sie einen angepassten Kernel erzeugen sollten und gibt ausführliche Anweisungen wie Sie einen angepassten Kernel konfigurieren, bauen und installieren.

### *Drucken*

Beschreibt, wie Sie Drucker unter FreeBSD verwalten. Diskutiert werden Deckblätter, das Einrichten eines Druckers und ein Abrechnungssystem für ausgedruckte Seiten.

### *Linux®-Binärkompatibilität*

Beschreibt die binäre Kompatibilität zu Linux®. Weiterhin werden ausführliche Installationsanleitungen für Oracle® und Mathematica® gegeben.

### *Konfiguration und Tuning*

Beschreibt die Einstellungen, die ein Systemadministrator vornehmen kann, um die Leistungsfähigkeit eines FreeBSD Systems zu verbessern. In diesem Kapitel werden auch verschiedene Konfigurationsdateien besprochen.

### *FreeBSDs Bootvorgang*

Erklärt den Bootprozess von FreeBSD und beschreibt die Optionen, mit denen sich der Bootprozess beeinflussen lässt.

### *Sicherheit*

Beschreibt die Werkzeuge mit denen Sie Ihr FreeBSD-System absichern. Unter Anderem werden Kerberos, IPsec und OpenSSH besprochen.

### *Jails*

Dieses Kapitel beschreibt das Jails-Framework sowie die Vorteile von Jails gegenüber der traditionellen chroot-Unterstützung von FreeBSD.

### *Verbindliche Zugriffskontrolle*

Erklärt vorgeschriebene Zugriffskontrollen (MACs) und wie mit ihrer Hilfe FreeBSD-Systeme gesichert werden.

### *Security Event Auditing*

Beschreibt, was FreeBSD Event Auditing ist, wie Sie diese Funktion installieren und konfigurieren und die damit erzeugten Audit-Trails überwachen und auswerten können.

### *Speichermedien*

Erläutert den Umgang mit Speichermedien und Dateisystemen. Behandelt werden Plattenlaufwerke, RAID-Systeme, optische Medien, Bandlaufwerke, speicherbasierte Laufwerke und verteilte Dateisysteme.

### *GEOM: Modulares Framework zur Plattentransformation*

Beschreibt das GEOM-Framework von FreeBSD sowie die Konfiguration der verschiedenen unterstützten RAID-Level.

### *Dateisystemunterstützung*

Beschreibt die Unterstützung nicht-nativer Dateisysteme (beispielsweise des Z-Dateisystems (zfs)

von Sun™) durch FreeBSD.

### *Virtualisierung*

Dieses Kapitel beschreibt verschiedene Virtualisierungslösungen und wie diese mit FreeBSD zusammenarbeiten.

### *Lokalisierung – I18N/L10N einrichten und benutzen*

Zeigt wie Sie FreeBSD mit anderen Sprachen als Englisch einsetzen. Es wird sowohl die Lokalisierung auf der System-Ebene wie auch auf der Anwendungs-Ebene betrachtet.

### *FreeBSD aktualisieren*

Erklärt die Unterschiede zwischen FreeBSD-STABLE, FreeBSD-CURRENT und FreeBSD-Releases. Das Kapitel enthält Kriterien anhand derer Sie entscheiden können, ob es sich lohnt, ein Entwickler-System zu installieren und aktuell zu halten. Außerdem wird beschrieben, wie Sie ein System durch das Einspielen neuer Sicherheits-Patches absichern.

### *DTrace*

Beschreibt, wie das von Sun™ entwickelte DTrace-Werkzeug unter FreeBSD konfiguriert und eingesetzt werden kann. Dynamisches Tracing kann Ihnen beim Aufspüren von Leistungsproblemen helfen, indem Sie Echtzeit-Systemanalysen durchführen.

### *Serielle Datenübertragung*

Erläutert, wie Sie Terminals und Modems an Ihr FreeBSD-System anschließen und sich so ein- und auswählen können.

### *PPP*

Erklärt wie Sie mit PPP, SLIP oder PPP über Ethernet ein FreeBSD-System mit einem entfernten System verbinden.

### *Elektronische Post (E-Mail)*

Erläutert die verschiedenen Bestandteile eines E-Mail Servers und zeigt einfache Konfigurationen für sendmail, dem meist genutzten E-Mail-Server.

### *Netzwerkserver*

Bietet ausführliche Informationen und Beispielkonfigurationen, die es Ihnen ermöglichen, Ihren FreeBSD-Rechner als Network File System Server, Domain Name Server, Network Information Server, oder als Zeitsynchronisationsserver einzurichten.

### *Firewalls*

Erklärt die Philosophie hinter softwarebasierten Firewalls und bietet ausführliche Informationen zur Konfiguration der verschiedenen, für FreeBSD verfügbaren Firewalls.

### *Weiterführende Netzwerkthemen*

Behandelt viele Netzwerkthemen, beispielsweise das Verfügbarmachen einer Internetverbindung für andere Rechner eines LANs, Routing, drahtlose Netzwerke, Bluetooth®, IPv6, ATM und andere mehr.

*Bezugsquellen für FreeBSD*

Enthält eine Aufstellung der Quellen von denen Sie FreeBSD beziehen können: CD-ROM, DVD sowie Internet-Sites.

*Bibliografie*

Dieses Buch behandelt viele Themen und kann nicht alle Fragen erschöpfend beantworten. Die Bibliografie enthält weiterführende Bücher, die im Text zitiert werden.

*Ressourcen im Internet*

Enthält eine Aufstellung der Foren, die FreeBSD Benutzern für Fragen und Diskussionen zur Verfügung stehen.

*OpenPGP-Schlüssel*

Enthält PGP-Fingerabdrücke von etlichen FreeBSD Entwicklern.

# Konventionen in diesem Buch

Damit der Text einheitlich erscheint und leicht zu lesen ist, werden im ganzen Buch die nachstehenden Konventionen beachtet:

## Typographie

***Kursiv***

Für Dateinamen, URLs, betonte Teile eines Satzes und das erste Vorkommen eines Fachbegriffs wird ein *kursiver* Zeichensatz benutzt.

`Fixschrift`

Fehlermeldungen, Kommandos, Umgebungsvariablen, Namen von Ports, Hostnamen, Benutzernamen, Gruppennamen, Gerätenamen, Variablen und Code-Ausschnitte werden in einer `Fixschrift` dargestellt.

**Fett**

**Fett** kennzeichnet Anwendungen, Kommandozeilen und Tastensymbole.

## Benutzereingaben

Tasten werden **fett** dargestellt, um sie von dem umgebenden Text abzuheben. Tasten, die gleichzeitig gedrückt werden müssen, werden durch ein + zwischen den einzelnen Tasten dargestellt:

`Ctrl` + `Alt` + `Del`

Im gezeigten Beispiel soll der Benutzer die Tasten `Ctrl`, `Alt` und `Del` gleichzeitig drücken.

Tasten, die nacheinander gedrückt werden müssen, sind durch Kommas getrennt:

`Ctrl` + `X`, `Ctrl` + `S`

Das letzte Beispiel bedeutet, dass die Tasten `Ctrl` und `X` gleichzeitig betätigt werden und danach die

Tasten `Ctrl` und `S` gleichzeitig gedrückt werden müssen.

### Beispiele

Beispiele, die durch C:\> eingeleitet werden, zeigen ein MS-DOS® Kommando. Wenn nichts Anderes angezeigt wird, können diese Kommandos unter neuen Versionen von Microsoft® Windows® auch in einem DOS-Fenster ausgeführt werden.

```
E:\> tools\fdimage floppies\kern.flp A:
```

Beispiele, die mit # beginnen, müssen unter FreeBSD mit Superuser-Rechten ausgeführt werden. Dazu melden Sie sich entweder als `root` an oder Sie wechseln von Ihrem normalen Account mit su(1) zu dem Benutzer `root`.

```
# dd if=kern.flp of=/dev/fd0
```

Beispiele, die mit % anfangen, werden unter einem normalen Benutzer-Account ausgeführt. Sofern nichts Anderes angezeigt wird, verwenden die Beispiele die Syntax der C-Shell.

```
% top
```

# Danksagung

Dieses Buch ist aus Beiträgen von vielen Leuten aus allen Teilen der Welt entstanden. Alle eingegangen Beiträge, zum Beispiel Korrekturen oder vollständige Kapitel, waren wertvoll.

Einige Firmen haben dieses Buch dadurch unterstützt, dass Sie Autoren in Vollzeit beschäftigt und die Veröffentlichung des Buchs finanziert haben. Besonders BSDi (das später von Wind River Systems übernommen wurde) beschäftigte Mitglieder des FreeBSD Documentation Projects, um dieses Buch zu erstellen. Dadurch wurde die erste (englische) gedruckte Auflage im März 2000 möglich (ISBN 1-57176-241-8). Wind River Systems bezahlte dann weitere Autoren, die die zum Drucken nötige Infrastruktur verbesserten und zusätzliche Kapitel beisteuerten. Das Ergebnis dieser Arbeit ist die zweite (englische) Auflage vom November 2001 (ISBN 1-57176-303-1). Zwischen 2003 und 2004 bezahlte FreeBSD Mall, Inc mehrere Mitarbeiter für die Vorbereitung der gedruckten dritten Auflage.

params: path: "/books/handbook/parti/" :leveloffset: +1

# Teil I: Einleitung

# Kapitel 1. Überblick

Herzlichen Dank für Ihr Interesse an FreeBSD! Das folgende Kapitel behandelt verschiedene Aspekte des FreeBSD Projekts wie dessen geschichtliche Entwicklung, seine Ziele oder das Entwicklungsmodell.

Nach dem Durcharbeiten des Kapitels wissen Sie über folgende Punkte Bescheid:

- Wo FreeBSD im Vergleich zu anderen Betriebssystemen steht
- Die Geschichte des FreeBSD Projekts
- Die Ziele des FreeBSD Projekts
- Die Grundlagen des FreeBSD-Open-Source-Entwicklungsmodells
- Und natürlich woher der Name "FreeBSD" kommt.

# Kapitel 2. Willkommen zu FreeBSD!

FreeBSD ist ein offenes, standardkonformes Unix-ähnliches Betriebssystem für x86 (32 und 64 Bit) ARM®, AARch64, RISC-V®, MIPS®, POWER®, PowerPC® und Sun UltraSPARC® Rechner. Es bietet alle Funktionen, die heutzutage als selbstverständlich angesehen werden, wie präemptives Multitasking, Speicherschutz, virtueller Speicher, Mehrbenutzerfunktionen, SMP-Unterstützung, Open Source Entwicklungswerkzeuge für verschiedene Sprachen und Frameworks sowie Desktop-Funktionen rund um das X Window System, KDE und GNOME. Besondere Eigenschaften sind:

- *Liberale Open Source Lizenz*, die Ihnen das Recht einräumt, den Quellcode frei zu modifizieren und zu erweitern und ihn in freien oder proprietären Produkten zu integrieren, ohne dabei den für Copyleft-Lizenzen typischen Einschränkungen zu unterliegen. Ebenso sollen mögliche Inkompatibilitätsprobleme vermieden werden.
- *Starke TCP/IP-Netzwerkfähigkeit* - FreeBSD implementiert Industrie-Standardprotokolle mit immer höherer Leistung und Skalierbarkeit. Dies macht FreeBSD zu einer exzellenten Lösung sowohl für Server, als auch für Routing/Firewall Aufgaben. Tatsächlich nutzen viele Unternehmen und Anbieter FreeBSD zu genau diesem Zweck.
- *Vollständig integrierte OpenZFS-Unterstützung*, einschließlich root-on-ZFS, ZFS Boot Environments, Fehlermanagement, administrative Delegation, Unterstützung für Jails, FreeBSD-spezifische Dokumentation und Unterstützung im System-Installationsprogramm.
- *Umfangreiche Sicherheitsfunktionen*, vom System für die verbindliche Zugriffskontrolle (Mandatory Access Control, MAC), bis hin zu Capsicum und Sandbox-Mechanismen.
- *Über 30.000 vorkonfigurierte Pakete* für alle unterstützten Architekturen und die Ports-Sammlung, die es Benutzern einfach macht, eigene, angepasste Software zu erstellen.
- *Dokumentation* - Zusätzlich zu diesem Handbuch und Büchern von verschiedenen Autoren, die Themen von Systemadministration bis hin zu Kernel-Interna behandeln, gibt es auch die man(1) Seiten, nicht nur für Daemonen, Dienstprogramme und Konfigurationsdateien, sondern auch für Kernel-APIs (Sektion 9) und individuelle Treiber (Sektion 4).
- *Einfache und konsistente Repository-Struktur und Build-System* - FreeBSD benutzt ein einziges Repository für alle seine Komponenten, sowohl den Kernel als auch das Basissystem. Dies, zusammen mit einem einheitlichen und leicht anpassbaren Build-System und einem gut durchdachten Entwicklungsprozess, macht es einfach, FreeBSD in die Build-Infrastruktur für Ihr eigenes Produkt zu integrieren.
- *Der Unix-Philosophie treu bleiben* und Kombinierbarkeit den monolithischen "all in one"-Daemonen mit hartkodiertem Verhalten vorziehen.
- *Binärkompatibilität* mit Linux, die es möglich macht, viele Linux-Binärdateien ohne Virtualisierung auszuführen.

FreeBSD basiert auf dem 4.4BSD-LiteRelease der Computer Systems Research Group (CSRG) der Universität von Kalifornien in Berkeley und führt die namenhafte Tradition der Entwicklung von BSD-Systemen fort. Zusätzlich zu der herausragenden Arbeit CSRG hat das FreeBSD Projekt tausende weitere Arbeitsstunden investiert, um das System zu erweitern, verfeinern und maximale Leistung und Zuverlässigkeit bei Alltagslast zu bieten. FreeBSD bietet Leistung und Zuverlässigkeit auf dem Niveau von Open Source und kommerziellen Angeboten, und kombiniert innovative

Funktionen, die woanders nicht verfügbar sind.

## 2.1. Was kann FreeBSD?

Die Anwendungsmöglichkeiten von FreeBSD werden nur durch Ihre Vorstellungskraft begrenzt. Von Software-Entwicklung bis zu Produktionsautomatisierung, von Lagerverwaltung über Abweichungskorrektur bei Satelliten; Falls etwas mit kommerziellen UNIX® Produkten machbar ist, dann ist es höchstwahrscheinlich auch mit FreeBSD möglich. FreeBSD profitiert stark von tausenden hochwertigen Anwendungen aus wissenschaftlichen Instituten und Universitäten in aller Welt. Häufig sind diese für wenig Geld oder sogar kostenlos zu bekommen.

Durch den freien Zugang zum Quellcode von FreeBSD ist es in unvergleichbarer Weise möglich, das System für spezielle Anwendungen oder Projekte anzupassen. Dies ist mit den meisten kommerziellen Betriebssystemen einfach nicht möglich. Beispiele für Anwendungen, die unter FreeBSD laufen, sind:

- *Internet-Dienste:* Die robuste TCP/IP-Implementierung in FreeBSD macht es zu einer idealen Plattform für verschiedenste Internet-Dienste, wie zum Beispiel:
  - Webserver
  - IPv4- und IPv6-Routing
  - Firewall NAT ("IP-Masquerading")-Gateways
  - FTP-Server
  - E-Mail-Server
  - Und mehr…
- *Bildung:* Sind Sie Informatikstudent oder Student eines verwandten Studiengangs? Die praktischen Einblicke in FreeBSD sind die beste Möglichkeit etwas über Betriebssysteme, Rechnerarchitektur und Netzwerke zu lernen. Einige frei erhältliche CAD-, mathematische und grafische Anwendungen sind sehr nützlich, gerade für diejenigen, deren Hauptinteresse in einem Computer darin besteht, *andere* Arbeit zu erledigen!
- *Forschung:* Mit dem frei verfügbaren Quellcode für das gesamte System bildet FreeBSD ein exzellentes Studienobjekt in der Disziplin der Betriebssysteme, wie auch in anderen Zweigen der Informatik. Es ist beispielsweise denkbar, das räumlich getrennte Gruppen gemeinsam an einer Idee oder Entwicklung arbeiten. Das Konzept der freien Verfügbarkeit und -nutzung von FreeBSD ermöglicht so die freie Verwendung, ohne sich gross Gedanken über Lizenzbedingungen zu machen oder aufgrund von Beschränkungen evtl. in einem offenen Forum bestimmte Dinge nicht diskutieren zu dürfen.
- *Netzwerkfähigkeit:* Brauchen Sie einen neuen Router? Oder einen Name-Server (DNS)? Eine Firewall zum Schutze Ihres Intranets vor Fremdzugriff? FreeBSD macht aus dem in der Ecke verstaubenden 386- oder 486-PC im Handumdrehen einen leistungsfähigen Router mit anspruchsvollen Paketfilter-Funktionen.
- *Embedded:* FreeBSD ist eine exzellente Plattform, um auf embedded Systemen aufzubauen. Mit der Unterstützung für die ARM®-, MIPS®- und PowerPC®-Plattformen, verbunden mit dem robusten Netzwerkstack, aktuellen Neuerungen und der freizügigen BSD-Lizenz stellt FreeBSD eine ausgezeichnete Basis für embedded Router, Firewalls und andere Geräte dar.

- *Desktop:* FreeBSD ist eine gute Wahl für kostengünstige X-Terminals mit dem frei verfügbaren X11-Server. FreeBSD bietet die Auswahl aus vielen Open Source Desktop Umgebungen, dazu gehören auch die GNOME und KDE GUIs. FreeBSD kann sogar "plattenlos" booten, was einzelne Workstations sogar noch günstiger macht und die Verwaltung erleichtert.
- *Software-Entwicklung:* Das Standard-FreeBSD-System wird mit einem kompletten Satz an Entwicklungswerkzeugen bereitgestellt, unter anderem einem vollständigen C/C++-Compiler und -Debugger. Entwicklungswerkzeugen. Viele zusätzliche Programmiersprachen für Wissenschaft und Entwicklung sind aus der Ports- und Paket-Sammlung zu haben.

FreeBSD ist sowohl in Form von Quellcode als auch in Binärform auf CD-ROM, DVD und über Anonymous FTP erhältlich. Lesen Sie dazu Bezugsquellen für FreeBSD, um weitere Informationen erhalten.

## 2.2. Wer verwendet FreeBSD?

FreeBSD ist bekannt für seine Stärken als Webserver - zu den Webseiten, die unter FreeBSD laufen, gehören Hacker News, Netcraft, NetEase, Netflix, Sina, Sony Japan, Rambler, Yahoo!, und Yandex.

FreeBSDs fortgeschrittene Eigenschaften, bewährte Sicherheit und vorhersehbare Release-Zyklen, genauso wie seine tolerante Lizenz haben dazu geführt, dass es als Plattform zum Aufbau vieler kommerzieller und quelloffener Geräte und Produkte verwendet wird. Viele der weltgrössten IT-Unternehmen benutzen FreeBSD:

- Apache- Die Apache Software Foundation lässt den Grossteil seiner der Öffentlichkeit zugänglichen Infrastruktur, inklusive des möglicherweise grössten SVN-Repositories der Welt mit über 1,4 Millionen Commits, auf FreeBSD laufen.
- Apple- OS X verwendet viel von FreeBSDs eigenem Netzwerkstack, virtuellem Dateisystem und den Benutzerumgebungskomponenten für sein eigenes System. Apple iOS nutzt ebenso Elemente, die es von FreeBSD übernommen hat
- Cisco- IronPort Network Sicherheits- und Anti-Spam-Appliance verwendet einen modifizierten FreeBSD-Kernel.
- Citrix- Die NetScaler Reihe von Sicherheits-Appliances bietet auf den Schichten 4-7 Load-Balancing, Content Caching, Anwendungsfirewall, gesichertes VPN und mobilen Cloud-Netzwerkzugriff, gepaart mit der Mächtigkeit der FreeBSD-Shell.
- Isilon- Isilons Unternehmens-Speicherappliances basieren auf FreeBSD. Die extrem liberale FreeBSD-Lizenz erlaubt Isilon ihr intellektuelles Eigentum durch den gesamten Kernel zu integrieren und kann sich so auf das Erstellen ihres Produktes und nicht des Betriebssystems fokussieren.
- Quest KACE- Die KACE Systemmanagement-Appliances nutzen FreeBSD wegen seiner Zuverlässigkeit, Skalierbarkeit und Gemeinschaft, welche deren zukünftige Weiterentwicklung fördert.
- iXsystems- Die TrueNAS-Linie von vereinheitlichten Speicherappliances beruht auf FreeBSD. Zusätzlich zu deren kommerziellen Produkten, managed iXsystems auch noch die beiden Open Source Projekte TrueOS und FreeNAS.
- Juniper- Das JunOS Betriebssystem, welches alle Juniper Netzwerkgeräte (inklusive Router,

Switche, Sicherheits- und Netzwerkappliances) antreibt, verwendet FreeBSD Juniper ist einer der vielen Hersteller, welcher das symbolische Verhältnis zwischen dem Projekt und dem Hersteller von kommerziellen Produkten darstellt. Verbesserungen, die Juniper entwickelt hat, werden ebenso in FreeBSD aufgenommen, um die Komplexität der Integration neuer Eigenschaften von FreeBSD zurück in zukünftige JunOS Versionen zu vereinfachen.

- McAfee- SecurOS, die Basis von McAfee Enterprise-Firewallprodukten inkl. Sidewinder basiert auf FreeBSD.
- NetApp- Die Data ONTAP GX Reihe von Speicherappliances basieren auf FreeBSD. Zusätzlich hat NetApp viele Neuheiten beigesteuert, inklusive des neuen BSD-lizensierten Hypervisors bhyve.
- Netflix- Die OpenConnect-Appliance, die Netflix verwendet, um Filme zu seinen Kunden zu streamen basiert auf FreeBSD. Netflix hat weitreichende Beiträge zum Quellcode von FreeBSD beigetragen und arbeitet daran, ein möglichst geringes Delta zur normalen Version beizubehalten. Netflix OpenConnect-Appliances sind für mehr als 32% vom gesamten Internetverkehr in Nordamerika verantwortlich.
- Sandvine- Sandvine nutzt FreeBSD as die Basis für deren Echtzeit Hochgeschwindigkeits-Netzwerkplattform, welche den Kern deren intelligenter Netzwerkpolicy-Kontrollprodukte darstellt.
- Sony- Die PlayStation 4 Spielekonsole verwendet eine modifizierte Version von FreeBSD.
- Sophos- Das Sophos Email-Appliance Produkt basiert auf einem abgesicherten FreeBSD und scannt eingehende E-Mail auf Spam und Viren, während es gleichzeitig ausgehende Mail auf Schadsoftware und versehentlichen Versand von vertraulichen Informationen überwacht.
- Spectra Logic- Die nTier Reihe von archivspeicherfähigen Appliances nutzt FreeBSD und OpenZFS.
- Stormshield - Stormshield Network Security Appliances basieren auf einer abgesicherten Version von FreeBSD. Die BSD-Lizenz erlaubt es ihnen, ihr geistiges Eigentum in das System zu integrieren und gleichzeitig interessante Entwicklungen an die Gemeinschaft zurückzugeben.
- The Weather Channel- Die IntelliStar Appliance, welche am Kopfende eines jeden Kabelversorgers installiert ist und für das Einspeisen von lokalen Wettervorhersagen in das Kabelfernsehprogramm verantwortlich ist, läuft auf FreeBSD.
- Verisign- Verisign ist für den Betrieb der .com und .net Root-Domainregistries genauso verantwortlich wie für die dazugehörige DNS-Infrastruktur. Sie verlassen sich auf einen Reihe von verschiedenen Netzwerkbetriebssystemen inklusive FreeBSD, um zu gewährleisten, dass es keine gemeinsame Fehlerstelle in deren Infrastruktur gibt.
- Voxer- Voxer verwendet ZFS auf FreeBSD für ihre mobile Voice-Messaging-Platform. Voxer wechselte von einem Solaris-Derivat zu FreeBSD, wegen der ausgezeichneten Dokumentation und wegen der größeren, aktiveren und sehr Entwickler freundlichen Gemeinschaft. Neben entscheidenen Merkmalen wie ZFS und DTrace bietet FreeBSD auch TRIM-Unterstützung für ZFS.
- Fudo Security- Die FUDO Sicherheitsappliance erlaubt es Unternehmen, Vertragspartner und Administratoren, die an ihren Systemen arbeiten durchführen, zu überwachen, zu kontrollieren, aufzuzeichnen und zu begutachten. Dies basiert auf all den besten Sicherheitseigenschaften von FreeBSD, inklusive ZFS, GELI, Capsicum, HAST und auditdistd.

FreeBSD hat ebenfalls eine Reihe von verwandten Open Source Projekten hervorgebracht:

- BSD Router- Einen FreeBSD-basierten Ersatz für grosse Unternehmensrouter, der entwickelt wurde, um auf Standard PC-Hardware zu laufen.
- FreeNAS- Ein eigens dafür entworfenes FreeBSD für den Zweck als Netzwerk-Dateiserver Appliance zu fungieren. Es enthält eine Python-basierte Webschnittstelle, um das Management von sowohl UFS- als auch ZFS-Systemen zu vereinfachen. Enthalten sind NFS, SMB/CIFS, AFP, FTP und iSCSI. Ebenfalls enthalten ist ein erweiterteres Plugin-System basierend auf FreeBSD-Jails.
- GhostBSD- basiert auf FreeBSD und verwendet die GTK-Umgebung, um ein schönes Aussehen und eine komfortable Erfahrung auf der modernen BSD-Plattform zu liefern, die eine natürliche und native UNIX®-Arbeitsumgebung bietet.
- mfsBSD- Eine Sammlung von Werkzeugen zum Erstellen von FreeBSD-Systemimages, welches ausschliesslich im Hauptspeicher läuft.
- NAS4Free- Eine Dateiserverdistribution basierend auf FreeBSD mit einer von PHP-getriebenen Webschnittstelle.
- OPNSense- OPNSense ist eine quelloffene, einfach zu benutzende und auf FreeBSD basierende Firewall- und Router-Plattform. OPNSense enthält viele Funktionen die sonst nur in kommerziellen Firewalls enthalten sind und manchmal sogar mehr. Es kombiniert die vielfältigen Funktionen kommerzieller Angebote mit den Vorteilen von offenen und nachprüfbaren Quellen.
- TrueOS- TrueOS basiert auf der legendären Sicherheit und Stabilität von FreeBSD. TrueOS basiert auf FreeBSD-CURRENT und bietet die aktuellsten Treiber, Sicherheitsaktualisierungen und Pakete.
- FuryBSD - ein brandneuer, quelloffener FreeBSD Desktop. FuryBSD ist eine Hommage an die Desktop-BSD-Projekte der Vergangenheit wie PC-BSD und TrueOS mit seiner graphischen Oberfläche und beinhaltet zusätzliche Werkzeuge wie ein hybrides USB/DVD-Abbild hinzu. FuryBSD ist vollständig frei nutzbar und wird unter der BSD-Lizenz vertrieben.
- MidnightBSD - ist ein auf FreeBSD basierendes Betriebssystem, das mit Blick auf Desktop-Benutzer entwickelt wurde. Es enthält die gesamte Software, die Sie für Ihre täglichen Aufgaben erwarten: Mail, Web-Browsing, Textverarbeitung, Spiele und vieles mehr.
- pfSense- Eine Firewalldistribution basierend auf FreeBSD mit eine grossen Menge von Fähigkeiten und ausgedehnter IPv6-Unterstützung.
- ZRouter- Eine Open Source Firmware-Alternative für eingebettete Geräte, die auf FreeBSD basiert. Entwickelt wurde sie, um die proprietäre Firmware von Standard-Routern zu ersetzen.

Eine Liste von Referenzen von Unternehmen, dessen Produkte und Dienstleistungen auf FreeBSD basieren, finden Sie auf der Webseite der FreeBSD Foundation. Wikipedia pflegt eine Liste von Produkten, die auf FreeBSD basieren.

# Kapitel 3. Über das FreeBSD Projekt

Der folgende Abschnitt bietet einige Hintergrundinformationen zum FreeBSD Projekt, einschließlich einem kurzen geschichtlichen Abriss, den Projektzielen und dem Entwicklungsmodell.

## 3.1. Kurzer geschichtlicher Abriss zu FreeBSD

Das FreeBSD Projekt wurde Anfang 1993 ins Leben gerufen, zum Teil als Ergebnis der Arbeit der letzten drei Koordinatoren des "Unofficial 386BSD Patchkit": Nate Williams, Rod Grimes und Jordan Hubbard.

Das ursprüngliche Ziel war es, einen zwischenzeitlichen Abzug von 386BSD zu erstellen, um ein paar Probleme zu beseitigen, die das Patchkit-Verfahren nicht lösen konnte. Der frühe Arbeitstitel für das Projekt war "386BSD 0.5" oder "386BSD Interim" als Referenz darauf.

386BSD war das Betriebssystem von Bill Jolitz, welches bis zu diesem Zeitpunkt heftig unter fast einjähriger Vernachlässigung litt. Als das Patchkit mit jedem Tag anschwoll und unhandlicher wurde, entschied man sich, Bill Jolitz zu helfen, indem ein übergangsweise "bereinigter" Abzug zur Verfügung gestellt wurde. Diese Pläne wurden durchkreuzt, als Bill Jolitz plötzlich seine Zustimmung zu diesem Projekt zurückzog, ohne einen Hinweis darauf, was stattdessen geschehen sollte.

Das Trio entschied, dass das Ziel sich weiterhin lohnen würde, selbst ohne die Unterstützung von Bill und so wurde entschieden, den Namen FreeBSD zu verwenden, der von David Greenman geprägt wurde. Die anfänglichen Ziele wurden festgelegt, nachdem man sich mit den momentanen Benutzern des Systems besprach und abzusehen war, dass das Projekt die Chance hatte, Realität zu werden, kontaktierte Jordan Walnut Creek CDROM mit dem Vorhaben, FreeBSDs Verteilung auch auf diejenigen auszuweiten, die noch keinen Internetzugang besaßen. Walnut Creek CDROM unterstützte nicht nur die Idee durch die Verbreitung von FreeBSD auf CD, sondern ging auch so weit dass es dem Projekt eine Maschine mit schneller Internetverbindung zur Verfügung stellte, um damit zu arbeiten. Ohne den von Walnut Creek bisher nie dagewesenen Grad von Vertrauen in ein, zur damaligen Zeit, komplett unbekanntes Projekt, wäre es unwahrscheinlich, dass FreeBSD so weit gekommen wäre, wie es heute ist.

Die erste auf CD-ROM (und netzweit) verfügbare Veröffentlichung war FreeBSD 1.0 im Dezember 1993. Diese basierte auf dem Band der 4.3BSD-Lite ("Net/2") der Universität von Kalifornien in Berkeley. Viele Teile stammten aus 386BSD und von der Free Software Foundation. Gemessen am ersten Angebot, war das ein ziemlicher Erfolg und Sie ließen dem das extrem erfolgreiche FreeBSD 1.1 im Mai 1994 folgen.

Zu dieser Zeit formierten sich unerwartete Gewitterwolken am Horizont, als Novell und die Universität von Kalifornien in Berkeley (UCB) ihren langen Rechtsstreit über den rechtlichen Status des Berkeley Net/2-Bandes mit einem Vergleich beilegten. Eine Bedingung dieser Einigung war es, dass die UCB große Teile des Net/2-Quellcodes als "belastet" zugestehen musste, und dass diese Besitz von Novell sind, welches den Code selbst einige Zeit vorher von AT&T bezogen hatte. Im Gegenzug bekam die UCB den "Segen" von Novell, dass sich das 4.4BSD-Lite-Release bei seiner endgültigen Veröffentlichung als unbelastet bezeichnen darf. Alle Net/2-Benutzer sollten auf das

neue Release wechseln. Das betraf auch FreeBSD. Dem Projekt wurde eine Frist bis Ende Juli 1994 eingeräumt, das auf Net/2-basierende Produkt nicht mehr zu vertreiben. Unter den Bedingungen dieser Übereinkunft war es dem Projekt noch erlaubt ein letztes Release vor diesem festgesetzten Zeitpunkt herauszugeben. Das war FreeBSD 1.1.5.1.

FreeBSD machte sich dann an die beschwerliche Aufgabe, sich Stück für Stück aus einem neuen und ziemlich unvollständigen Satz von 4.4BSD-Lite-Teilen, wieder aufzubauen. Die "Lite" -Veröffentlichungen waren deswegen leicht, weil Berkeleys CSRG große Code-Teile, die für ein start- und lauffähiges System gebraucht wurden, aufgrund diverser rechtlicher Anforderungen entfernen musste und weil die 4.4-Portierung für Intel-Rechner extrem unvollständig war. Das Projekt hat bis November 1994 gebraucht diesen Übergang zu vollziehen. Im Dezember wurde dann FreeBSD 2.0 veröffentlicht. Obwohl FreeBSD gerade die ersten Hürden genommen hatte, war dieses Release ein maßgeblicher Erfolg. Diesem folgte im Juni 1995 das robustere und einfacher zu installierende FreeBSD 2.0.5.

Seit dieser Zeit hat FreeBSD eine Reihe von Releases veröffentlicht, die jedes mal die Stabilität, Geschwindigkeit und Menge an verfügbaren Eigenschaften der vorherigen Version verbessert.

Momentan werden langfristige Entwicklungsprojekte im 10.X-CURRENT (Trunk)-Zweig durchgeführt, und Abzüge (Snapshots) der Releases von 10.X werden regelmässig auf den Snapshot-Servern zur Verfügung gestellt.

## 3.2. Ziele des FreeBSD-Projekts

Das FreeBSD Projekt stellt Software her, die ohne Einschränkungen für beliebige Zwecke eingesetzt werden kann. Viele von uns haben beträchtlich in Quellcode und das Projekt investiert und hätten sicher nichts dagegen, hin und wieder ein wenig finanziellen Ausgleich dafür zu bekommen. Aber in keinem Fall bestehen wir darauf. Wir glauben unsere erste und wichtigste "Mission" ist es, Software für jeden Interessierten und zu jedem Zweck zur Verfügung zu stellen, damit die Software größtmögliche Verbreitung erlangt und größtmöglichen Nutzen stiftet. Das ist, glaube ich, eines der grundlegenden Ziele freier Software, welche wir mit größter Begeisterung unterstützen.

Der Code in unserem Quellbaum, der unter die General Public License (GPL) oder die Library General Public License (LGPL) fällt, stellt geringfügig mehr Bedingungen. Das aber vielmehr im Sinne von eingefordertem Zugriff, als das übliche Gegenteil der Beschränkungen. Aufgrund zusätzlicher Abhängigkeiten, die sich durch die Verwendung von GPL-Software bei kommerziellem Gebrauch ergeben, bevorzugen wir daher Software unter der transparenteren BSD-Lizenz, wo immer es angebracht ist.

## 3.3. Das FreeBSD-Entwicklungsmodell

Die Entwicklung von FreeBSD ist ein offener und flexibler Prozess, der durch den Beitrag von buchstäblich tausenden Leuten rund um die Welt ermöglicht wird, wie an der Liste der Beitragenden ersehen können. Die vielen Entwickler können aufgrund der Entwicklungs-Infrastruktur von FreeBSD über das Internet zusammenarbeiten. Wir suchen ständig nach neuen Entwicklern, Ideen und jenen, die sich in das Projekt tiefer einbringen wollen. Nehmen Sie einfach auf der Mailingliste FreeBSD technical discussions Kontakt mit uns auf. Die Mailingliste FreeBSD announcements steht für wichtige Ankündigungen, die alle FreeBSD-Benutzer betreffen, zur

Verfügung.

Unabhängig davon ob Sie alleine oder mit anderen eng zusammen arbeiten, enthält die folgende Aufstellung nützliche Informationen über das FreeBSD Projekt und dessen Entwicklungsabläufe.

**Die SVN-Repositories**

Der Hauptquellbaum von FreeBSD wurde über viele Jahre ausschließlich mit CVS (Concurrent-Versions-System) gepflegt, einem frei erhältlichen Versionskontrollsystem. Im Juni 2008 begann das FreeBSD Project mit dem Umstieg auf SVN (Subversion). Dieser Schritt wurde notwendig, weil durch technische Einschränkungen von CVS aufgrund des rapide wachsenden Quellcodebaumes und dem Umfang der bereits gespeichterten Revisisionsinformationen an dessen Grenzen zu stoßen begann. Die Repositories des Dokumentationsprojekts und die Ports-Sammlung wurden ebenfalls von CVS zu SVN im Mai und Juli 2012 umgezogen. Lesen Sie dazu Synchronisation der Quellen für weitere Informationen zur Synchronisation des FreeBSD `src/` -Repositories und Die Ports-Sammlung verwenden für Details zum Beziehen der FreeBSD Ports-Sammlung.

**Die Committer-Liste**

Die *Committer* sind diejenigen Leute, welche *schreibenden* Zugriff auf den Subversion-Baum besitzen und berechtigt sind, Änderungen an den FreeBSD-Quellen (der Begriff "Committer" stammt aus dem Versionskontrollbefehl `commit` , der dazu verwendet wird, Änderungen in das Repository zu bringen). Jeder hat die Möglichkeit über die die Datenbank für Problemberichte einen Fehlerreport einzureichen. Bevor Sie einen Fehlerreport einreichen, sollten Sie auf den FreeBSD Mailinglisten, den IRC-Kanälen oder in Foren überprüfen, ob das Problem tatsächlich ein Fehler ist.

**The FreeBSD core team**

Die *FreeBSD core team* ist mit dem Vorstand vergleichbar, wenn das FreeBSD Projekt ein Unternehmen wäre. Die Hauptaufgabe des Core Teams ist es sicherzustellen, dass sich das Projekt als Ganzes in einem guten Zustand befindet und sich in die richtige Richtung bewegt. Das Einladen von engagierten und verantwortungsvollen Entwicklern zu dem Zweck, sich der Gruppe von Committern anzuschliessen, ist eine der Funktionen des Core Teams, genauso wie neue Mitglieder des Core Teams zu rekrutieren, wenn andere ausscheiden. Das aktuelle Core Team wurde aus einer Menge von Kandidaten aus dem Kreis der Committer im Juni 2020 gewählt. Wahlen werden alle zwei Jahre abgehalten.



Wie die meisten Entwickler auch, sind die Mitglieder des Core Teams Freiwillige, wenn es um die Entwicklung von FreeBSD geht und erhalten keinerlei finanziellen Vorteil aus dem Projekt, deshalb sollte "Verpflichtung" nicht fehlverstanden werden mit "garantierter Unterstützung". Die "Vorstands"-Analogie oben ist nicht sehr akkurat und kann vielleicht besser damit umschrieben werden, dass diese Leute ihr Leben für FreeBSD gegen jedwede Vernunft geopfert haben.

**Aussenstehende Beitragende**

Schliesslich stellt die grösste, aber nichtsdestotrotz wichtigste Gruppe von Entwicklern die der Benutzer selbst dar, die stetig Rückmeldungen und Fehlerbehebungen liefert. Der hauptsächliche Weg mit FreeBSDs nicht-zentralisierter Entwicklung Kontakt zu halten, ist, die

FreeBSD technical discussions Mailingliste zu abonnieren, auf der solche Dinge diskutiert werden. Lesen Sie dazu Ressourcen im Internet für weitere Informationen über die verschiedenen FreeBSD-Mailinglisten.

Liste der Beitragenden ist eine, die lang ist und stetig wächst, also warum nicht FreeBSD beitreten und noch heute etwas zurückgeben?

Code ist nicht die einzige Art, zu dem Projekt etwas beizutragen. Für eine ausführlichere Liste von Dingen die getan werden müssen, lesen Sie auf der FreeBSD Projektwebseite.

Zusammenfassend ist unser Entwicklungsmodell als eine lose Menge von konzentrischen Kreisen organisiert. Das zentralisierte Modell ist mit der Praktikabilität der *Anwender* von FreeBSD entworfen worden, die mit der einfachen Art einhergeht, eine zentrale Basis für den Code zu haben und keine potentiellen Beiträge auszuschliessen! Unser Ansporn ist es, ein stabiles Betriebssystem mit einer grossen Menge von kohärenten Anwendungsprogrammen, welches die Benutzer einfach installieren und verwenden können - dieses Modell funktioniert darin sehr gut, dieses Ziel zu erreichen.

Alles was wir von denen verlangen, die uns als FreeBSD-Entwickler beitreten ist, etwas von der gleichen Hingabe an den Erfolg, die seine momentanen Gemeinschaft inne hat, zu besitzen.

## 3.4. Programme von Drittherstellern

Zusätzlich zur Basisdistribution bietet FreeBSD eine Sammlung von portierter Software mit tausenden der am meisten nachgefragten Programme an. Als diese Zeilen geschrieben wurden, gab es über 37000 Ports! Die Liste der Ports reicht von HTTP-Servern, zu Spielen, Sprachen, Editoren und so ziemlich alles, was dazwischen liegt. Die gesamte Port-Sammlung ist geschätzt 3 GB gross. Um einen Port zu übersetzen, wechseln Sie einfach in das Verzeichnis des Programms, das sie installieren möchten und geben `make install` ein und das System erledigt den Rest. Die gesamte Originaldistribution für jeden Port, den Sie bauen wird dynamisch heruntergeladen, so dass sie nur genügend Plattenplatz zum bauen des Ports, den sie haben möchten, zur Verfügung stellen müssen. Fast jeder Port ist auch als vorkompiliertes"Paket", das über das folgende einfache Kommando (`pkg install`) für diejenigen, die keine kompilierten Port aus den Quellen wünschen. Weitere Informationen zu Ports und Paketen finden Sie in Installieren von Anwendungen: Pakete und Ports.

## 3.5. Zusätzliche Dokumentation

Alle unterstützten FreeBSD Versionen bieten eine Option, um zusätzliche Dokumentation unter /usr/local/shared/doc/freebsd während des initialen Systemsetups zu installieren. Dokumentation kann auch zu einem späteren Zeitpunkt über Pakete installiert werden, wie es “Die Dokumentation aus den Ports aktualisieren” beschreibt. Sie können ebenso die lokal installierten Anleitungen mit jedem HTML-fähigen Browser lesen, indem Sie die folgende URL verwenden:

**Das FreeBSD Handbuch**

/usr/local/shared/doc/freebsd/handbook/index.html

**Die FreeBSD FAQ**

/usr/local/shared/doc/freebsd/faq/index.html

Genauso erhalten Sie auch die Master (und am häufigsten aktualisierten) Kopien von https://www.FreeBSD.org/.

# Kapitel 4. FreeBSD installieren

## 4.1. Übersicht

Es gibt verschiedene Möglichkeiten, FreeBSD zu installieren, abhängig von der Einsatzumgebung. Dazu gehören:

- Abbilder von virtuellen Maschinen, die Sie herunterladen und in einer virtuellen Umgebung einsetzen können. Diese Abbilder können von der FreeBSD Downloadseite heruntergeladen werden. Es gibt Abbilder für KVM ("qcow2"), VMWare ("vmdk"), Hyper-V ("vhd"), sowie Raw-Device Abbilder, die durchgängig unterstützt werden. Dies sind keine Installationsabbilder, sondern vorkonfigurierte ("bereits installierte") Instanzen, die sofort gestartet und konfiguriert werden können.
- Abbilder von virtuellen Maschinen, die auf Amazon's AWS Marketplace, Microsoft Azure Marketplace und Google Cloud Platform verfügbar sind, um auf den jeweiligen Hosting-Diensten ausgeführt zu werden. Weitere Informationen zur Bereitstellung von FreeBSD auf Azure finden Sie im entsprechenden Kapitel der Azure Dokumentation.
- SD-Karten Abbilder für eingebettete Systeme wie den Raspberry Pi oder BeagleBone Black. Diese Abbilder können von der FreeBSD Downloadseite heruntergeladen werden. Die Dateien müssen unkomprimiert und als Raw-Image auf eine SD-Karte geschrieben werden, von der das System dann booten wird.
- Installationsabbilder, um FreeBSD auf einer Festplatte für die üblichen Desktop-, Laptop- oder Serversysteme zu installieren.

Der Rest dieses Kapitels beschreibt den vierten Fall und erklärt, wie man FreeBSD mit dem textbasierten Installationsprogramm bsdinstall installiert.

Die Installationsanweisungen in diesem Kapitel gelten für die i386™- und AMD64-Architekturen. Gegebenenfalls werden spezifische Anweisungen für andere Plattformen erwähnt. Möglicherweise gibt es auch geringfügige Unterschiede zwischen dem Installationsprogramm und dem, was hier gezeigt wird. Sie sollten dieses Kapitel daher als eine Art Wegweiser und nicht als exakte Anleitung betrachten.



Benutzer, die es vorziehen, FreeBSD mit einem graphischen Installationsprogramm zu installieren, sind vielleicht an FuryBSD, GhostBSD oder MidnightBSD interessiert.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie wissen:

- welche Mindestanforderungen an die Hardware gestellt werden und welche Architekturen FreeBSD unterstützt.
- wie man FreeBSD Installationsmedien erstellt.
- wie man bsdinstall startet.
- welche Fragen bsdinstall stellt, was sie bedeuten und wie man diese beantwortet.
- wie Sie Fehler bei der Installation beheben.

- wie Sie eine Live-Version von FreeBSD ausprobieren können, bevor Sie die Installation starten.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Die Liste von unterstützter Hardware lesen, die mit der zu installierenden Version von FreeBSD ausgeliefert wird, um sicherzustellen, dass die Hardware auch unterstützt wird.

## 4.2. Minimale Hardwareanforderungen

Die Hardwareanforderungen zur Installation von FreeBSD variieren mit der Architektur. Hardwarearchitekturen und von FreeBSD unterstützte Geräte sind auf der Seite FreeBSD Release Informationen aufgelistet. Die FreeBSD Download Seite enthält Informationen zur Auswahl des richtigen Abbilds für verschiedene Architekturen.

Für die Installation von FreeBSD sind mindestens 96 MB RAM und 1.5 GB freier Festplattenspeicher erforderlich. Allerdings ist eine solch geringe Menge an Arbeitsspeicher und Speicherplatz nur für spezifische Anwendungen ausreichend, beispielsweise für Embedded-Geräte. Desktop-Systeme benötigen weitaus mehr Ressourcen. 2-4 GB RAM und mindestens 8 GB Speicherplatz sind ein guter Anfang.

Dies sind die Anforderungen an den Prozessor für jede Architektur:

**amd64**

Dies ist die gängigste Art von Prozessor für Desktop- und Laptop-Systeme. Andere Anbieter nennen diese Architektur auch x86-64.

Beispiele für amd64-kompatible Prozessoren umfassen: AMD Athlon™64, AMD Opteron™, multi-core Intel® Xeon™ und Intel® Core™ 2 sowie neuere Prozessoren.

**i386**

Ältere Desktop- und Laptop-Systeme verwenden oft die 32-Bit x86-Architektur.

Fast alle i386-kompatiblen Prozessoren mit einer Floating-Point-Einheit werden unterstützt. Alle Intel®-Prozessoren 486 oder neuer werden unterstützt.

FreeBSD nutzt die Physical Adress Extensions (PAE), falls die CPU diese Funktion unterstützt. Wenn PAE im Kernel aktiviert ist, wird Speicher über 4 GB vom Kernel erkannt und kann von System verwendet werden. PAE schränkt allerdings auch die Gerätetreiber und anderen Komponenten von FreeBSD ein.

**powerpc**

Alle New Word ROM Apple® Mac®-Systeme mit integriertem USB werden unterstützt. SMP wird auf Maschinen mit mehreren CPUs unterstützt.

Ein 32-Bit Kernel kann jedoch nur die ersten 2 GB RAM verwenden.

**sparc64**

Systeme, die von FreeBSD/sparc64 unterstützt werden, sind auf der FreeBSD/sparc64-Projektseite aufgelistet.

SMP wird auf allen Systemen mit mehr als einem Prozessor unterstützt. Eine dedizierte Platte wird benötigt, da es nicht möglich ist, eine Platte mit einem anderen Betriebssystem zur gleichen Zeit zu teilen.

## 4.3. Vor der Installation

Wenn das System die Mindestanforderungen für die Installation von FreeBSD erfüllt, sollte die Installationsdatei heruntergeladen und die Installationsmedien vorbereitet werden. Bevor Sie dies tun, prüfen Sie mit Hilfe dieser Checkliste, ob das System für die Installation bereit ist:

1. **Sichern Sie wichtige Daten**

   Erstellen Sie *immer* eine Sicherung aller wichtigen Daten, *bevor* Sie ein Betriebssystem installieren. Speichern Sie die Daten jedoch nicht auf dem System, auf dem das Betriebssystem installiert wird, sondern nutzen Sie einen Wechseldatenträger, wie beispielsweise ein USB-Laufwerk, oder sichern Sie auf einem anderen System im Netzwerk, oder nutzen einen Online-Backup-Dienst. Überprüfen Sie die Sicherungen, bevor Sie mit der Installation beginnen. Sobald das Installationsprogramm die Festplatte des Systems formatiert, gehen alle gespeicherten Daten unwiderruflich verloren.

2. **Den Installationsort von FreeBSD festlegen**

   Falls FreeBSD das einzige installierte Betriebssystem sein wird, kann dieser Schritt übersprungen werden. Sollte FreeBSD allerdings die Platte mit anderen Betriebssystemen teilen, müssen Sie entscheiden, welche Platte oder Partition für FreeBSD verwendet werden soll.

   Für die Architekturen i386 und amd64 können die Platten in mehrere Partitionen aufgeteilt werden. Dazu stehen Ihnen zwei Partitionsschemas zur Verfügung. Traditionell enthält ein *Master Boot Record* (MBR) eine Partitionstabelle, welche bis zu vier *primäre Partitionen* aufnehmen kann. Aus historischen Gründen werden diese primären Partitionen in FreeBSD *slices* genannt. Eine Begrenzung von nur vier Partitionen ist für große Platten sehr beschränkt, so dass eine dieser primären Partitionen als *erweiterte Partition* eingesetzt wird. Mehrere *logische Partitionen* können dann innerhalb der erweiterten Partition angelegt werden. Die *GUID-Partitionstabelle* (GPT) ist eine neuere und einfachere Methode zur Partition einer Festplatte. Geläufige GPT-Implementierungen erlauben bis zu 128 Partitionen pro Platte, was die Notwendigkeit von logischen Partitionen eliminiert.

   FreeBSDs Standard-Bootloader benötigt entweder eine primäre oder eine GPT-Partition. Wenn alle primären oder GPT-Partitionen bereits in Verwendung sind, muss eine davon für FreeBSD zur Verfügung gestellt werden. Benutzen Sie ein Werkzeug zur Veränderung der Partitionsgrößen, wenn Sie eine Partition erstellen möchten, ohne dabei vorhandene Daten zu löschen. Den freigegebenen Platz können Sie dann für die Installation verwenden.

   Eine Vielzahl freier und kommerzieller Werkzeuge zur Veränderung der Partitionsgrößen finden Sie unter http://en.wikipedia.org/wiki/List_of_disk_partitioning_software. GParted Live (http://gparted.sourceforge.net/livecd.php) ist eine freie Live-CD, die den GParted-Partitionseditor enthält. GParted ist auch in einer Vielzahl von anderen Linux Live-CD Distributionen enthalten.



Bei richtiger Anwendung können Werkzeuge zur Veränderung von Partitionsgrößen auf sichere Art und Weise Platz für eine neue Partition schaffen. Erstellen Sie trotzdem eine Vollsicherung und überprüfen Sie deren Integrität bevor Sie die Partitionen auf der Platte verändern.

Festplattenpartitionen, die unterschiedliche Betriebssysteme enthalten, ermöglichen es, jeweils eines dieser Systeme zu verwenden. Eine alternative Möglichkeit, mehrere Betriebssysteme gleichzeitig einzusetzen, ohne dabei Partitionen ändern zu müssen, wird im Virtualisierung behandelt.

3. **Netzwerkparameter ermitteln**

   Manche FreeBSD Installationsarten benötigen eine Netzwerkverbindung, um Installationsdateien herunter zu laden. Nach jeder Installation bietet das Installationsprogramm die Möglichkeit, die Netzwerkschnittstellen des Systems zu konfigurieren.

   Steht im Netzwerk ein DHCP-Server zur Verfügung, wird dieser im Allgemeinen verwendet, um automatisch Netzwerkeinstellungen vorzunehmen. Falls DHCP nicht verfügbar ist, müssen die folgenden Netzwerkeinstellungen beim lokalen Netzwerkadministrator oder Provider erfragt werden:

   a. IP-Adresse
   b. Subnetz-Maske
   c. IP-Adresse des Default-Gateway
   d. Domänenname des Netzwerks
   e. IP-Adressen der DNS-Server im Netzwerk

4. **Lesen Sie die FreeBSD-Errata**

   Obwohl das FreeBSD Projekt sich bemüht, jede veröffentlichte Version von FreeBSD so stabil wie möglich zu machen, können sich doch gelegentlich Fehler in den Veröffentlichungsprozess einschleichen. In sehr seltenen Fällen betreffen diese Fehler den Installationsvorgang. Wenn diese Probleme entdeckt und behoben sind, werden dazu Hinweise in der FreeBSD Errata (https://www.freebsd.org/releases/12.1r/errata/) auf der FreeBSD Webseite veröffentlicht. Prüfen Sie die Errata vor der Installation, um sicherzustellen, dass es keine Probleme gibt, welche die Installation betreffen.

   Informationen und Errata für all diese Veröffentlichungen finden Sie unter den Release Informationen auf der FreeBSD Webseite (https://www.freebsd.org/releases/).

### 4.3.1. Die Installationsmedien vorbereiten

Das FreeBSD-Installationsprogramm ist keine Anwendung, das aus einem anderen Betriebssystem heraus gestartet werden kann. Laden Sie stattdessen eine Installationsdatei für FreeBSD herunter und brennen Sie den Dateityp auf einen entsprechenden Datenträger (CD, DVD, oder USB). Starten Sie dann das System mit diesem Datenträger.

Die FreeBSD-Installationsmedien sind unter www.freebsd.org/where/ verfügbar. Der Name der Installationsdatei enthält die Version von FreeBSD, die Architektur sowie den Dateityp. Wenn Sie beispielsweise FreeBSD 12.1 auf einem amd64-System von DVD installieren wollen, laden Sie FreeBSD-12.1-RELEASE-amd64-dvd1.iso und brennen Sie die Datei auf eine DVD. Starten Sie dann das System mit dieser DVD.

Die Installationsdateien stehen in verschiedenen Formaten zur Verfügung und variieren je nach Rechnerarchitektur und Medientyp.

Für Rechner, die mit UEFI (Unified Extensible Firmware Interface) booten, stehen zusätzliche Installationsdateien zur Verfügung. Die Namen dieser Dateien enthalten die Zeichenkette uefi.

Dateitypen:

- `-bootonly.iso`: Dies ist die kleinste Installation, die lediglich das Installationsprogramm enthält. Hierzu ist während der Installation eine funktionierende Internetverbindung erforderlich, da das Installationsprogramm die benötigen Dateien für die FreeBSD-Installation herunter laden muss. Diese Datei sollte mit einem CD-Brennprogramm auf CD gebrannt werden.
- `-disc1.iso`: Diese Datei enthält alle benötigten Dateien für eine FreeBSD-Installation, den Quellcode und die Ports-Sammlung. Die Datei sollte mit einem CD-Brennprogramm auf CD gebrannt werden.
- `-dvd1.iso`: Diese Datei enthält alle benötigen Dateien für eine FreeBSD-Installation, den Quellcode und die Ports-Sammlung. Darüber hinaus enthält sie eine Reihe von beliebten Binärpaketen zur Installation eines Window-Managers, sodass Sie ein komplettes System installieren können, ohne dass Sie eine Verbindung zum Internet benötigen. Die Datei sollte mit einem DVD-Brennprogramm auf eine DVD gebrannt werden.
- `-memstick.img`: Diese Datei enthält alle benötigten Dateien für eine FreeBSD-Installation, den Quellcode und die Ports-Sammlung. Die Datei sollte mit den nachstehenden Anweisungen auf einen USB-Stick geschrieben werden.
- `-mini-memstick.img`: Diese Datei enthält, wie `-bootonly.iso`, keine Installationsdateien, sondern lädt diese bei Bedarf nach. Während der Installation wird eine funktionierende Internetverbindung benötigt. Schreiben Sie die Datei, wie in Eine Installationsdatei auf einen USB-Stick schreiben beschrieben, auf einen USB-Stick.

Nachdem Sie die Datei heruntergeladen haben, laden Sie CHECKSUM.SHA256 aus dem gleichen Verzeichnis herunter. Berechnen Sie dann die *Prüfsumme* für die Datei. FreeBSD bietet hierfür sha256(1), das Sie als `sha256 Dateiname` aufrufen können. Andere Betriebssysteme haben ähnliche Programme.

Vergleichen Sie die berechnete Prüfsumme mit der in CHECKSUM.SHA256. Die beiden Prüfsummen müssen übereinstimmen, ansonsten ist die Datei beschädigt und muss erneut heruntergeladen werden.

#### Eine Installationsdatei auf einen USB-Stick schreiben

Die *.img-Datei ist ein komplettes *Abbild* (engl. Image) des späteren USB-Sticks. Die Datei kann *nicht* auf das Zielgerät kopiert werden. Es existieren jedoch mehrere Programme, mit denen die *.img -Datei auf einen USB-Stick geschrieben werden kann. In diesem Abschnitt werden zwei dieser

Programme vorgestellt.



Bevor Sie fortfahren, machen Sie Sicherungskopien der Daten auf dem USB-Stick. Diese Prozedur wird alle Daten auf dem Stick löschen.

**Procedure: Das Image mit `dd` auf einen USB-Stick schreiben**



Dieses Beispiel verwendet /dev/da0 als das Zielgerät, auf welches das Image geschrieben werden soll. Seien Sie *sehr vorsichtig*, dass das richtige Gerät benutzt wird, da das Kommando alle vorhandenen Daten auf dem Zielgerät zerstört.

1. Das Werkzeug dd(1) steht unter BSD, Linux® und Mac OS®-Systemen zur Verfügung. Um das Image zu brennen, verbinden Sie den USB-Stick mit dem System und bestimmen Sie dessen Gerätenamen. Geben Sie dann den Namen der Installationsdatei und den Gerätenamen des USB-Sticks an. Dieses Beispiel schreibt die Installation für amd64 auf das erste USB-Gerät im FreeBSD-System.

   ```
   # dd if=FreeBSD-12.1-RELEASE-amd64-memstick.img of=/dev/da0 bs=1M conv=sync
   ```

   Wenn dieser Befehl fehlschlägt, stellen Sie sicher, dass der USB-Stick nicht eingehangen ist und prüfen Sie den Gerätenamen. Auf einigen Systemen muss der Befehl vielleicht mit Hilfe von sudo(8) ausgeführt werden. Die Syntax von dd(1) variiert leicht zwischen verschiedenen Plattformen. Zum Beispiel erfordert Mac OS® ein kleingeschriebenes `bs=1m`. Einige Systeme wie Linux® verwenden vielleicht einen Puffer. Verwenden Sie dann sync(8), um die Daten zu schreiben.

**Procedure: Das Image unter Windows® schreiben**



Versichern Sie sich, dass Sie den korrekten Laufwerksbuchstaben angeben, da die bestehenden Daten des Laufwerks überschrieben und zerstört werden.

1. Image Writer für Windows® herunterladen

   Image Writer für Windows® ist eine frei verfügbare Anwendung, welche eine Imagedatei korrekt auf einen USB-Stick schreiben kann. Laden Sie diese von https://sourceforge.net/projects/win32diskimager/ herunter und entpacken Sie sie in ein Verzeichnis.

2. Das Image mit Image Writer auf den USB-Stick schreiben

   Klicken Sie doppelt auf das Win32DiskImager-Icon, um das Programm zu starten. Prüfen Sie dabei, dass der Laufwerksbuchstabe unter `Device` dem Gerät entspricht, in dem sich der USB-Stick befindet. Klicken Sie auf das Ordnersymbol und wählen Sie das Image aus,

welches auf den USB-Stick geschrieben werden soll. Um den Image-Dateinamen zu akzeptieren, klicken Sie auf **[ Save ]**. Überprüfen Sie erneut, ob alles stimmt und dass keine Ordner auf dem USB-Stick in anderen Fenstern geöffnet sind. Sobald alles bereit ist, klicken Sie auf **[ Write ]**, um die Imagedatei auf den USB-Stick zu schreiben.

Sie sind jetzt dazu bereit, mit der Installation von FreeBSD zu beginnen.

# 4.4. Die Installation starten



Es werden bei Installation so lange keine Änderungen an den Festplatten durchgeführt, bis die folgende Meldung erscheint:

```
Your changes will now be written to disk. If you
have chosen to overwrite existing data, it will
be PERMANENTLY ERASED. Are you sure you want to
commit your changes?
```

Die Installation kann vor dieser Warnung zu jeder Zeit abgebrochen werden. Falls Zweifel bestehen, dass etwas falsch konfiguriert wurde, schalten Sie einfach den Computer vor diesem Punkt aus und es werden keine Änderungen an der Festplatte vorgenommen.

Dieser Abschnitt beschreibt, wie das System vom Installationsmedium, das nach den Anweisungen in Die Installationsmedien vorbereiten erstellt wurde, gebootet wird. Wenn Sie einen bootfähigen USB-Stick einsetzen, verbinden Sie diesen mit dem System, bevor Sie den Computer einschalten. Falls die Installation von einer CD startet, müssen Sie den Computer einschalten und die CD so bald wie möglich einlegen. Wie das System konfiguriert werden muss, um von dem verwendeten Installationsmedium zu booten, hängt von der Architektur ab.

## 4.4.1. Systemstart von i386™ und amd64

Diese Architekturen beinhalten ein BIOS-Menü zur Auswahl des Boot-Gerätes. Abhängig von dem verwendeten Installationsmedium können Sie CD/DVD oder USB als erstes Boot-Gerät auswählen. Die meisten Systeme erlauben es auch, das Boot-Gerät während des Startvorgangs zu wählen, typischerweise durch drücken von `F10`, `F11`, `F12` oder `Esc`.

Falls der Computer wie normal startet und das bestehende Betriebssystem lädt, befolgen Sie einen der hier aufgeführten Schritte:

1. Das Installationsmedium wurde während des Startvorgangs nicht früh genug eingelegt. Lassen Sie das Medium eingelegt und versuchen Sie, den Rechner neu zu starten.
2. Die Änderungen am BIOS waren nicht richtig oder wurden nicht gespeichert. Überprüfen Sie, dass das richtige Boot-Gerät als erstes Boot-Gerät ausgewählt ist.
3. Das verwendete System ist zu alt und unterstützt das starten vom gewählten Medium nicht. In diesem Fall kann der Plop Boot Manager (http://www.plop.at/de/bootmanagers.html) verwendet werden, um ältere Computer von CD oder USB-Medien zu starten.

### 4.4.2. Systemstart beim PowerPC®

Auf den meisten Maschinen können Sie `C` auf der Tastatur gedrückt halten, um von der CD zu starten. Andernfalls, halten Sie `Command` + `Option` + `O` + `F`, oder `Windows` + `Alt` + `O` + `F` auf nicht-Apple® Tastaturen gedrückt. Geben Sie an der `0 >`-Eingabeaufforderung folgendes ein:

```
boot cd:,\ppc\loader cd:0
```

### 4.4.3. FreeBSD Bootmenü

Wenn das System vom Installationsmedium gestartet wird, erscheint folgendes Menü auf dem Bildschirm:




*Abbildung 1. FreeBSD Boot Loader Menü*

In der Voreinstellung wird das Menü zehn Sekunden auf Benutzereingaben warten, bevor das Installationsprogramm gestartet wird. Drücken Sie die Leertaste, um den Timer anzuhalten. Um eine Option auszuwählen, drücken Sie die entsprechende Nummer bzw. Buchstaben. Die folgenden Optionen stehen zur Verfügung.

- `Boot Multi User`: Dies wird den Boot-Prozess von FreeBSD fortsetzen. Wenn der Timer angehalten wurde, drücken Sie entweder die `1`, `B`, oder `Enter`.
- `Boot Single User`: Dieser Modus kann verwendet werden, um eine bestehende FreeBSD-Installation zu reparieren. Dies wird in "Der Single-User Modus" beschrieben. Drücken Sie die `2` oder `S` um in diesen Modus zu gelangen.
- `Escape to loader prompt`: Dieser Modus startet einen Prompt, welcher nur eine begrenzte Anzahl an Low-Level-Befehlen enthält. Dies wird in "Phase Drei" beschrieben. Drücken Sie die `3` oder `Esc` um in diesen Modus zu gelangen.
- `Reboot`: Startet das System neu.

- `Kernel`: Lädt einen anderen Kernel.
- `Configure Boot Options`: Öffnet das Menü, welches in FreeBSD Boot-Optionen Menü beschrieben ist.




*Abbildung 2. FreeBSD Boot-Optionen Menü*

Das Boot-Optionen Menü ist in zwei Abschnitte unterteilt. Der erste Abschnitt wird verwendet, um zurück zum Hauptmenü zu gelangen, oder um Optionen zurück auf die Standardwerte zu setzen.

Im zweiten Abschnitt können verschiedene Optionen auf `On` oder `Off` gesetzt werden. Das System wird bei einem Neustart immer mit den Einstellungen für diese Optionen booten:

- `ACPI Support`: Wenn das System während des Bootens hängt, setzen Sie diese Option auf `Off`.
- `Safe Mode`: Wenn das System trotz deaktiviertem `ACPI Support` immer noch hängt, setzen Sie diese Option auf `On`.
- `Single User`: Setzen Sie die Option auf `On`, um eine bestehende FreeBSD-Installation zu reparieren. Dieser Prozess wird in "Der Single-User Modus" beschrieben. Sobald das Problem behoben ist, setzen Sie die Option wieder auf `Off`.
- `Verbose`: Wenn Sie während des Bootens ausführliche Meldungen sehen möchten, zum Beispiel für die Fehlersuche bei Hardwareproblemen, setzen Sie diese Option auf `On`.

Nachdem Sie die benötigten Auswahlen getroffen haben, drücken Sie die `1` oder die Rücktaste, um zum Hauptmenü zurückzukehren. Drücken Sie dann `Enter` um den FreeBSD Bootprozess fortzusetzen. Eine Reihe von Boot-Meldungen werden nun im Rahmen der Geräteerkennung von FreeBSD angezeigt. Sobald dieser Prozess abgeschlossen ist, erscheint das Menü aus Willkommen-Menü.




*Abbildung 3. Willkommen-Menü*

Wählen Sie hier **[ Install ]** und drücken Sie `Enter`, um in das Installationsprogramm zu gelangen. Der Rest dieses Kapitels beschreibt das Installationsprogramm. Andernfalls verwenden Sie die Pfeiltasten um einen anderen Menüpunkt auszuwählen. **[ Shell ]** kann verwendet werden, um eine Shell zu starten und Zugriff auf die Kommandozeilenprogramme zu erhalten, damit beispielsweise die Platten vor der Installation vorbereitet werden können. **[ Live CD ]** kann verwendet werden um FreeBSD vor der Installation auszuprobieren. Die Live-Version wird in Verwendung der Live-CD beschrieben.



Um sich die Boot-Meldungen und die Ergebnisse der Geräteerkennung erneut anzeigen zu lassen, drücken Sie `S` gefolgt von `Enter`. Dadurch wird eine Shell gestartet, in der Sie die Ereignisse seitenweise mit `more /var/run/dmesg.boot` lesen können. Geben Sie `exit` ein, um zum Willkommen-Menü zurückzukehren.

# 4.5. Verwendung von bsdinstall

Dieser Abschnitt zeigt die Reihenfolge der Menüs von bsdinstall sowie die Informationen, die während der Installation abgefragt werden. Benutzen Sie die Pfeiltasten zur Navigation und die Leertaste, um einen Menüpunkt zu aktivieren oder zu deaktivieren. Wenn Sie fertig sind, drücken Sie `Enter`, um die Auswahl zu speichern und zum nächsten Bildschirm zu gelangen.

## 4.5.1. Die Tastaturbelegung auswählen

Bevor die Installation gestartet wird, lädt bsdinstall die Tastaturbelegung, wie in Laden der Tastaturbelegung gezeigt.

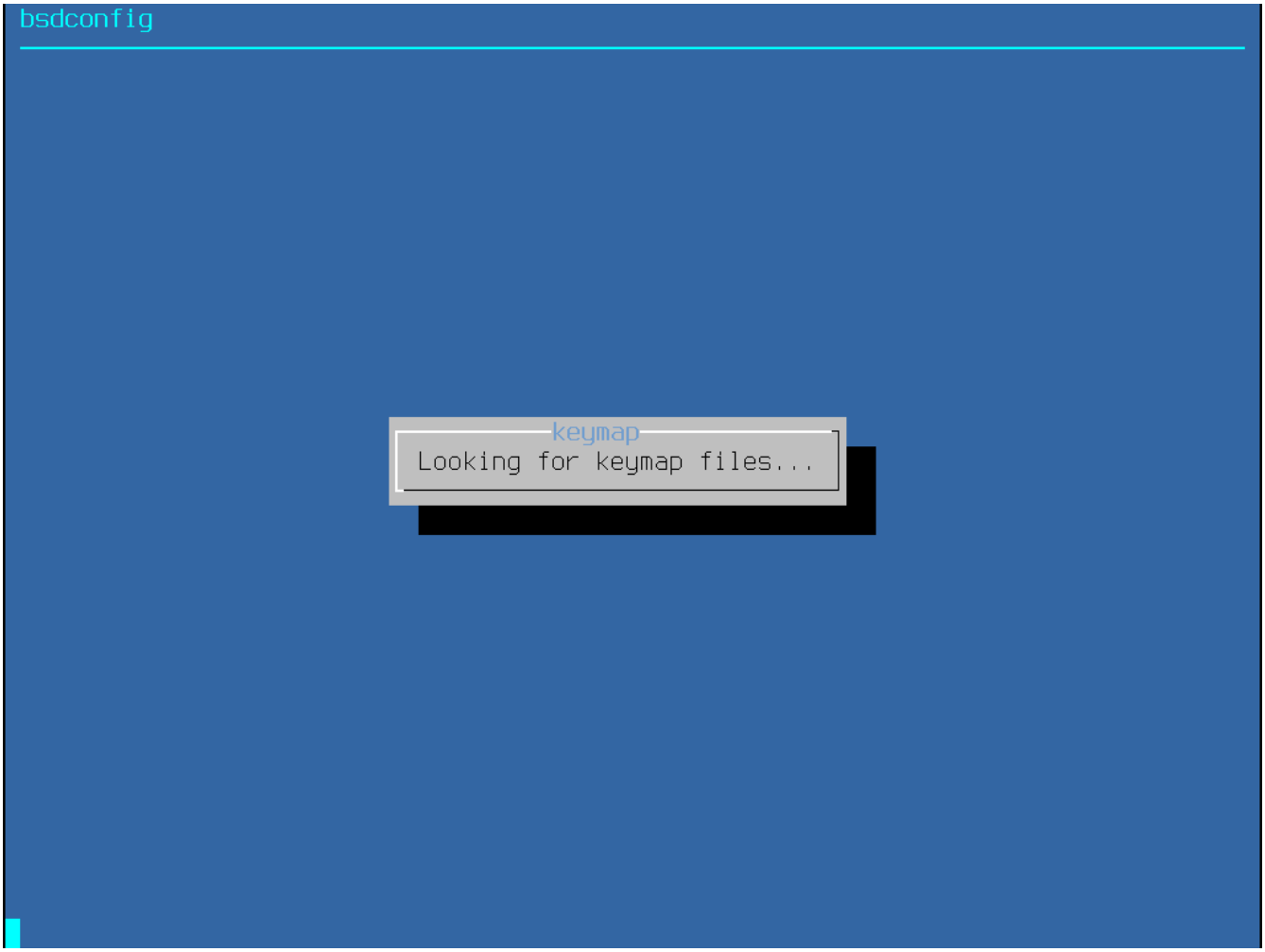



*Abbildung 4. Laden der Tastaturbelegung*

Nachdem die Tastaturbelegung geladen wurde, zeigt bsdinstall das Menü aus Bildschirm zur Auswahl der Tastaturbelegung an. Wählen Sie die Tastenbelegung, die der am System angeschlossenen Tastatur am nächsten kommt, indem Sie die Pfeiltasten Hoch/Runter verwenden und anschließend `Enter` drücken.

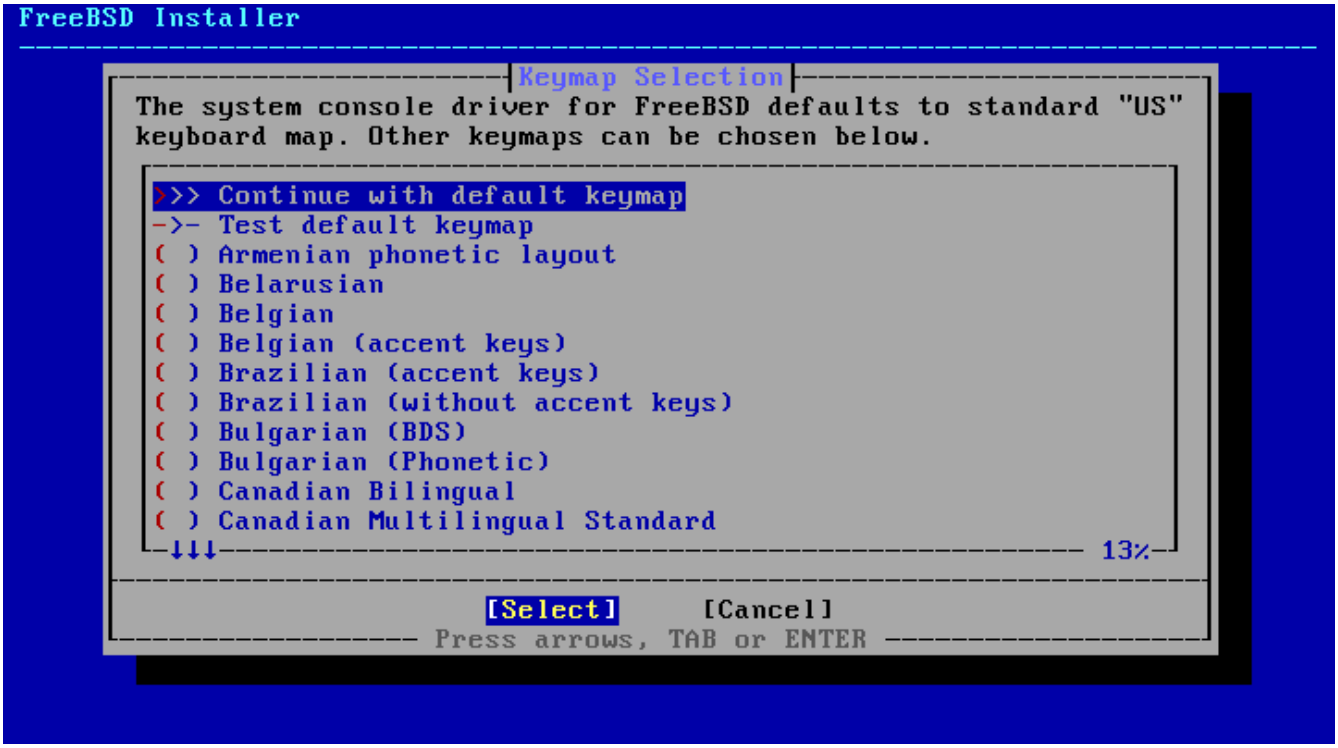



*Abbildung 5. Bildschirm zur Auswahl der Tastaturbelegung*



Durch drücken von `Esc` wird das Menü verlassen und die Standardbelegung eingestellt. United States of America ISO-8859-1 ist eine sichere Option, falls Sie sich unsicher sind, welche Auswahl Sie treffen sollen.

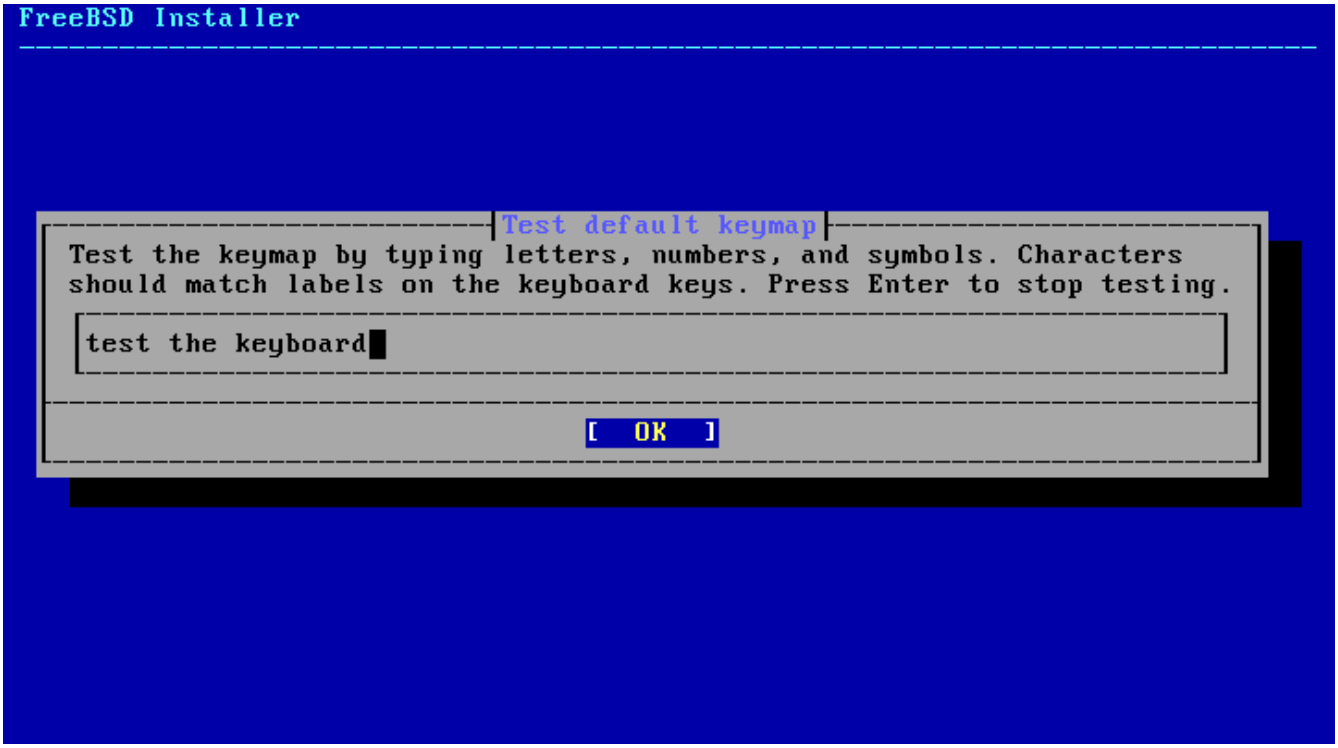



*Abbildung 6. Bildschirm zum Testen der Tastaturbelegung*

### 4.5.2. Den Rechnernamen festlegen

Das nächste bsdinstall-Menü konfiguriert den Rechnernamen, der für das neu zu installierende System verwendet werden soll.

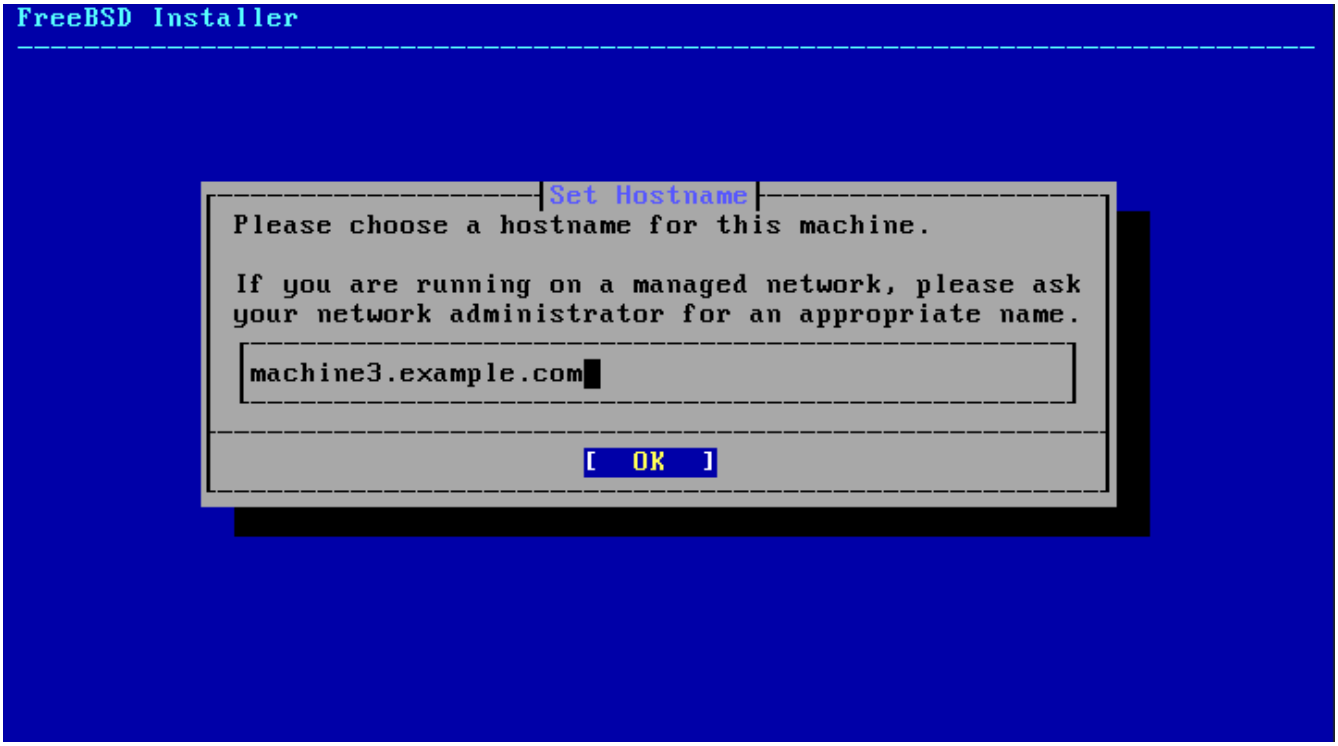



*Abbildung 7. Festlegen des Rechnernamens*

Geben Sie einen für das Netzwerk eindeutigen Rechnernamen an. Der eingegebene Rechnername sollte ein voll-qualifizierter Rechnername sein, so wie z.B. `machine3.example.com`.

### 4.5.3. Auswahl der zu installierenden Komponenten

Im nächsten Schritt fragt Sie bsdinstall, die optionalen Komponenten für die Installation auszuwählen.

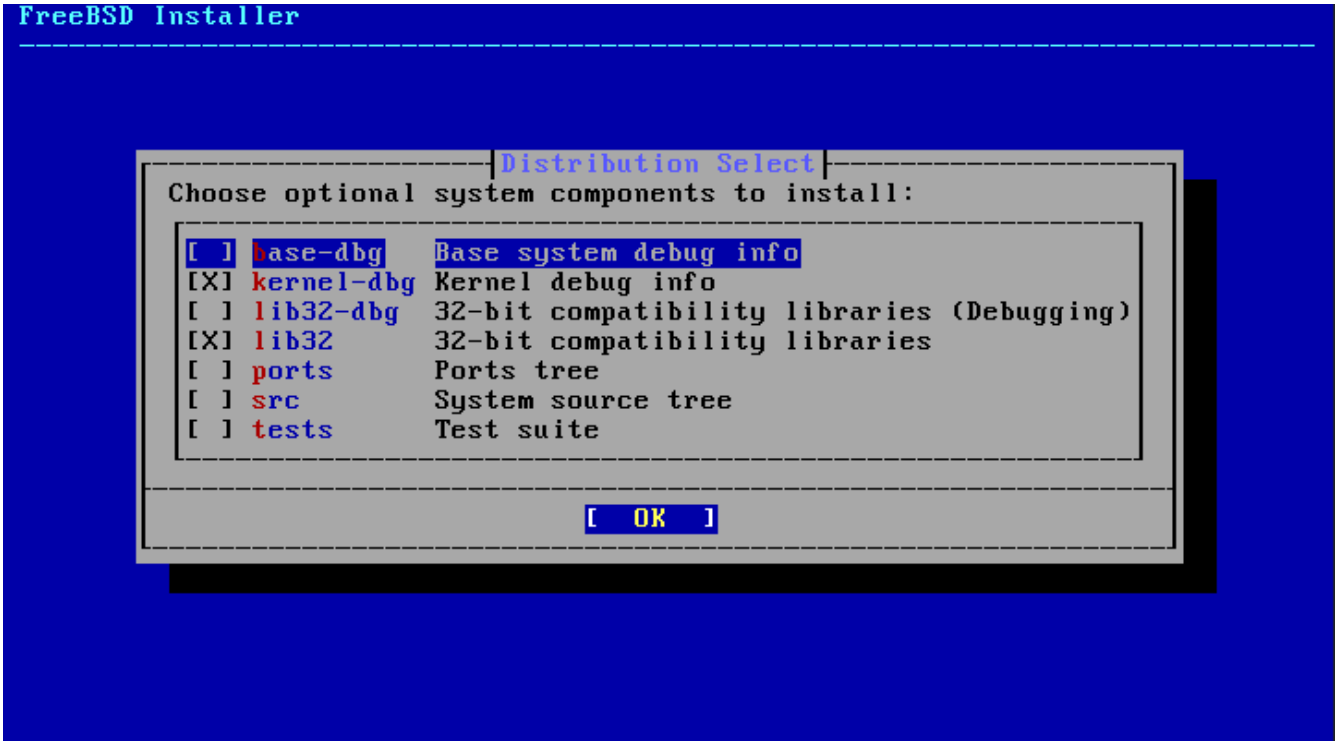



*Abbildung 8. Komponenten für die Installation auswählen*

Die Entscheidung, welche Komponenten auszuwählen sind, hängt größtenteils davon ab, für was

das System künftig eingesetzt werden soll und der zur Verfügung stehende Plattenplatz. Der FreeBSD-Kernel und die Systemprogramme (zusammengenommen auch als *Basissystem* bezeichnet) werden immer installiert. Abhängig vom Typ der Installation, werden manche dieser Komponenten nicht erscheinen.

- `base-dbg` - Basiswerkzeuge wie cat, ls und vielte weitere mit aktiviertem Debugging.
- `kernel-dbg` - Kernel und Module mit aktiviertem Debugging.
- `lib32-dbg` - Kompatibilitäts-Bibliotheken mit aktiviertem Debugging, für die Ausführung von 32-bit-Anwendungen auf einer 64-bit-Version von FreeBSD.
- `lib32` - Kompatibilitäts-Bibliotheken, um 32-bit-Anwendungen auf der 64-bit Version von FreeBSD laufen zu lassen.
- `ports` - Die FreeBSD Ports-Sammlung ist eine Sammlung von Dateien, die das herunterladen, erstellen und installieren von Drittanbietersoftware automatisiert. Installieren von Anwendungen: Pakete und Ports behandelt die Verwendung der Ports-Sammlung.



Das Installationsprogramm prüft nicht, ob genügend Plattenplatz zur Verfügung steht. Wählen Sie diese Option nur, wenn die Festplatte über ausreichend Speicher verfügt. Die Ports-Sammlung nimmt etwa 3 GB Plattenplatz ein.

- `src` - Der vollständige FreeBSD Quellcode für den Kernel und die Systemprogramme. Obwohl dies für die meisten Anwendungen nicht benötigt wird, kann es doch für manche Gerätetreiber, Kernelmodule und einigen Anwendungen aus der Ports-Sammlung erforderlich sein. Der Quellcode wird auch benötigt um an FreeBSD selbst mitzuentwickeln. Der komplette Quellcodebaum benötigt 1 GB Plattenplatz und um das gesamte Betriebssystem neu zu erstellen, werden zusätzliche 5 GB Platz benötigt.
- `tests` - FreeBSD Test-Suite.

### 4.5.4. Installation aus dem Netzwerk

Das Menü in Installation über das Netzwerk erscheint nur bei der Installation von einer -bootonly.iso-CD, da dieses Installationsmedium keine Kopien der Installationsdateien enthält. Da die Installationsdateien über eine Netzwerkverbindung abgerufen werden müssen, weist dieses Menü darauf hin, dass zunächst die Netzwerkschnittstelle konfiguriert werden muss. Falls dieses Menü während der Installation angezeigt wird, befolgen Sie die Anweisungen in Die Netzwerkschnittstelle konfigurieren.

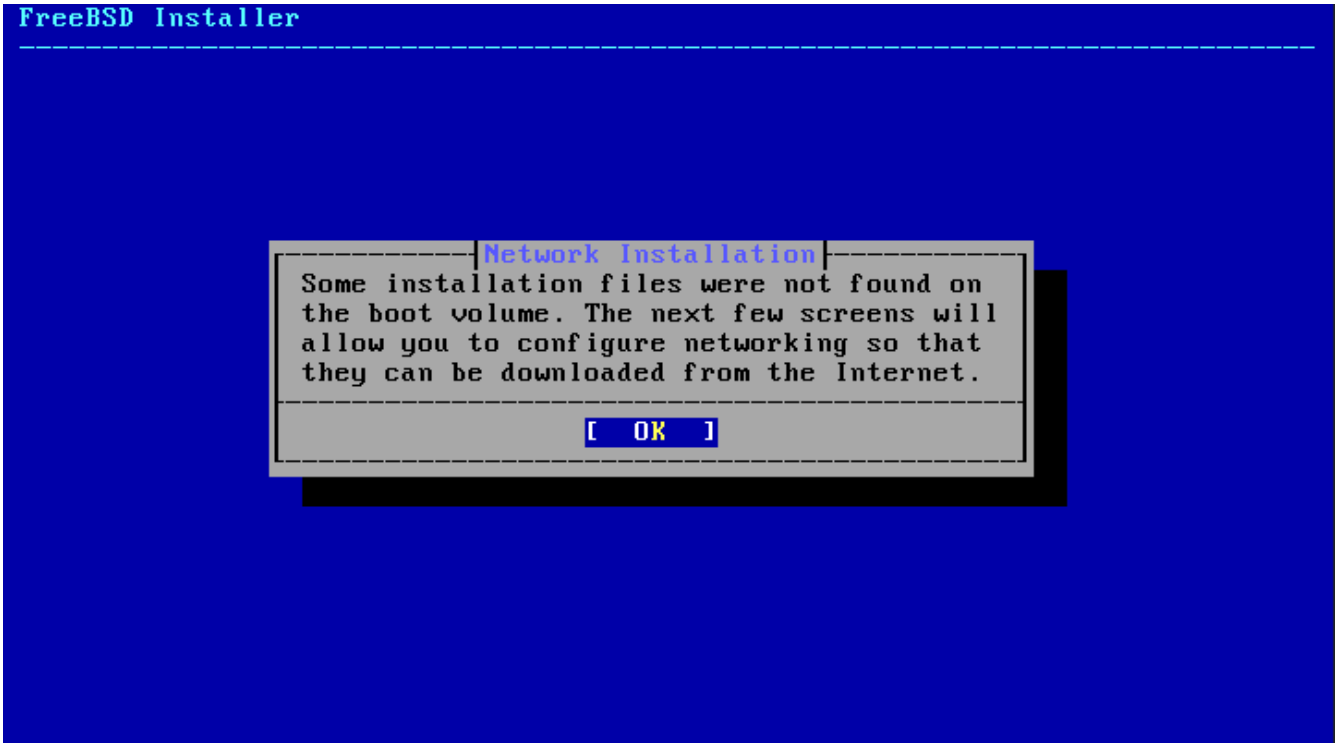



*Abbildung 9. Installation über das Netzwerk*

## 4.6. Plattenplatz bereitstellen

Im nächsten Menü wird die Methode bestimmt, um den Plattenplatz zuzuweisen.

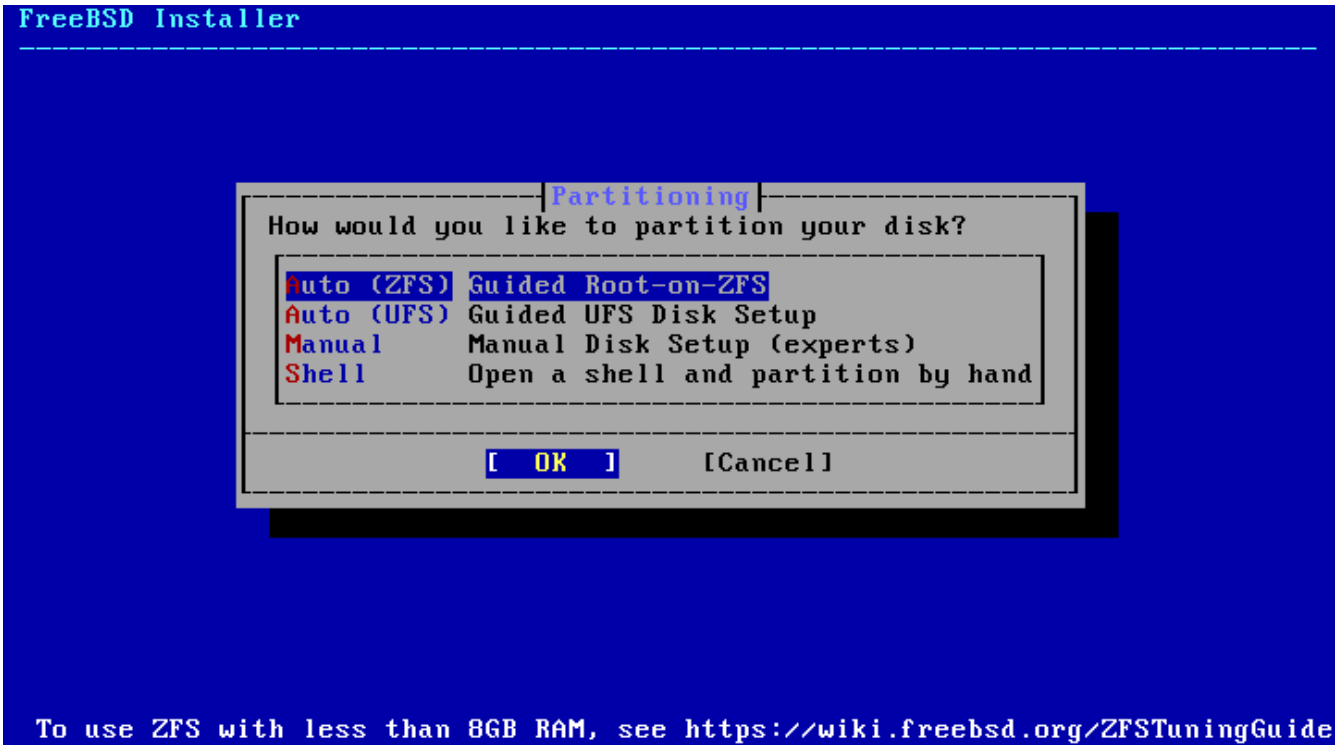



*Abbildung 10. Partitionierung unter FreeBSD*

bsdinstall bietet dem Benutzer vier Methoden zur Zuweisung von Plattenplatz:

- `Auto (UFS)` richtet die Partitionen automatisch mit dem `UFS`-Dateisystems ein.
- `Manual` ermöglicht es fortgeschrittenen Benutzern, angepasste Partitionen über Menüoptionen

zu erstellen.

- `Shell` öffnet eine Eingabeaufforderung, in der fortgeschrittene Benutzer angepasste Partitionen mit Werkzeugen wie gpart(8), fdisk(8) und bsdlabel(8) erstellen können.
- `Auto (ZFS)` erzeugt ein root-on-ZFS-System mit optionaler GELI-Verschlüsselung für Boot Environments.

Dieser Abschnitt beschreibt, was bei der Partitionierung der Platten zu beachten ist und wie die einzelnen Methoden zur Partitionierung angewendet werden.

### 4.6.1. Ein Partitionslayout entwerfen

Wenn Sie Dateisysteme anlegen, sollten Sie beachten, dass Festplatten auf Daten in den äußeren Spuren schneller zugreifen können als auf Daten in den inneren Spuren. Daher sollten die kleineren und oft benutzten Dateisysteme an den äußeren Rand der Platte gelegt werden. Die größeren Partitionen wie /usr sollten in die inneren Bereiche gelegt werden. Es empfiehlt sich, die Partitionen in folgender Reihenfolge anzulegen: /, swap, /var und /usr.

Die Größe der /var-Partition ist abhängig vom Zweck der Maschine. Diese Partition enthält hauptsächlich Postfächer, Logdateien und Druckwarteschlangen. Abhängig von der Anzahl an Systembenutzern und der Aufbewahrungszeit für Logdateien, können Postfächer und Logdateien unerwartete Größen annehmen. Die meisten Benutzer benötigen nur selten mehr als ein Gigabyte für /var.



Ein paar Mal wird es vorkommen, dass viel Festplattenspeicher in /var/tmp benötigt wird. Wenn neue Software mit pkg_add(1) installiert wird, extrahieren die Paketwerkzeuge eine vorübergehende Kopie der Pakete unter /var/tmp. Die Installation großer Softwarepakete wie Firefox oder LibreOffice kann sich wegen zu wenig Speicherplatz in /var/tmp als trickreich herausstellen.

Die /usr Partition enthält viele der Hauptbestandteile des Systems, einschließlich der FreeBSD Ports-Sammlung und den Quellcode des Systems. Für diese Partition werden mindestens zwei Gigabyte empfohlen.

Behalten Sie bei der Auswahl der Partitionsgrößen den Platzbedarf im Auge. Wenn Sie den Platz auf einer Partition vollständig aufgebraucht haben, eine andere Partition aber kaum benutzen, kann die Handhabung des Systems schwierig werden.

Als Daumenregel sollten Sie doppelt soviel Speicher für die Swap-Partition vorsehen, als Sie Hauptspeicher haben, da die VM-Paging-Algorithmen im Kernel so eingestellt sind, dass sie am besten laufen, wenn die Swap-Partition mindestens doppelt so groß wie der Hauptspeicher ist. Zu wenig Swap kann zu einer Leistungsverminderung im VM page scanning Code führen, sowie Probleme verursachen, wenn später mehr Speicher in die Maschine eingebaut wird.

Auf größeren Systemen mit mehreren SCSI-, oder IDE-Laufwerken an unterschiedlichen Controllern, wird empfohlen, Swap-Bereiche auf bis zu vier Laufwerken einzurichten. Diese Swap-Partitionen sollten ungefähr dieselbe Größe haben. Der Kernel kann zwar mit beliebigen Größen umgehen, aber die internen Datenstrukturen skalieren bis zur vierfachen Größe der größten Partition. Ungefähr gleich große Swap-Partitionen erlauben es dem Kernel, den Swap-Bereich

optimal über die Laufwerke zu verteilen. Große Swap-Bereiche, auch wenn sie nicht oft gebraucht werden, sind nützlich, da sich ein speicherfressendes Programm unter Umständen auch ohne einen Neustart des Systems beenden lässt.

Indem Sie ein System richtig partitionieren, verhindern Sie, dass eine Fragmentierung in den häufig beschriebenen Partitionen auf die meist nur gelesenen Partitionen übergreift. Wenn Sie die häufig beschriebenen Partitionen an den Rand der Platte legen, dann wird die I/O-Leistung dieser Partitionen steigen. Die I/O-Leistung ist natürlich auch für große Partitionen wichtig, doch erzielen Sie eine größere Leistungssteigerung, wenn Sie /var an den Rand der Platte legen.

### 4.6.2. Geführte Partitionierung für UFS

Bei dieser Methode wird ein Menü die verfügbaren Platten anzeigen. Sollten mehrere Platten angeschlossen sein, wählen Sie diejenige aus, auf der FreeBSD installiert werden soll.

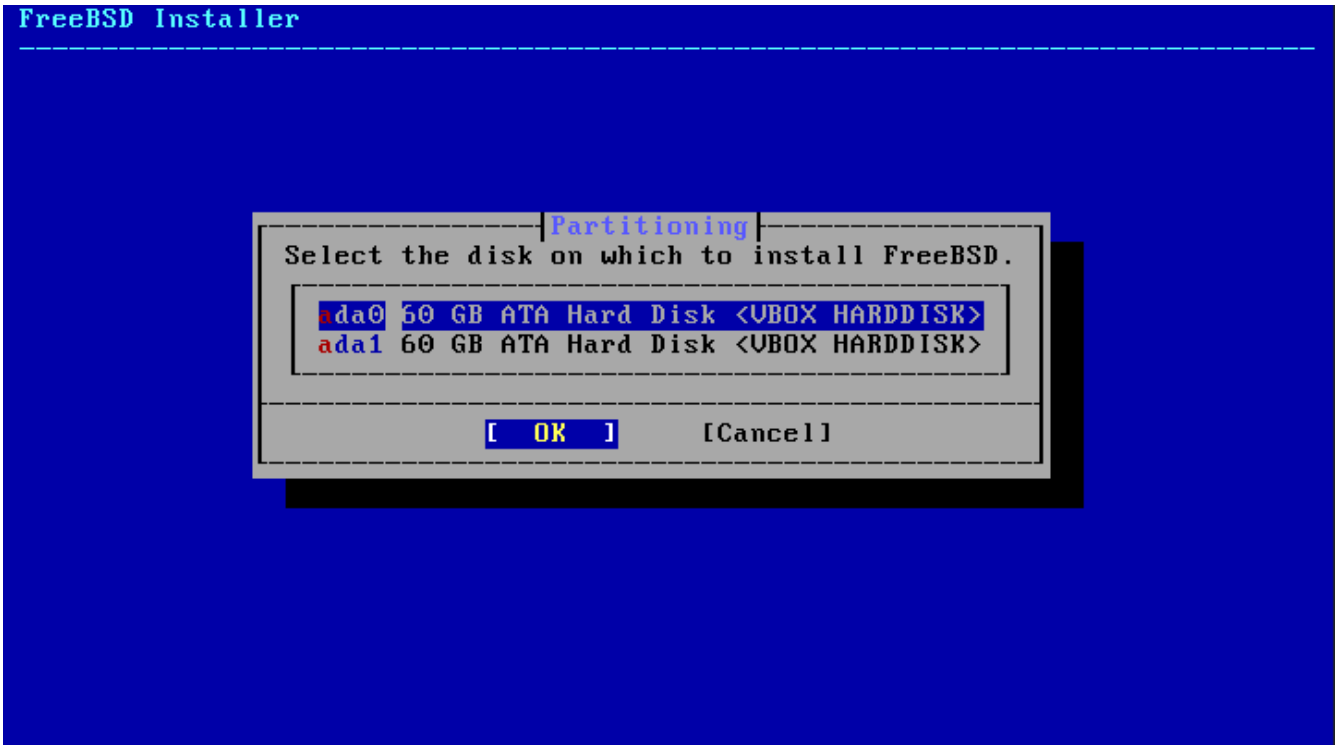



*Abbildung 11. Aus mehreren Platten eine auswählen*

Nachdem Sie die Platte ausgewählt haben, fordert das nächste Menü dazu auf, entweder die gesamte Festplatte für die Installation zu nutzen oder eine Partition aus unbenutzten Speicherplatz zu erstellen. Ein allgemeines Partitionslayout, das die gesamte Platte einnimmt wird erstellt, wenn **[ Entire Disk ]** ausgewählt wird. Durch die Wahl von **[ Partition ]** wird ein Partitionslayout aus dem unbenutzten Speicherplatz der Platte erstellt.




*Abbildung 12. Auswahl der gesamten Platte oder einer Partition*

Wenn **[ Entire Disk ]** gewählt wurde, weist bsdinstall darauf hin, dass die Festplatte gelöscht wird.

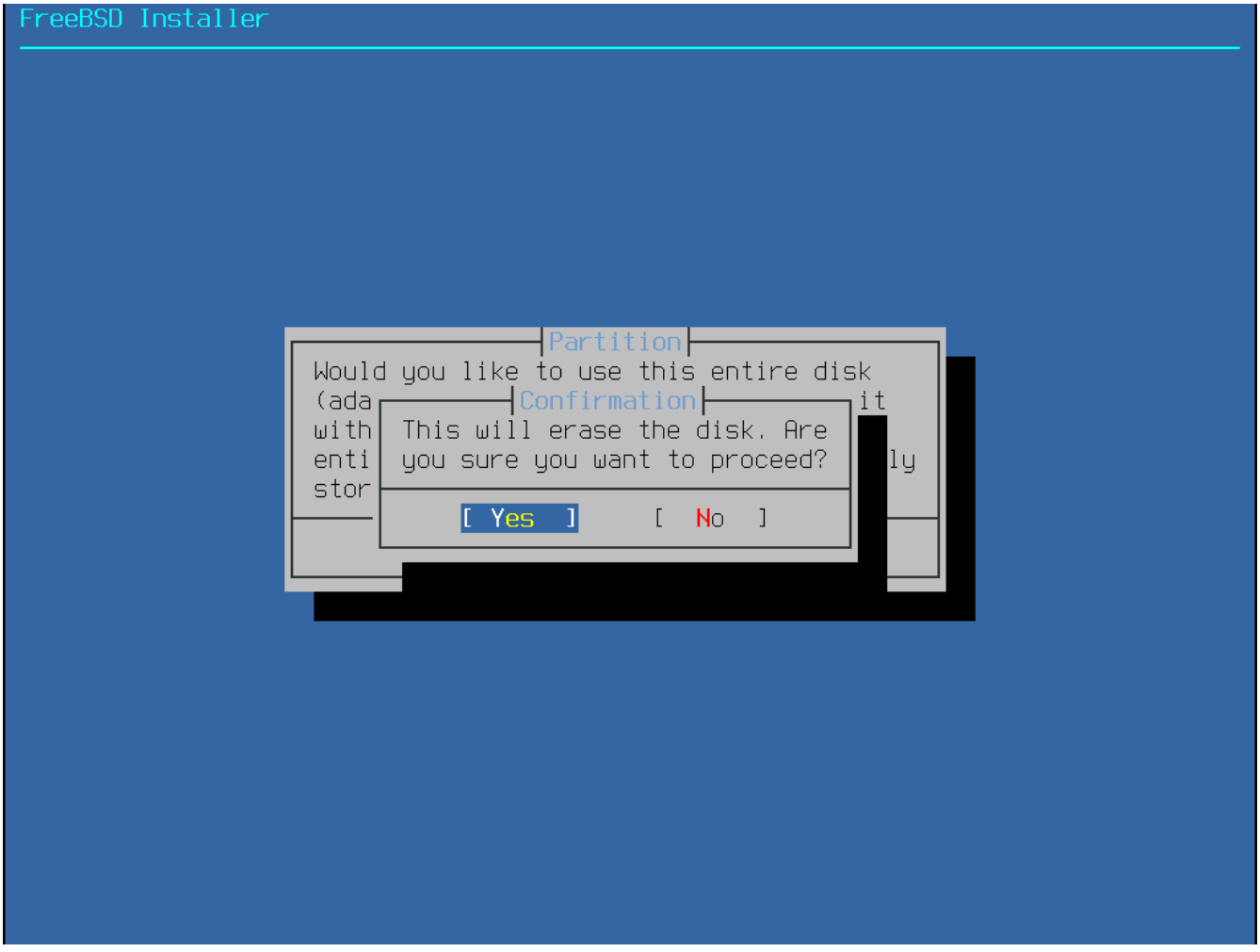
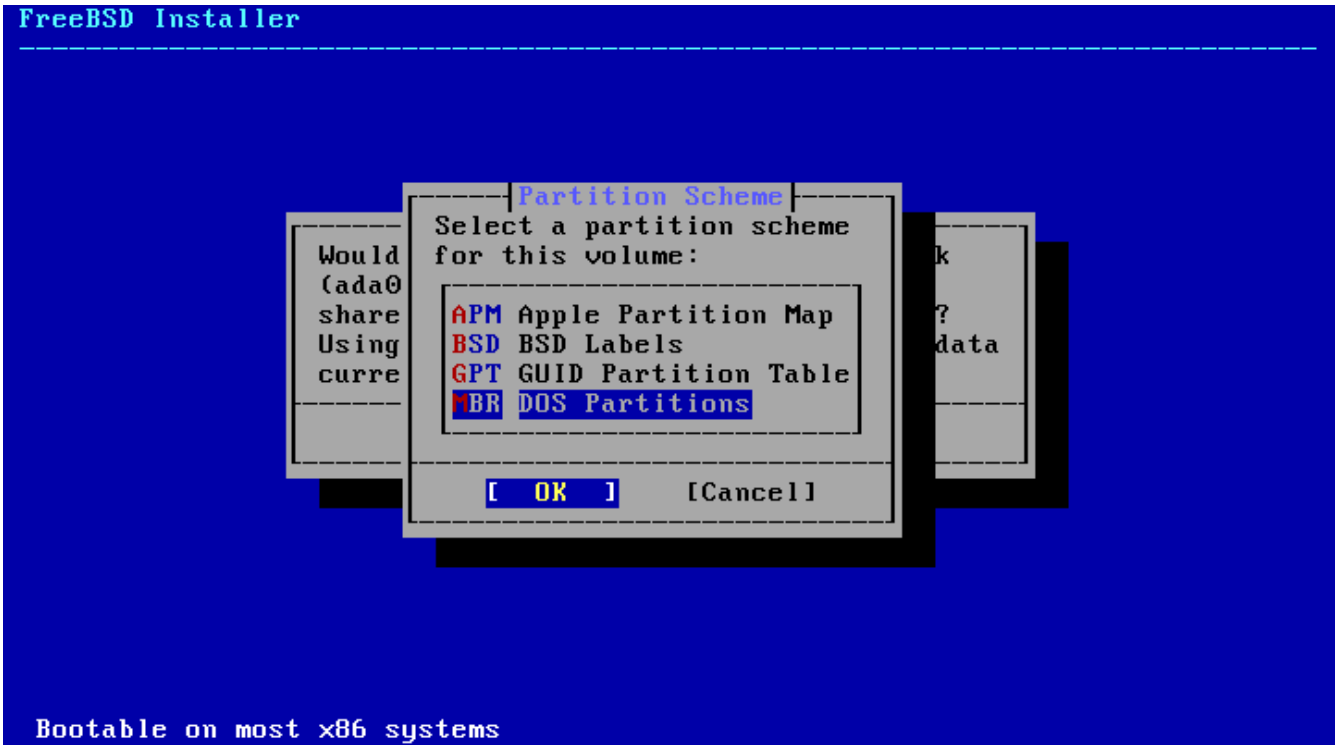



*Abbildung 13. Bestätigung*

Das nächste Menü zeigt eine Liste der verfügbaren Partitionsschemas. GPT ist ist normalerweise die

geeignetste Wahl für amd64-Rechner. Ältere Rechner, die nicht mit GPT kompatibel sind, sollten MBR benutzen. Die anderen Partitionsschemas werden im Allgemeinen für ungewöhnliche oder ältere Rechner benutzt. Weitere Informationen finden Sie in Partitionierungsschemas.




Nachdem das Partitionslayout nun erstellt wurde, sollten Sie es überprüfen, um sicherzustellen, dass es die Bedürfnisse der Installation erfüllt. Durch die Auswahl von **[ Revert ]** können die Partitionen wieder auf den ursprünglichen Wert zurückgesetzt werden und durch **[ Auto ]** werden die automatischen FreeBSD Partitionen wiederhergestellt. Partitionen können auch manuell erstellt, geändert oder gelöscht werden. Sollte die Partitionierung richtig sein, wählen Sie **[ Finish ]** aus, um mit der Installation fortzufahren.

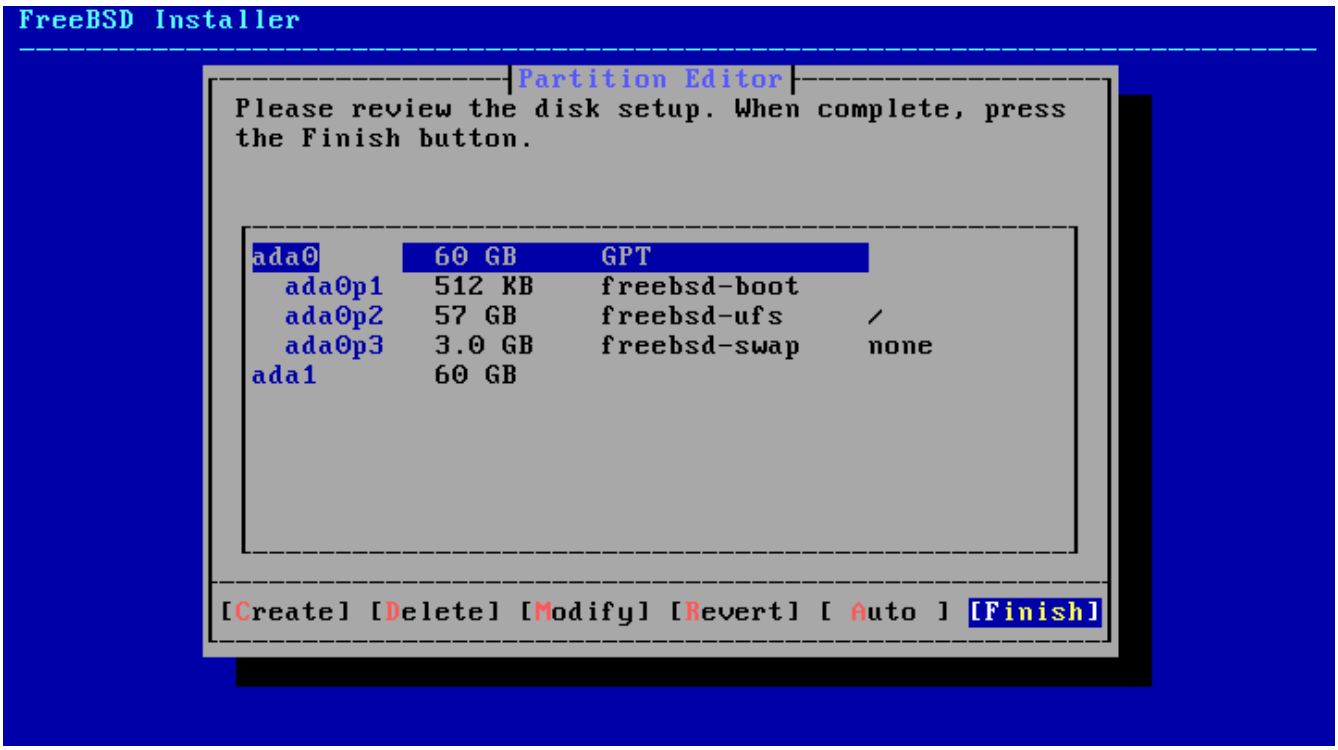



*Abbildung 14. Überprüfen der erstellten Partitionen*

Sobald die Festplatten konfiguriert sind, bietet das nächste Menü die letzte Möglichkeit, Änderungen vorzunehmen, bevor die ausgewählten Laufwerke formatiert werden. Wenn Änderungen vorgenommen werden müssen, wählen Sie **[ Back ]**, um zum Hauptmenü zurückzukehren. Mit **[ Revert & Exit ]** wird das Installationsprogramm beendet, ohne Änderungen am Laufwerk vorzunehmen. Wählen Sie **[ Commit ]**, um die Installation zu starten.

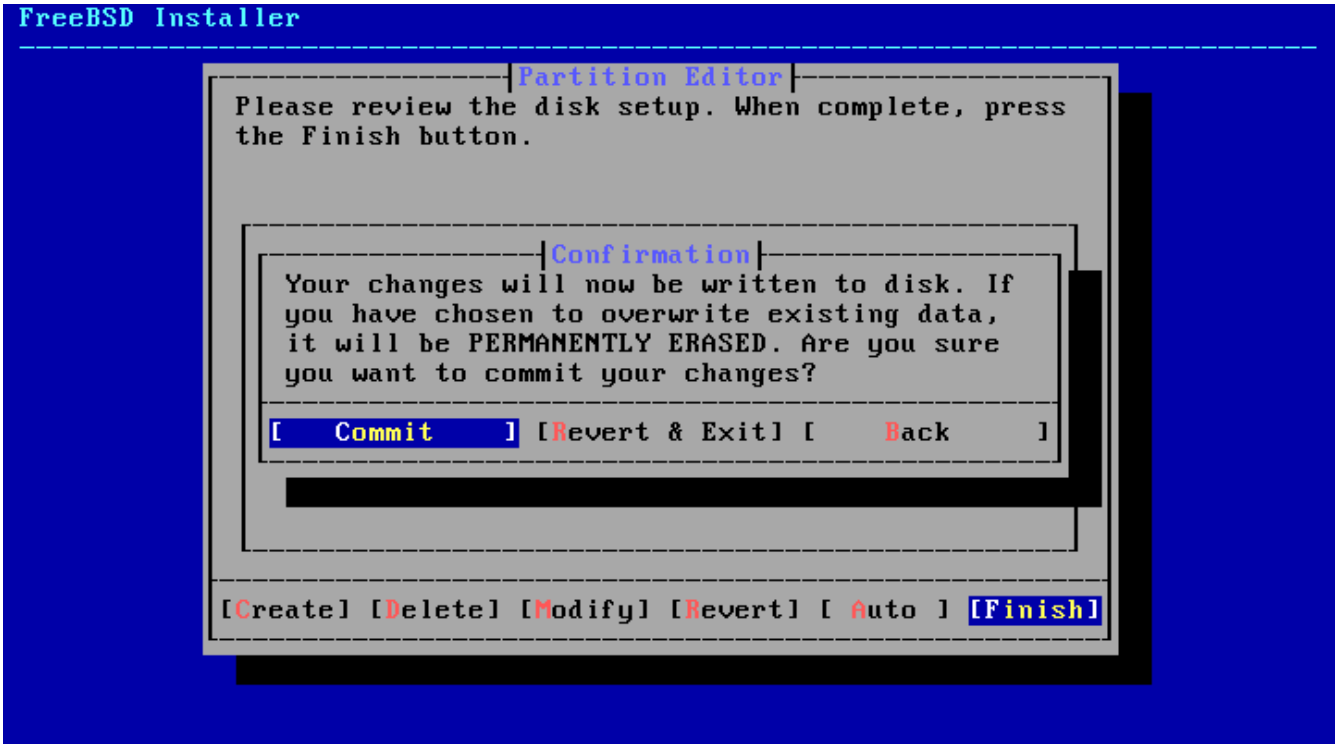



*Abbildung 15. Abschließende Konfiguration*

Um mit der Installation fortzufahren, gehen Sie zu Abrufen der Distributionen.

### 4.6.3. Manuelle Partitionierung

Diese Methode öffnet den Partitionseditor:

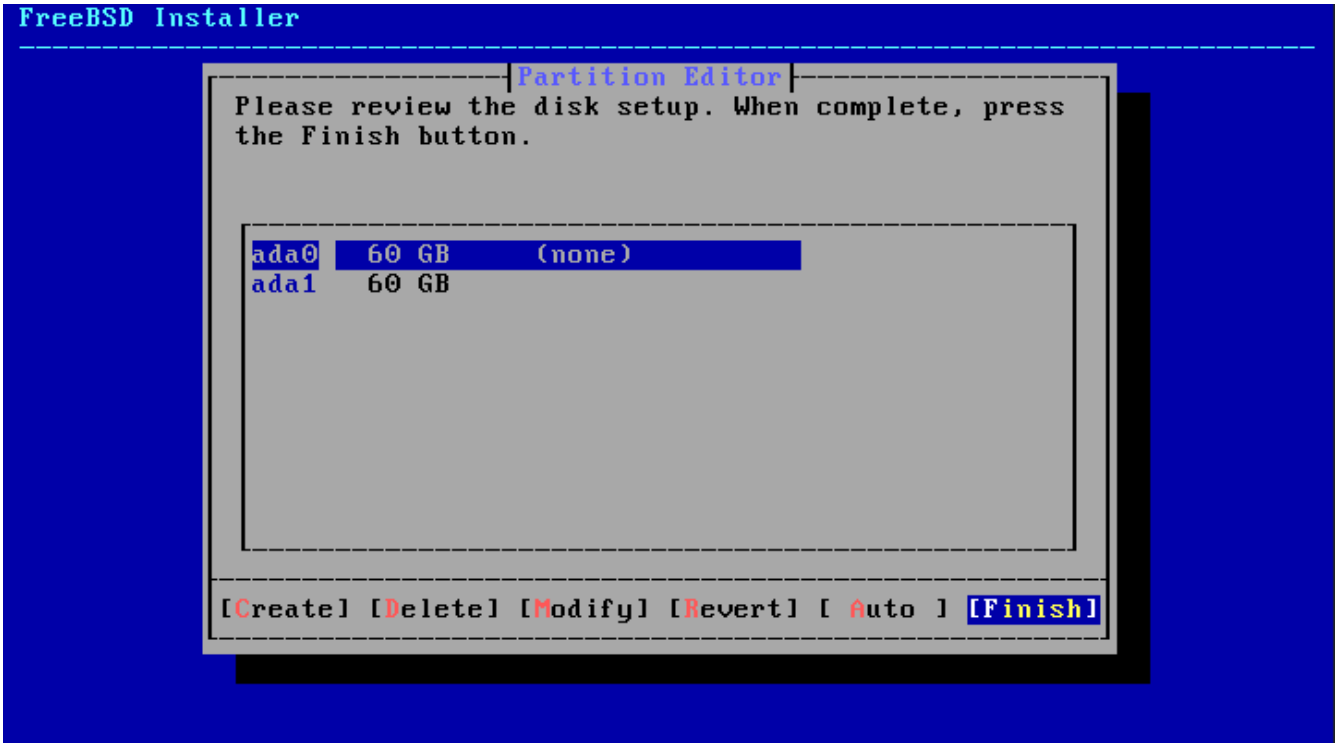



*Abbildung 16. Partitionen manuell erstellen*

Durch hervorheben einer Platte (in diesem Fall ada0) und die Auswahl von **[ Create ]**, wird ein Menü mit den verfügbaren Partitionierungsschemas angezeigt.

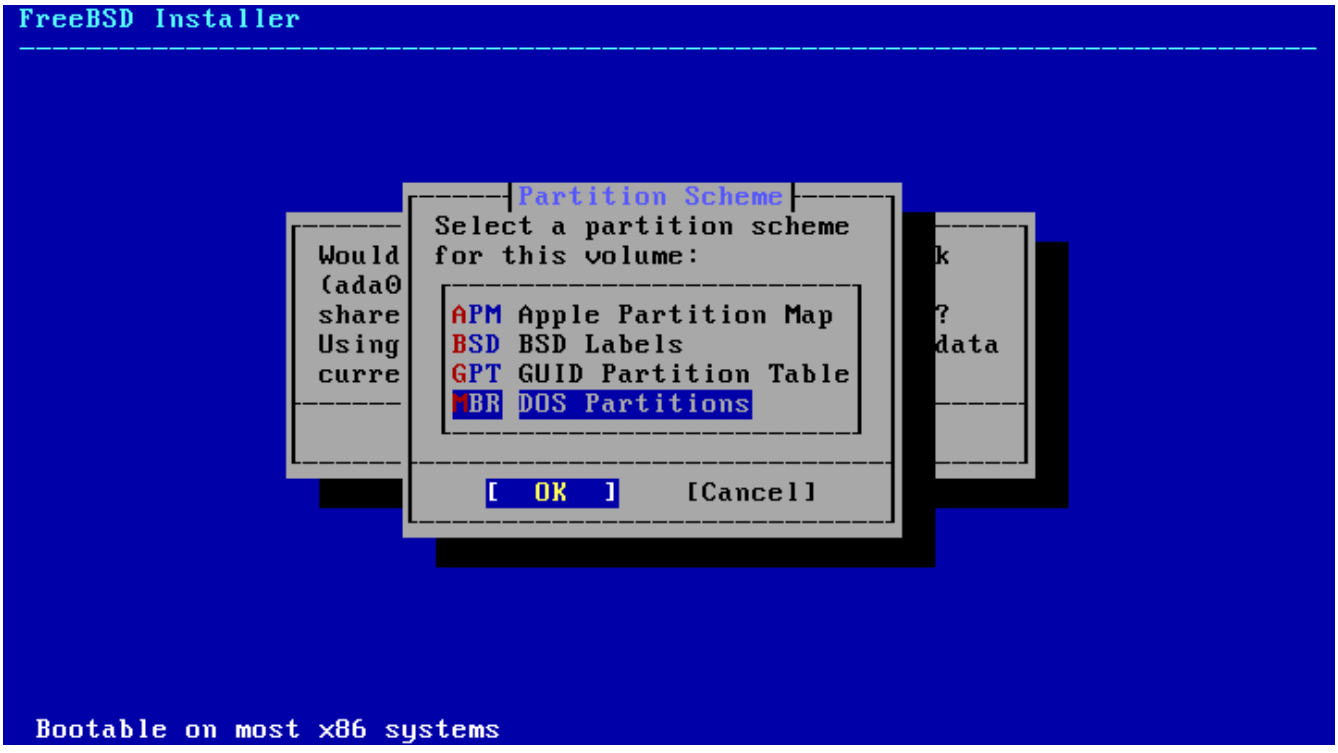



*Abbildung 17. Partitionen manuell anlegen*

GPT ist normalerweise die beste Wahl für amd64-Computer. Ältere Computer, die nicht mit GPT kompatibel sind, sollten MBR verwenden. Die anderen Partitionsschemas werden für gewöhnlich für ältere Computersysteme benutzt.

*Tabelle 1. Partitionierungsschemas*

| Abkürzung | Beschreibung |
| --- | --- |
| APM | Apple Partition Map, verwendet von PowerPC®. |
| BSD | BSD-Labels ohne einen MBR, manchmal auch "dangerously dedicated mode" genannt, da nicht-BSD Festplatten-Werkzeuge dies vielleicht nicht erkennen können. |
| GPT | GUID Partition Table (http://en.wikipedia.org/wiki/GUID_Partition_Table). |
| MBR | Master Boot Record (http://en.wikipedia.org/wiki/Master_boot_record). |
| VTOC8 | Volume Table Of Contents, von Sun SPARC64 und UltraSPARC Computern verwendet. |

Nachdem das Partitionierungsschema ausgewählt und erstellt wurde, werden durch erneute Auswahl von **[ Create ]** die Partitionen erzeugt. Mit der `Tab`-Taste können Sie den Cursor zwischen den Feldern bewegen.

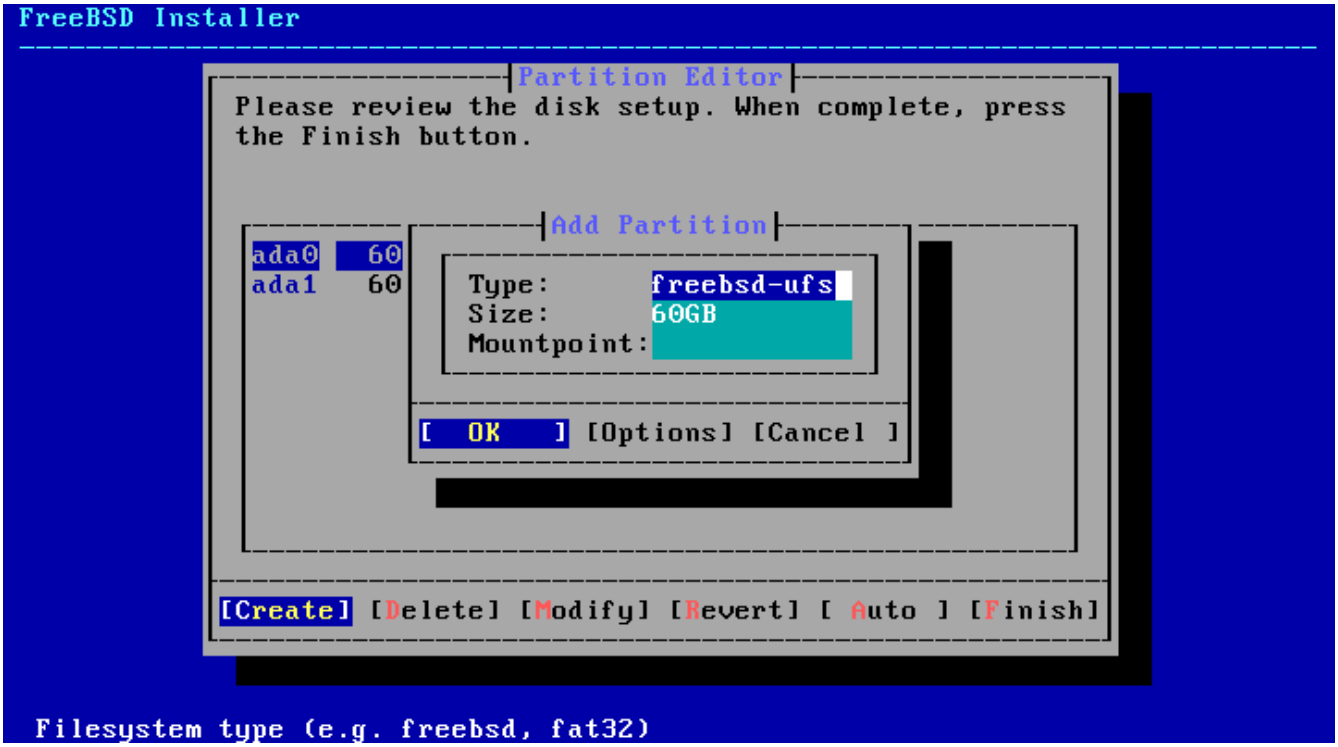



*Abbildung 18. Partitionen manuell erzeugen*

Eine FreeBSD-Standardinstallation mit GPT legt mindestens die folgenden drei Partitionen an:

- `freebsd-boot` - Enthält den FreeBSD-Bootcode.
- `freebsd-ufs` - Ein FreeBSD UFS-Dateisystem.
- `freebsd-zfs` - Ein FreeBSD ZFS-Dateisystem. Weitere Informationen finden Sie in Das Z-Dateisystem (ZFS).

- `freebsd-swap` - FreeBSD Auslagerungsbereich (swap space).

Die einzelnen GPT-Partitionstypen sind in gpart(8) dokumentiert.

Es können mehrere Dateisystempartitionen erzeugt werden und manche Leute ziehen es vor, ein traditionelles Layout mit getrennten Partitionen für die Dateisysteme /, /var, /tmp und /usr zu erstellen. Lesen Sie dazu Ein traditionelles, partitioniertes Dateisystem erstellen, um ein Beispiel zu erhalten.

Größenangaben (`Size`) können mit gängigen Abkürzungen eingegeben werden: *K* für Kilobytes, *M* für Megabytes oder *G* für Gigabytes.



Korrekte Sektorausrichtung ermöglicht größtmögliche Geschwindigkeit und das Anlegen von Partitionsgrößen als vielfaches von 4K-Bytes hilft, die passende Ausrichtung auf Platten mit entweder 512-Bytes oder 4K-Bytes Sektorgrößen, festzulegen. Generell sollte die Verwendung von Partitionsgrößen, die sogar vielfache von 1M oder 1G sind, den einfachsten Weg darstellen, um sicher zu stellen, dass jede Partition an einem vielfachen von 4K beginnt. Eine Ausnahme gibt es: momentan sollte die *freebsd-boot*-Partition aufgrund von Beschränkungen im Bootcode nicht größer sein als 512K.

Ein Einhägepunkt (`Mountpoint`) wird benötigt, falls diese Partition ein Dateisystem enthält. Falls nur eine einzelne UFS-Partition erstellt wird, sollte der Einhängepunkt / lauten.

Ein `label` ist ein Name, durch den diese Partition angesprochen wird. Festplattennamen oder -nummern können sich ändern, falls die Platte einmal an einem anderen Controller oder Port angeschlossen sein sollte, doch das Partitionslabel ändert sich dadurch nicht. Anstatt auf Plattennamen und Partitionsnummern in Dateien wie /etc/fstab zu verweisen, sorgen Labels dafür, dass das System Hardwareänderungen eher toleriert. GPT-Labels erscheinen in /dev/gpt/, wenn eine Platte angeschlossen wird. Andere Partitionierungsschemas besitzen unterschiedliche Fähigkeiten, Labels zu verwenden und diese erscheinen in anderen /dev/-Verzeichnissen.



Vergeben Sie ein einzigartiges Label für jede Partition, um Konflikte mit identischen Labels zu verhindern. Ein paar Buchstaben des Computernamens, dessen Verwendungszweck oder Ortes kann dem Label hinzugefügt werden. Beispielsweise `labroot` oder `rootfslab` für die UFS Root-Partition auf einem Laborrechner namens `lab`.

*Beispiel 1. Ein traditionelles, partitioniertes Dateisystem erstellen*

Für ein traditionelles Partitionslayout, in dem sich /, /var, /tmp und /usr in getrennten Partitionen befinden sollen, erstellen Sie ein GPT-Partitionsschema und anschließend die Partitionen selbst. Die gezeigten Partitionsgrößen sind typisch für eine Festplatte von 20 G. Falls mehr Platz verfügbar ist, sind größere Swap oder /var-Partitionen nützlich. Den hier gezeigten Beschreibungen sind `bsp` für "Beispiel" vorangestellt, jedoch sollten Sie andere, einzigartige Beschreibungen verwenden, wie oben beschrieben.

Standardmäßig erwartet FreeBSDs gptboot, dass die erste UFS-Partition die /-Partition ist.

| Partitionstyp | Grösse | Eingehängt als | Beschreibung |
|---|---|---|---|
| freebsd-boot | 512K | | |
| freebsd-ufs | 2G | / | bsprootfs |
| freebsd-swap | 4G | bspswap | |
| freebsd-ufs | 2G | /var | bspvarfs |
| freebsd-ufs | 1G | /tmp | bsptmpfs |
| freebsd-ufs | Akzeptieren Sie die Standardeinstellungen (Rest der Platte) | /usr | bspusrfs |

Nachdem die Partitionen erzeugt wurden, wählen Sie **[ Finish ]**, um die Installation mit Abrufen der Distributionen fortzusetzen.

### 4.6.4. Geführte Partitionierung mit Root-on-ZFS

Dieser Modus funktioniert nur mit ganzen Laufwerken und wird alle vorhandenen Daten auf der Platte löschen. Das Konfigurationsmenü für ZFS bietet einige Optionen, um die Erstellung des Pools zu beeinflussen.

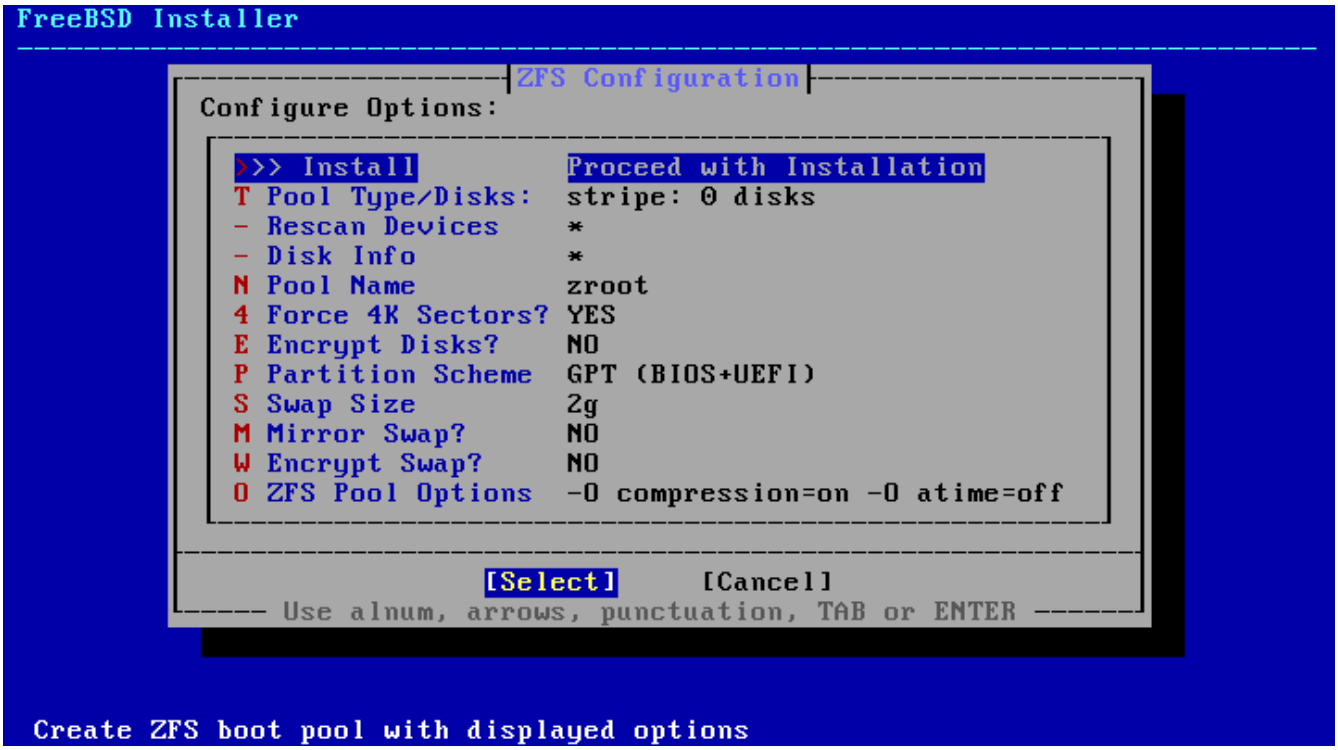



*Abbildung 19. ZFS Konfigurationsmenü*

Hier eine Zusammenfassung der Optionen, die in diesem Menü benutzt werden können:

- `Install` - Setzt die Installation mit den ausgewählten Optionen fort.
- `Pool Type/Disks` - Erlaubt die Konfiguration des `Pool Type` und der Festplatte(n), die den Pool bilden werden. Das ZFS-Installationsprogramm unterstützt derzeit nur die Erstellung eines einzelnen Top-Level-Vdev, außer im Stripe-Modus. Um komplexere Pools zu erstellen, folgen Sie

den Anweisungen in Shell Partitionierung, um den Pool zu erstellen.

- `Rescan Devices` - Aktualisiert die Liste der verfügbaren Festplatten.
- `Disk Info` - Dieses Menü wird verwendet, um Datenträger zu inspizieren, einschließlich ihrer Partitionstabelle und weitere Informationen wie die Modell- und Seriennummer des Geräts.
- `Pool Name` - Legt den Namen des Pools fest. Der Standard ist *zroot*.
- `Force 4K Sectors?` - Erzwingt die Verwendung von 4K-Sektoren. Im Standard erstellt die Installation automatisch Partitionen, die an 4K-Grenzen ausgerichtet sind. Bei ZFS wird die Verwendung von 4K-Sektoren erzwungen. Dies ist selbst bei Festplatten mit 512-Byte-Sektoren sicher und hat den zusätzlichen Vorteil, dass Pools, die auf solchen Festplatten mit erstellt werden, auch in Zukunft 4K-Sektoren haben können, entweder als zusätzlicher Speicherplatz oder als Ersatz für ausgefallene Platten. Drücken Sie `Enter`, um die Verwendung von 4K-Sektoren zu konfigurieren.
- `Encrypt Disks?` - Das Verschlüsseln der Datenträger mit GELI. Weitere Informationen zur Datenträgerverschlüsselung finden Sie in "Plattenverschlüsselung mit geli". Drücken Sie `Enter` um eine Auswahl zu treffen.
- `Partition Scheme` - Erlaubt die Auswahl des Partitionsschemas. GPT ist die empfohlene Option. Drücken Sie `Enter`, um zwischen den verschiedenen Optionen zu wählen.
- `Swap Size` - Legt die Größe des Swap-Speichers fest.
- `Mirror Swap?` - Erlaubt es, den Swap-Speicher zwischen den Platten zu spiegeln. Beachten Sie jedoch, dass die Aktivierung dazu führt, dass Crash Dumps nicht mehr funktionieren. Drücken Sie `Enter`, um diese Option zu aktivieren/deaktivieren.
- `Encrypt Swap?` - Erlaubt es, den Swap-Speicher zu verschlüsseln. Der Swap-Speicher wird bei jedem Systemstart mit einem temporären Schlüssel verschlüsselt, der bei einem Neustart des Systems verworfen wird. Drücken Sie `Enter`, diese Option zu aktivieren/deaktivieren. Weitere Informationen zur Verschlüsselung des Swap-Speichers finden Sie in "Den Auslagerungsspeicher verschlüsseln".

Wählen Sie `T` um den Pool Typ und die Festplatte(n) zu konfigurieren, die den Pool bilden werden.

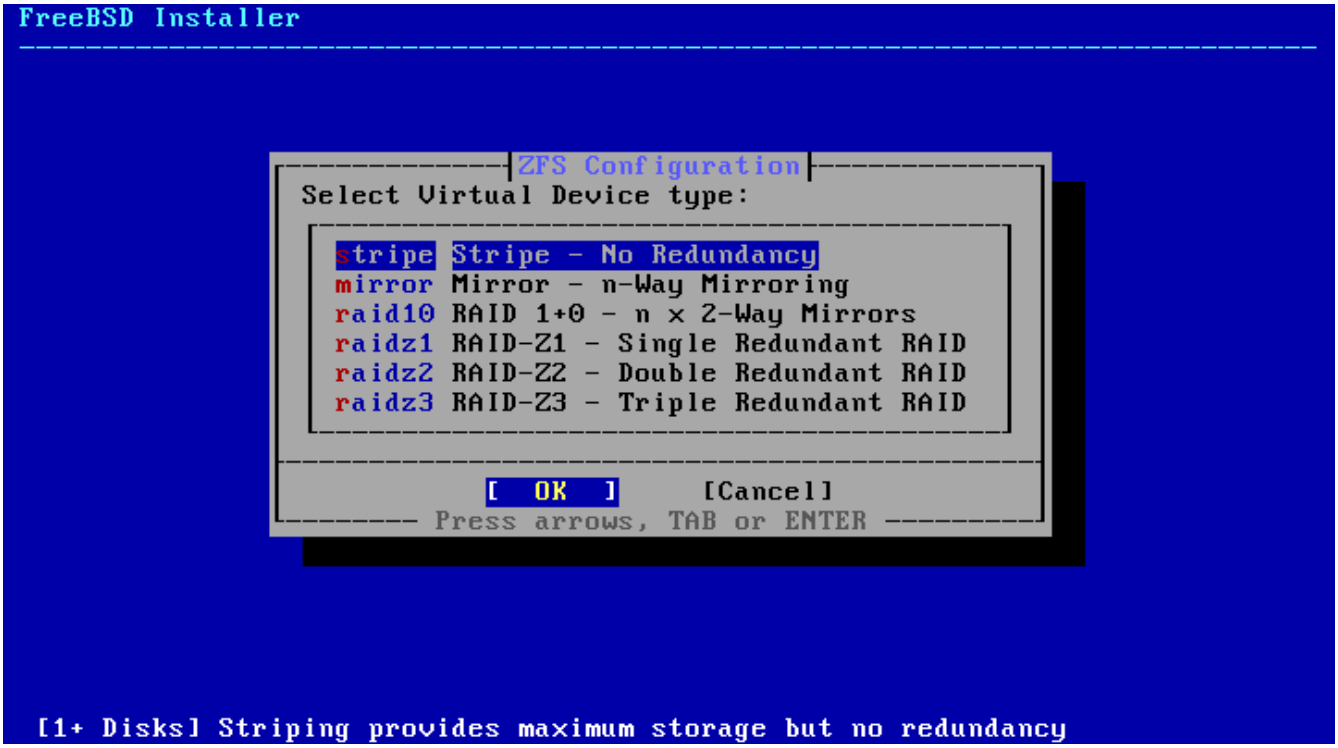



*Abbildung 20. ZFS Pool Typen*

Hier eine Zusammenfassung der Pool-Typen, die in diesem Menü ausgewählt werden können:

- `stripe` - Striping bietet maximalen Speicherplatz für alle angeschlossenen Geräte, aber keine Redundanz. Fällt eine Platte aus, sind die Daten im Pool unwiderruflich verloren.
- `mirror` - Bei der Spiegelung wird eine vollständige Kopie aller Daten auf jeder Platte gespeichert. Die Spiegelung bietet eine gute Leistung beim Lesen, da die Daten von allen Platten parallel gelesen werden. Die Leistung beim Schreiben ist langsamer, da die Daten auf alle Platten im Pool geschrieben werden müssen. Hiermit können alle Platten bis auf eine ausfallen. Diese Option erfordert mindestens zwei Platten.
- `raid10` - Striped Mirrors. Bieten die beste Leistung, aber den geringsten Speicherplatz. Diese Option erfordert mindestens eine gerade Anzahl von Platten und mindestens vier Platten.
- `raidz1` - Einzelnes redundantes RAID. Ermöglicht den Ausfall einer Platte. Für diese Option sind mindestens drei Festplatten erforderlich.
- `raidz2` - Doppeltes redundantes RAID. Ermöglicht den Ausfall von zwei Platten. Für diese Option sind mindestens vier Festplatten erforderlich.
- `raidz3` - Dreifaches redundantes RAID. Ermöglicht den Ausfall von drei Platten. Für diese Option sind mindestens fünf Festplatten erforderlich.

Sobald ein Pool-Typ (`Pool Type`) ausgewählt wurde, wird eine Liste der verfügbaren Laufwerke angezeigt und der Benutzer wird aufgefordert, eine oder mehrere Laufwerke für die Erstellung des Pools auszuwählen. Anschließend wie die Konfiguration geprüft um zu gewährleisten, dass genug Laufwerke ausgewählt wurden. Wählen Sie **[ <Change Selection> ]** um zur Auswahl der Laufwerke zurückzukehren, oder **[ <Back> ]** um den `Pool Type` zu ändern.

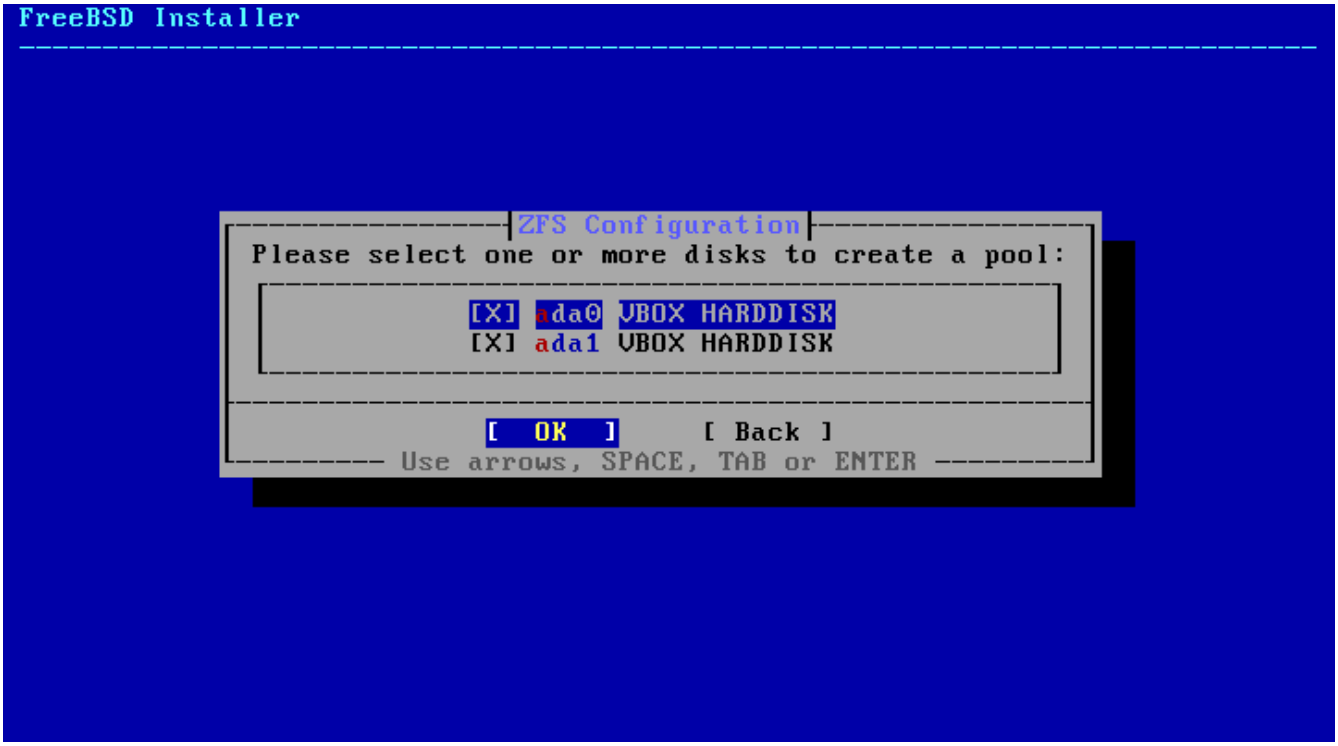



*Abbildung 21. Auswahl der Laufwerke*

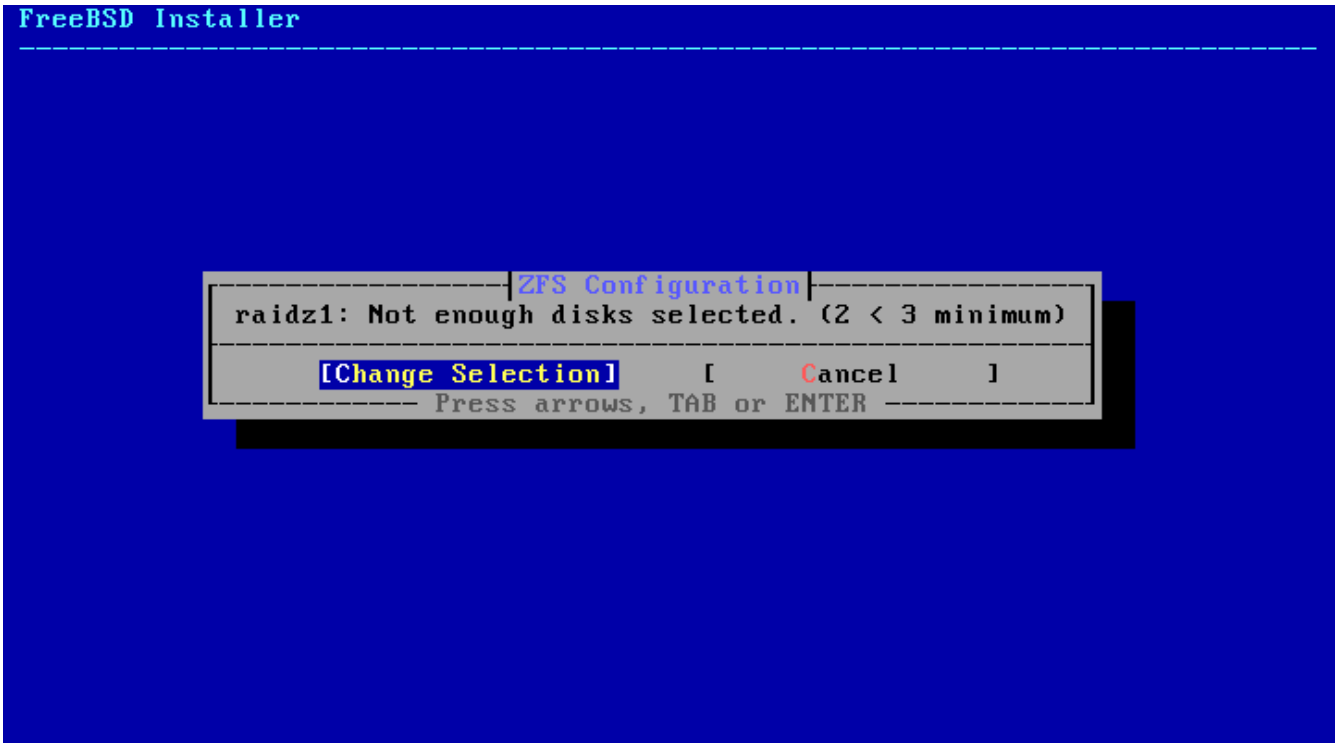



*Abbildung 22. Ungültige Auswahl*

Wenn eine oder mehrere Platten in der Liste fehlen, oder wenn Festplatten angebunden wurden, nachdem das Installationsprogramm gestartet wurde, wählen Sie **[ - Rescan Devices ]** um die Laufwerke nochmals zu suchen und anzuzeigen.

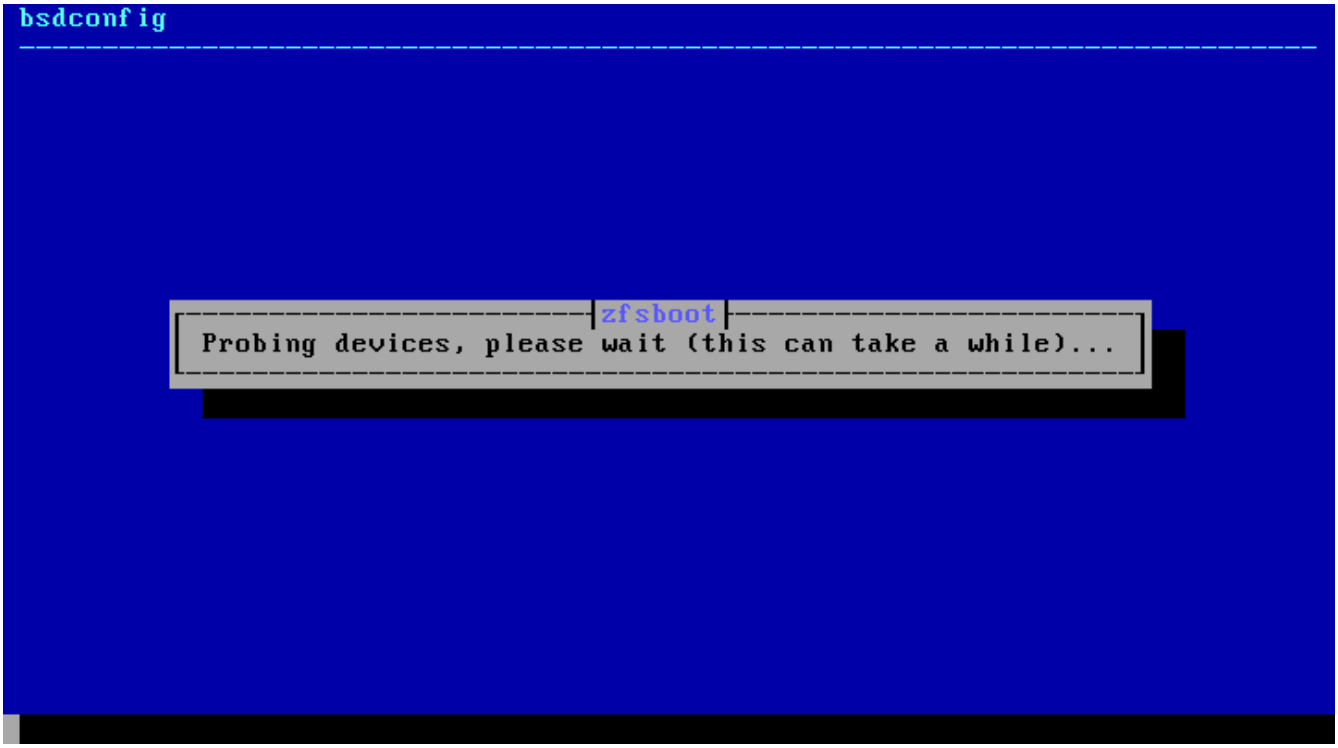



*Abbildung 23. Rescan Devices*

Um zu vermeiden, dass versehentlich die falsche Platte gelöscht wird, können Sie das **[ - Disk-Info ]** -Menü verwenden. Dieses Menü zeigt verschiedene Informationen, einschließlich der Partitionstabelle, der Modellnummer und der Seriennummer, falls verfügbar.

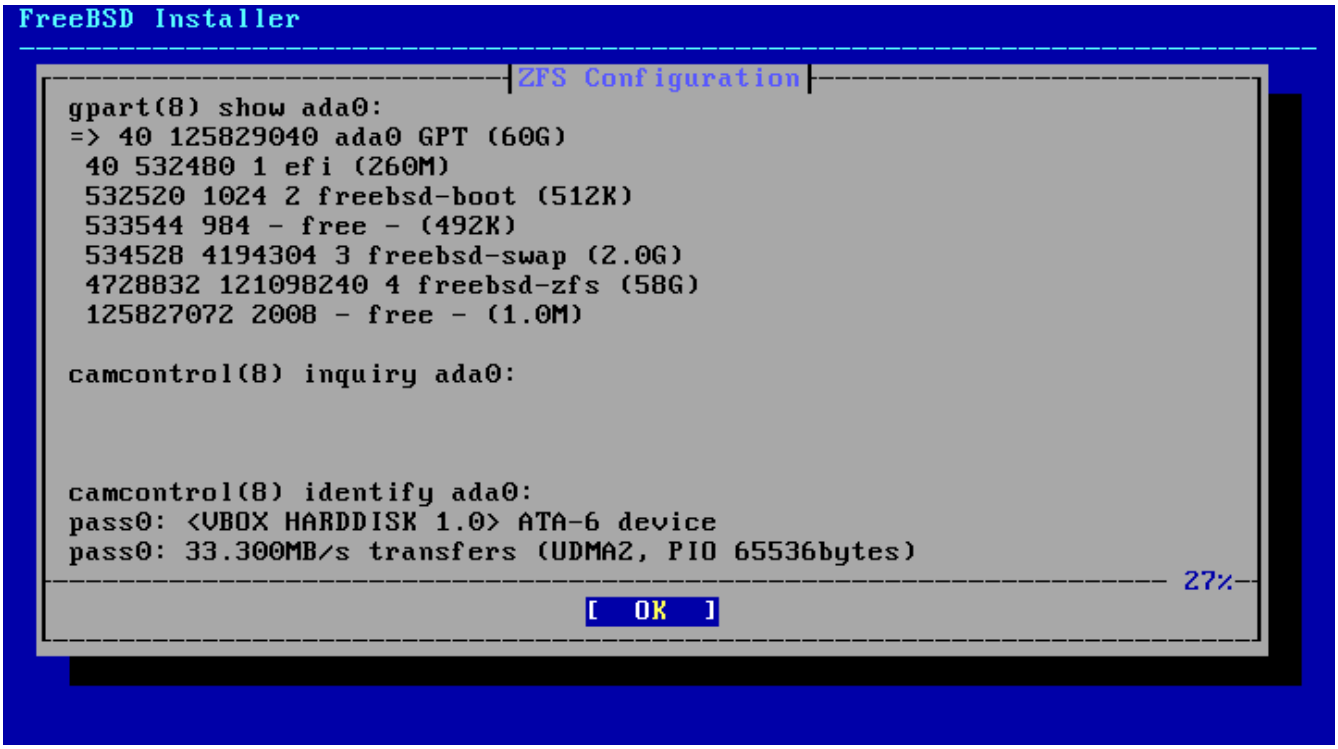



*Abbildung 24. Informationen zum Laufwerk*

Wählen Sie `N`, um den Pool-Namen zu konfigurieren. Geben Sie den gewünschten Namen ein und wählen Sie dann **[ OK ]**, um den Namen zu speichern, oder **[ <Cancel> ]**, um zum Hauptmenü zurückzukehren und den Standard zu belassen.




*Abbildung 25. Pool-Name*

Wählen Sie `S`, um die Größe des Swap-Speichers festzulegen. Geben Sie die gewünschte Größe ein und wählen Sie dann **[ OK ]**, um die Einstellung zu speichern, oder **[ <Cancel> ]**, um zum Hauptmenü zurückzukehren und den Standard zu belassen.

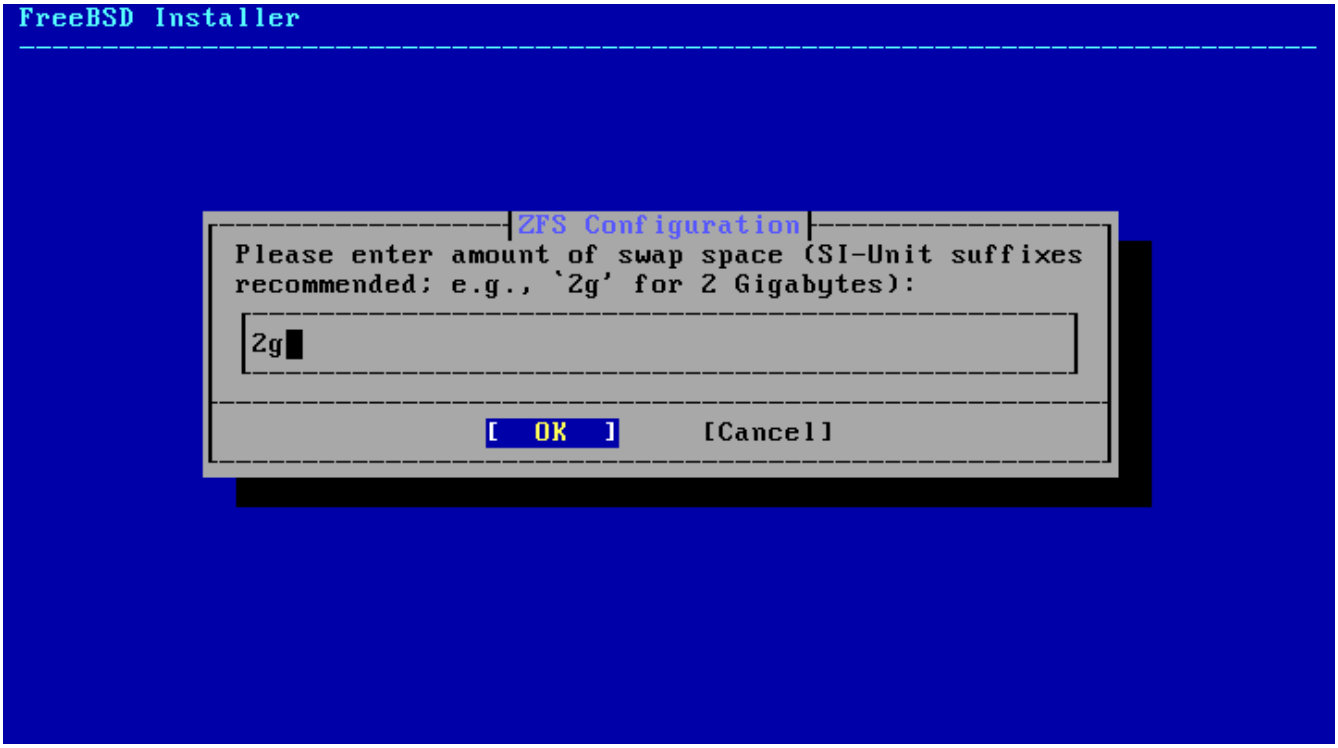



*Abbildung 26. Größe des Swap-Speichers*

Wenn alle Optionen wie gewünscht konfiguriert sind, wählen Sie oben im Menü die Option **[ >>> Install ]**. Das Installationsprogramm bietet dann eine letzte Chance zum Abbrechen, bevor der Inhalt der ausgewählten Laufwerke zerstört wird, um den ZFS-Pool zu erstellen.

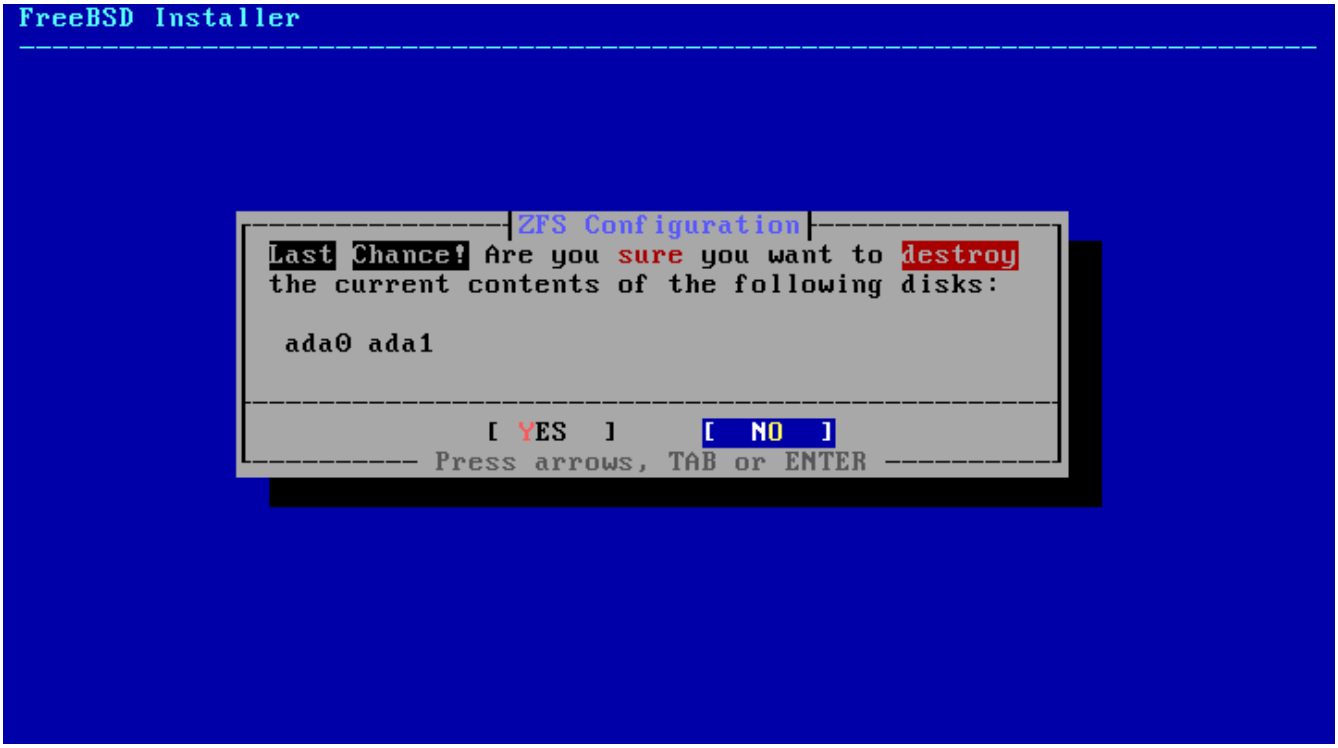



*Abbildung 27. Letzte Chance!*

Wenn die GELI Plattenverschlüsselung aktiviert wurde, fordert das Installationsprogramm zweimal zur Eingabe der Passphrase auf. Anschließend beginnt die Initialisierung der Verschlüsselung.

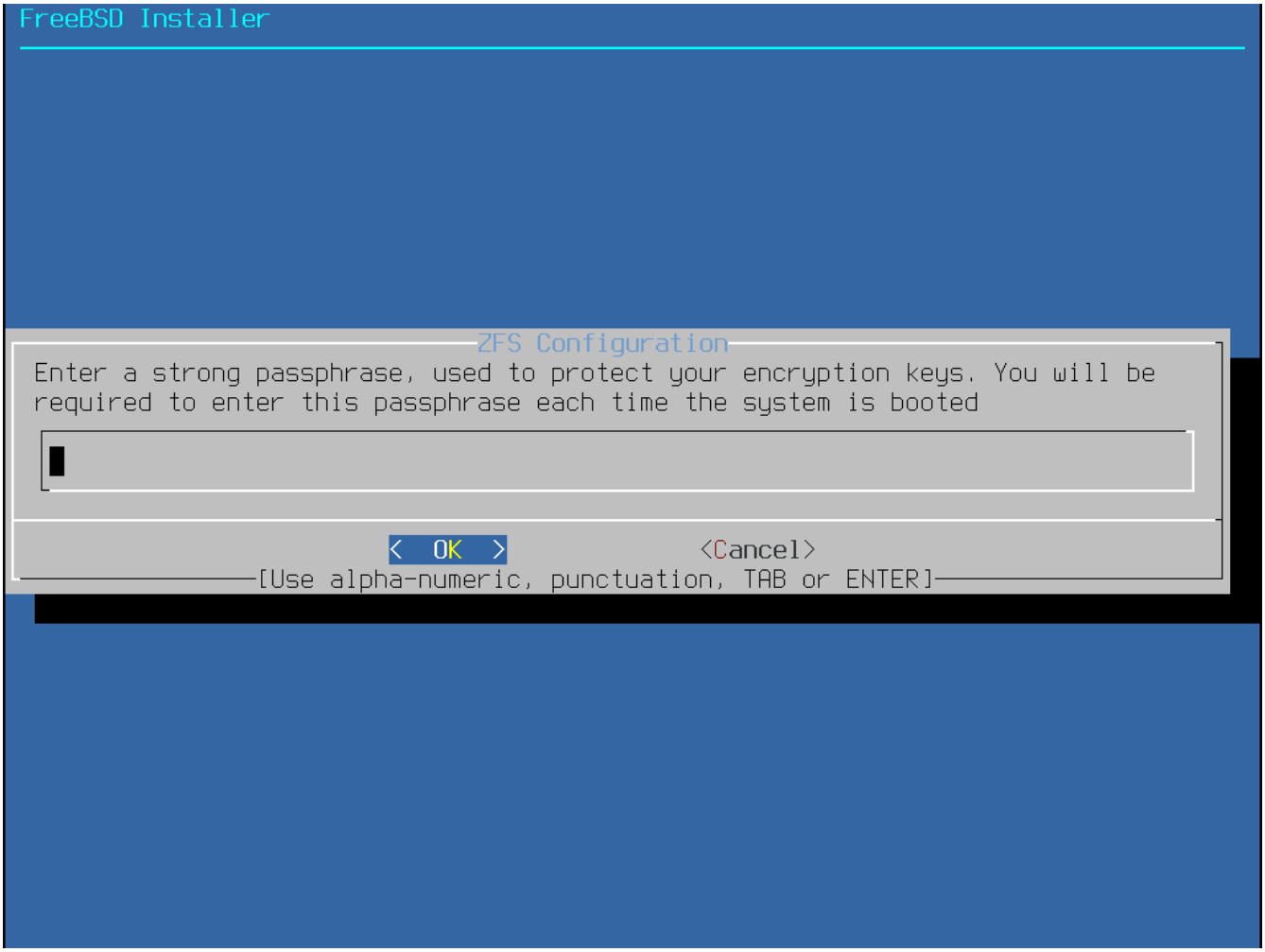



*Abbildung 28. Passwort für die Verschlüsselung der Platte*

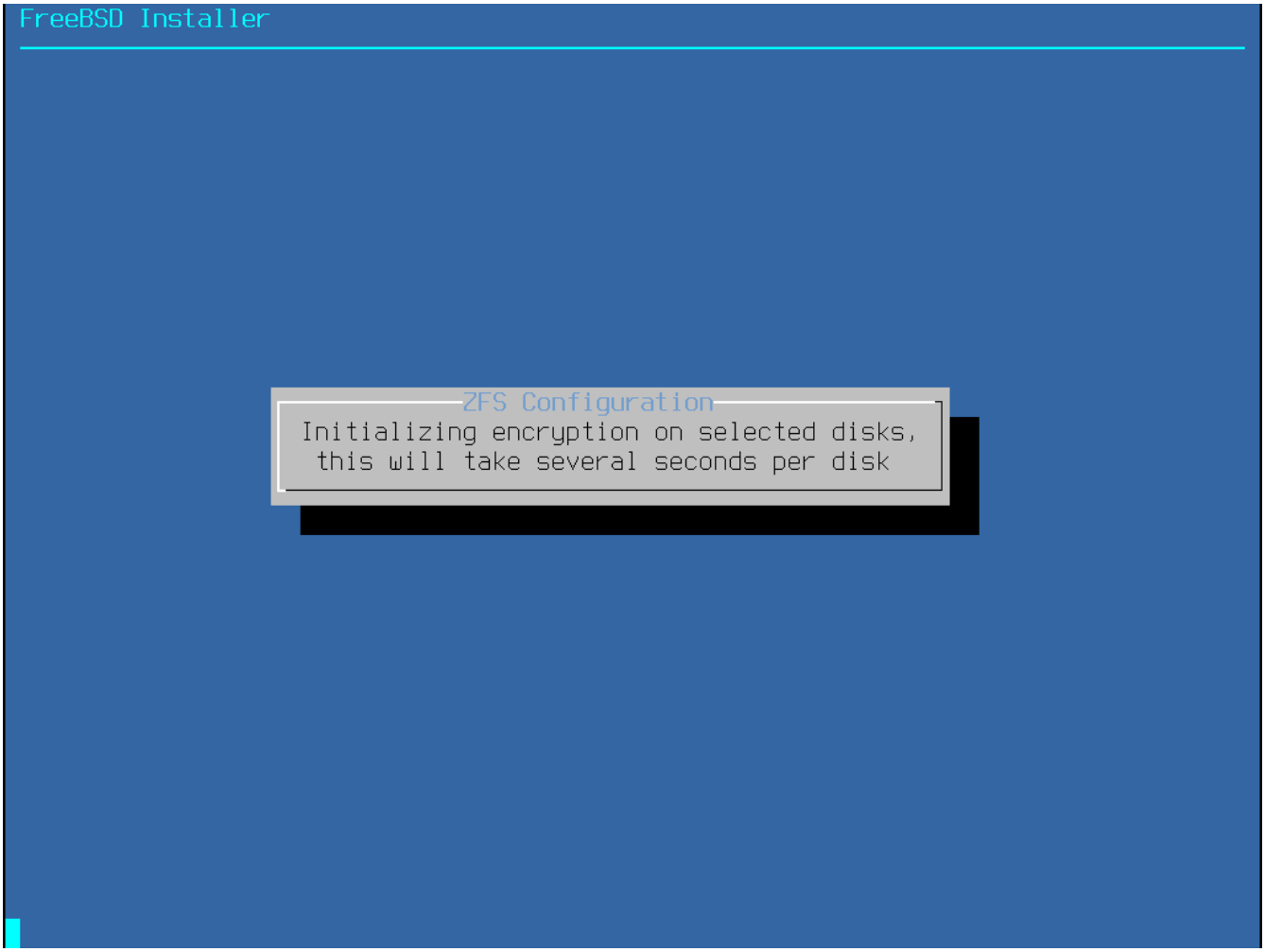



*Abbildung 29. Initialisierung der Verschlüsselung*

Danach wird die Installation normal weitergeführt. Um mit der Installation fortzufahren, lesen Sie Abrufen der Distributionen.

### 4.6.5. Shell Partitionierung

bsdinstall bietet bei fortgeschrittenen Installationen womöglich nicht die benötigte Flexibilität. Erfahrene Benutzer können die Option **[ Shell ]** im Menü auswählen, um die Laufwerke manuell zu partitionieren, Dateisysteme zu erstellen, /tmp/bsdinstall_etc/fstab zu befüllen und Dateisysteme unter /mnt einzuhängen. Geben Sie anschließend `exit` ein, um zu bsdinstall zurückzukehren und die Installation fortzusetzen.

## 4.7. Abrufen der Distributionen

Die Installationsdauer hängt von den gewählten Distributionen, dem Installationsmedium und der Geschwindigkeit des Computers ab. Eine Reihe von Nachrichten werden angezeigt, um den Fortschritt darzustellen.

Zunächst formatiert das Installationsprogramm die ausgewählten Platten und initialisiert die Partitionen. Bei `bootonly media` oder `mini memstick` werden als nächstes die benötigten Komponenten heruntergeladen:

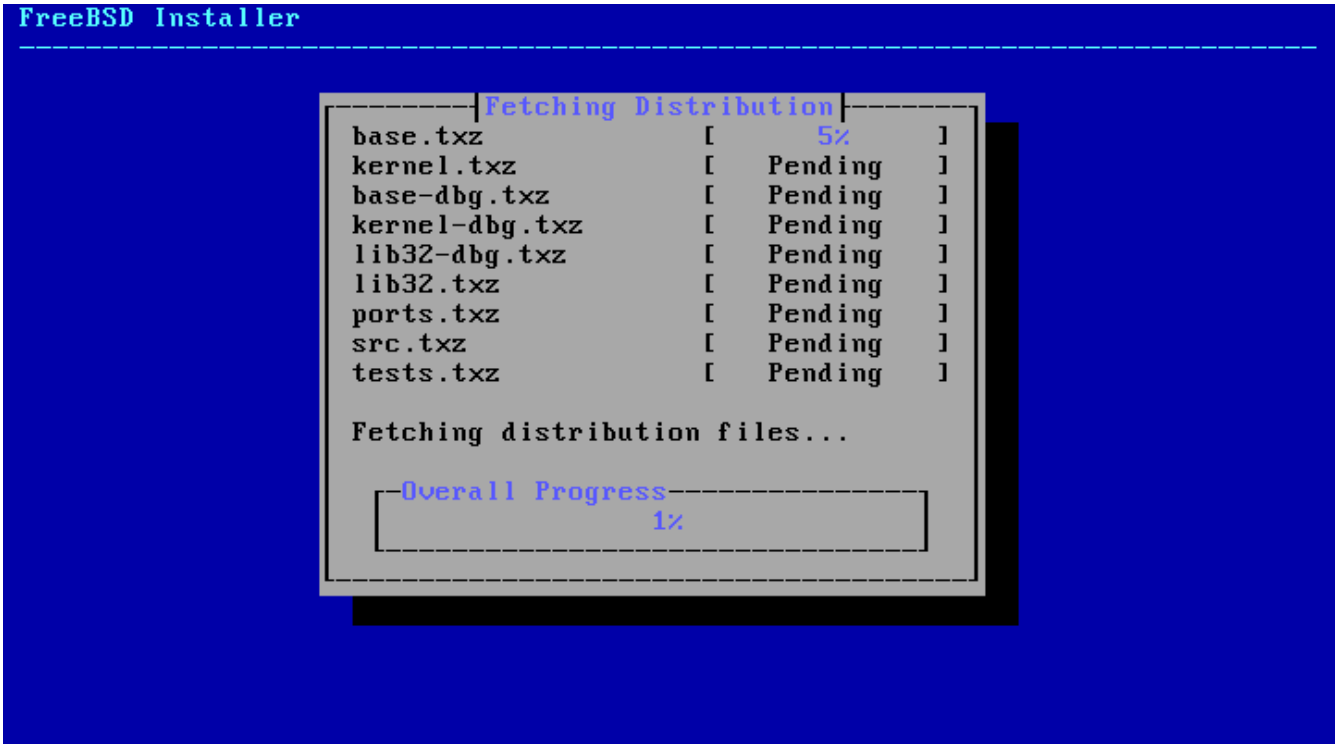



*Abbildung 30. Herunterladen der Distributionsdateien*

Als nächstes wird die Integrität der Distributionsdateien überprüft, um sicherzustellen, dass diese während des Ladevorgangs nicht beschädigt oder unsauber vom Installationsmedium gelesen wurden:

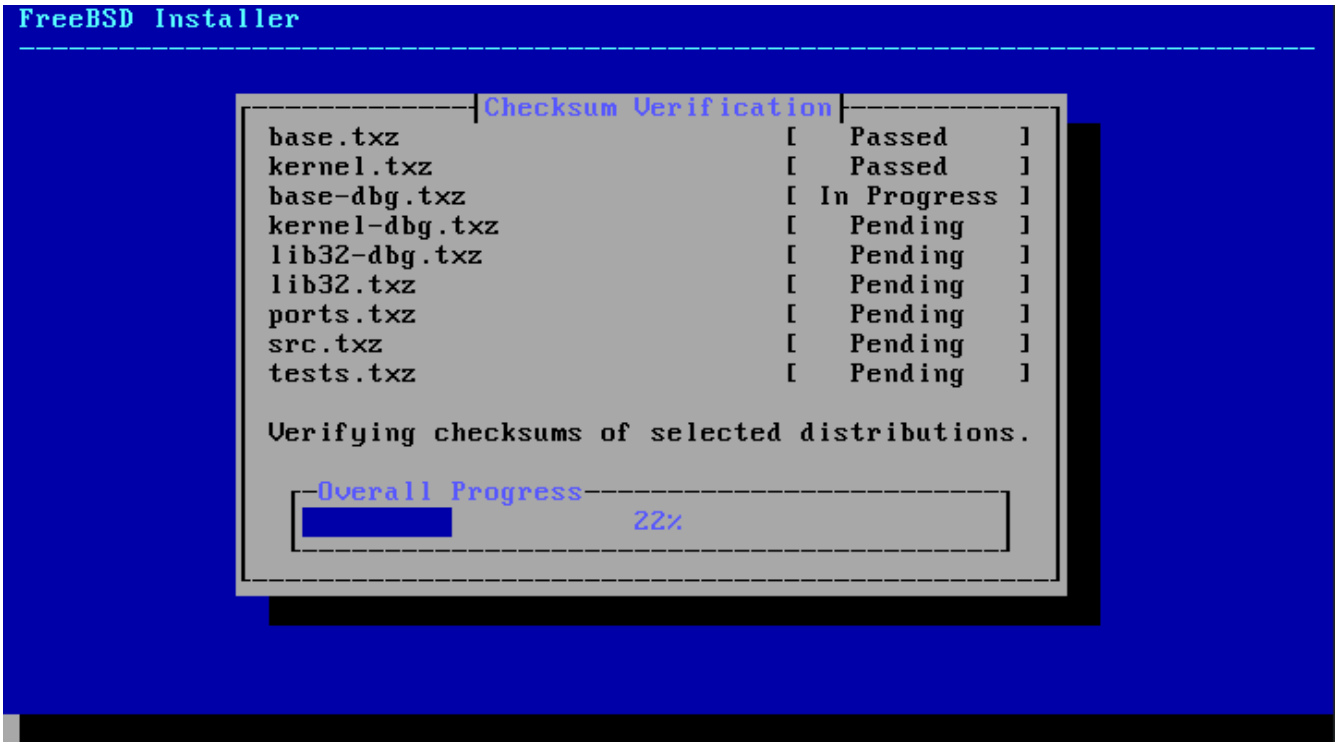



*Abbildung 31. Überprüfen der Distributionsdateien*

Zum Schluss werden die überprüften Distributionsdateien auf die Festplatte entpackt:

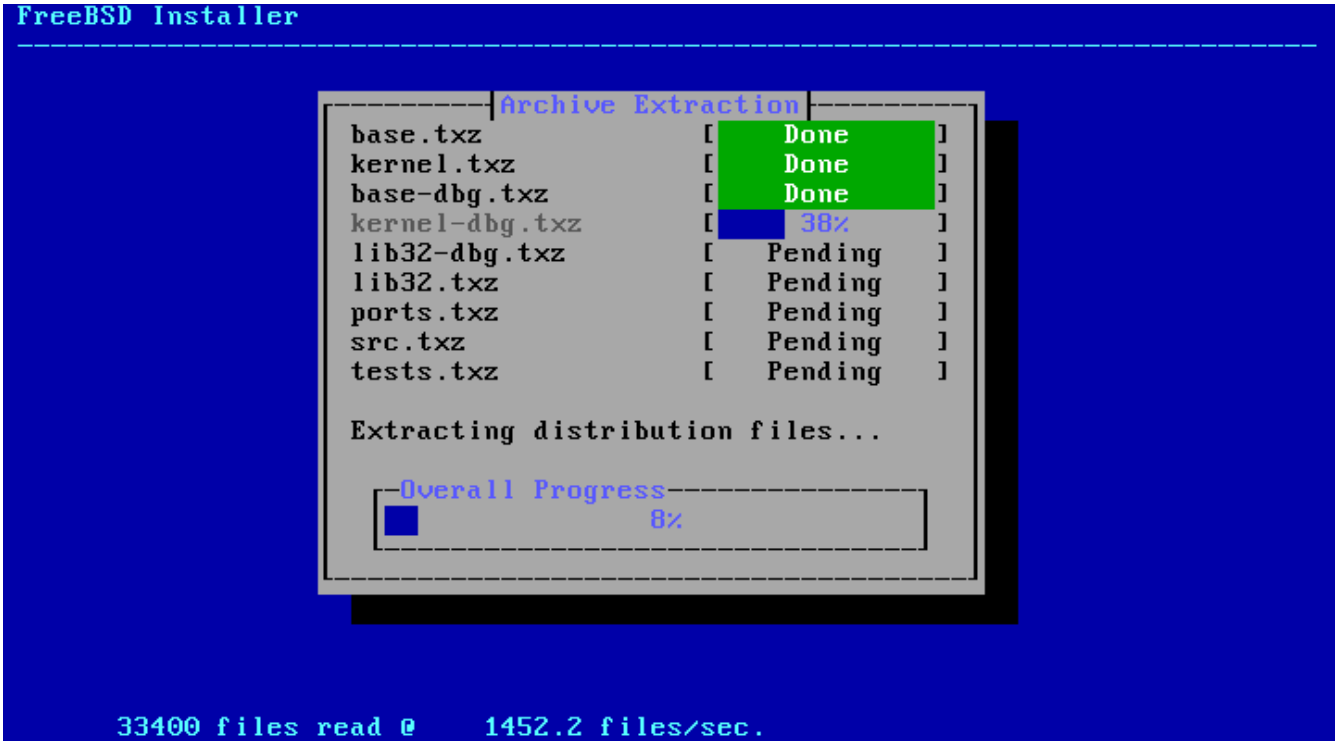



*Abbildung 32. Entpacken der Distributionsdateien*

Sobald alle benötigten Distributionsdateien entpackt wurden, wird bsdinstall das erste Menü für die Arbeiten nach der Installation anzeigen. Die zur Verfügung stehenden Konfigurationsoptionen werden im nächsten Abschnitt beschrieben.

# 4.8. Benutzerkonten, Zeitzone, Dienste und Sicherheitsoptionen

## 4.8.1. Setzen des `root`-Passworts

Zuerst muss das `root`-Passwort gesetzt werden. Die eingegebenen Zeichen werden dabei nicht auf dem Bildschirm angezeigt. Nachdem das Passwort eingegeben wurde, muss es zur Bestätigung erneut eingetippt werden. Damit werden auch Tippfehler verhindert.

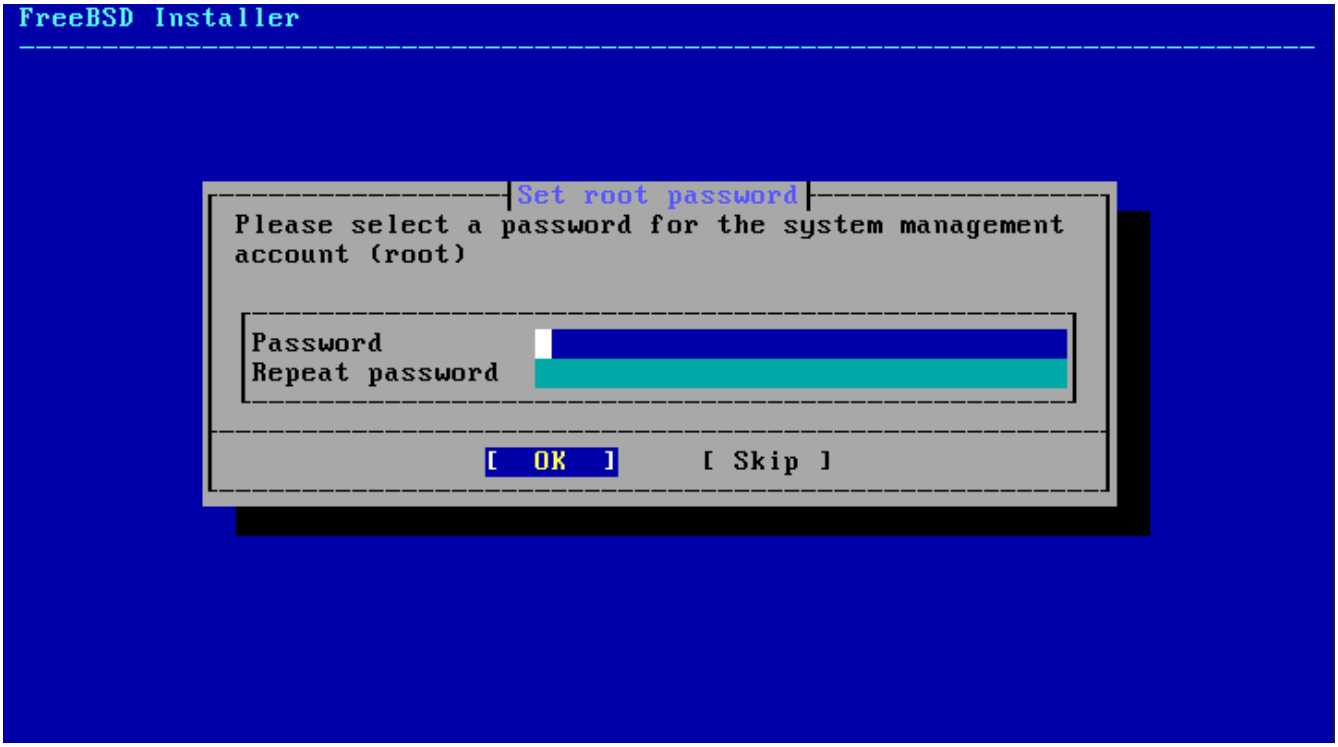



*Abbildung 33. Das `root`-Passwort setzen*

### 4.8.2. Setzen der Zeitzone

Die nächsten Menüs werden verwendet, um die korrekte Ortszeit zu ermitteln. Dazu muss die gewünschte geographische Region, das Land und die Zeitzone ausgewählt werden. Das Setzen der Zeitzone erlaubt es dem System automatische Korrekturen vorzunehmen, beispielsweise beim Wechsel von Sommer- auf Winterzeit.

Das hier gezeigte Beispiel bezieht sich auf einen Rechner in der Zeitzone des spanischen Festlands. Die Auswahl ist je nach geographischer Lage unterschiedlich.




*Abbildung 34. Auswahl der geographischen Region*




*Abbildung 35. Das Land auswählen*

Wählen Sie das zutreffende Land mit den Pfeiltasten und durch anschließendes drücken von `Enter` aus.

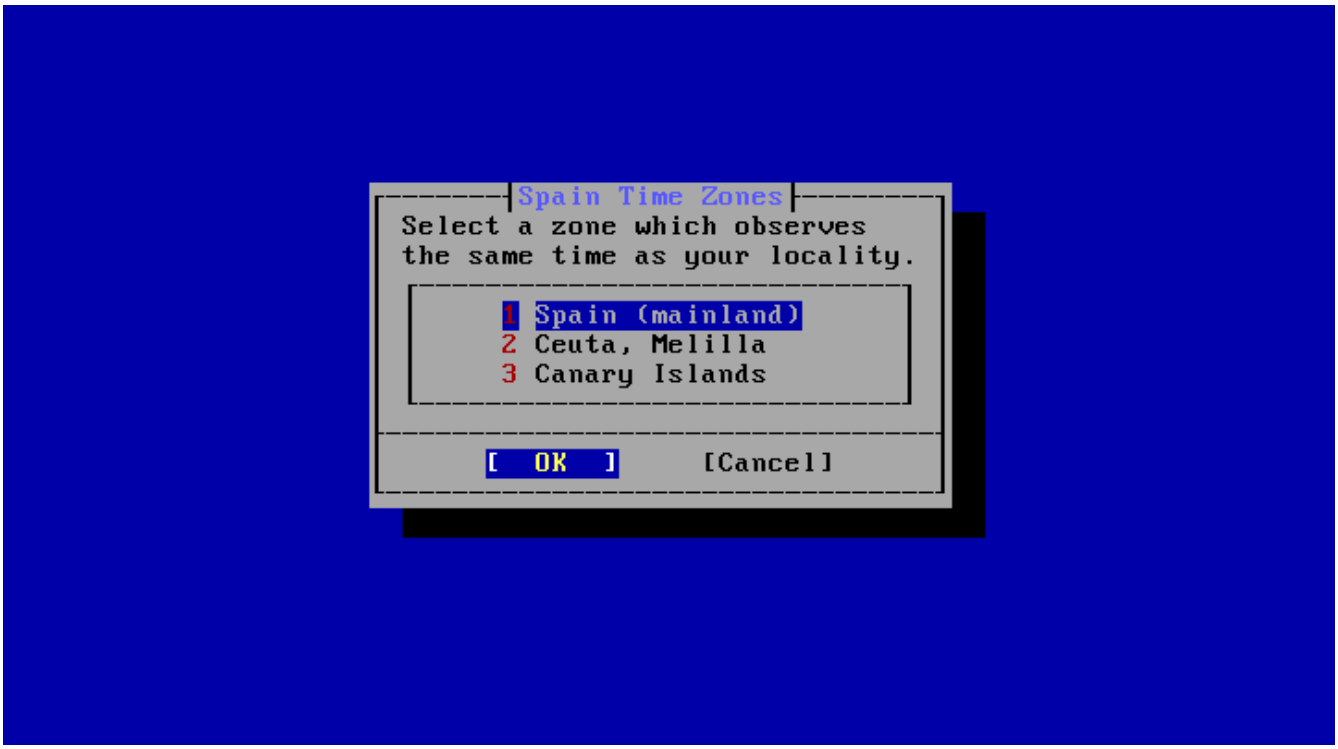



*Abbildung 36. Wählen einer Zeitzone*

Die passende Zeitzone wird durch die Pfeiltasten und anschließendes drücken von `Enter` ausgewählt.




*Abbildung 37. Bestätigen der Zeitzone*

Bestätigen Sie, dass die Abkürzung für die Zeitzone korrekt ist.

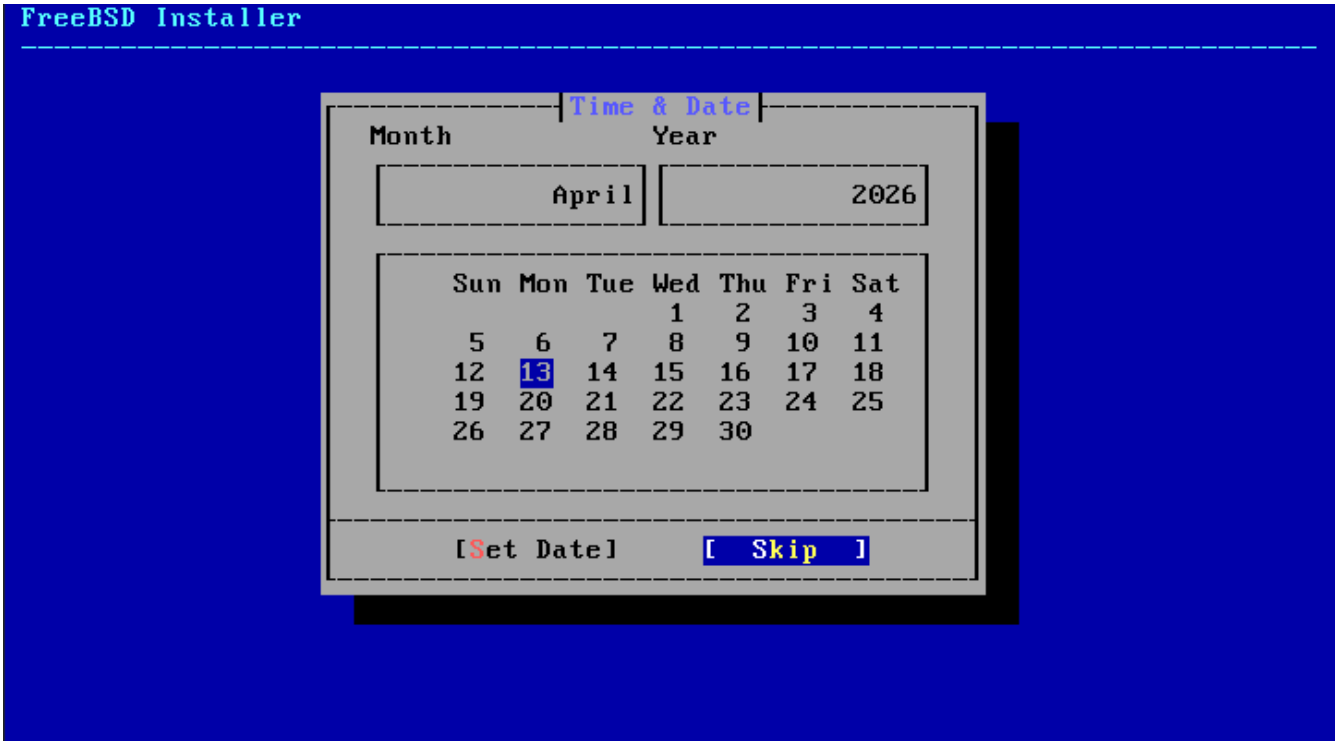



*Abbildung 38. Datum auswählen*

Das entsprechende Datum wird mit den Pfeiltasten und das anschließende Drücken von **[ Set Date ]** gewählt. Andernfalls kann die Auswahl durch Drücken von **[ Skip ]** übersprungen werden.

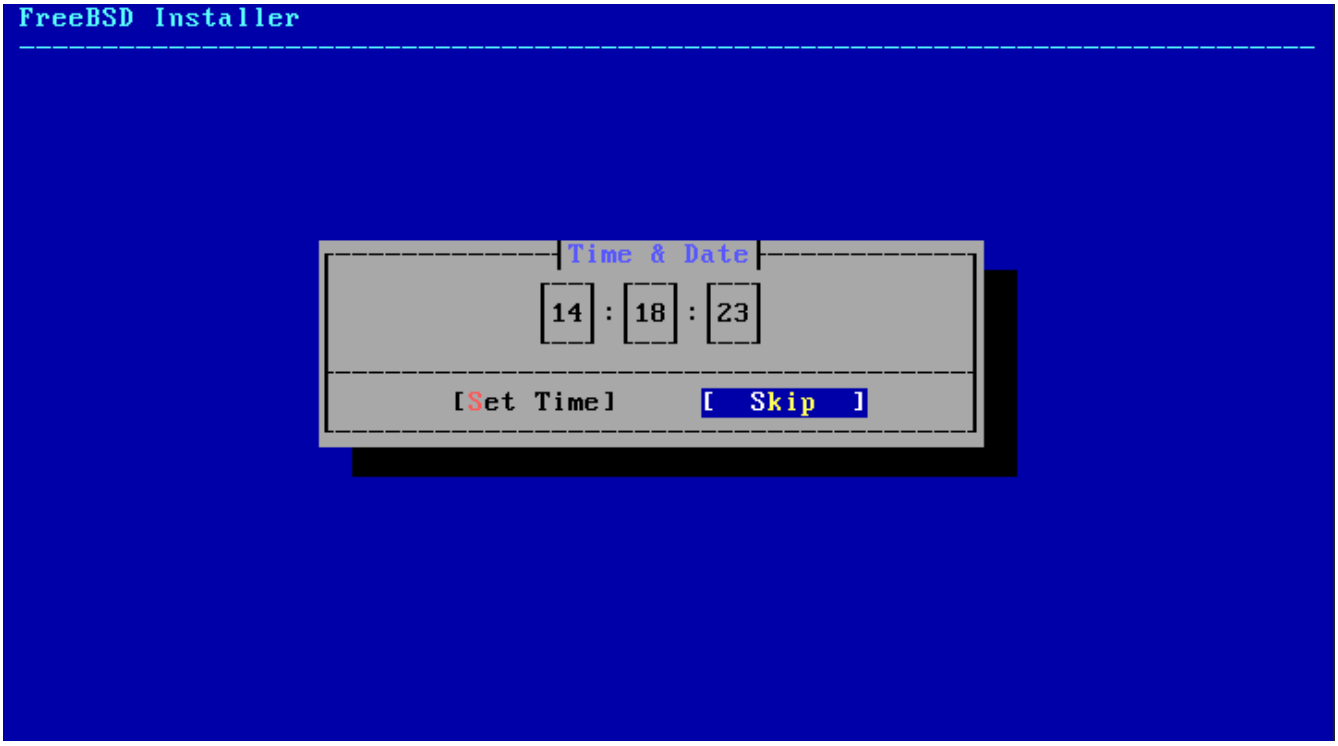



*Abbildung 39. Uhrzeit auswählen*

Die entsprechende Uhrzeit wird mit den Pfeiltasten und das anschließende Drücken von **[ Set Time ]** gewählt. Andernfalls kann die Auswahl durch Drücken von **[ Skip ]** übersprungen werden.

### 4.8.3. Dienste aktivieren

Zusätzliche Systemdienste, die zur Startzeit aktiviert werden sollen, können im folgenden Menü eingeschaltet werden. All diese Dienste sind optional. Starten Sie nur die Dienste, die zur korrekten Funktion des Systems benötigt werden.

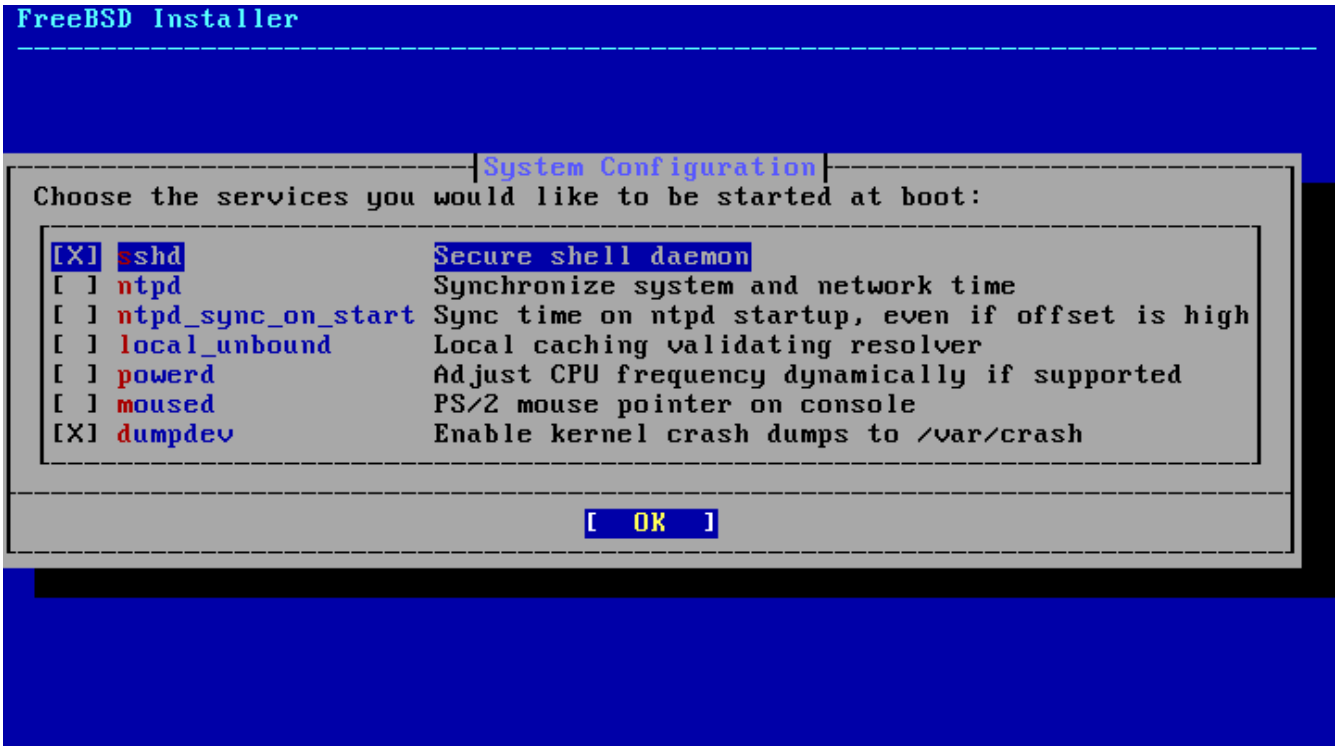



*Abbildung 40. Auswahl zusätzlicher Dienste*

Die folgenden Dienste können über dieses Menü aktiviert werden:

- `local_unbound` - Aktiviert den lokalen unbound DNS-Cache. Bedenken Sie, dass dies der Unbound des Basissystems ist und nur als lokaler Cache-Forwarding-Resolver gedacht ist. Möchten Sie einen DNS-Server für das gesamte Netzwerk einrichten, installieren Sie bitte dns/unbound.
- `sshd` - Der Secure Shell (SSH)-Daemon für Fernzugriff über eine verschlüsselte Verbindung. Aktivieren Sie diesen Dienst nur dann, wenn das System für Fernzugriff zur Verfügung stehen soll.
- `moused` - Aktivieren Sie diesen Dienst, wenn Sie Mausunterstützung auf der Systemkonsole benötigen.
- `ntpdate` - Aktiviert die automatische Synchronisation der Uhrzeit beim booten. Diese Funktionalität ist ebenfalls im ntpd(8)-Daemon verfügbar. In naher Zukunft soll das Programm ntpdate(8) entfernt werden.
- `ntpd` - Der Network Time Protocol (NTP)-Daemon zur automatischen Uhrzeitsynchronisation. Aktivieren Sie diesen Dienst, wenn es im Netzwerk einen Windows®-, Kerberos- oder LDAP-Server gibt.
- `powerd` - Systemwerkzeug zur Leistungsregelung und für Stromsparfunktionen.
- `dumpdev` - Aktiviert die Absturzaufzeichnung, welche sehr nützlich sein kann, um Systemfehler aufzuspüren. Daher wird Anwendern empfohlen, diese Option zu aktivieren.

### 4.8.4. Aktivieren von Sicherheitsoptionen

Im nächsten Menü können Sicherheitsoptionen aktiviert werden. Alle diese Optionen sind optional. Es wird jedoch empfohlen, sie zu aktivieren.

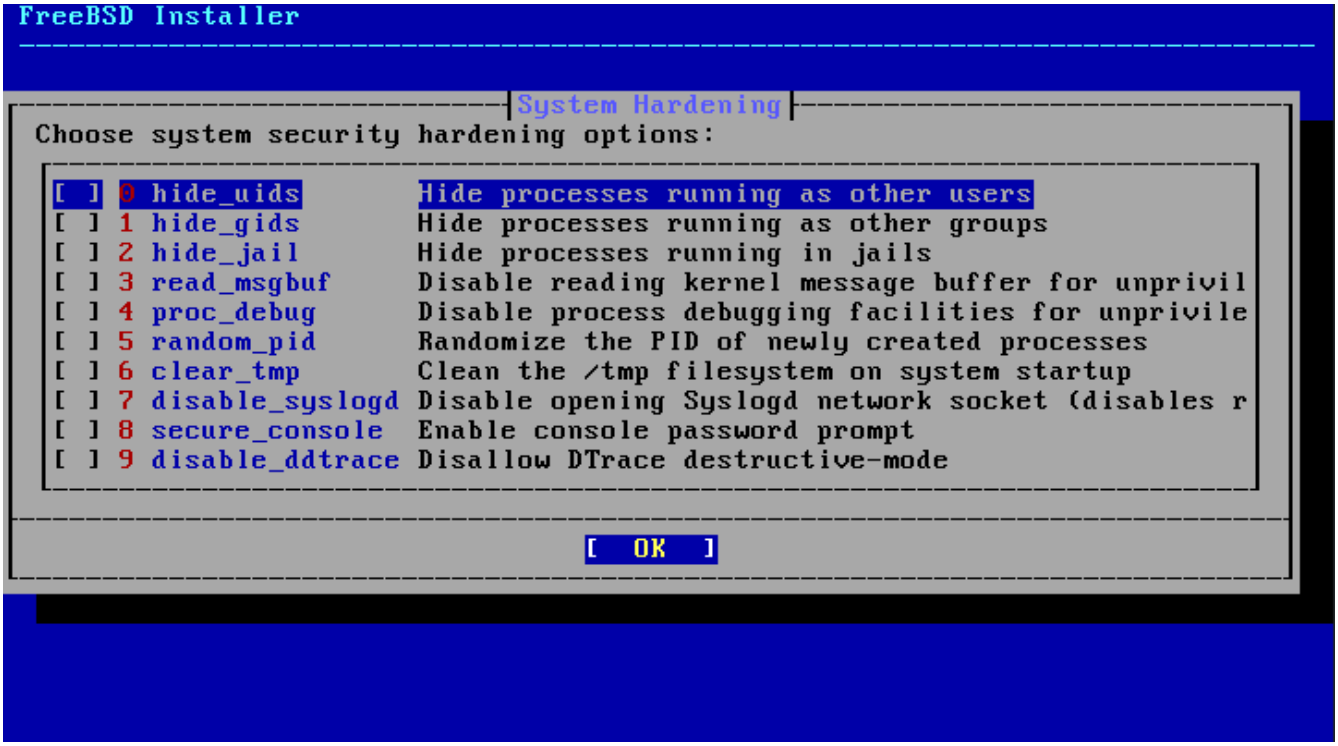



*Abbildung 41. Auswahl der Sicherheitsoptionen*

Folgende Optionen können in diesem Menü aktiviert werden:

- `hide_uids` - Versteckt die Prozesse von anderen Benutzern, um zu verhindern, dass unprivilegierte Benutzer laufende Prozesse von anderen Benutzern (UID) sehen können.
- `hide_gids` - Versteckt die Prozesse anderer Gruppen, um zu verhindern, dass unprivilegierte Benutzer laufende Prozesse von anderen Gruppen (GID) sehen können.
- `hide_jails` - Versteckt Jail-Prozesse, um zu verhindern, dass unprivilegierte Benutzer die in den Jails laufenden Prozesse sehen können.
- `read_msgbuf` - Deaktiviert den Lesezugriff auf den Nachrichtenpuffer des Kernels für nicht privilegierte Benutzer. Dadurch wird verhindert, dass dmesg(8) zum Anzeigen von Nachrichten aus dem Nachrichtenpuffer des Kernels verwendet wird.
- `proc_debug` - Die Deaktivierung von Prozess-Debugging-Funktionen für unprivilegierte Benutzer deaktiviert einige IPC-Dienste und procfs-Funktionen, ptrace() und ktrace(). Beachten Sie, dass dadurch auch die Nutzung von Werkzeugen wie lldb(1), truss(1), procstat(1) und einige Debugging-Funktionen von Skriptsprachen wie PHP, für unprivilegierte Benutzer unterbunden wird.
- `random_pid` - Zufällig generierte PID für neu erstellte Prozesse.
- `clear_tmp` - Bereinigt das Verzeichnis /tmp beim Systemstart.
- `disable_syslogd` - Diese Option verhindert, dass syslogd einen Netzwerk-Socket öffnet. In der Voreinstellung startet FreeBSD syslogd auf sichere Weise mit `-s`. Das verhindert, dass der Daemon auf Port 514 auf UDP-Anfragen lauscht. Wenn diese Option aktiviert ist, läuft syslogd mit dem Schalter `-ss`, dass syslogd daran hindert, einen Port zu öffnen. Weitere Informationen finden Sie in syslogd(8).
- `disable_sendmail` - Deaktiviert den sendmail MTA.
- `secure_console` - Wenn diese Option aktiviert ist, fragt das System im Single-User-Modus nach dem `root"`-Passwort.
- `disable_ddtrace` - DTrace kann in einem Modus laufen, der sich tatsächlich auf den laufenden Kernel auswirkt. Destruktive Aktionen dürfen nicht benutzt werden, es sei denn, sie wurden explizit aktiviert. Um diese Option bei der Verwendung von DTrace zu aktivieren, benutzen Sie `-w`. Weitere Informationen finden Sie in dtrace(1).

### 4.8.5. Benutzer hinzufügen

Das nächste Menü fordert Sie dazu auf, mindestens ein Benutzerkonto zu erstellen. Es wird empfohlen, sich als normaler Benutzer am System anzumelden und nicht als `root`-Benutzer. Wenn man als `root` angemeldet ist, gibt es so gut wie keine Beschränkungen oder Schutz vor dem, was man tun kann. Die Anmeldung als normaler Benutzer ist daher sicherer und bietet mehr Schutz.

Wählen Sie **[ Yes ]**, um neue Benutzer hinzuzufügen.




*Abbildung 42. Benutzerkonten hinzufügen*

Folgen Sie den Anweisungen und geben Sie die angeforderten Informationen für das Benutzerkonto ein. Das Beispiel in Benutzerinformationen eingeben erstellt ein Konto für den Benutzer `asample`.

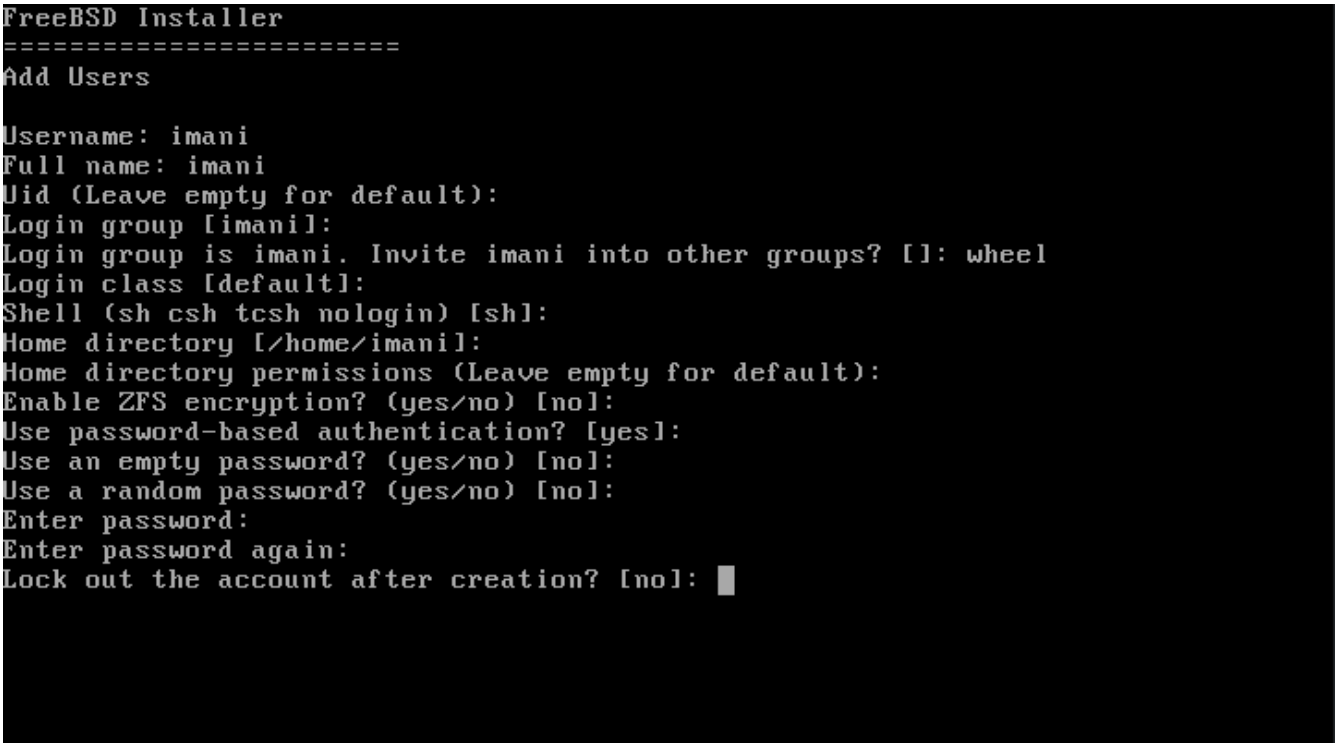



*Abbildung 43. Benutzerinformationen eingeben*

Die folgenden Informationen müssen eingegeben werden:

- `Username` - Der Name des Benutzers, den man zur Anmeldung eingeben muss. Es ist üblich, den ersten Buchstaben des Vornamens zusammen mit dem Nachnamen zu kombinieren. Jeder Benutzername ist möglich, solange er für das System einzigartig ist. Es wird zwischen Groß-

und Kleinschreibung unterschieden und der Benutzername sollte keine Leerzeichen enthalten.

- `Full name` - Der volle Name des Benutzers. Dieser darf auch Leerzeichen enthalten und dient als Beschreibung für das Benutzerkonto.
- `Uid` - User ID. Normalerweise wird dieses Feld leer gelassen, so dass das System einen Wert vergibt.
- `Login group` - Die Benutzergruppe. Normalerweise bleibt dieses Feld leer, um die Standardgruppe zu akzeptieren.
- `Invite user into other groups?` - Zusätzliche Gruppen zu denen der Benutzer als Mitglied hinzugefügt werden soll. Falls der Benutzer administrativen Zugriff benötigt, tragen Sie hier `wheel` ein.
- `Login class` - In der Regel bleibt dieses Feld leer.
- `Shell` - Die interaktive Shell für diesen Benutzer. Tragen Sie hier eine der aufgeführten Shells ein. Weitere Informationen über Shells finden Sie im “Shells”.
- `Home directory` - Das Heimatverzeichnis des Benutzers. Die Vorgabe ist für gewöhnlich richtig.
- `Home directory permissions` - Zugriffsrechte auf das Heimatverzeichnis des Benutzers. Die Vorgabe ist normalerweise die passende.
- `Use password-based authentication?` - Normalerweise `yes`, damit der Benutzer bei der Anmeldung sein Passwort eingeben muss.
- `Use an empty password?` - Normalerweise `no`, da ein leeres Passwort unsicher ist.
- `Use a random password?` - Normalerweise `no`, damit der Benutzer sein Passwort am nächsten Prompt selber vergeben kann.
- `Enter password` - Das Passwort für diesen Benutzer. Eingegebene Zeichen werden nicht am Bildschirm angezeigt.
- `Enter password again` - Das Passwort muss zur Überprüfung erneut eingegeben werden.
- `Lock out the account after creation?` - Normalerweise `no`, damit sich der Benutzer anmelden kann.

Nachdem alles eingegeben wurde, wird eine Zusammenfassung angezeigt und das System fragt Sie, dies so korrekt ist. Falls ein Eingabefehler gemacht wurde, geben Sie `no` ein und versuchen es erneut. Falls alles in Ordnung ist, geben Sie `yes` ein, um den neuen Benutzer anzulegen.

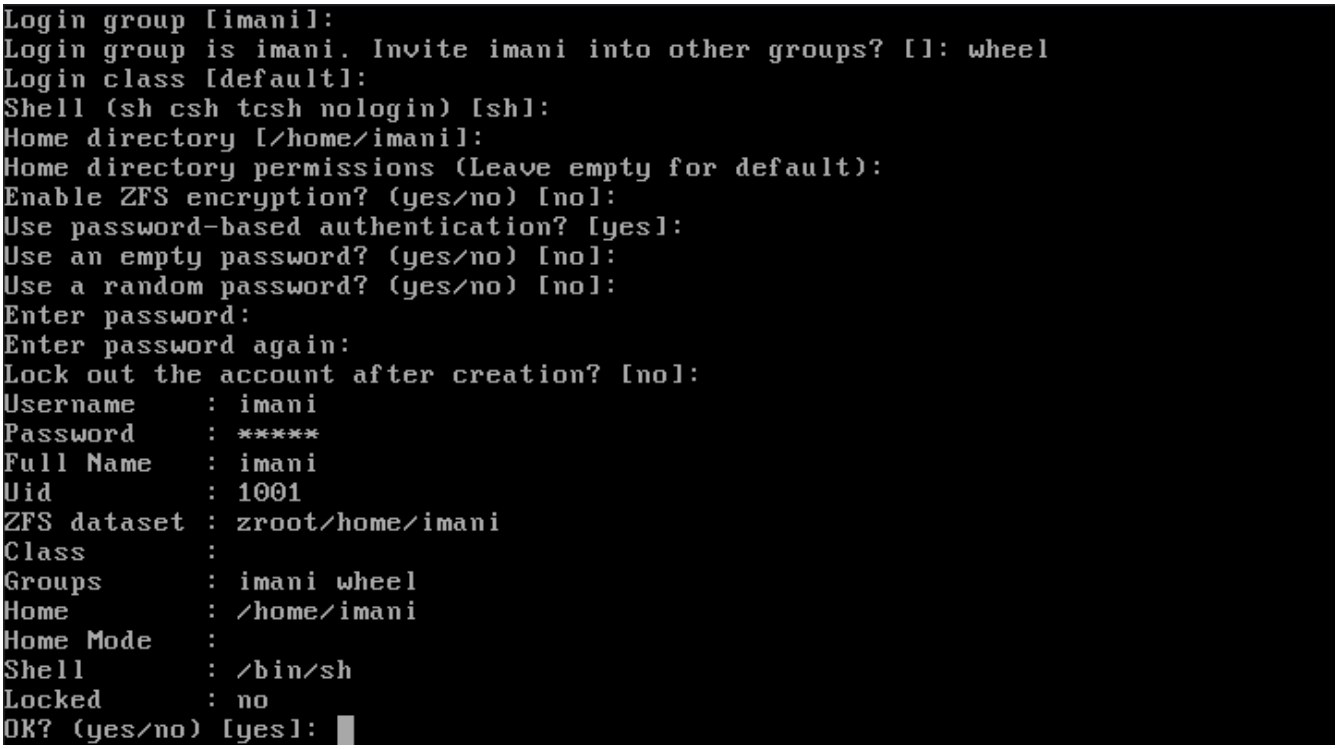



*Abbildung 44. Verlassen der Benutzer- und Gruppenverwaltung*

Falls es mehr Benutzer hinzuzufügen gibt, beantworten Sie die Frage `Add another user?` mit `yes`. Geben Sie `no` ein, wird das hinzufügen von Benutzern beendet und die Installation fortgesetzt.

Für weitere Informationen zum hinzufügen von Benutzern und deren Verwaltung, lesen Sie “Benutzer und grundlegende Account-Verwaltung”.

### 4.8.6. Letzte Konfigurationsschritte

Nachdem alles installiert und konfiguriert wurde, bekommen Sie noch eine letzte Chance, um Einstellungen zu verändern.

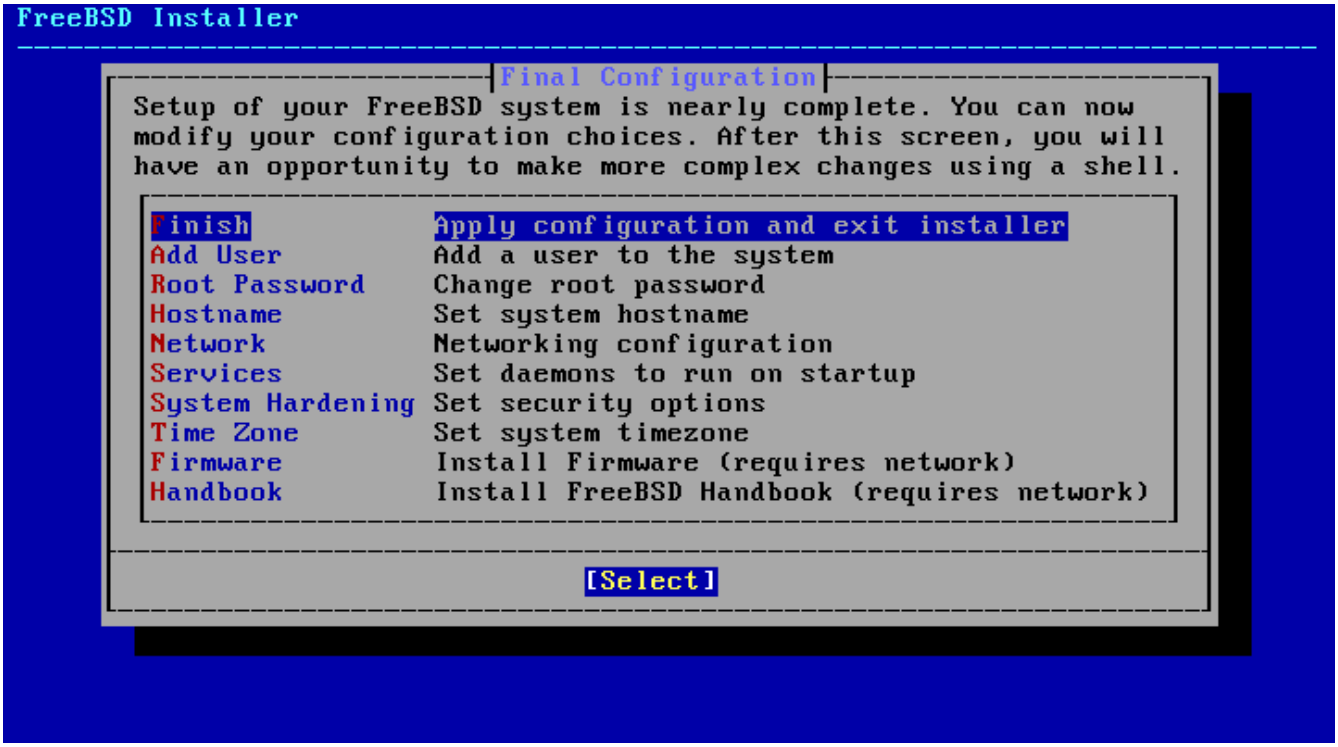



*Abbildung 45. Letzte Schritte der Konfiguration*

Verwenden Sie dieses Menü, um noch letzte Änderungen oder zusätzliche Konfigurationen vor dem Abschließen der Installation zu tätigen.

- `Add User` - Beschrieben in Benutzer hinzufügen.
- `Root Password` - Beschrieben in Setzen des `root`-Passworts.
- `Hostname` - Beschrieben in Den Rechnernamen festlegen.
- `Network` - Beschrieben in Die Netzwerkschnittstelle konfigurieren.
- `Services` - Beschrieben in Dienste aktivieren.
- `Time Zone` - Beschrieben in Setzen der Zeitzone.
- `Handbook` - Herunterladen und installieren des FreeBSD Handbuchs.

Nachdem die letzten Konfigurationsschritte beendet sind, wählen Sie **[ Exit ]**.




*Abbildung 46. Manuelle Konfiguration*

bsdinstall wird nach zusätzlichen Konfigurationen, die noch zu tätigen sind, fragen, bevor in das neue System gebootet wird. Wählen Sie **[ Yes ]**, um in eine Shell innerhalb des neuen Systems zu wechseln oder **[ No ]**, um mit dem letzten Schritt der Installation zu beginnen.

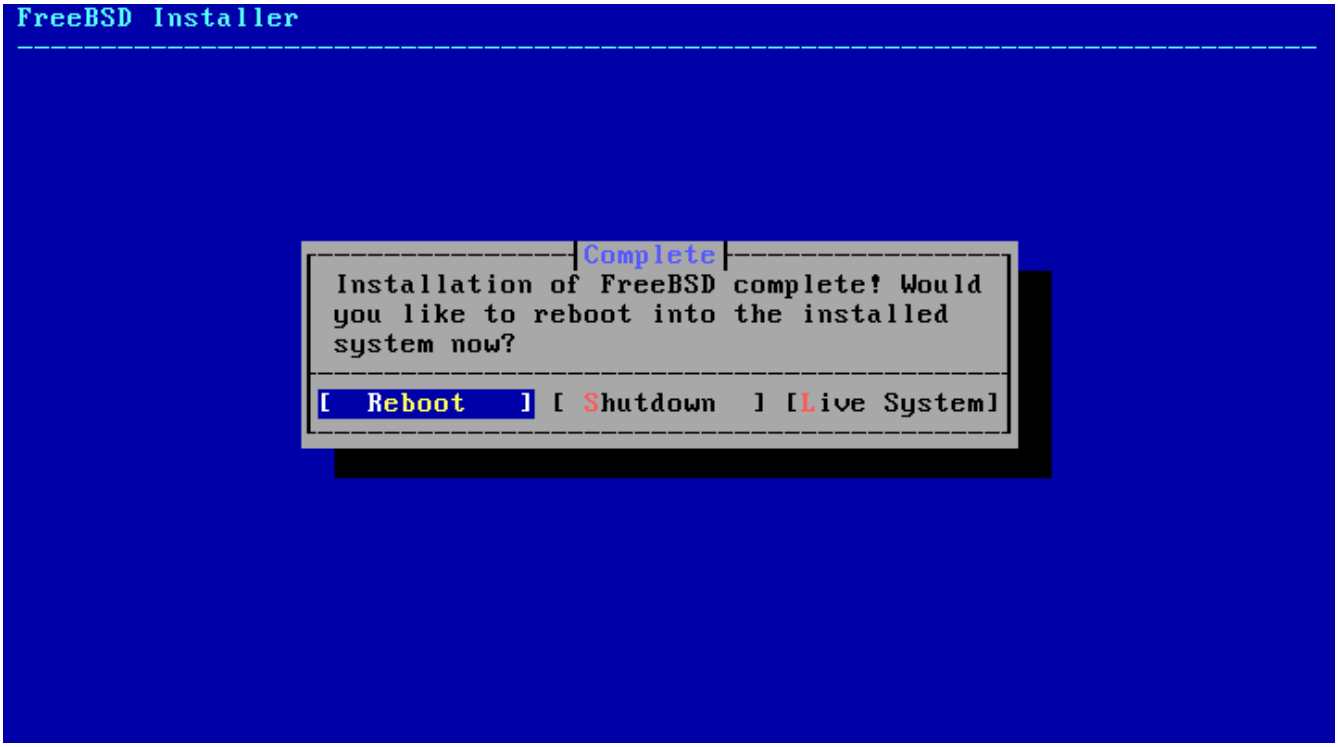



*Abbildung 47. Die Installation vervollständigen*

Wenn weitere Konfigurationen oder besondere Einstellungen benötigt werden, wählen Sie **[ Live CD ]**, um das Installationsmedium im Live-CD Modus zu starten.

Wenn die Installation vollständig ist, wählen Sie **[ Reboot ]**, um den Computer neu zu starten und das neu installierte FreeBSD-System zu booten. Vergessen Sie nicht, das FreeBSD

Installationsmedium zu entfernen, oder der Computer wird erneut davon starten.

Wenn FreeBSD startet, werden viele Informationsmeldungen ausgegeben. Nachdem das System den Startvorgang abgeschlossen hat, wird eine Anmeldeaufforderung angezeigt. Geben Sie am `login:` den Benutzernamen ein, den Sie während der Installation hinzugefügt haben. Vermeiden Sie es, sich als `root` anzumelden. Lesen Sie “Der Superuser-Account”, wenn Sie administrativen Zugriff benötigen.

Um Nachrichten, die während des Bootens angezeigt wurden, zu sehen, aktivieren Sie durch drücken von `Scroll-Lock` den *scroll-back buffer*. Die Tasten `PgUp`, `PgDn` und die Pfeiltasten dienen zur Navigation durch die Nachrichten. Durch erneutes drücken von `Scroll-Lock` wird der Bildschirm wieder entsperrt und kehrt zur normalen Anzeige zurück. Mit `less /var/run/dmesg.boot` können Sie sich diese Nachrichten im laufenden Betrieb ansehen. Durch drücken von `q` kehren Sie wieder zur Kommandozeile zurück.

Wenn sshd in Auswahl zusätzlicher Dienste aktiviert wurde, ist der erste Start ein bisschen langsamer, weil das System die RSA- und DSA-Schlüssel erzeugen muss. Die nachfolgenden Startvorgänge werden dann wieder schneller sein. Wie in diesem Beispiel zu sehen ist, werden die Fingerabdrücke der Schlüssel am Bildschirm ausgegeben:

```
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ssh/ssh_host_key.pub.
The key fingerprint is:
10:a0:f5:af:93:ae:a3:1a:b2:bb:3c:35:d9:5a:b3:f3 root@machine3.example.com
The key's randomart image is:
+--[RSA1 1024]----+
|    o..          |
|   o . .         |
|  .   o          |
|       o         |
|    o   S        |
|   + + o         |
|o . + *          |
|o+ ..+ .         |
|==o..o+E         |
+-----------------+
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
7e:1c:ce:dc:8a:3a:18:13:5b:34:b5:cf:d9:d1:47:b2 root@machine3.example.com
The key's randomart image is:
+--[ DSA 1024]----+
|       ..     . .|
|      o  .   . + |
|     . ..   . E .|
|    . .  o o . . |
|     +  S = .    |
|    +  . = o     |
```

```
|     +  . * .    |
|    . .  o .     |
|      .o. .      |
+-----------------+
Starting sshd.
```

Lesen Sie OpenSSH für weitere Informationen zu Fingerabdrücken und SSH.

FreeBSD installiert standardmäßig keine graphische Umgebung. Das X-Window-System enthält Informationen zur Installation und Konfiguration eines graphischen Window Managers.

Das korrekte herunterfahren eines FreeBSD-Computers hilft, beugt dem Datenverlust vor und schützt sogar die Hardware vor Schäden. *Schalten Sie nicht den Strom ab, bevor das System ordnungsgemäß heruntergefahren wurde!* Wenn der Benutzer ein Mitglied der `wheel`-Gruppe ist, können Sie zum Superuser durch die Eingabe von `su` und der anschließenden Eingabe des Passworts von `root` werden. Geben Sie dann `shutdown -p now` ein. Das System wird jetzt sauber heruntergefahren und, falls die Hardware es unterstützt, den Rechner ausschalten.

# 4.9. Netzwerkschnittstellen

## 4.9.1. Die Netzwerkschnittstelle konfigurieren

Als nächstes wird eine Liste der gefundenen Netzwerkschnittstellen gezeigt. Wählen Sie die Schnittstelle aus, die Sie konfigurieren möchten.

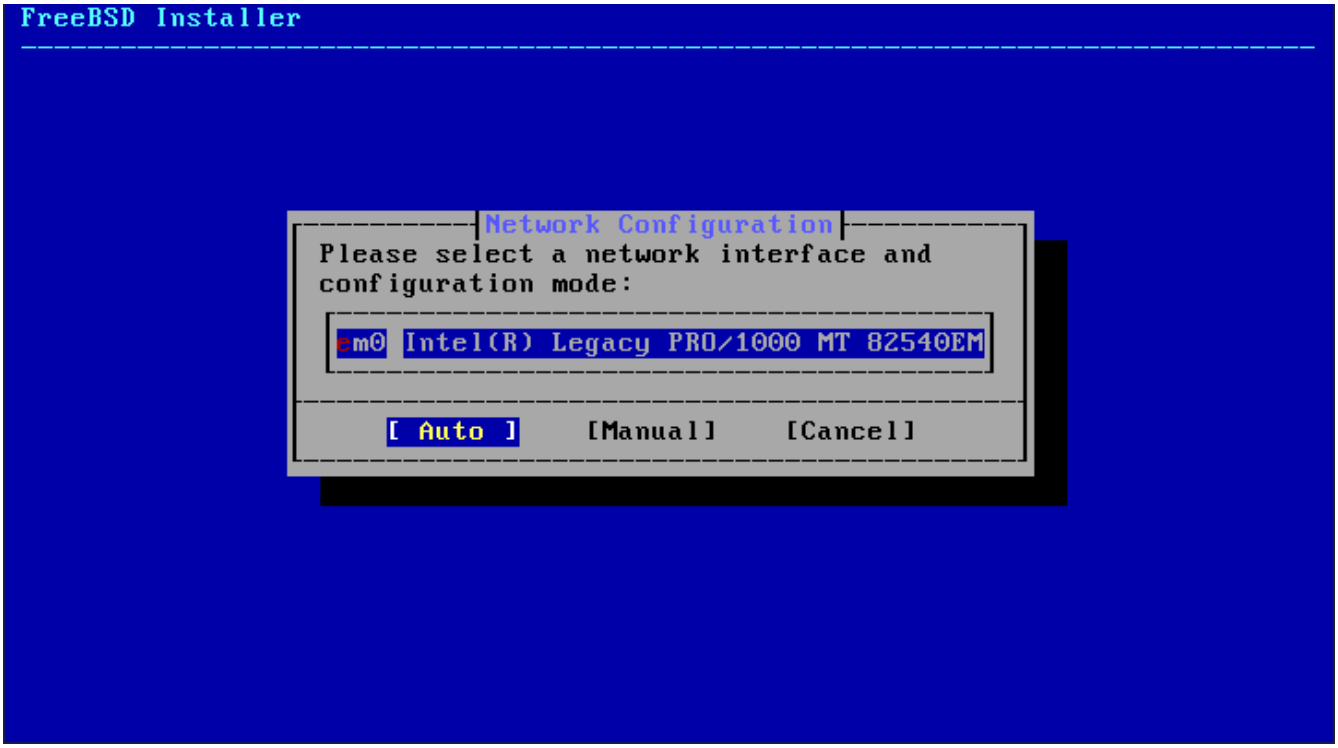



*Abbildung 48. Eine zu konfigurierende Netzwerkschnittstelle auswählen*

Wenn Sie eine Ethernet-Schnittstelle ausgewählt haben, fährt das Installationsprogramm mit dem Menü aus Auswahl von IPv4 fort. Wenn Sie eine drahtlose Netzwerkschnittstelle ausgewählt haben, wird das System nach drahtlosen Zugriffspunkten (Access Points) suchen:

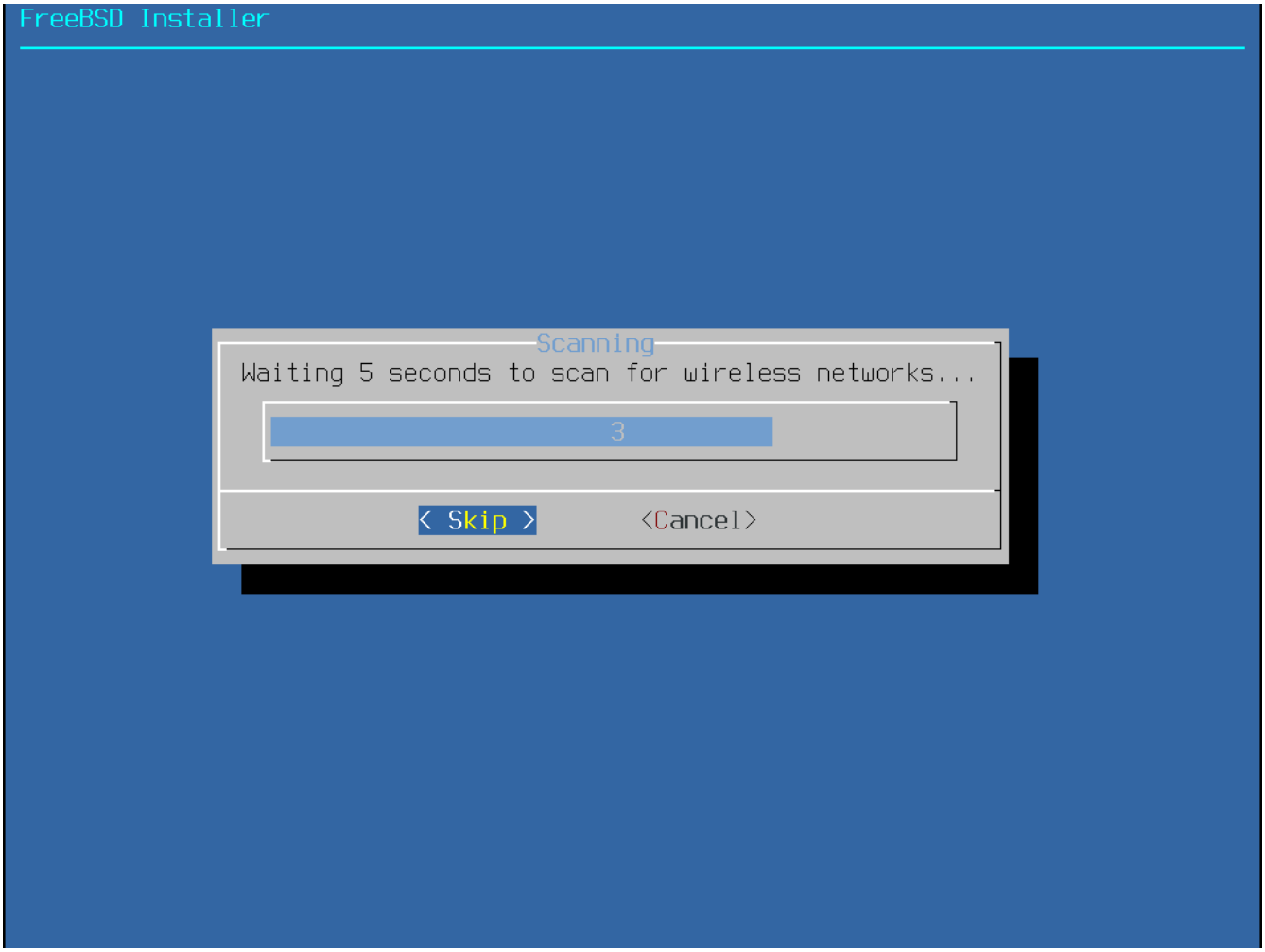



*Abbildung 49. Nach drahtlosen Access Points scannen*

Drahtlose Netzwerke werden durch einen Service Set Identifier (SSID) identifiziert. Der SSID ist ein kurzer, eindeutiger Name, der für jedes Netzwerk vergeben wird. SSIDs, die während des Scans gefunden wurden, werden aufgelistet, gefolgt von einer Beschreibung der Verschlüsselungsarten, die für dieses Netzwerk verfügbar sind. Falls die gewünschte SSID nicht in der Liste auftaucht, wählen Sie **[ Rescan ]**, um erneut einen Scanvorgang durchzuführen. Falls dann das gewünschte Netzwerk immer noch nicht erscheint, überprüfen Sie die Antenne auf Verbindungsprobleme oder versuchen Sie, näher an den Access point zu gelangen. Scannen Sie erneut nach jeder vorgenommenen Änderung.

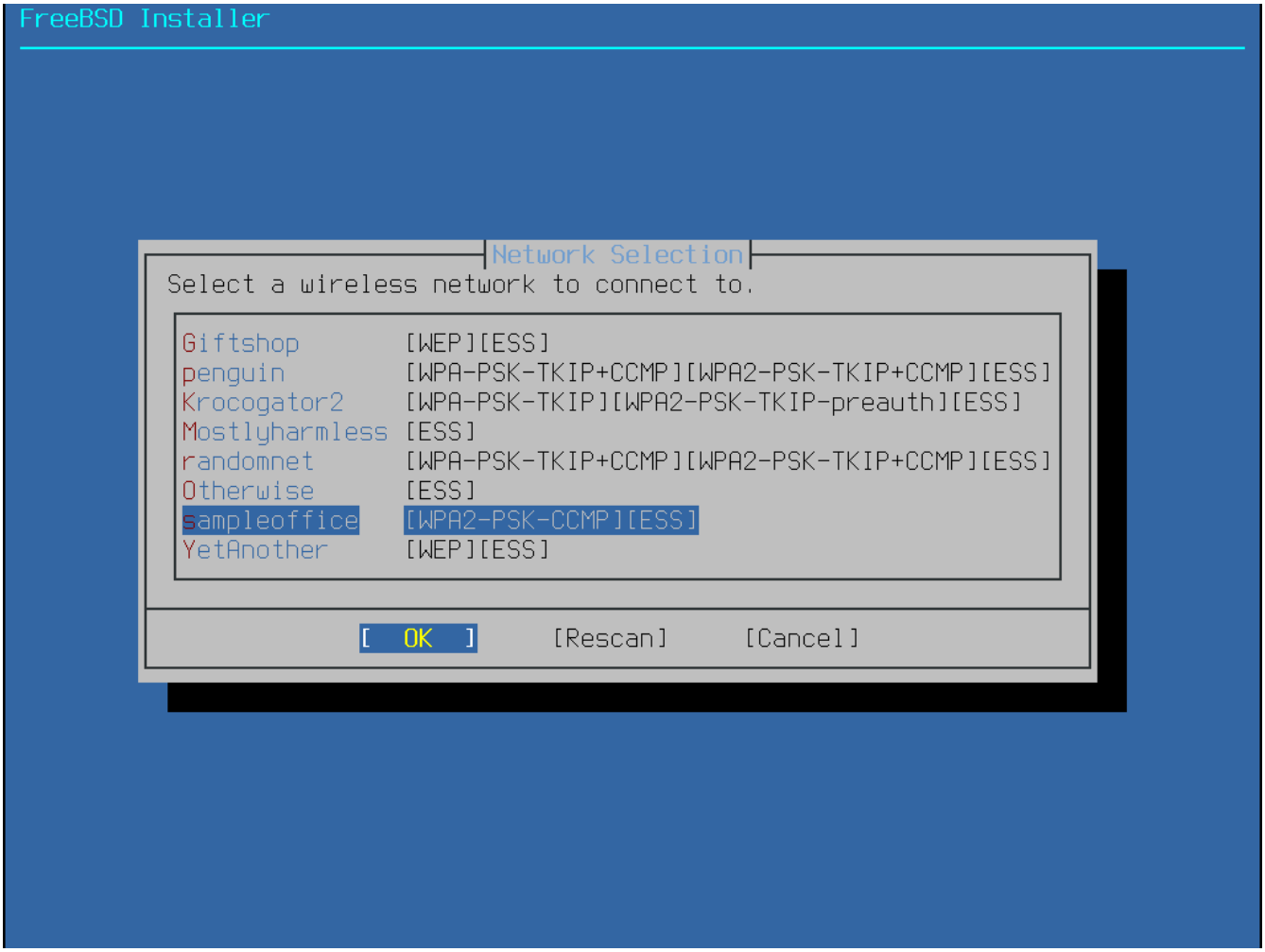



*Abbildung 50. Ein drahtloses Netzwerk auswählen*

Geben Sie nun die Verschlüsselungsinformationen ein, um sich mit dem drahtlosen Netzwerk zu verbinden. WPA2 wird als Verschlüsselung dringend empfohlen, da ältere Verschlüsselungsmethoden, wie WEP, nur wenig Sicherheit bieten. Wenn das Netzwerk WPA2 verwendet, geben Sie das Passwort (auch bekannt als Pre-Shared Key PSK) ein. Aus Sicherheitsgründen werden die in das Eingabefeld eingegeben Zeichen nur als Sternchen angezeigt.

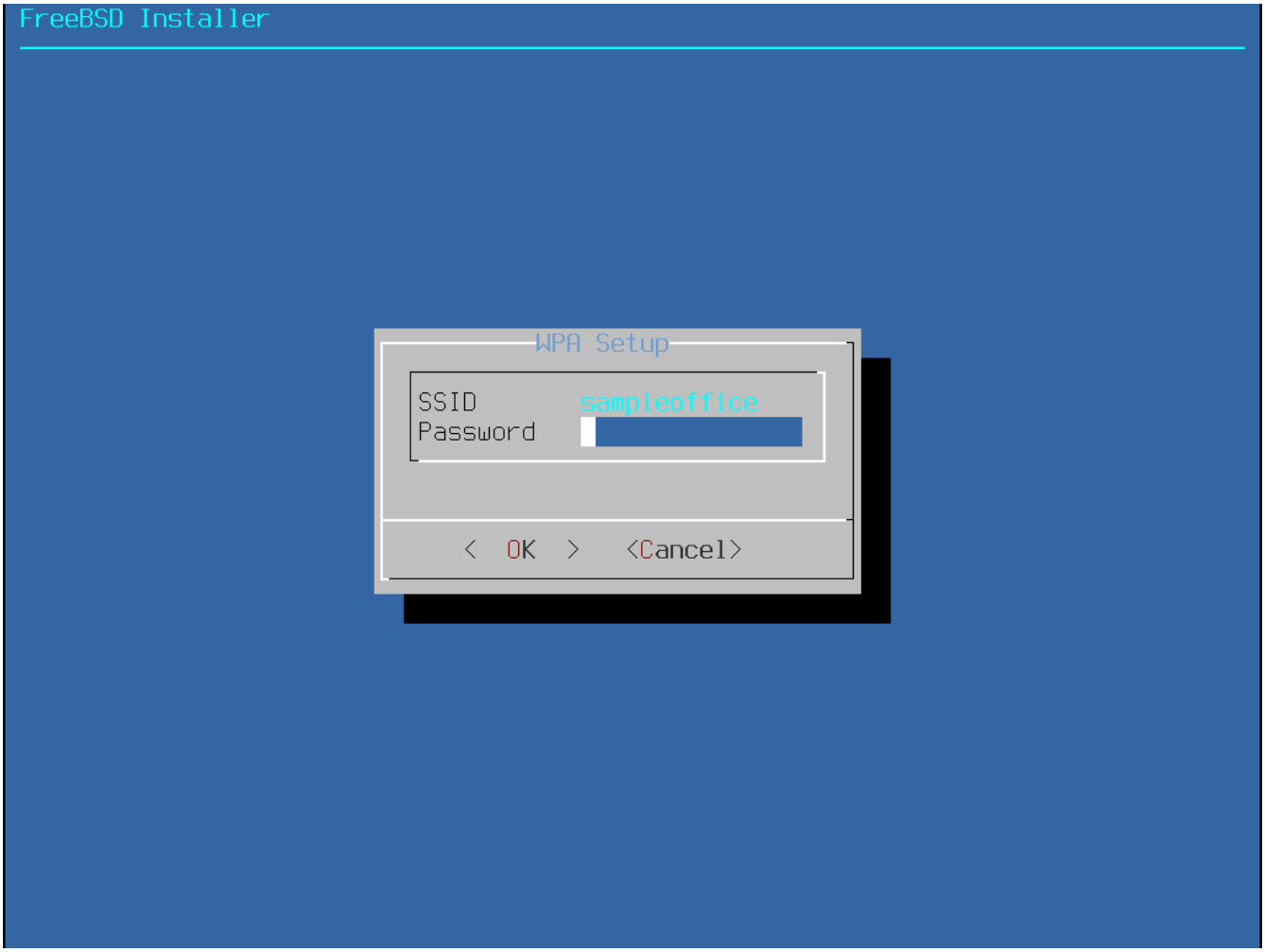



*Abbildung 51. Verbindungsaufbau mit WPA2*

Wählen Sie, ob eine IPv4-Adresse auf der Ethernet-Schnittstelle oder der drahtlosen Schnittstelle konfiguriert werden soll.

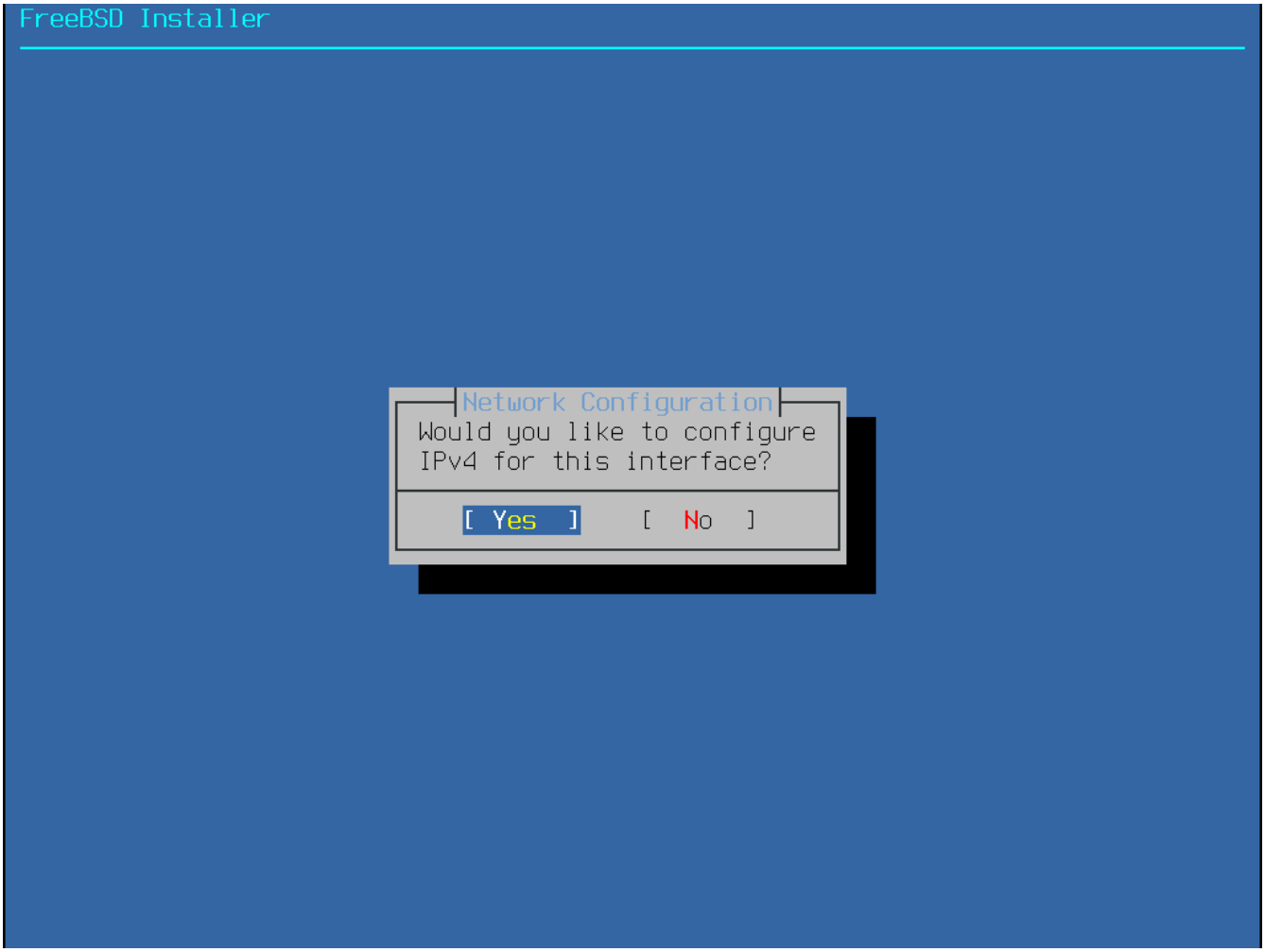



*Abbildung 52. Auswahl von IPv4*

Es gibt zwei Arten, ein IPv4-Netzwerk zu konfigurieren. DHCP wird automatisch die Netzwerkschnittstelle richtig konfigurieren und sollte verwendet werden, wenn das Netzwerk über einen DHCP-Server verfügt. Eine *statische* IP-Konfiguration erfordert die manuelle Eingabe von Netzwerkinformationen.



Geben Sie keine zufällig gewählten Netzwerkinformationen ein, da dies nicht funktionieren wird. Holen Sie sich die in Erforderliche Informationen zum Netzwerk gezeigten Informationen vom Netzwerkadministrator oder Serviceprovider, falls kein DHCP-Server verfügbar ist.

Falls ein DHCP-Server zur Verfügung steht, wählen Sie im nächsten Menü **[ Yes ]**, um die Netzwerkschnittstelle automatisch einrichten zu lassen. Dieser Vorgang kann einige Sekunden dauern.

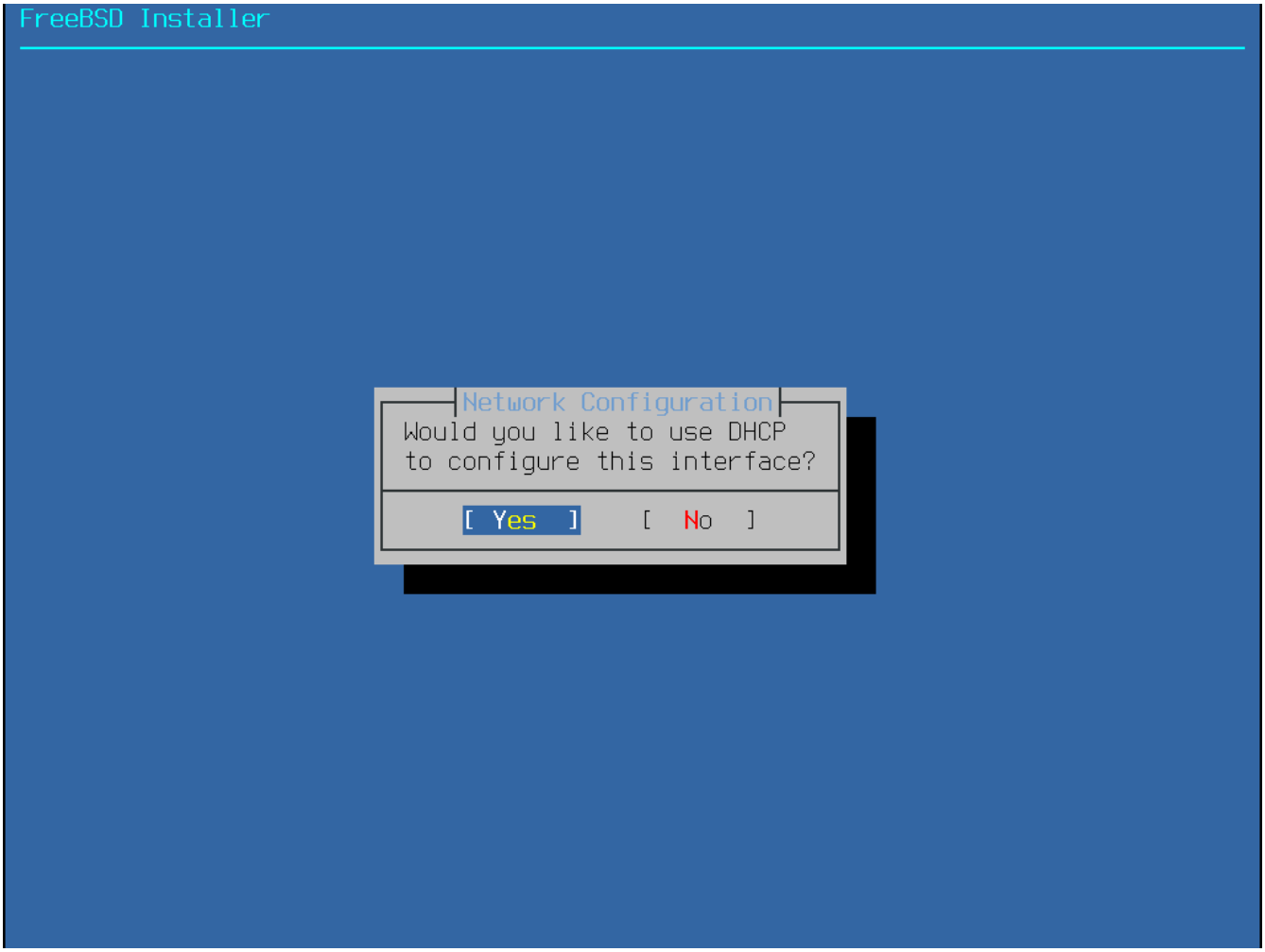



*Abbildung 53. Auswählen der IPv4-Konfiguration über DHCP*

Wenn kein DHCP-Server zur Verfügung steht, wählen Sie **[No]** und tragen Sie die folgenden Informationen in das Menü ein:

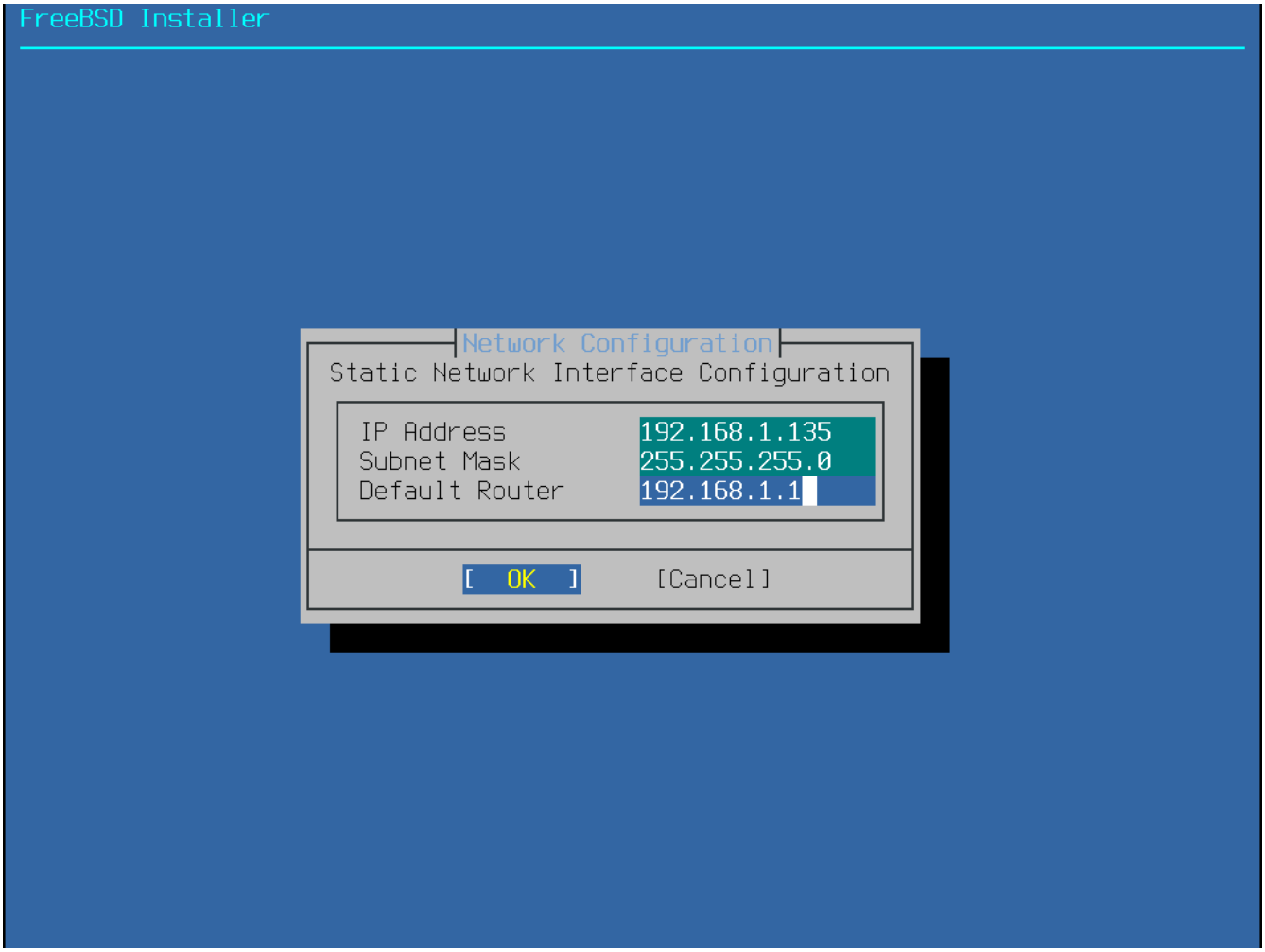



*Abbildung 54. Statische IPv4-Konfiguration*

- `IP Address` - Die IPv4-Adresse, welche diesem Computer zugewiesen werden soll. Diese Adresse muss eindeutig sein und darf nicht bereits von einem anderen Gerät im lokalen Netzwerk verwendet werden.
- `Subnet Mask` - Die Subnetzmaske des Netzwerks.
- `Default Router` - Die IP-Adresse des Defaultrouters im Netzwerk.

Das nächste Menü fragt, ob die Schnittstelle für IPv6 konfiguriert werden soll. Falls IPv6 verfügbar ist und verwendet werden soll, wählen Sie **[ Yes ]** aus.

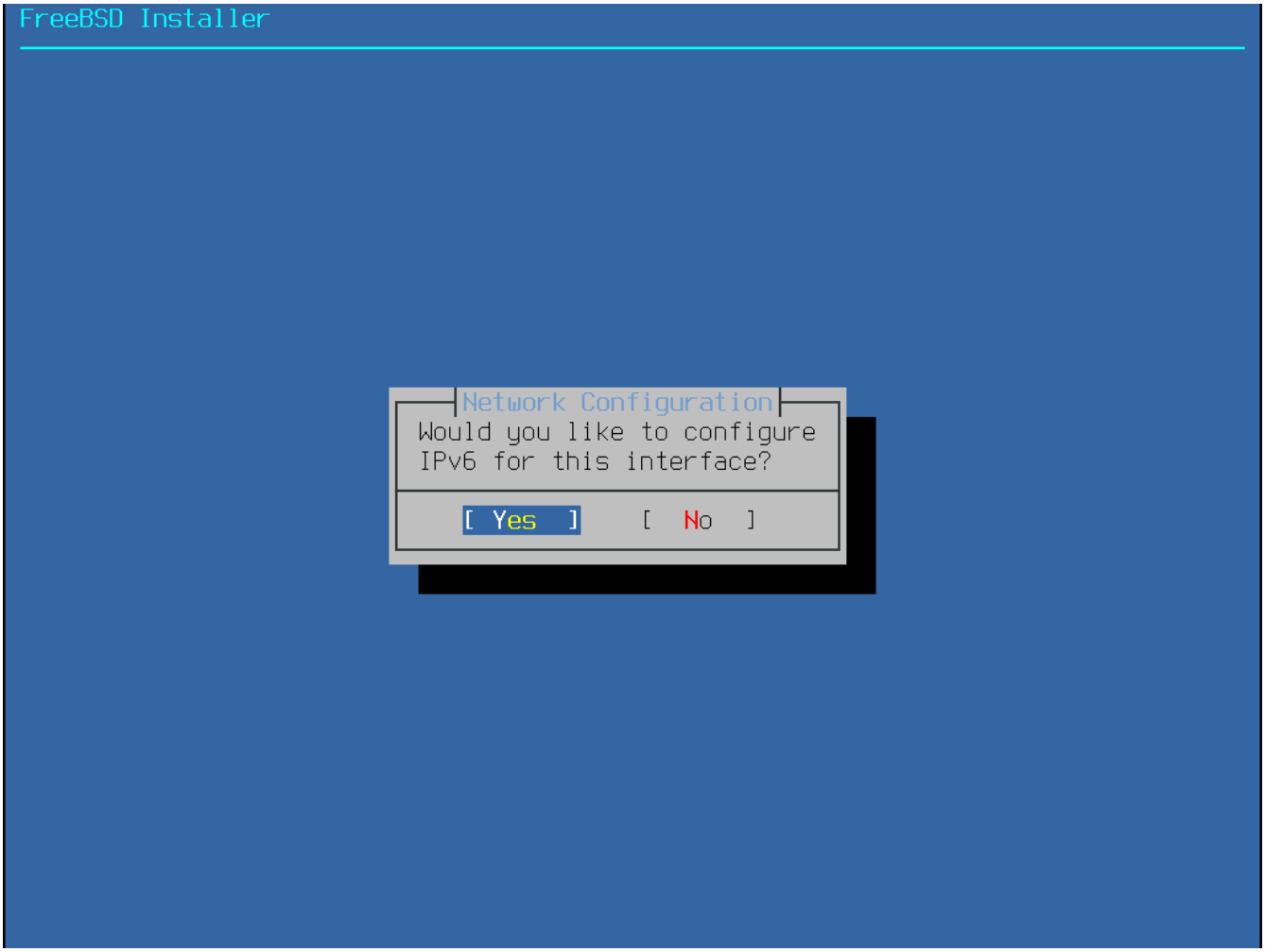



*Abbildung 55. Auswahl von IPv6*

IPv6 besitzt ebenfalls zwei Arten der Konfiguration. *StateLess Address AutoConfiguration*, (SLAAC) wird automatisch die richtigen Informationen von einem lokalen Router abfragen. Lesen Sie http://tools.ietf.org/html/rfc4862 für weitere Informationen. Eine *statische* Konfiguration verlangt die manuelle Eingabe von Netzwerkinformationen.

Wenn ein IPv6-Router verfügbar ist, wählen Sie im nächsten Menü **[ Yes ]**, um die Netzwerkschnittstelle automatisch konfigurieren zu lassen.

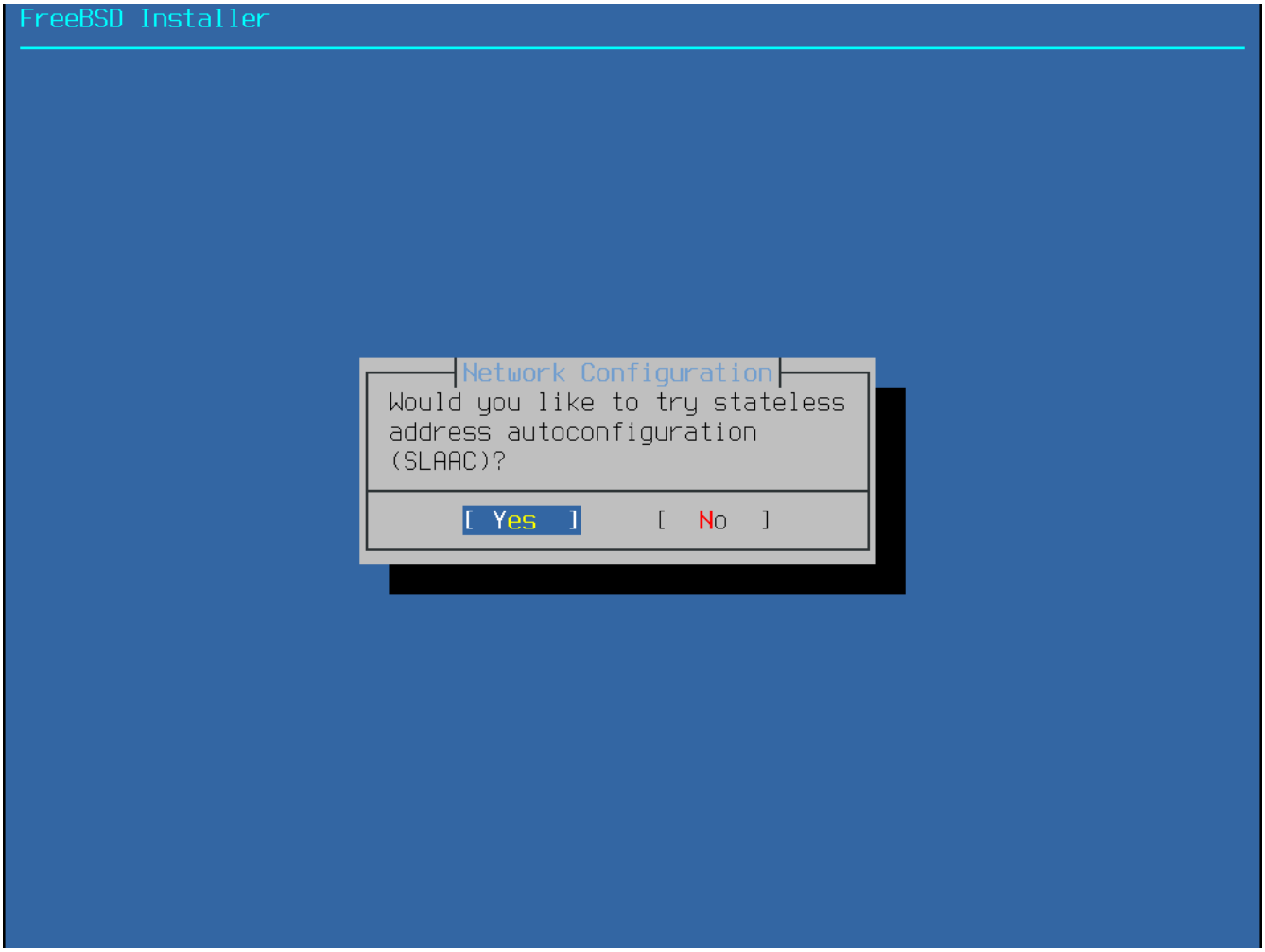



*Abbildung 56. Auswahl der IPv6SLAAC-Konfiguration*

Wenn kein IPv6-Router zur Verfügung steht, wählen Sie **[No]** und tragen Sie die folgenden Adressinformationen in dieses Menü ein:

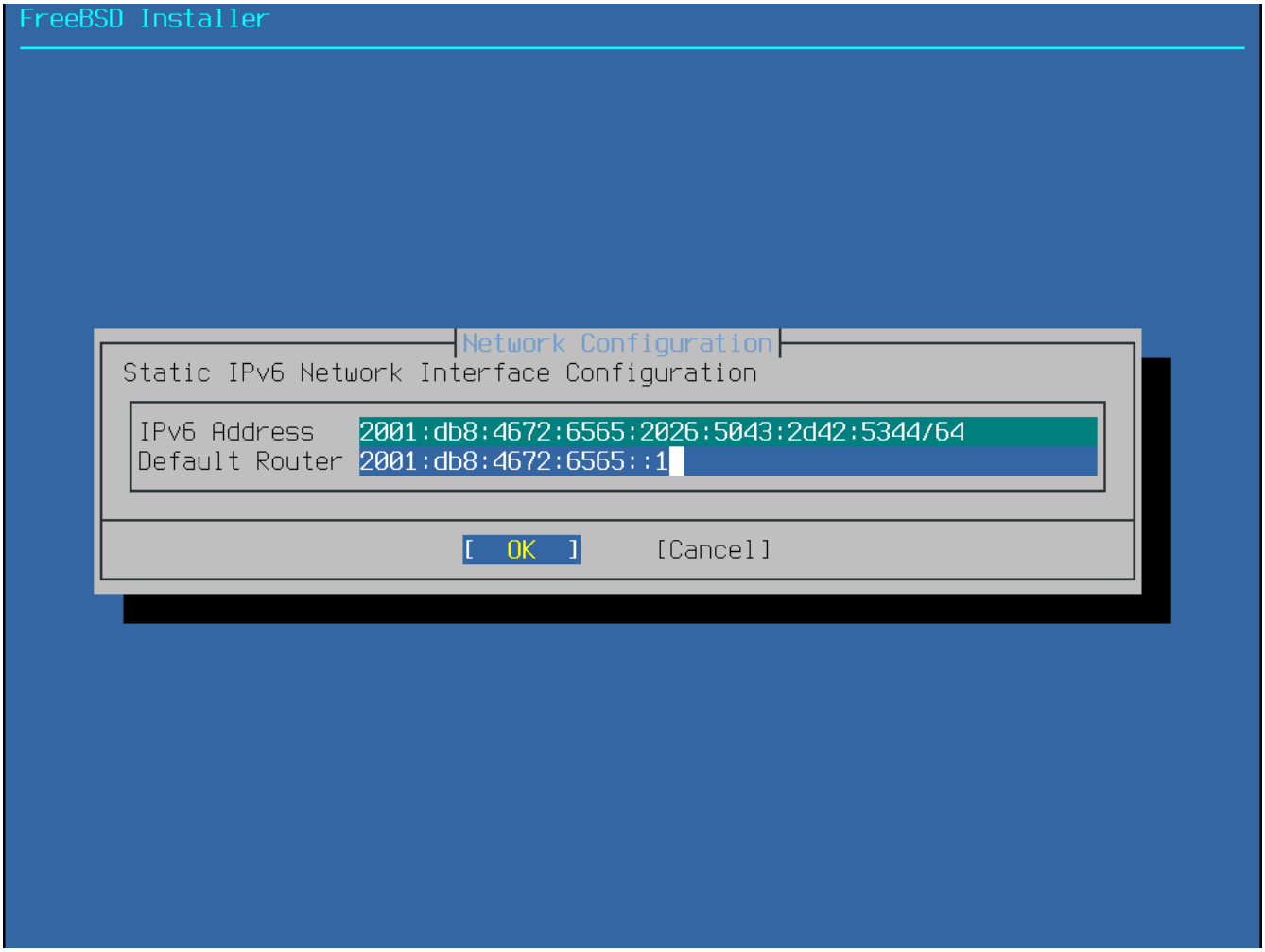



*Abbildung 57. Statische IPv6-Konfiguration*

- `IPv6 Address` - Die zugewiesene IPv6-Adresse, welche dem Computer zugeteilt werden soll. Diese Adresse muss eindeutig sein und nicht bereits von einer anderen Netzwerkkomponente im lokalen Netzwerk verwendet werden.
- `Default Router` - Die IPv6-Adresse des Defaultrouters im Netzwerk.

Das letzte Menü der Netzwerkkonfiguration konfiguriert den *Domain Name System* (DNS) Resolver, welcher Hostnamen von und zu Netzwerkadressen umwandelt. Falls DHCP oder SLAAC verwendet wurde, um die Netzwerkschnittstelle zu konfigurieren, ist die Konfiguration für den Resolver möglicherweise bereits eingetragen. Andernfalls geben Sie den lokalen Netzwerkdomänennamen in das Feld `Search` ein. `DNS #1` und `DNS #2` sind die IPv4- und/oder IPv6-Adressen der lokalen DNS-Server. Zumindest ein DNS-Server wird benötigt.

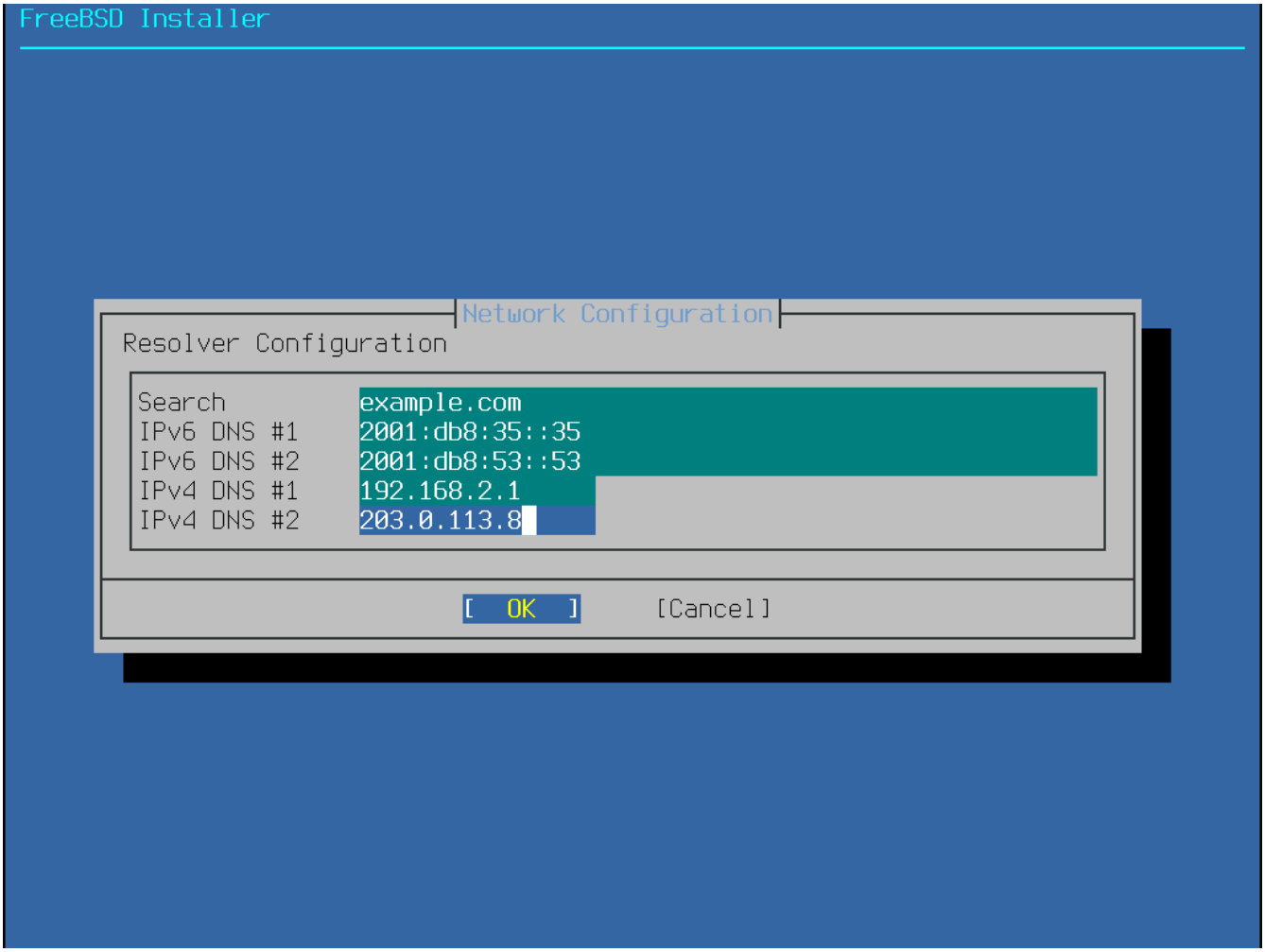



*Abbildung 58. DNS-Konfiguration*

Sobald die Schnittstelle konfiguriert ist, bestimmen Sie einen Spiegelserver, welcher in der gleichen Region auf der Welt beheimatet ist, wie der Computer, auf dem FreeBSD installiert wird. Dateien können so viel schneller übertragen werden, wenn der Spiegelserver sich näher am Zielcomputer befindet und die Installationszeit wird somit reduziert.




*Abbildung 59. Einen Spiegelserver wählen*

# 4.10. Fehlerbehebung

Dieser Abschnitt behandelt einfache Fehlerbehebungen für die Installation, wie beispielsweise häufig auftretende Fehler, die von Anwendern berichtet wurden.

Überprüfen Sie die Hardware Notes (https://www.freebsd.org/releases/) nach der Version von FreeBSD, um sicher zu stellen, dass die Hardware auch unterstützt wird. Wenn die Hardware unterstützt wird und Sie immer noch Abstürze oder andere Probleme erleben, müssen Sie einen eigenen Kernel bauen. Diese Prozedur wird in Konfiguration des FreeBSD-Kernels beschrieben. Das erlaubt es, Unterstützung für Geräte, die im GENERIC-Kernel nicht vorhanden sind, hinzuzufügen. Der Kernel ist mit der Annahme konfiguriert, dass die Hardwaregeräte sich in ihren Fabrikeinstellungen in Bezug auf IRQs, I/O-Adressen und DMA-Kanälen befinden. Wenn die Hardware neu konfiguriert wurde, werden Sie möglicherweise die Konfiguration des Kernels bearbeiten und diesen neu erstellen müssen, um FreeBSD mitzuteilen, wo es gewisse Dinge finden kann.



Manche Installationsprobleme können Aktualisierung der Firmware auf verschiedenen Hardwarekomponenten verhindert oder verringert werden, meistens am Mainboard. Mit Mainboard-Firmware ist für gewöhnlich das BIOS gemeint. Die meisten Mainboard- und Computerhersteller haben eine Webseite mit Aktualisierungen und Informationen zur Durchführung.

Hersteller raten meist von einer Aktualisierung des Mainboard-BIOS ab, außer es gibt einen guten Grund dafür, wie beispielsweise eine kritische Aktualisierung. Der Aktualisierungsvorgang *kann* schiefgehen, was das BIOS unvollständig macht und den Computer nicht mehr starten lässt.

Wenn das System während der Geräteerkennung beim Starten hängt oder sich während der

Installation merkwürdig verhält, ist ACPI vielleicht der Übeltäter. FreeBSD macht auf i386- und amd64-Plattformen starken Gebrauch vom ACPI-Dienst, um dem System bei der Konfiguration während des Startvorgangs zu helfen. Leider existieren immer noch Fehler im ACPI-Treiber, in den Mainboards und der BIOS-Firmware. ACPI kann durch setzen der Einstellung `hint.acpi.0.disabled` im dritten Teil des Bootloaders deaktiviert werden:

```
set hint.acpi.0.disabled="1"
```

Dies wird nach jedem Neustart des Systems wieder zurückgesetzt, also ist es notwendig, die Zeile `hint.acpi.0.disabled="1"` zu der Datei /boot/loader.conf hinzuzufügen. Weitere Informationen über den Bootloader lassen sich in “Übersicht” nachlesen.

## 4.11. Verwendung der Live-CD

Das Willkommensmenü von bsdinstall, welches in Willkommen-Menü gezeigt wird, enthält eine **[ Live CD ]** Option. Die Live-CD ist für Benutzer, die sich fragen, ob FreeBSD das richtige Betriebssystem für sie ist und die vor der Installation noch einige Merkmale und Eigenschaften testen wollen.

Die folgenden Punkte sollten beachtet werden, bevor die **[ Live CD ]** benutzt wird:

- Um Zugriff auf das System zu bekommen, wird eine Authentifizierung benötigt. Der Benutzername ist `root` und das Kennwort bleibt leer.
- Da das System direkt von dem Installationsmedium ausgeführt wird, ist die Geschwindigkeit deutlich langsamer als bei einem System, das auf einer Festplatte installiert ist.
- Diese Option enthält nur eine Eingabeaufforderung und keine graphische Oberfläche.

# Kapitel 5. Grundlagen des FreeBSD Betriebssystems

## 5.1. Übersicht

Dieses Kapitel umfasst die grundlegenden Kommandos und Funktionsweisen des FreeBSD-Betriebssystems. Viel von diesem Material gilt auch für jedes andere UNIX®-artige System. Neue Benutzer von FreeBSD sollten dieses Kapitel aufmerksam lesen.

Dieser Abschnitt behandelt die folgenden Themen:

- virtuelle Konsolen,
- Erstellung und Verwaltung von Benutzern und Gruppen in FreeBSD,
- Zugriffsrechte unter UNIX® sowie Datei-Flags unter FreeBSD,
- Zugriffskontrolllisten für Dateisysteme,
- die Verzeichnisstruktur von FreeBSD,
- Organisation von Dateisystemen unter FreeBSD,
- Ein- und Abhängen von Dateisystemen,
- Prozesse, Dämonen und Signale,
- Shells und die Login-Umgebung,
- Texteditoren,
- Geräte und Gerätedateien,
- wie Sie in den Manualpages nach weiteren Informationen suchen können.

## 5.2. Virtuelle Konsolen und Terminals

Wenn das FreeBSD-System so konfiguriert wurde, dass es ohne eine grafische Benutzeroberfläche startet, wird das System nach dem Start einen Anmeldeprompt ausgeben, wie in diesem Beispiel zu sehen:

```
FreeBSD/amd64 (pc3.example.org) (ttyv0)

login:
```

Die erste Zeile enthält einige Informationen über das System. `amd64` zeigt an, dass auf dem System in diesem Beispiel eine 64-Bit Version von FreeBSD läuft. Der Hostname ist `pc3.example.org` und ttyv0 gibt an, dass dies die "Systemkonsole" ist. Die zweite Zeile zeigt den Anmeldeprompt.

Da FreeBSD ein Mehrbenutzersystem ist, muss es die verschiedenen Benutzer voneinander unterscheiden können. Dies wird dadurch erreicht, dass sich jeder Benutzer zuerst am System anmelden muss, um Zugriff auf die Programme zu bekommen. Jeder Benutzer hat einen

eindeutigen "Benutzernamen" und ein persönliches "Kennwort".

Um sich auf der Systemkonsole anzumelden, geben Sie den Benutzernamen ein, der während der Systeminstallation, wie in Benutzer hinzufügen beschrieben, konfiguriert wurde und drücken Sie `Enter`. Geben Sie dann das zum Benutzernamen zugeordnete Passwort ein und drücken `Enter`. Das Passwort wird aus Sicherheitsgründen *nicht angezeigt*.

Sobald das richtige Passwort eingegeben wird, wird die Nachricht des Tages (MOTD) gefolgt von einer Eingabeaufforderung ausgegeben. In Abhängigkeit der verwendeten Shell des Benutzers wird der Prompt mit dem Zeichen `#`, `$` oder `%` dargestellt. Der Prompt zeigt an, dass der Benutzer jetzt an der FreeBSD Systemkonsole angemeldet ist und nun alle verfügbaren Befehle probieren kann.

### 5.2.1. Virtuelle Konsolen

Obwohl die Systemkonsole dazu verwendet werden kann, um mit dem System zu interagieren, wird sich ein Benutzer in der Regel an einer virtuellen Konsole im FreeBSD-System anmelden. Das liegt daran, dass die Systemmeldungen standardmäßig auf der Systemkonsole angezeigt werden und somit die Meldungen des Befehls oder einer Datei, die der Benutzer gerade bearbeitet, überschrieben werden.

In der Voreinstellung ist FreeBSD so konfiguriert, dass viele virtuelle Konsolen zur Eingabe von Befehlen zur Verfügung stehen. Jede virtuelle Konsole verfügt über einen eigenen Anmeldeprompt und eine Shell. Sie können ganz einfach zwischen den virtuellen Konsolen umschalten. Dies ist vergleichbar mit mehreren geöffneten Fenstern in einer graphischen Umgebung.

Die Tastenkombinationen `Alt` + `F1` bis `Alt` + `F8` sind in FreeBSD zum Umschalten zwischen virtuellen Konsolen reserviert. Verwenden Sie `Alt` + `F1` um auf die Systemkonsole (ttyv0) zu wechseln, `Alt` + `F2` für die erste virtuelle Konsole (ttyv1, `Alt` + `F3` für die zweite virtuelle Konsole (ttyv2, und so weiter. Wenn Sie Xorg als graphische Oberfläche benutzen, können Sie mit `Strg` `Alt` `F1` zur virtuellen Konsole zurückkehren.

Beim Wechsel von einer Konsole zur nächsten wird die Bildschirmausgabe von FreeBSD verwaltet. Dies erzeugt die Illusion mehrerer Bildschirme und Tastaturen, an denen Kommandos abgesetzt werden können. Die Programme, die in einer virtuellen Konsole gestartet werden, laufen auch dann weiter, wenn der Benutzer auf eine andere virtuelle Konsole wechselt.

Lesen Sie kbdcontrol(1), vidcontrol(1), atkbd:(4), syscons(4) sowie vt(4) für eine recht technische Beschreibung der FreeBSD-Konsole und der Tastatur-Treiber.

In FreeBSD wird die Anzahl der verfügbaren virtuellen Konsolen in diesem Abschnitt von /etc/ttys konfiguriert:

```
# name    getty                           type  status comments
#
ttyv0   "/usr/libexec/getty Pc"         xterm   on  secure
# Virtual terminals
ttyv1   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv2   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv3   "/usr/libexec/getty Pc"         xterm   on  secure
```

```
ttyv4   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv5   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv6   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv7   "/usr/libexec/getty Pc"         xterm   on  secure
ttyv8   "/usr/X11R6/bin/xdm -nodaemon"  xterm   off secure
```

Um eine virtuelle Konsole zu deaktivieren, setzen Sie ein Kommentarzeichen ( an den Anfang der Zeile für die entsprechende Konsole. Um bspw. die Anzahl der verfügbaren virtuellen Konsolen von acht auf vier zu reduzieren, setzen Sie ein an den Anfang der letzten vier Zeilen, den virtuellen Konsolen ttyv5 bis ttyv8. Kommentieren Sie nicht die Zeile für die Systemkonsole ttyv0 aus! Beachten Sie, dass die letzte virtuelle Konsole (ttyv8) zum Wechsel auf die graphische Oberfläche gedacht ist, wenn Xorg wie im Das X-Window-System installiert und konfiguriert ist.

ttys(5) enthält eine ausführliche Beschreibung der Spalten dieser Datei und der verfügbaren Optionen für virtuelle Konsolen.

### 5.2.2. Single-User-Modus

Das FreeBSD Boot-Menü verfügt über eine Option "Boot Single User". Wird diese Option gewählt, bootet das System in einen speziellen Modus, der als "Single-User-Modus" bekannt ist. Dieser Modus wird normalerweise zur Reparatur des Systems verwendet, bspw. wenn das System nicht mehr startet, oder das root-Passwort zurückgesetzt werden muss. Im Single-User-Modus haben Sie keinen Zugriff auf das Netzwerk und es stehen Ihnen keine weiteren virtuellen Konsolen zur Verfügung. Allerdings haben Sie vollen Zugriff auf das System und in der Voreinstellung wird das root -Passwort nicht benötigt. Aus diesem Grund wird ein physischer Zugriff auf die Tastatur benötigt, um in diesem Modus zu booten. Zur Absicherung eines FreeBSD-Systems sollte ermittelt werden, welche Personen physischen Zugriff auf die Tastatur bekommen sollen.

Die Einstellungen für den Single-User-Modus befinden sich diesem Abschnitt von /etc/ttys:

```
# name  getty                           type  status  comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown  off  secure
```

In der Voreinstellung ist der Status auf secure eingestellt. Das setzt voraus, dass der physische Zugriff auf die Tastatur entweder unwichtig ist, oder über eine Sicherheitsrichtlinie geregelt wird. Wenn der Status auf insecure eingestellt wird, wird davon ausgegangen, dass die Umgebung selbst unsicher ist, da jeder Zugriff auf die Tastatur hat. FreeBSD wird dann nach dem root-Passwort fragen, wenn ein Benutzer versucht in den Single-User-Modus zu booten.



*Setzen Sie* insecure *nicht leichtfertig ein*! Wenn das root-Passwort vergessen wird, wird es schwierig in den Single-User-Modus zu gelangen, wenn man den Bootprozess von FreeBSD nicht genau versteht.

### 5.2.3. Den Videomodus der Konsole anpassen

Der Standard-Videomodus der FreeBSD-Konsole kann auf jeden Modus eingestellt werden, der von der Grafikkarte und dem Monitor unterstützt wird (beispielsweise 1024x768 oder 1280x1024). Um eine andere Einstellung zu verwenden, muss das VESA-Modul geladen werden:

```
# kldload vesa
```

Um festzustellen, welche Video-Modi von der Hardware unterstützt werden, nutzen Sie vidcontrol(1). Um eine Liste aller unterstützten Modi zu sehen, verwenden Sie diesen Befehl:

```
# vidcontrol -i mode
```

Die Ausgabe dieses Befehls listet alle Videomodi, die von der Hardware unterstützt werden. Um einen neuen Video-Modi zu wählen, wird der entsprechende Modus als root-Benutzer an vidcontrol(1) übergeben:

```
# vidcontrol MODE_279
```

Um diese Einstellung dauerhaft zu speichern, muss folgende Zeile in /etc/rc.conf hinzugefügt werden:

```
allscreens_flags="MODE_279"
```

## 5.3. Benutzer und grundlegende Account-Verwaltung

FreeBSD ermöglicht es mehreren Benutzern, den Computer zur selben Zeit zu benutzen. Es kann immer nur ein Benutzer vor der Konsole sitzen, aber es können sich beliebig viele Benutzer über das Netzwerk am System anmelden. Jeder Benutzer muss einen Account haben, um das System benutzen zu können.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- die verschiedenen Account-Typen von FreeBSD kennen,
- wissen, wie Sie Accounts angelegen, verändern oder löschen,
- wissen, wie Sie Limits für einen Benutzer oder eine Gruppe setzen, um beispielsweise Ressourcen, wie Speicher oder CPU-Zeit einzuschränken,
- wissen, wie Sie Gruppen erstellen und Benutzer zu diesen Gruppen hinzufügen.

### 5.3.1. Account-Typen

Jeder Zugriff auf das FreeBSD-System geschieht über Accounts und alle Prozesse werden von Benutzern gestartet, also sind Benutzer- und Account-Verwaltung von wesentlicher Bedeutung.

Es gibt drei Haupttypen von Accounts: Systembenutzer, Benutzer-Accounts und der Superuser-Account.

### Systembenutzer

Systembenutzer starten Dienste wie DNS, Mail-Server und Web-Server. Der Grund dafür ist die Sicherheit; wenn die Programme von dem Superuser gestartet werden, können Sie ohne Einschränkungen handeln.

Beispiele von Systembenutzern sind `daemon`, `operator`, `bind`, `news` und `www`.



Bei der Verwendung der Gruppe `operator` ist Vorsicht geboten, da dem Benutzer unbeabsichtigt Privilegien gewährt werden könnten, beispielsweise zum Herunterfahren oder Neustarten des Systems, oder der Zugriff auf alle Geräte in /dev.

`nobody` ist der generische unprivilegierte Systembenutzer. Bedenken Sie aber, dass je mehr Dienste `nobody` benutzen, desto mehr Dateien und Prozesse diesem Benutzer gehören und dieser Benutzer damit umso privilegierter wird.

### Benutzer-Accounts

Benutzer-Accounts sind realen Personen zugeordnet und sind das primäre Mittel des Zugriffs das System. Jede Person, die Zugriff auf das System bekommt, sollte einen eindeutigen Benutzer-Account besitzen. Dies erlaubt es dem Administrator herauszufinden, wer was macht. Gleichzeitig werden die Benutzer daran gehindert, die Einstellungen anderer Benutzer zu zerstören.

Jeder Benutzer kann die eigene Umgebung anpassen, bspw. seine voreingestellte Shell, Editor, Tastenbelegungen und Spracheinstellungen.

Mit jedem Account eines FreeBSD-Systems sind bestimmte Informationen verknüpft:

**Loginnamen**
: Der Loginname wird am `login:` Prompt eingegeben. Jeder Benutzer muss einen eindeutigen Benutzernamen haben. Es gibt eine Reihe von Regeln für die Erstellung von gültigen Loginnamen, die in passwd(5) dokumentiert sind. Es wird aus Kompatibilitätsgründen empfohlen, Benutzernamen zu verwenden, die aus Kleinbuchstaben bestehen und bis zu acht Zeichen lang sind.

**Passwort**
: Jeder Account ist mit einem Passwort verknüpft.

**User ID (UID)**
: Die User ID (UID) ist eine Zahl, die verwendet wird, um die Benutzer auf dem FreeBSD-System eindeutig zu identifizieren. Programme, die einen Loginnamen akzeptieren, wandeln diesen zuerst in eine UID um. Es wird empfohlen, nur UIDs kleiner 65535 zu verwenden, da höhere Werte Kompatibilitätsprobleme mit einigen Anwendungen verursachen können.

**Group ID (GID)**

Die Group ID (GID) ist eine Zahl, die verwendet wird, um die primäre Gruppe eines Benutzers eindeutig zu identifizieren. Gruppen sind ein Mechanismus zur Steuerung des Zugriffs auf Ressourcen über die GID eines Benutzers anstelle der UID. Dies kann die Größe einiger Konfigurationsdateien signifikant reduzieren und ermöglicht es Benutzern, Mitglied mehreren Gruppen zu sein. Es wird empfohlen, GIDs kleiner 65535 zu verwenden, da höhere Werte bei einigen Anwendungen große Probleme verursachen können.

**Login-Klasse**

Login-Klassen erweitern das Gruppenkonzept. Sie erhöhen die Flexibilität des Systems in der Handhabung der verschiedenen Accounts. Login-Klassen werden auch im Login-Klassen konfigurieren diskutiert.

**Gültigkeit von Passwörtern**

In der Voreinstellung verfallen Passwörter nicht. Allerdings können Passwortwechsel nach einer gewissen Zeit auf Basis einzelner Accounts erzwungen werden.

**Verfallszeit eines Accounts**

In der Voreinstellung verfallen unter FreeBSD keine Accounts. Wenn Sie Accounts einrichten, die nur für eine bestimmte Zeit gültig sein sollen, beispielsweise Accounts für Teilnehmer eines Praktikums, können Sie mit pw(8) die Gültigkeitsdauer des Accounts angeben. Nachdem die angegebene Zeitspanne verstrichen ist, kann dieser Account nicht mehr zum Anmelden verwendet werden, obwohl alle Verzeichnisse und Dateien, die diesem Account gehören, noch vorhanden sind.

**vollständiger Benutzername**

FreeBSD identifiziert einen Account eindeutig über den Loginnamen, der aber keine Ähnlichkeit mit dem richtigen Namen des Benutzers haben muss. Ähnlich wie bei einem Kommentar, kann diese Information Leerzeichen, Großbuchstaben und mehr als 8 Zeichen enthalten.

**Heimatverzeichnis**

Das Heimatverzeichnis gibt den vollständigen Pfad zu dem Verzeichnis an, in dem sich der Benutzer nach erfolgreicher Anmeldung befindet. Es ist üblich, alle Heimatverzeichnisse unter /home/Loginname oder /usr/home/Loginname anzulegen. Im Heimatverzeichnis oder in dort angelegten Verzeichnissen werden die Dateien eines Benutzers gespeichert.

**Login-Shell**

Grundsätzlich ist die Shell, von denen es viele unterschiedliche gibt, eine Schnittstelle zum System. Die bevorzugte Shell eines Benutzers kann seinem Account zugeordnet werden.

**Der Superuser-Account**

Der Superuser-Account, normalerweise `root` genannt, ist vorkonfiguriert und erleichtert die Systemverwaltung, sollte aber nicht für alltägliche Aufgaben wie das Verschicken und Empfangen von Mails, Erforschen des Systems oder Programmierung benutzt werden.

Der Superuser kann, im Gegensatz zu normalen Benutzer-Accounts, ohne Beschränkungen operieren und die falsche Anwendung des Superuser-Accounts kann in spektakulären

Katastrophen resultieren. Benutzer-Accounts sind nicht in der Lage, das System versehentlich zu zerstören, deswegen wird empfohlen, normale Benutzer-Accounts zu verwenden, solange nicht zusätzliche Privilegien benötigt werden.

Kommandos, die Sie als Superuser eingeben, sollten Sie immer doppelt und dreifach überprüfen, da ein zusätzliches Leerzeichen oder ein fehlender Buchstabe irreparablen Datenverlust bedeuten kann.

Es gibt mehrere Möglichkeiten Superuser-Rechte zu bekommen. Obwohl man sich direkt als `root` anmelden kann, wird von dieser Methode dringend abgeraten.

Verwenden Sie stattdessen su(1) um zum Superuser zu werden. Wenn Sie noch ein `-` eingeben, wird der Benutzer auch die Umgebung des Root-Benutzers erben. Der Benutzer, der diesen Befehl ausführt muss Mitglied der Gruppe `wheel` sein, oder der Befehl schlägt fehl. Zudem muss der Benutzer das Kennwort für den Benutzer-Account `root` kennen.

In diesem Beispiel wird der Benutzer nur zum Superuser, um `make install` auszuführen, da dieser Befehl Superuser-Rechte erfordert. Nachdem der Befehl ausgeführt wurde, kann der Benutzer `exit` eingeben, um den Superuser-Account zu verlassen und zu den Privilegien des Benutzer-Accounts zurückkehren.

*Beispiel 2. Ein Programm als Superuser installieren*

```
% configure
% make
% su -
Password:
# make install
# exit
%
```

Das in FreeBSD enthaltene su(1) funktioniert gut für einzelne Systeme oder in kleineren Netzwerken, mit nur einem Administrator. Eine Alternative ist es, das Paket oder den Port security/sudo zu installieren. Diese Software bietet eine Protokollierung von Aktivitäten und ermöglicht es dem Administrator zu bestimmen, welche Benutzer welche Befehle als Superuser ausführen dürfen.

### 5.3.2. Accounts verändern

FreeBSD stellt eine Vielzahl an Programmen bereit, um Accounts zu verändern. Die gebräuchlichsten Kommandos sind in Programme zur Verwaltung von Benutzer-Accounts gefolgt von einer detaillierten Beschreibung, zusammengefasst. Weitere Informationen und Anwendungsbeispiele finden Sie in der Manualpage des jeweiligen Programms.

*Tabelle 2. Programme zur Verwaltung von Benutzer-Accounts*

| Programm | Zusammenfassung |
|---|---|
| adduser(8) | Das empfohlene Werkzeug, um neue Accounts zu erstellen. |
| rmuser(8) | Das empfohlene Werkzeug, um Accounts zu löschen. |
| chpass(1) | Ein flexibles Werkzeug, um Informationen in der Account-Datenbank zu verändern. |
| passwd(1) | Ein Werkzeug, um Passwörter von Accounts zu ändern. |
| pw(8) | Ein mächtiges und flexibles Werkzeug um alle Informationen über Accounts zu ändern. |

#### adduser

Das empfohlene Programm zum Hinzufügen neuer Benutzer ist adduser(8). Wenn ein neuer Benutzer hinzugefügt wird, aktualisiert das Programm automatisch /etc/passwd und /etc/group. Es erstellt auch das Heimatverzeichnis für den Benutzer, kopiert die Standardkonfigurationsdateien aus /usr/shared/skel und kann optional eine „Willkommen``-Nachricht an den neuen Benutzer versenden. Das Programm muss als Superuser ausgeführt werden.

Das Werkzeug adduser(8) arbeitet interaktiv und führt durch die einzelnen Schritte, wenn ein neues Benutzerkonto erstellt wird. Wie in Einen Benutzer unter FreeBSD anlegen zu sehen ist, müssen Sie entweder die benötigte Information eingeben oder `Return` drücken, um den Vorgabewert in eckigen Klammern zu akzeptieren. In diesem Beispiel wird der Benutzer in die Gruppe `wheel` aufgenommen, was es ihm erlaubt mit su(1) zum Superuser zu werden. Wenn Sie fertig sind, können Sie entweder einen weiteren Benutzer erstellen oder das Programm beenden.

*Beispiel 3. Einen Benutzer unter FreeBSD anlegen*

```
# adduser
Username: jru
Full name: J. Random User
Uid (Leave empty for default):
Login group [jru]:
Login group is jru. Invite jru into other groups? []: wheel
Login class [default]:
Shell (sh csh tcsh zsh nologin) [sh]: zsh
Home directory [/home/jru]:
Home directory permissions (Leave empty for default):
Use password-based authentication? [yes]:
Use an empty password? (yes/no) [no]:
Use a random password? (yes/no) [no]:
Enter password:
Enter password again:
Lock out the account after creation? [no]:
Username   : jru
Password   : ****
```

```
Full Name  : J. Random User
Uid        : 1001
Class      :
Groups     : jru wheel
Home       : /home/jru
Shell      : /usr/local/bin/zsh
Locked     : no
OK? (yes/no): yes
adduser: INFO: Successfully added (jru) to the user database.
Add another user? (yes/no): no
Goodbye!
#
```



Wenn Sie das Passwort eingeben, werden weder Passwort noch Sternchen angezeigt. Passen Sie auf, dass Sie das Passwort korrekt eingeben.

#### `rmuser`

Benutzen Sie rmuser(8) als Superuser, um einen Account vollständig aus dem System zu entfernen. Dieses Programm führt die folgenden Schritte durch:

1. Entfernt den crontab(1) Eintrag des Benutzers, wenn dieser existiert.
2. Entfernt alle at(1) jobs, die dem Benutzer gehören.
3. Schließt alle Prozesse des Benutzers.
4. Entfernt den Benutzer aus der lokalen Passwort-Datei des Systems.
5. Entfernt optional das Heimatverzeichnis des Benutzers, falls es dem Benutzer gehört.
6. Entfernt eingegangene E-Mails des Benutzers aus /var/mail.
7. Entfernt alle Dateien des Benutzers aus temporären Dateispeicherbereichen wie /tmp.
8. Entfernt den Loginnamen von allen Gruppen, zu denen er gehört, aus /etc/group. Wenn eine Gruppe leer wird und der Gruppenname mit dem Loginnamen identisch ist, wird die Gruppe entfernt. Das ergänzt sich mit den einzelnen Benutzer-Gruppen, die von adduser(8) für jeden neuen Benutzer erstellt werden.

Der Superuser-Account kann nicht mit rmuser(8) entfernt werden, da dies in den meisten Fällen das System unbrauchbar macht.

Als Vorgabe wird ein interaktiver Modus benutzt.

*Beispiel 4. Interaktives Löschen von Accounts mit `rmuser`*

```
# rmuser jru
Matching password entry:
jru:*:1001:1001::0:0:J. Random User:/home/jru:/usr/local/bin/zsh
Is this the entry you wish to remove? y
Remove user's home directory (/home/jru)? y
```

```
Removing user (jru): mailspool home passwd.
#
```

### chpass

Jeder Benutzer kann chpass(1) verwenden, um die Shell und persönliche Informationen des Benutzerkontos zu verändern. Der Superuser kann dieses Werkzeug benutzen, um zusätzliche Kontoinformationen für alle Benutzer zu ändern.

Werden neben dem optionalen Loginnamen keine weiteren Optionen angegeben, zeigt chpass(1) einen Editor mit Account-Informationen an. Wenn der Benutzer den Editor verlässt, wird die Account-Datenbank mit den neuen Informationen aktualisiert.



Dieses Programm fragt nach dem Verlassen des Editors nach dem Passwort, es sei denn, man ist als Superuser angemeldet.

In `chpass` als Superuser verwenden hat der Superuser `chpass jru` eingegeben. Es werden die Felder ausgegeben, die für diesen Benutzer geändert werden können. Wenn stattdessen `jru` diesen Befehl aufruft, werden nur die letzten sechs Felder ausgegeben. Dies ist in `chpass` als normaler Benutzer verwenden zu sehen.

*Beispiel 5. `chpass` als Superuser verwenden*

```
#Changing user database information for jru.
Login: jru
Password: *
Uid [#]: 1001
Gid [# or name]: 1001
Change [month day year]:
Expire [month day year]:
Class:
Home directory: /home/jru
Shell: /usr/local/bin/zsh
Full Name: J. Random User
Office Location:
Office Phone:
Home Phone:
Other information:
```

*Beispiel 6. `chpass` als normaler Benutzer verwenden*

```
#Changing user database information for jru.
Shell: /usr/local/bin/tcsh
Full Name: J. Random User
Office Location:
Office Phone:
```

```
Home Phone:
Other information:
```



Die Kommandos chfn(1) und chsh(1) sind nur Verweise auf chpass(1), genauso wie ypchpass(1), ypchfn(1) und ypchsh(1). Da NIS automatisch unterstützt wird, ist es nicht notwendig das `yp` vor dem Kommando einzugeben. NIS wird später im Netzwerkserver besprochen.

**passwd**

Jeder Benutzer kann mit passwd(1) einfach sein Passwort ändern. Um eine versehentliche oder unbefugte Änderung zu verhindern, muss bei einem Passwortwechsel zunächst das ursprüngliche Passwort eingegeben werden, bevor das neue Passwort festgelegt werden kann.

*Beispiel 7. Das eigene Passwort wechseln*

```
% passwd
Changing local password for jru.
Old password:
New password:
Retype new password:
passwd: updating the database...
passwd: done
```

Der Superuser kann jedes beliebige Passwort ändern, indem er den Benutzernamen an passwd(1) übergibt. Das Programm fordert den Superuser nicht dazu auf, das aktuelle Passwort des Benutzers einzugeben. Dadurch kann das Passwort geändert werden, falls der Benutzer sein ursprüngliches Passwort vergessen hat.

*Beispiel 8. Als Superuser das Passwort eines anderen Accounts verändern*

```
# passwd jru
Changing local password for jru.
New password:
Retype new password:
passwd: updating the database...
passwd: done
```



Wie bei chpass(1) ist yppasswd(1) nur ein Verweis auf passwd(1). NIS wird von jedem dieser Kommandos unterstützt.

`pw`

Mit dem Werkzeug pw(8) können Accounts und Gruppen erstellt, entfernt, verändert und angezeigt

werden. Dieses Kommando dient als Schnittstelle zu den Benutzer- und Gruppendateien des Systems. pw(8) besitzt eine Reihe mächtiger Kommandozeilenschalter, die es für die Benutzung in Shell-Skripten geeignet machen, doch finden neue Benutzer die Bedienung des Kommandos komplizierter, als die der anderen hier vorgestellten Kommandos.

### 5.3.3. Gruppen

Eine Gruppe ist einfach eine Zusammenfassung von Accounts. Gruppen werden durch den Gruppennamen und die GID identifiziert. Der Kernel von FreeBSD entscheidet anhand der UID und der Gruppenmitgliedschaft eines Prozesses, ob er dem Prozess etwas erlaubt oder nicht. Wenn jemand von der GID eines Benutzers oder Prozesses spricht, meint er damit meistens die erste Gruppe der Gruppenliste.

Die Zuordnung von Gruppennamen zur GID steht in /etc/group, einer Textdatei mit vier durch Doppelpunkte getrennten Feldern. Im ersten Feld steht der Gruppenname, das zweite enthält ein verschlüsseltes Passwort, das dritte gibt die GID an und das vierte besteht aus einer Komma separierten Liste der Mitglieder der Gruppe. Eine ausführliche Beschreibung der Syntax dieser Datei finden Sie in group(5).

Wenn Sie /etc/group nicht von Hand editieren möchten, können Sie pw(8) zum Editieren benutzen. Das folgende Beispiel zeigt das Hinzufügen einer Gruppe mit dem Namen `teamtwo`:

*Beispiel 9. Setzen der Mitgliederliste einer Gruppe mit pw(8)*

```
# pw groupadd teamtwo
# pw groupshow teamtwo
teamtwo:*:1100:
```

`1100` ist die GID der Gruppe `teamtwo`. Momentan hat `teamtwo` noch keine Mitglieder. Mit dem folgenden Kommando wird der Benutzer `jru` in die Gruppe `teamtwo` aufgenommen.

*Beispiel 10. Ein Gruppenmitglied mit pw(8) hinzufügen*

```
# pw groupmod teamtwo -M jru
# pw groupshow teamtwo
teamtwo:*:1100:jru
```

Als Argument von `-M` geben Sie eine Komma separierte Liste von Mitgliedern an, die in die Gruppe aufgenommen werden sollen. Aus den vorherigen Abschnitten ist bekannt, dass die Passwort-Datei ebenfalls eine Gruppe für jeden Benutzer enthält. Das System teilt dem Benutzer automatisch eine Gruppe zu, die aber vom `groupshow` Kommando von pw(8) nicht angezeigt wird. Diese Information wird allerdings von id(1) und ähnlichen Werkzeugen angezeigt. Das heißt, dass pw(8) nur /etc/group manipuliert, es wird nicht versuchen, zusätzliche Informationen aus /etc/passwd zu lesen.

*Beispiel 11. Hinzufügen eines neuen Gruppenmitglieds mittels pw(8)*

```
# pw groupmod teamtwo -m db
# pw groupshow teamtwo
teamtwo:*:1100:jru,db
```

Die Argumente zur Option `-m` ist eine durch Komma getrennte Liste von Benutzern, die der Gruppe hinzugefügt werden sollen. Anders als im vorherigen Beispiel werden diese Benutzer in die Gruppe aufgenommen und ersetzen nicht die bestehenden Benutzer in der Gruppe.

*Beispiel 12. Mit `id` die Gruppenzugehörigkeit bestimmen*

```
% id jru
uid=1001(jru) gid=1001(jru) groups=1001(jru), 1100(teamtwo)
```

In diesem Beispiel ist `jru` Mitglied von `jru` und `teamtwo`.

Weitere Informationen zu diesem Befehl und dem Format von /etc/group finden Sie in pw(8) und group(5).

## 5.4. Zugriffsrechte

In FreeBSD besitzt jede Datei und jedes Verzeichnis einen Satz von Zugriffsrechten. Es stehen mehrere Programme zum Anzeigen und Bearbeiten dieser Rechte zur Verfügung. Ein Verständnis für die Funktionsweise von Zugriffsrechten ist notwendig, um sicherzustellen, dass Benutzer nur auf die von ihnen benötigten Dateien zugreifen können und nicht auf die Dateien des Betriebssystems oder von anderen Benutzern.

In diesem Abschnitt werden die traditionellen Zugriffsrechte von UNIX® beschrieben. Informationen zu feingranularen Zugriffsrechten für Dateisysteme finden Sie im Zugriffskontrolllisten für Dateisysteme (ACL).

In UNIX® werden die grundlegenden Zugriffsrechte in drei Typen unterteilt: Lesen, Schreiben und Ausführen. Diese Zugriffstypen werden verwendet, um den Dateizugriff für den Besitzer der Datei, die Gruppe und alle anderen zu bestimmen. Die Lese-, Schreib- und Ausführungsberechtigungen werden mit den Buchstaben `r`, `w` und `x` dargestellt. Alternativ können die Berechtigungen als binäre Zahlen dargestellt werden, da jede Berechtigung entweder aktiviert oder deaktiviert (`0`) ist. Wenn die Berechtigung als Zahl dargestellt wird, ist die Reihenfolge immer als `rwx` zu lesen, wobei `r` den Wert `4` hat, `w` den Wert `2` und `x` den Wert `1`.

In Tabelle 4.1 sind die einzelnen nummerischen und alphabetischen Möglichkeiten zusammengefasst. Das Zeichen `-` in der Spalte "Auflistung im Verzeichnis" besagt, dass eine Berechtigung deaktiviert ist.

*Tabelle 3. UNIX® Zugriffsrechte*

| Wert | Zugriffsrechte | Auflistung im Verzeichnis |
|---|---|---|
| 0 | Kein Lesen, Kein Schreiben, Kein Ausführen | `---` |
| 1 | Kein Lesen, Kein Schreiben, Ausführen | `--x` |
| 2 | Kein Lesen, Schreiben, Kein Ausführen | `-w-` |
| 3 | Kein Lesen, Schreiben, Ausführen | `-wx` |
| 4 | Lesen, Kein Schreiben, Kein Ausführen | `r--` |
| 5 | Lesen, Kein Schreiben, Ausführen | `r-x` |
| 6 | Lesen, Schreiben, Kein Ausführen | `rw-` |
| 7 | Lesen, Schreiben, Ausführen | `rwx` |

Benutzen Sie das Argument `-l` mit ls(1), um eine ausführliche Verzeichnisauflistung zu sehen, die in einer Spalte die Zugriffsrechte für den Besitzer, die Gruppe und alle anderen enthält. Die Ausgabe von `ls -l` könnte wie folgt aussehen:

```
% ls -l
total 530
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 myfile
-rw-r--r--  1 root  wheel     512 Sep  5 12:31 otherfile
-rw-r--r--  1 root  wheel    7680 Sep  5 12:31 email.txt
```

Das erste Zeichen (ganz links) der ersten Spalte zeigt an, ob es sich um eine normale Datei, ein Verzeichnis, ein zeichenorientiertes Gerät, ein Socket oder irgendeine andere Pseudo-Datei handelt. In diesem Beispiel zeigt `-` eine normale Datei an. Die nächsten drei Zeichen, dargestellt als `rw-`, ergeben die Rechte für den Datei-Besitzer. Die drei Zeichen danach `r--` die Rechte der Gruppe, zu der die Datei gehört. Die letzten drei Zeichen, `r--`, geben die Rechte für den Rest der Welt an. Ein Minus bedeutet, dass das Recht nicht gegeben ist. In diesem Beispiel sind die Zugriffsrechte also: der Eigentümer kann die Datei lesen und schreiben, die Gruppe kann lesen und alle anderen können auch nur lesen. Entsprechend obiger Tabelle wären die Zugriffsrechte für diese Datei `644`, worin jede Ziffer die drei Teile der Zugriffsrechte dieser Datei verkörpert.

Wie kontrolliert das System die Rechte von Hardware-Geräten? FreeBSD behandelt die meisten Hardware-Geräte als Dateien, welche Programme öffnen, lesen und mit Daten beschreiben können. Diese speziellen Gerätedateien sind in /dev gespeichert.

Verzeichnisse werden ebenfalls wie Dateien behandelt. Sie haben Lese-, Schreib- und Ausführ-Rechte. Das Ausführungs-Bit hat eine etwas andere Bedeutung für ein Verzeichnis als für eine Datei. Die Ausführbarkeit eines Verzeichnisses bedeutet, dass in das Verzeichnis, zum Beispiel mit cd(1), gewechselt werden kann. Das bedeutet auch, dass in dem Verzeichnis auf Dateien, deren Namen bekannt sind, zugegriffen werden kann, vorausgesetzt die Zugriffsrechte der Dateien lassen dies zu.

Das Leserecht auf einem Verzeichnis erlaubt es, sich den Inhalt des Verzeichnisses anzeigen zu lassen. Um eine Datei mit bekanntem Namen in einem Verzeichnis zu löschen, müssen auf dem Verzeichnis Schreib- *und* Ausführ-Rechte gesetzt sein.

Es gibt noch mehr Rechte, aber die werden vor allem in speziellen Umständen benutzt, wie zum Beispiel bei SetUID-Binaries und Verzeichnissen mit gesetztem Sticky-Bit. Mehr über Zugriffsrechte von Dateien und wie sie gesetzt werden, finden Sie in chmod(1).

### 5.4.1. Symbolische Zugriffsrechte

Symbolische Zugriffsrechte verwenden Zeichen anstelle von oktalen Werten, um die Berechtigungen für Dateien oder Verzeichnisse festzulegen. Zugriffsrechte verwenden die Syntax *Wer*, *Aktion* und *Berechtigung*. Die folgenden Werte stehen zur Auswahl:

| Option | Symbol | Bedeutung |
|---|---|---|
| *Wer* | u | Benutzer (user) |
| *Wer* | g | Gruppe (group) |
| *Wer* | o | Andere (other) |
| *Wer* | a | Alle |
| *Aktion* | + | Berechtigungen hinzufügen |
| *Aktion* | - | Berechtigungen entziehen |
| *Aktion* | = | Berechtigungen explizit setzen |
| *Berechtigung* | r | lesen (read) |
| *Berechtigung* | w | schreiben (write) |
| *Berechtigung* | x | ausführen (execute) |
| *Berechtigung* | t | Sticky-Bit |
| *Berechtigung* | s | Set-UID oder Set-GID |

Diese symbolischen Werte werden zusammen mit chmod(1) verwendet. Beispielsweise würde der folgende Befehl den Zugriff auf *FILE* für alle anderen Benutzer verbieten:

```
% chmod go= FILE
```

Wenn Sie mehr als eine Änderung der Rechte einer Datei vornehmen wollen, können Sie eine durch Kommata getrennte Liste der Rechte angeben. Das folgende Beispiel entzieht der Gruppe und der Welt die Schreibberechtigung auf *FILE* und fügt für jeden Ausführungsrechte hinzu:

```
% chmod go-w,a+x FILE
```

### 5.4.2. FreeBSD Datei-Flags

Zusätzlich zu den Zugriffsrechten unterstützt FreeBSD auch die Nutzung von "Datei-Flags". Diese erhöhen die Sicherheit des Systems, indem sie eine verbesserte Kontrolle von Dateien erlauben. Verzeichnisse werden allerdings nicht unterstützt. Mit dem Einsatz von Datei-Flags kann sogar `root` daran gehindert werden, Dateien zu löschen oder zu verändern.

Datei-Flags werden mit chflags(1) verändert. Um beispielsweise auf der Datei file1 das "unlöschbar"-Flag zu aktivieren, geben Sie folgenden Befehl ein:

```
# chflags sunlink file1
```

Um dieses Flag zu deaktivieren, setzen Sie ein "no" vor `sunlink`:

```
# chflags nosunlink file1
```

Um die Flags einer Datei anzuzeigen, verwenden Sie ls(1) zusammen mit `-lo`:

```
# ls -lo file1
```

```
-rw-r--r--  1 trhodes  trhodes  sunlnk 0 Mar  1 05:54 file1
```

Einige Datei-Flags können nur vom `root`-Benutzer gesetzt oder gelöscht werden. Andere wiederum können auch vom Eigentümer der Datei gesetzt werden. Weitere Informationen hierzu finden sich in chflags(1) und chflags(2).

### 5.4.3. Die Berechtigungen `setuid`, `setgid`, und `sticky`

Anders als die Berechtigungen, die bereits angesprochen wurden, existieren drei weitere Einstellungen, über die alle Administratoren Bescheid wissen sollten. Dies sind die Berechtigungen `setuid`, `setgid` und `sticky`.

Diese Einstellungen sind wichtig für manche UNIX®-Operationen, da sie Funktionalitäten zur Verfügung stellen, die normalerweise nicht an gewöhnliche Anwender vergeben wird. Um diese zu verstehen, muss der Unterschied zwischen der realen und der effektiven Benutzer-ID erwähnt werden.

Die reale Benutzer-ID ist die UID, welche den Prozess besitzt oder gestartet hat. Die effektive UID ist diejenige, als die der Prozess läuft. Beispielsweise wird passwd(1) mit der realen ID des Benutzers ausgeführt, der sein Passwort ändert. Um jedoch die Passwortdatenbank zu bearbeiten, wird es effektiv als `root`-Benutzer ausgeführt. Das ermöglicht es normalen Benutzern, ihr Passwort zu ändern, ohne einen `Permission Denied`-Fehler angezeigt zu bekommen.

Die setuid-Berechtigung kann durch das Voranstellen bei einer Berechtigungsgruppe mit der Nummer Vier (4) gesetzt werden, wie im folgenden Beispiel gezeigt wird:

```
# chmod 4755 suidexample.sh
```

Die Berechtigungen auf suidexample.sh sehen jetzt wie folgt aus:

```
-rwsr-xr-x   1 trhodes  trhodes    63 Aug 29 06:36 suidexample.sh
```

Beachten Sie, dass ein `s` jetzt Teil der Berechtigungen des Dateibesitzers geworden ist, welches das Ausführen-Bit ersetzt. Dies ermöglicht es Werkzeugen mit erhöhten Berechtigungen zu laufen, wie beispielsweise `passwd`.



Die `nosuid` mount(8)-Option bewirkt, dass solche Anwendungen stillschweigend scheitern, ohne den Anwender darüber zu informieren. Diese Option ist nicht völlig zuverlässig, da ein `nosuid`-Wrapper in der Lage wäre, dies zu umgehen.

Um dies in Echtzeit zu beobachten, öffnen Sie zwei Terminals. Starten Sie auf einem `passwd` als normaler Benutzer. Während es auf die Passworteingabe wartet, überprüfen Sie die Prozesstabelle und sehen Sie sich die Informationen für passwd(1) an:

Im Terminal A:

```
Changing local password for trhodes
Old Password:
```

Im Terminal B:

```
# ps aux | grep passwd
```

```
trhodes  5232  0.0  0.2  3420  1608   0  R+    2:10AM   0:00.00 grep passwd
root     5211  0.0  0.2  3620  1724   2  I+    2:09AM   0:00.01 passwd
```

Obwohl passwd(1) als normaler Benutzer ausgeführt wird, benutzt es die effektive UID von `root`.

Die `setgid`-Berechtigung führt die gleiche Aktion wie die `setuid`-Berechtigung durch, allerdings verändert sie die Gruppenberechtigungen. Wenn eine Anwendung oder ein Werkzeug mit dieser Berechtigung ausgeführt wird, erhält es die Berechtigungen basierend auf der Gruppe, welche die Datei besitzt und nicht die des Benutzers, der den Prozess gestartet hat.

Um die `setgid`-Berechtigung auf einer Datei zu setzen, geben Sie chmod(1) eine führende Zwei (2) mit:

```
# chmod 2755 sgidexample.sh
```

Beachten Sie in der folgenden Auflistung, dass das `s` sich jetzt in dem Feld befindet, das für die Berechtigungen der Gruppe bestimmt ist:

```
-rwxr-sr-x   1 trhodes  trhodes    44 Aug 31 01:49 sgidexample.sh
```



Obwohl es sich bei dem in diesen Beispielen gezeigten Shellskript um eine ausführbare Datei handelt, wird es nicht mit einer anderen EUID oder effektiven Benutzer-ID ausgeführt. Das ist so, weil Shellskripte keinen Zugriff auf setuid(2) -Systemaufrufe erhalten.

Die `setuid` und `setgid` Berechtigungs-Bits können die Systemsicherheit verringern, da sie erhöhte Rechte ermöglichen. Das dritte Berechtigungs-Bit, das `sticky bit` kann die Sicherheit eines Systems erhöhen.

Wenn das `sticky bit` auf einem Verzeichnis angewendet wird, erlaubt es das Löschen von Dateien nur durch den Besitzer der Datei. Dies ist nützlich, um die Löschung von Dateien in öffentlichen Verzeichnissen wie /tmp, durch Benutzer denen diese Dateien nicht gehören, zu verhindern. Um diese Berechtigung anzuwenden, stellen Sie der Berechtigung eine Eins (1) voran:

```
# chmod 1777 /tmp
```

Das `sticky bit` kann anhand des `t` ganz am Ende der Berechtigungen abgelesen werden.

```
# ls -al / | grep tmp
```

```
drwxrwxrwt  10 root  wheel         512 Aug 31 01:49 tmp
```

## 5.5. Verzeichnis-Strukturen

Die FreeBSD-Verzeichnishierarchie ist die Grundlage, um ein umfassendes Verständnis des Systems zu erlangen. Das wichtigste Verzeichnis ist das Root-Verzeichnis "/". Dieses Verzeichnis ist das erste, das während des Bootens eingehangen wird. Es enthält das notwendige Basissystem, um das Betriebssystem in den Mehrbenutzerbetrieb zu bringen. Das Root-Verzeichnis enthält auch die Mountpunkte für Dateisysteme, die beim Wechsel in den Multiuser-Modus eingehängt werden.

Ein Mountpunkt ist ein Verzeichnis, in das zusätzliche Dateisysteme (in der Regel unterhalb des Wurzelverzeichnisses) eingehängt werden können. Dieser Vorgang wird in Festplatten, Slices und Partitionen ausführlich beschrieben. Standard-Mountpunkte sind /usr, /var, /tmp, /mnt sowie /cdrom. Auf diese Verzeichnisse verweisen üblicherweise Einträge in /etc/fstab. Diese Datei ist eine Tabelle mit verschiedenen Dateisystemen und Mountpunkten, vom System gelesen werden. Die meisten der Dateisysteme in /etc/fstab werden beim Booten automatisch durch das Skript rc(8) gemountet, wenn die zugehörigen Einträge nicht mit `noauto` versehen sind. Weitere Informationen zu diesem Thema finden Sie im Die fstab Datei.

Eine vollständige Beschreibung der Dateisystem-Hierarchie finden Sie in hier(7). Die folgende Aufstellung gibt einen kurzen Überblick über die am häufigsten verwendeten Verzeichnisse:

| Verzeichnis | Beschreibung |
|---|---|
| / | Wurzelverzeichnis des Dateisystems. |

| Verzeichnis | Beschreibung |
|---|---|
| /bin/ | Grundlegende Werkzeuge für den Single-User-Modus sowie den Mehrbenutzerbetrieb. |
| /boot/ | Programme und Konfigurationsdateien, die während des Bootens benutzt werden. |
| /boot/defaults/ | Vorgaben für die Boot-Konfiguration. Weitere Details finden Sie in loader.conf(5). |
| /dev/ | Gerätedateien. Weitere Details finden Sie in intro(4). |
| /etc/ | Konfigurationsdateien und Skripten des Systems. |
| /etc/defaults/ | Vorgaben für die System Konfigurationsdateien. Weitere Details finden Sie in rc(8). |
| /etc/mail/ | Konfigurationsdateien von MTAs wie sendmail(8). |
| /etc/periodic/ | Täglich, wöchentlich oder monatlich laufende Skripte, die von cron(8) gestartet werden. Weitere Details finden Sie in periodic(8). |
| /etc/ppp/ | Konfigurationsdateien von ppp(8). |
| /mnt/ | Ein leeres Verzeichnis, das von Systemadministratoren häufig als temporärer Mountpunkt genutzt wird. |
| /proc/ | Prozess Dateisystem. Weitere Details finden Sie in procfs(5) und mount_procfs(8). |
| /rescue/ | Statisch gelinkte Programme zur Wiederherstellung des Systems, wie in rescue(8) beschrieben. |
| /root/ | Home Verzeichnis von `root`. |
| /sbin/ | Systemprogramme und administrative Werkzeuge, die grundlegend für den Single-User-Modus und den Mehrbenutzerbetrieb sind. |
| /tmp/ | Temporäre Dateien, die für gewöhnlich bei einem Neustart des Systems verloren gehen. Häufig wird ein speicherbasiertes Dateisystem unter /tmp eingehängt. Dieser Vorgang kann automatisiert werden, wenn tmpmfs-bezogene Variablen von rc.conf(5) verwendet werden, oder ein entsprechender Eintrag in /etc/fstab existiert. Weitere Informationen finden Sie in mdmfs(8). |
| /usr/ | Der Großteil der Benutzerprogramme und Anwendungen. |
| /usr/bin/ | Gebräuchliche Werkzeuge, Programmierhilfen und Anwendungen. |
| /usr/include/ | Standard C include-Dateien. |
| /usr/lib/ | Bibliotheken. |
| /usr/libdata/ | Daten verschiedener Werkzeuge. |
| /usr/libexec/ | System-Dämonen und System-Werkzeuge, die von anderen Programmen ausgeführt werden. |

| Verzeichnis | Beschreibung |
|---|---|
| /usr/local/ | Lokale Programme und Bibliotheken. Die Ports-Sammlung von FreeBSD benutzt dieses Verzeichnis als Zielverzeichnis für Anwendungen. Innerhalb von /usr/local sollte das von hier(7) beschriebene Layout für /usr benutzt werden. Das man Verzeichnis wird direkt unter /usr/local anstelle unter /usr/local/share angelegt. Die Dokumentation der Ports findet sich in share/doc/port. |
| /usr/obj/ | Von der Architektur abhängiger Verzeichnisbaum, der durch das Bauen von /usr/src entsteht. |
| /usr/ports/ | Die FreeBSD-Ports-Sammlung (optional). |
| /usr/sbin/ | System-Dämonen und System-Werkzeuge, die von Benutzern ausgeführt werden. |
| /usr/shared/ | Von der Architektur unabhängige Dateien. |
| /usr/src/ | Quelldateien von BSD und/oder lokalen Ergänzungen. |
| /var/ | Wird für mehrere Zwecke genutzt und enthält Logdateien, temporäre Daten und Spooldateien. Manchmal wird ein speicherbasiertes Dateisystem unter /var eingehängt. Dieser Vorgang kann automatisiert werden, wenn die varmfs-bezogenen Variablen von rc.conf(5) verwendet werden, oder ein entsprechender Eintrag in /etc/fstab existiert. Weitere Informationen finden Sie in mdmfs(8). |
| /var/log/ | Verschiedene Logdateien des Systems. |
| /var/mail/ | Postfächer der Benutzer. |
| /var/spool/ | Verschiedene Spool-Verzeichnisse der Drucker- und Mailsysteme. |
| /var/tmp/ | Temporäre Dateien, die in der Regel auch bei einem Neustart des Systems erhalten bleiben, es sei denn, bei /var handelt es sich um ein speicherbasiertes Dateisystem. |
| /var/yp/ | NIS maps. |

## 5.6. Festplatten, Slices und Partitionen

FreeBSD identifiziert Dateien anhand eines Dateinamens. In Dateinamen wird zwischen Groß- und Kleinschreibung unterschieden: readme.txt und README.TXT bezeichnen daher zwei verschiedene Dateien. FreeBSD benutzt keine Dateiendungen, um den Typ der Datei zu bestimmen, egal ob es sich um ein Programm, ein Dokument oder um andere Daten handelt.

Dateien werden in Verzeichnissen gespeichert. In einem Verzeichnis können sich keine oder hunderte Dateien befinden. Ein Verzeichnis kann auch andere Verzeichnisse enthalten und so eine Hierarchie von Verzeichnissen aufbauen, die die Ablage von Daten erleichtert.

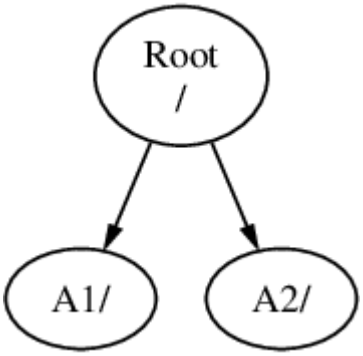
In Dateinamen werden Verzeichnisse durch einen Schrägstrich (/, Slash) getrennt. Wenn z.B. das Verzeichnis foo ein Verzeichnis bar enthält, in dem sich die Datei readme.txt befindet, lautet der vollständige Name der Datei (oder der *Pfad* zur Datei) foo/bar/readme.txt. Beachten Sie, dass sich dies von Windows® unterscheidet, wo der \ (Backslash für die Trennung von Datei- und

Verzeichnisnamen verwendet wird. FreeBSD benutzt keine Laufwerkbuchstaben oder Laufwerknamen im Pfad. Beispielsweise würde man unter FreeBSD nicht c:\foo\bar\readme.txt eingeben.

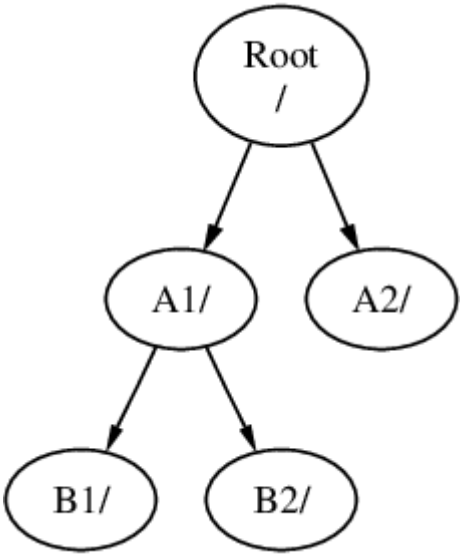
Verzeichnisse und Dateien werden in einem Dateisystem gespeichert. Jedes Dateisystem besitzt genau ein *Wurzelverzeichnis*, das so genannte Root-Directory. Dieses Wurzelverzeichnis kann weitere Verzeichnisse enthalten. Ein Dateisystem wird als Wurzeldateisystem festgelegt, und jedes weitere Dateisystem wird unter dem Wurzeldateisystem *eingehangen*. Daher scheint jedes Verzeichnis, unabhängig von der Anzahl der Platten, auf derselben Platte zu liegen.

Betrachten wir die drei Dateisysteme `A`, `B` und `C`. Jedes Dateisystem besitzt ein eigenes Wurzelverzeichnis, das zwei andere Verzeichnisse enthält: A1, A2, B1, B2, C1 und C2.

Das Wurzeldateisystem soll `A` sein. ls(1) zeigt darin die beiden Verzeichnisse A1 und A2 an. Der Verzeichnisbaum sieht wie folgt aus:

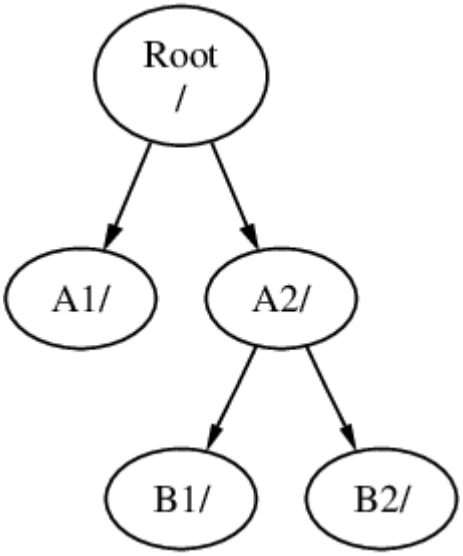



Ein Dateisystem wird in einem Verzeichnis eines anderen Dateisystems eingehangen. Wir hängen nun das Dateisystem `B` in das Verzeichnis A1 ein. Das Wurzelverzeichnis von `B` ersetzt nun das Verzeichnis A1 und die Verzeichnisse des Dateisystems `B` werden sichtbar:



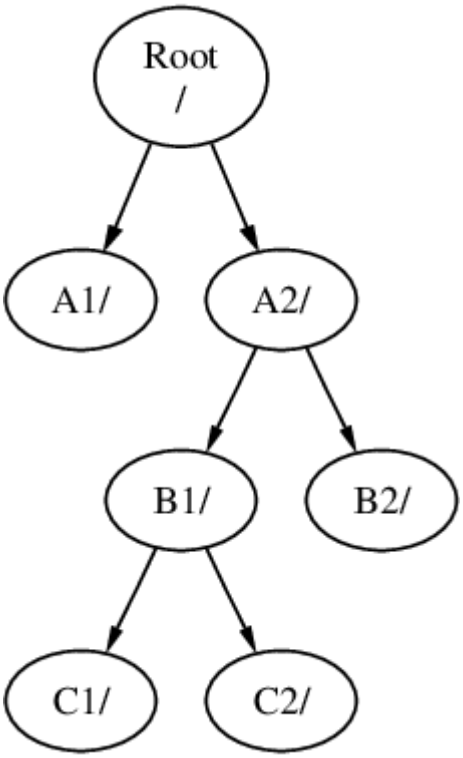

Jede Datei in den Verzeichnissen B1 oder B2 kann über den Pfad /A1/B1 oder /A1/B2 erreicht werden. Dateien aus dem Verzeichnis /A1 sind jetzt verborgen. Wenn das Dateisystem `B` wieder *abgehangen* wird (umount), erscheinen die verborgenen Dateien wieder.

Wenn das Dateisystem `B` unter dem Verzeichnis A2 eingehangen würde, sähe der Verzeichnisbaum so aus:

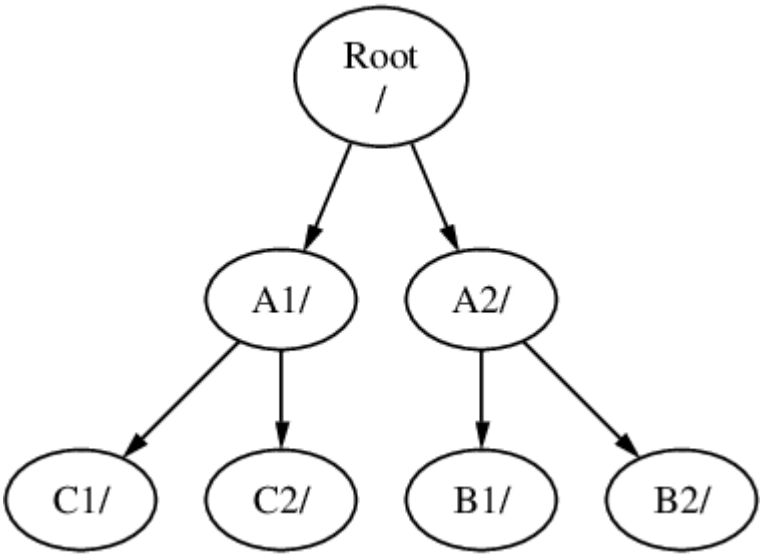



Die Dateien des Dateisystems `B` wären unter den Pfaden /A2/B1 und /A2/B2 erreichbar.

Dateisysteme können übereinander eingehangen werden. Der folgende Baum entsteht, wenn im letzten Beispiel das Dateisystem `C` in das Verzeichnis B1 des Dateisystems `B` eingehangen wird:




`C` könnte auch im Verzeichnis A1 eingehangen werden:




Sie können sogar mit nur einem großen Dateisystem auskommen. Dies hat mehrere Nachteile und einen Vorteil.

*Vorteile mehrerer Dateisysteme*

- Die Dateisysteme können mit unterschiedlichen Optionen (mount options) eingehangen werden. Beispielsweise kann das Wurzeldateisystem schreibgeschützt eingehangen werden, sodass es für Benutzer nicht möglich ist, versehentlich kritische Dateien zu editieren oder zu löschen. Von Benutzern beschreibbare Dateisysteme wie /home können mit der Option *nosuid* eingehangen werden, wenn sie von anderen Dateisystemen getrennt sind. Die *SUID*- und *GUID* -Bits verlieren auf solchen Dateisystemen ihre Wirkung und die Sicherheit des Systems kann dadurch erhöht werden.
- Die Lage von Dateien im Dateisystem wird, abhängig vom Gebrauch des Dateisystems, automatisch von FreeBSD optimiert. Ein Dateisystem mit vielen kleinen Dateien, die häufig geschrieben werden, wird anders behandelt als ein Dateisystem mit wenigen großen Dateien. Mit nur einem Dateisystem ist diese Optimierung unmöglich.
- In der Regel übersteht ein FreeBSD-Dateisystem auch einen Stromausfall. Allerdings kann ein Stromausfall zu einem kritischen Zeitpunkt das Dateisystem beschädigen. Wenn die Daten über mehrere Dateisysteme verteilt sind, lässt sich das System mit hoher Wahrscheinlichkeit noch starten. Dies erleichtert das Zurückspielen von Datensicherungen.

*Vorteil eines einzelnen Dateisystems*

- Dateisysteme haben eine festgelegte Größe. Es kann passieren, dass Sie eine Partition vergrößern müssen. Dies ist nicht leicht: Sie müssen die Daten sichern, das Dateisystem vergrößert anlegen und die gesicherten Daten zurückspielen.



FreeBSD kennt den Befehl growfs(8), mit dem man Dateisysteme im laufenden Betrieb vergrößern kann.

Dateisysteme befinden sich in Partitionen (damit sind nicht die normalen MS-DOS®-Partitionen gemeint). Jede Partition wird mit einem Buchstaben von `a` bis `h` bezeichnet und kann nur ein Dateisystem enthalten. Dateisysteme können daher über ihren Mount-Point, den Punkt an dem sie eingehangen sind, oder den Buchstaben der Partition, in der sie liegen, identifiziert werden.

FreeBSD benutzt einen Teil der Platte für den *Swap-Bereich*, um *virtuellen Speicher* zur Verfügung zu stellen. Dadurch kann der Rechner Anwendungen mehr Speicher zur Verfügung stellen als tatsächlich eingebaut ist. Wenn der Speicher knapp wird, kann FreeBSD nicht benutzte Daten in den Swap-Bereich auslagern. Die ausgelagerten Daten können später wieder in den Speicher geholt werden (dafür werden dann andere Daten ausgelagert).

Für einige Partitionen gelten besondere Konventionen:

| Partition | Konvention |
|---|---|
| a | Enthält normalerweise das Wurzeldateisystem. |
| b | Enthält normalerweise den Swap-Bereich. |
| c | Ist normalerweise genauso groß wie die Slice in der die Partition liegt. Werkzeuge, die auf der kompletten Slice arbeiten, wie ein Bad-Block-Scanner, können so die c-Partition benutzen. Für gewöhnlich wird in dieser Partition kein Dateisystem angelegt. |
| d | Früher hatte die d-Partition eine besondere Bedeutung. Heute ist dies nicht mehr der Fall und die Partition d kann wie jede andere Partition auch verwendet werden. |

In FreeBSD werden Festplatten in Slices, welche in Windows® als Partitionen bekannt sind, aufgeteilt und von 1 bis 4 durchnummeriert. Diese werden dann in Partitionen unterteilt, welche wiederum Dateisysteme enthalten und mit Buchstaben benannt werden.

Die Slice-Nummern werden mit vorgestelltem s hinter den Gerätenamen gestellt: "da0*s1*" ist die erste Slice auf dem ersten SCSI-Laufwerk. Auf einer Festplatte gibt es höchstens vier Slices. In einer Slice des passenden Typs kann es weitere logische Slices geben. Diese erweiterten Slices werden ab fünf durchnummeriert: "ada0*s5*" ist die erste erweiterte Slice auf einer SATA-Platte. Diese Geräte werden von Dateisystemen benutzt, die sich in einer kompletten Slice befinden müssen.

Slices, "dangerously dedicated"-Festplatten und andere Platten enthalten Partitionen, die mit Buchstaben von a bis h bezeichnet werden. Der Buchstabe wird an den Gerätenamen gehangen: "da0*a*" ist die a-Partition des ersten da-Laufwerks. Dieses Laufwerk ist "dangerously dedicated". "ada1s3*e*" ist die fünfte Partition in der dritten Slice der zweiten SATA-Platte.

Schließlich wird noch jede Festplatte des Systems eindeutig bezeichnet. Der Name einer Festplatte beginnt mit einem Code, der den Typ der Platte bezeichnet. Es folgt eine Nummer, die angibt, um welche Festplatte es sich handelt. Anders als bei Slices werden Festplatten von Null beginnend durchnummeriert. Gängige Festplatten-Namen sind in Laufwerk-Codes aufgeführt.

Wenn Sie eine Partition angeben, beinhaltet das den Plattennamen, s, die Slice-Nummer und den Buchstaben der Partition. Einige Beispiele finden Sie in Namen von Platten, Slices und Partitionen.

Der Aufbau einer Festplatte wird in Aufteilung einer Festplatte dargestellt.

Bei der Installation von FreeBSD legen Sie Slices auf der Festplatte an, erstellen Partitionen für FreeBSD innerhalb der Slice, erstellen ein Dateisystem oder Auslagerungsbereiche und entscheiden, welche Dateisysteme wo eingehangen werden.

*Tabelle 4. Laufwerk-Codes*

| Laufwerkstyp | Gerätename |
|---|---|
| SATA- und IDE-Festplatten | `ada` oder `ad` |
| SCSI-Festplatten und USB-Speichermedien | `da` |
| SATA- und IDE-CD-ROM-Laufwerke | `cd` oder `acd` |
| SCSI-CD-ROM-Laufwerke | `cd` |
| Diskettenlaufwerke | `fd` |
| Verschiedene proprietäre CD-ROM-Laufwerke | `mcd` für Mitsumi CD-ROM und `scd` für Sony CD-ROM |
| SCSI-Bandlaufwerke | `sa` |
| IDE-Bandlaufwerke | `ast` |
| RAID-Laufwerke | Beispiele sind `aacd` für Adaptec® AdvancedRAID, `mlxd` für Mylex®, `amrd` für AMI MegaRAID®, `idad` für Compaq Smart RAID, `twed` für 3ware® RAID. |

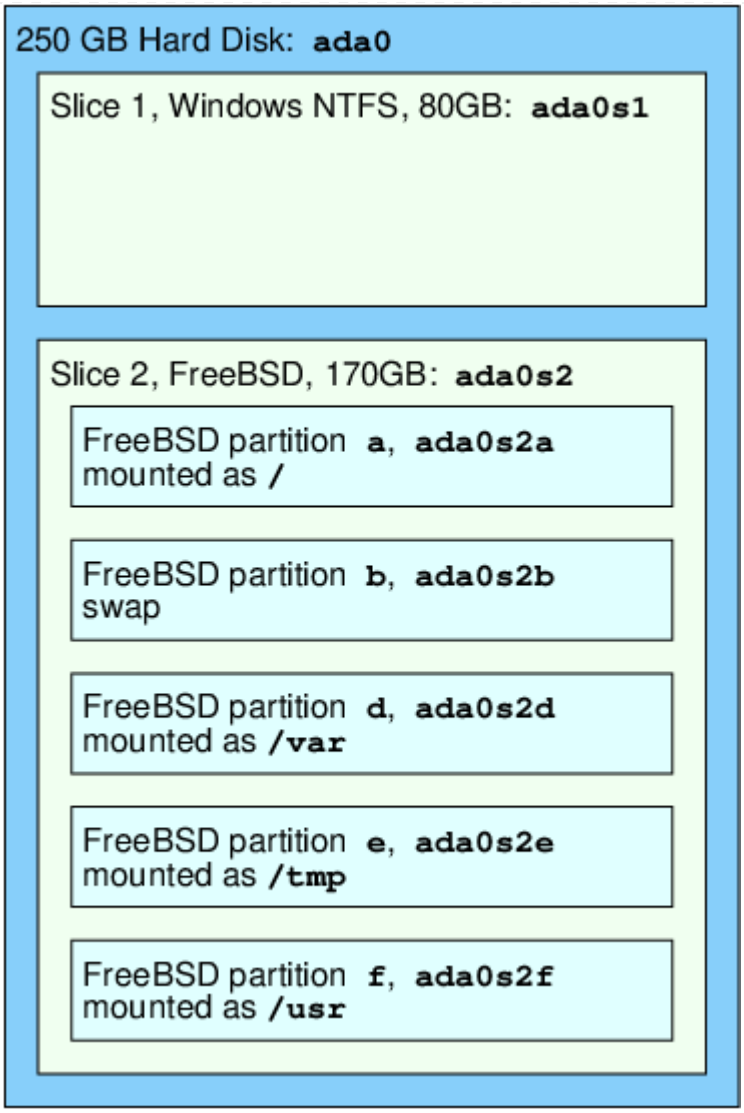
*Beispiel 13. Namen von Platten, Slices und Partitionen*

| Name | Bedeutung |
|---|---|
| ada0s1a | Die erste Partition (`a`) in der ersten Slice (`s1`) der ersten SATA-Festplatte (`ada0`). |
| da1s2e | Die fünfte Partition (`e`) der zweiten Slice (`s2`) auf der zweiten SCSI-Festplatte (`da1`). |

*Beispiel 14. Aufteilung einer Festplatte*

Das folgende Diagramm zeigt die Sicht von FreeBSD auf die erste SATA-Festplatte des Systems. Die Platte soll 250 GB groß sein und eine 80 GB große Slice (MS-DOS®-Partitionen) sowie eine 170 GB große Slice enthalten. Die erste Slice enthält ein Windows® NTFS-Dateisystem (C:), die zweite Slice enthält eine FreeBSD-Installation. Die FreeBSD-Installation in diesem Beispiel verwendet vier Datenpartitionen und einen Auslagerungsbereich.

Jede der vier Partitionen enthält ein Dateisystem. Das Wurzeldateisystem ist die `a`-Partition. In der `d`-Partition befindet sich /var und in der `f`-Partition befindet sich /usr. Die `c`-Partition bezieht sich auf die gesamte Slice und wird nicht für gewöhnliche Partitionen verwendet.




# 5.7. Anhängen und Abhängen von Dateisystemen

Ein Dateisystem wird am besten als ein Baum mit der Wurzel / veranschaulicht. /dev, /usr, und die anderen Verzeichnisse im Rootverzeichnis sind Zweige, die wiederum eigene Zweige wie /usr/local haben können.

Es gibt verschiedene Gründe, bestimmte dieser Verzeichnisse auf eigenen Dateisystemen anzulegen. /var enthält log/, spool/ sowie verschiedene andere temporäre Dateien und kann sich daher schnell füllen. Es empfiehlt sich, /var von / zu trennen, da es schlecht ist, wenn das Root-Dateisystem voll läuft.

Ein weiterer Grund bestimmte Verzeichnisbäume auf andere Dateisysteme zu legen, ist gegeben, wenn sich die Verzeichnisbäume auf gesonderten physikalischen oder virtuellen Platten, wie Network File System oder CD-ROM-Laufwerken, befinden.

## 5.7.1. Die fstab Datei

Während des Boot-Prozesses (FreeBSDs Bootvorgang) werden in /etc/fstab aufgeführte Verzeichnisse, sofern sie nicht mit der Option `noauto` versehen sind, automatisch angehangen. Diese Datei enthält Einträge in folgendem Format:

```
device  /mount-point    fstype  options dumpfreq    passno
```

`device`

Ein existierender Gerätename wie in Laufwerk-Codes beschrieben.

`mount-point`

Ein existierendes Verzeichnis, auf dem das Dateisystem gemountet wird.

`fstype`

Der Typ des Dateisystems, der an mount(8) weitergegeben wird. FreeBSDs Standarddateisystem ist `ufs`.

`options`

Entweder `rw` für beschreibbare Dateisysteme oder `ro` für schreibgeschützte Dateisysteme, gefolgt von weiteren benötigten Optionen. Eine häufig verwendete Option ist `noauto` für Dateisysteme, die während der normalen Bootsequenz nicht angehangen werden sollen. Weitere Optionen finden sich in mount(8).

`dumpfreq`

Wird von dump(8) benutzt, um bestimmen zu können, welche Dateisysteme gesichert werden müssen. Fehlt der Wert, wird `0` angenommen.

`passno`

Bestimmt die Reihenfolge, in der die Dateisysteme überprüft werden sollen. Für Dateisysteme, die übersprungen werden sollen, ist `passno` auf `0` zu setzen. Für das Root-Dateisystem, das vor allen anderen überprüft werden muss, sollte der Wert von `passno``1` betragen. Allen anderen Dateisystemen sollten Werte größer `1` zugewiesen werden. Wenn mehrere Dateisysteme den gleichen Wert besitzen, wird fsck(8) versuchen, diese parallel zu überprüfen.

Lesen Sie fstab(5) für weitere Informationen über das Format von /etc/fstab und dessen Optionen.

### 5.7.2. Verwendung von mount(8)

Dateisysteme werden mit mount(8) eingehängt. In der grundlegenden Form wird es wie folgt benutzt:

```
# mount device mountpoint
```

Dieser Befehl bietet viele Optionen, die in mount(8) beschrieben werden. Die am häufigsten verwendeten Optionen sind:

*Optionen von* `mount`

`-a`

Hängt alle Dateisysteme aus /etc/fstab an. Davon ausgenommen sind Dateisysteme, die mit

"noauto" markiert sind, die mit der Option `-t` ausgeschlossen wurden und Dateisysteme, die schon angehangen sind.

`-d`

Führt alles bis auf den `mount`-Systemaufruf aus. Nützlich ist diese Option in Verbindung mit `-v`. Damit wird angezeigt, was mount(8) tatsächlich versuchen würde, um das Dateisystem anzuhängen.

`-f`

Erzwingt das Anhängen eines unsauberen Dateisystems (riskant) oder die Rücknahme des Schreibzugriffs, wenn der Status des Dateisystems von beschreibbar auf schreibgeschützt geändert wird.

`-r`

Hängt das Dateisystem schreibgeschützt ein. Dies kann auch durch Angabe von `-o ro` erreicht werden.

**`-t`*fstype***

Hängt das Dateisystem mit dem angegebenen Typ an, oder hängt nur Dateisysteme mit dem angegebenen Typ an, wenn `-a` angegeben wurde. "ufs" ist das Standarddateisystem.

`-u`

Aktualisiert die Mountoptionen des Dateisystems.

`-v`

Geschwätzig sein.

`-w`

Hängt das Dateisystem beschreibbar an.

Die folgenden Optionen können durch eine Kommata separierte Liste an `-o` übergeben werden:

**nosuid**

SetUID und SetGID Bits werden auf dem Dateisystem nicht beachtet. Dies ist eine nützliche Sicherheitsfunktion.

### 5.7.3. Verwendung von umount(8)

umount(8) hängt ein Dateisystem ab. Dieser Befehl akzeptiert als Parameter entweder einen Mountpoint, einen Gerätenamen, `-a` oder `-A`.

Jede Form akzeptiert `-f`, um das Abhängen zu erzwingen, und `-v`, um etwas geschwätziger zu sein. Seien Sie bitte vorsichtig mit `-f`, da der Computer abstürzen kann oder es können Daten auf dem Dateisystem beschädigt werden.

Um alle Dateisysteme abzuhängen, oder nur diejenigen, die mit `-t` gelistet werden, wird `-a` oder `-A` benutzt. Beachten Sie, dass `-a` das Root-Dateisystem nicht aushängt.

# 5.8. Prozesse und Dämonen

FreeBSD ist ein Multitasking-Betriebssystem. Jedes Programm, das zu irgendeiner Zeit läuft wird als *Prozess* bezeichnet. Jedes laufende Kommando startet mindestens einen neuen Prozess. Dazu gibt es eine Reihe von Systemprozessen, die von FreeBSD ausgeführt werden.

Jeder Prozess wird durch eine eindeutige Nummer identifiziert, die *Prozess-ID* (*PID*) genannt wird. Prozesse haben ebenso wie Dateien einen Besitzer und eine Gruppe, die festlegen, welche Dateien und Geräte der Prozess benutzen kann. Die meisten Prozesse haben auch einen Elternprozess, der sie gestartet hat. Beispielsweise ist die Shell ein Prozess. Jedes in Shell gestartete Kommando ist dann ein neuer Prozess, der die Shell als Elternprozess besitzt. Die Ausnahme hiervon ist ein spezieller Prozess namens init(8), der beim booten immer als erstes gestartet wird und der immer die PID`1` hat.

Manche Programme erwarten keine Eingaben vom Benutzer und lösen sich bei erster Gelegenheit von ihrem Terminal. Ein Webserver zum Beispiel antwortet auf Web-Anfragen und nicht auf Benutzereingaben. Mail-Server sind ein weiteres Beispiel für diesen Typ von Anwendungen. Diese Programme sind als *Dämonen* bekannt. Der Begriff Dämon stammt aus der griechischen Mythologie und bezeichnet ein Wesen, das weder gut noch böse ist und welches unsichtbar nützliche Aufgaben verrichtet. Deshalb ist das BSD Maskottchen dieser fröhlich aussehende Dämon mit Turnschuhen und Dreizack.

Programme, die als Dämon laufen, werden entsprechend einer Konvention mit einem "d" am Ende benannt. BIND steht beispielsweise für Berkeley Internet Name Domain, das tatsächlich laufende Programm heißt aber `named`. Der Apache Webserver wird `httpd` genannt und der Druckerspool-Dämon heißt lpd(8). Dies ist allerdings nur eine Konvention. Der Dämon der Anwendung Sendmail heißt beispielsweise `sendmail` und nicht `maild`.

## 5.8.1. Prozesse beobachten

Um die Prozesse auf dem System zu sehen, benutzen Sie ps(1) und top(1). Eine statische Liste der laufenden Prozesse, deren PIDs, Speicherverbrauch und die Kommandozeile, mit der sie gestartet wurden, erhalten Sie mit ps(1). Um alle laufenden Prozesse in einer Anzeige zu sehen, die alle paar Sekunden aktualisiert wird, so dass Sie interaktiv sehen können was der Computer macht, benutzen Sie top(1).

In der Voreinstellung zeigt ps(1) nur die laufenden Prozesse, die dem Benutzer gehören. Zum Beispiel:

```
% ps
  PID TT  STAT    TIME COMMAND
 8203  0  Ss   0:00.59 /bin/csh
 8895  0  R+   0:00.00 ps
```

Die Ausgabe von ps(1) ist in einer Anzahl von Spalten organisiert. Die `PID` Spalte zeigt die Prozess-ID. PIDs werden von 1 beginnend bis 99999 zugewiesen und fangen wieder von vorne an. Ist eine PID bereits vergeben, wird diese allerdings nicht erneut vergeben. Die Spalte `TT` zeigt den Terminal, auf dem das Programm läuft. `STAT` zeigt den Status des Programms und `TIME` gibt die Zeit an, die das

Programm auf der CPU gelaufen ist. Dies ist nicht unbedingt die Zeit, die seit dem Start des Programms vergangen ist, da die meisten Programme hauptsächlich auf bestimmte Dinge warten, bevor sie wirklich CPU-Zeit verbrauchen. Unter der Spalte `COMMAND` findet sich schließlich die Kommandozeile, mit der das Programm gestartet wurde.

ps(1) besitzt viele Optionen, um die angezeigten Informationen zu beeinflussen. Eine nützliche Kombination ist `auxww`. `a` zeigt Information über alle laufenden Prozesse aller Benutzer. Der Name des Besitzers des Prozesses, sowie Informationen über den Speicherverbrauch werden mit `u` angezeigt. `x` zeigt auch Dämonen-Prozesse an, und `ww` veranlasst ps(1) die komplette Kommandozeile für jeden Befehl anzuzeigen, anstatt sie abzuschneiden, wenn sie zu lang für die Bildschirmausgabe wird.

Die Ausgabe von top(1) sieht in etwa so aus:

```
% top
last pid:  9609;  load averages:  0.56,  0.45,  0.36              up 0+00:20:03
10:21:46
107 processes: 2 running, 104 sleeping, 1 zombie
CPU:  6.2% user,  0.1% nice,  8.2% system,  0.4% interrupt, 85.1% idle
Mem: 541M Active, 450M Inact, 1333M Wired, 4064K Cache, 1498M Free
ARC: 992M Total, 377M MFU, 589M MRU, 250K Anon, 5280K Header, 21M Other
Swap: 2048M Total, 2048M Free

  PID USERNAME    THR PRI NICE   SIZE    RES STATE   C   TIME   WCPU COMMAND
  557 root          1 -21  r31   136M 42296K select  0   2:20  9.96% Xorg
 8198 dru           2  52    0   449M 82736K select  3   0:08  5.96% kdeinit4
 8311 dru          27  30    0  1150M   187M uwait   1   1:37  0.98% firefox
  431 root          1  20    0 14268K  1728K select  0   0:06  0.98% moused
 9551 dru           1  21    0 16600K  2660K CPU3    3   0:01  0.98% top
 2357 dru           4  37    0   718M   141M select  0   0:21  0.00% kdeinit4
 8705 dru           4  35    0   480M    98M select  2   0:20  0.00% kdeinit4
 8076 dru           6  20    0   552M   113M uwait   0   0:12  0.00% soffice.bin
 2623 root          1  30   10 12088K  1636K select  3   0:09  0.00% powerd
 2338 dru           1  20    0   440M 84532K select  1   0:06  0.00% kwin
 1427 dru           5  22    0   605M 86412K select  1   0:05  0.00% kdeinit4
```

Die Ausgabe ist in zwei Abschnitte geteilt. In den ersten fünf Kopfzeilen finden sich die zuletzt zugeteilte PID, die Systemauslastung (engl. load average), die Systemlaufzeit (die Zeit seit dem letzten Reboot) und die momentane Zeit. Die weiteren Zahlen im Kopf beschreiben wie viele Prozesse momentan laufen, wie viel Speicher und Swap verbraucht wurde und wie viel Zeit das System in den verschiedenen CPU-Modi verbringt. Wenn das ZFS-Kernelmodul geladen ist, dann zeigt die Zeile `ARC`, wie viele Daten aus dem Cache gelesen wurden.

Darunter befinden sich einige Spalten mit ähnlichen Informationen wie in der Ausgabe von ps(1), beispielsweise die PID, den Besitzer, die verbrauchte CPU-Zeit und das Kommando, das den Prozess gestartet hat. top(1) zeigt in zwei Spalten den Speicherverbrauch des Prozesses an. Die erste Spalte gibt den gesamten Speicherverbrauch des Prozesses an, in der zweiten Spalte wird der aktuelle Verbrauch angegeben.

Die Anzeige wird von top(1) automatisch alle zwei Sekunden aktualisiert. Ein anderer Intervall kann mit `-s` spezifiziert werden.

### 5.8.2. Stoppen von Prozessen

Eine Möglichkeit mit einem laufenden Prozess zu kommunizieren, ist über das Versenden von *Signalen* mittels kill(1). Es gibt eine Reihe von verschiedenen Signalen. Manche haben eine feste Bedeutung, während andere in der Dokumentation der Anwendung beschrieben sind. Ein Benutzer kann ein Signal nur an einen Prozess senden, welcher ihm gehört. Wird versucht ein Signal an einen Prozess eines anderen Benutzers zu senden, resultiert dies in einem Zugriffsfehler mangels fehlender Berechtigungen. Die Ausnahme ist der `root`-Benutzer, welcher jedem Prozess Signale senden kann.

FreeBSD kann auch ein Signal an einen Prozess senden. Wenn eine Anwendung schlecht geschrieben ist und auf Speicher zugreift, auf den sie nicht zugreifen soll, so sendet FreeBSD dem Prozess das *Segmentation Violation* Signal (`SIGSEGV`). Wenn eine Anwendung programmiert wurde, den alarm(3) Systemaufruf zu benutzen, um nach einiger Zeit benachrichtigt zu werden, bekommt sie das "Alarm"-Signal (`SIGALRM`) gesendet.

Zwei Signale können benutzt werden, um einen Prozess zu stoppen: `SIGTERM` und `SIGKILL`. `SIGTERM` fordert den Prozess höflich zum Beenden auf. Der Prozess kann das Signal abfangen und hat dann Gelegenheit Logdateien zu schließen und die Aktion, die er durchführte, abzuschließen. In manchen Situationen kann der Prozess `SIGTERM` ignorieren, wenn er eine Aktion durchführt, die nicht unterbrochen werden darf.

`SIGKILL` kann von keinem Prozess ignoriert werden. Wird einem Prozess `SIGKILL` geschickt, dann wird FreeBSD diesen sofort beenden.

Andere häufig verwendete Signale sind `SIGHUP`, `SIGUSR1` und `SIGUSR2`. Da diese Signale für allgemeine Zwecke vorgesehen sind, werden verschiedene Anwendungen unterschiedlich auf diese Signale reagieren.

Ändern Sie beispielsweise die Konfiguration eines Webservers, so muss dieser angewiesen werden, seine Konfiguration neu zu lesen. Ein Neustart von `httpd` würde dazu führen, dass der Server für kurze Zeit nicht erreichbar ist. Senden Sie dem Dämon stattdessen das `SIGHUP`-Signal. Es sei erwähnt, dass verschiedene Dämonen sich anders verhalten. Lesen Sie die Dokumentation des entsprechenden Dämonen um zu überprüfen, ob der Dämon bei einem `SIGHUP` die gewünschten Ergebnisse erzielt.

*Procedure: Verschicken von Signalen*

Das folgende Beispiel zeigt, wie Sie inetd(8) ein Signal schicken. Die Konfigurationsdatei von inetd(8) ist /etc/inetd.conf. Diese Konfigurationsdatei liest inetd(8) ein, wenn er `SIGHUP` empfängt.

1. Suchen Sie mit pgrep(1) die PID des Prozesses, dem Sie ein Signal schicken wollen. In diesem Beispiel ist die PID von inetd(8) 198:

```
% pgrep -l inetd
```

```
198 inetd -wW
```

2. Benutzen Sie kill(1), um ein Signal zu senden. Da inetd(8) dem Benutzer `root` gehört, müssen Sie zuerst mit su(1) `root` werden:

```
% su
Password:
# /bin/kill -s HUP 198
```

kill(1) wird, wie andere UNIX® Kommandos auch, keine Ausgabe erzeugen, wenn das Kommando erfolgreich war. Wird versucht, einem Prozess der nicht dem Benutzer gehört, ein Signal zu senden, dann wird die Meldung `kill: PID: Operation not permitted` ausgegeben. Ein Tippfehler bei der Eingabe der PID führt dazu, dass das Signal an einen falschen Prozess gesendet wird, was zu negativen Ergebnissen führen kann, oder das Signal wird an eine PID gesendet die derzeit nicht in Gebrauch ist, was zu dem Fehler `kill: PID: No such process` führt.



*Warum sollte man `/bin/kill` benutzen?*

Viele Shells stellen `kill` als internes Kommando zur Verfügung, das heißt die Shell sendet das Signal direkt, anstatt /bin/kill zu starten. Beachten Sie, dass die unterschiedlichen Shells eine andere Syntax benutzen, um die Namen der Signale anzugeben. Anstatt jede Syntax zu lernen, kann es einfacher sein, `/bin/kill` direkt aufzurufen.

Beim Versenden von anderen Signalen, ersetzen Sie `TERM` oder `KILL` in der Kommandozeile mit dem Namen des Signals.



Das zufällige Beenden eines Prozesses kann gravierende Auswirkungen haben. Insbesondere init(8), mit der PID 1, ist ein Spezialfall. `/bin/kill -s KILL 1` ist ein schneller, jedoch nicht empfohlener Weg, das System herunterzufahren. Überprüfen Sie die Argumente von kill(1) *immer* zweimal *bevor* Sie `Return` drücken.

## 5.9. Shells

Eine *Shell* stellt eine Kommandozeilen-Schnittstelle zur Interaktion mit dem Betriebssystem zur Verfügung. Sie empfängt Befehle von einem Eingabekanal und führt diese aus. Viele Shells bieten eingebaute Funktionen, die die tägliche Arbeit erleichtern, beispielsweise eine Dateiverwaltung, die Vervollständigung von Dateinamen (Globbing), Kommandozeilen-Editor, sowie Makros und Umgebungsvariablen. FreeBSD enthält einige Shells, darunter die Bourne Shell (sh(1)) und die verbesserte C-Shell (tcsh(1)). Weitere Shells, wie `zsh` oder `bash`, befinden sich in der Ports-Sammlung.

Die verwendete Shell ist letztlich eine Frage des Geschmacks. Ein C-Programmierer, findet vielleicht eine C-artige Shell wie tcsh(1) angenehmer. Ein Linux®-Benutzer bevorzugt vielleicht `bash`. Jede

Shell hat ihre speziellen Eigenschaften, die mit der bevorzugten Arbeitsumgebung des Benutzers harmonieren kann oder nicht. Deshalb stehen mehrere Shells zur Auswahl.

Ein verbreitetes Merkmal in Shells ist die Dateinamen-Vervollständigung. Nachdem der Benutzer einige Buchstaben eines Kommandos oder eines Dateinamen eingeben hat, vervollständigt die Shell den Rest durch drücken der `Tab`-Taste. Angenommen, Sie haben zwei Dateien foobar und football. Um foobar zu löschen, kann der Benutzer `rm foo` eingeben und `Tab` drücken um den Dateinamen zu vervollständigen.

Die Shell wird lediglich `rm foo` anzeigen. Sie konnte den Dateinamen nicht vervollständigen, da sowohl foobar als auch football mit `foo` anfangen. Einige Shells geben einen Signalton aus, oder zeigen alle Möglichkeiten an, wenn mehr als ein Name mit dem gegebenen Muster übereinstimmt. Der Benutzer muss dann weitere Zeichen eingeben, damit die Shell den gewünschten Dateinamen bestimmen kann. Durch Eingabe von `t` und erneutes Drücken von `Tab` ist die Shell in der Lage, den gewünschten Dateinamen zu vervollständigen.

Ein weiteres Merkmal der Shell ist der Gebrauch von Umgebungsvariablen. Dies sind veränderbare Schlüsselpaare im Umgebungsraum der Shell, die jedes von der Shell aufgerufene Programm lesen kann. Daher enthält der Umgebungsraum viele Konfigurationsdaten für Programme. Gebräuchliche Umgebungsvariablen zeigt verbreitete Umgebungsvariablen und deren Bedeutung. Beachten Sie, dass die Namen der Umgebungsvariablen immer in Großbuchstaben geschrieben sind:

*Tabelle 5. Gebräuchliche Umgebungsvariablen*

| Variable | Beschreibung |
|---|---|
| `USER` | Name des angemeldeten Benutzers. |
| `PATH` | Liste mit Verzeichnissen (getrennt durch Doppelpunkt) zum Suchen nach Programmen. |
| `DISPLAY` | Der Name des Xorg-Bildschirms, auf dem Ausgaben erfolgen sollen. |
| `SHELL` | Die aktuelle Shell. |
| `TERM` | Name des Terminaltyps des Benutzers. Benutzt, um die Fähigkeiten des Terminals zu bestimmen. |
| `TERMCAP` | Datenbankeintrag der Terminal Escape Codes, benötigt um verschieden Terminalfunktionen auszuführen. |
| `OSTYPE` | Typ des Betriebssystems. |
| `MACHTYPE` | Die CPU-Architektur des Systems. |
| `EDITOR` | Vom Benutzer bevorzugter Text-Editor. |
| `PAGER` | Vom Benutzer bevorzugter Text-Betrachter. |
| `MANPATH` | Liste mit Verzeichnissen (getrennt durch Doppelpunkt) zum Suchen nach Manualpages. |

Das Setzen von Umgebungsvariablen unterscheidet sich von Shell zu Shell. In tcsh(1) und csh(1) wird dazu `setenv` benutzt. sh(1) und `bash` benutzen `export` um Umgebungsvariablen zu setzen. Dieses Beispiel für die tcsh(1)-Shell setzt die Variable `EDITOR` auf /usr/local/bin/emacs:

```
% setenv EDITOR /usr/local/bin/emacs
```

Der entsprechende Befehl für `bash` wäre:

```
% export EDITOR="/usr/local/bin/emacs"
```

Um eine Umgebungsvariable zu expandieren, geben Sie in der Kommandozeile das Zeichen `$` vor dessen Namen ein. Zum Beispiel gibt `echo $TERM` den aktuellen Wert von `$TERM` aus.

Shells behandeln Spezialzeichen, so genannte Metazeichen, als besondere Darstellungen für Daten. Das häufigste Zeichen ist *, das eine beliebige Anzahl Zeichen in einem Dateinamen repräsentiert. Metazeichen können zur Vervollständigung von Dateinamen (Globbing) benutzt werden. Beispielsweise liefert `echo *` nahezu das gleiche wie `ls`, da die Shell alle Dateinamen die mit * übereinstimmen, an `echo` weitergibt.

Um zu verhindern, dass die Shell ein Sonderzeichen interpretiert, schützt man es, indem man einen Backslash (\) voranstellt. Zum Beispiel zeigt `echo $TERM` die Einstellung des Terminals an, wohingegen `echo \$TERM` einfach die Zeichenfolge `$TERM` ausgibt.

### 5.9.1. Ändern der Shell

Der einfachste Weg die Standard Shell zu ändern, ist `chsh` zu benutzen. `chsh` startet den Editor, welcher durch die Umgebungsvariable `EDITOR` gesetzt ist. Standardmäßig ist dies vi(1). Tragen Sie in die Zeile die mit `Shell:` beginnt, den absoluten Pfad der neuen Shell ein.

Alternativ setzt `chsh -s` die Shell, ohne dabei einen Editor aufzurufen. Um die Shell zum Beispiel auf `bash` zu ändern, geben Sie folgenden Befehl ein:

```
% chsh -s /usr/local/bin/bash
```



Die neue Shell *muss* in /etc/shells aufgeführt sein. Wurde die Shell aus der FreeBSD Ports-Sammlung installiert, so wie in Installieren von Anwendungen: Pakete und Ports beschrieben, sollte sie automatisch zu dieser Datei hinzugefügt worden sein. Wenn der Eintrag fehlt, nutzen Sie folgenden Befehl, und ersetzen Sie den Pfad mit dem Pfad zur gewünschten Shell:

```
# echo /usr/local/bin/bash >> /etc/shells
```

Danach kann chsh(1) erneut aufgerufen werden.

### 5.9.2. Fortgeschrittene Shell Techniken

Die UNIX®-Shell ist nicht nur ein Kommandozeileninterpreter, sie ist ein leistungsfähiges Werkzeug, das Benutzern die Ausführung von Befehlen ermöglicht. Es kann die Ein- und Ausgabe umleiten und Befehle miteinander verketten, um die finale Ausgabe zu verbessern. Diese Funktionalität, gepaart mit den eingebauten Befehlen, bietet dem Benutzer eine Umgebung, welche die Effizienz erheblich steigern kann.

Als Redirection bezeichnet man die Umleitung der Ein- oder Ausgabe in einen anderen Befehl oder Datei. Um beispielsweise die Ausgabe des Befehls ls(1) in eine Datei zu schreiben, muss die Ausgabe umgeleitet werden:

```
% ls > Verzeichnis_Ausgabe.txt
```

Die Datei Verzeichnis_Ausgabe.txt enthält nun den Verzeichnisinhalt. Einige Befehle, wie beispielsweise sort(1), können verwendet werden um von der Eingabe zu lesen. Wenn Sie die Ausgabe sortieren möchten, müssen Sie die Eingabe umleiten:

```
% sort < Verzeichnis_Ausgabe.txt
```

Die Eingabe wird sortiert und auf dem Bildschirm ausgegeben. Um diese Ausgabe wiederum in eine Datei umzuleiten, kann die Ausgabe von sort(1) umgeleitet werden:

```
% sort < Verzeichnis_Ausgabe.txt > Sortierte_Ausgabe.txt
```

In den bisherigen Beispielen wurden für die Umleitung Dateideskriptoren verwendet. Jedes UNIX®-System verfügt über drei Dateideskriptoren: Standardeingabe (stdin), Standardausgabe (stdout) und Stardardfehlerausgabe (stderr). Jeder Deskriptor hat einen bestimmten Zweck. Die Eingabe könnte von einer Tastatur, einer Maus oder einem anderen Eingabegerät stammen. Die Ausgabe könnte der Bildschirm oder ein Drucker sein. Die Standardfehlerausgabe wird zur Diagnose und für Fehlermeldungen verwendet. Alle drei Deskriptoren arbeiten I/O basiert und werden häufig als Streams bezeichnet.

Die Verwendung von Deskriptoren erlaubt es der Shell, die Ein- und Ausgabe von verschiedenen Kommandos umzuleiten und zu teilen. Eine weitere Möglichkeit zur Umleitung bietet der Pipe-Operator.

Der UNIX® Pipe-Operator "|" wird verwendet, um die Ausgabe eines Kommandos an ein anderes Programm zu übergeben. Grundsätzlich bedeutet dies, dass die Standardausgabe eines Programms als Standardeingabe für ein weiteres Programm verwendet wird. Ein Beispiel:

```
% cat Verzeichnis_Auflistung.txt | sort | less
```

In diesem Beispiel wird der Inhalt von Verzeichnis_Auflistung.txt sortiert und die Ausgabe an less(1) übergeben. Dies erlaubt es dem Benutzer, die Ausgabe Schritt für Schritt und im eigenen

Tempo zu betrachten.

## 5.10. Text-Editoren

Die meiste Konfiguration unter FreeBSD wird durch das Editieren von Textdateien erledigt. Deshalb ist es eine gute Idee, mit einem Texteditor vertraut zu werden. FreeBSD hat ein paar davon im Basissystem und sehr viel mehr in der Ports-Sammlung.

Ein einfach zu erlernender Editor ist ee(1), was für easy editor steht. Um diesen Editor zu starten, gibt man in der Kommandozeile `ee filename` ein, wobei *filename* den Namen der zu editierenden Datei darstellt. Einmal im Editor, finden sich alle Editor-Funktionen oben im Display aufgelistet. Das Einschaltungszeichen (^) steht für die `Ctrl` (oder `Strg`) Taste, mit `^e` ist also die Tastenkombination `Ctrl` + `e` gemeint. Um ee(1) zu verlassen, drücken Sie `Esc` und wählen dann im Hauptmenü `leave editor` aus. Der Editor fragt nach, ob Sie speichern möchten, wenn die Datei verändert wurde.

FreeBSD verfügt über leistungsfähigere Editoren wie vi(1) als Teil des Basissystems. Andere Editoren wie editors/emacs und editors/vim sind Teil der Ports-Sammlung. Diese Editoren bieten höhere Funktionalität, jedoch auf Kosten einer etwas schwierigeren Erlernbarkeit. Das Erlernen eines leistungsfähigeren Editors, wie vim oder Emacs, kann auf lange Sicht Zeit einsparen.

Viele Anwendungen, die Dateien verändern oder Texteingabe erwarten, werden automatisch einen Texteditor öffnen. Um den Standardeditor zu ändern, wird die Umgebungsvariable `EDITOR` gesetzt, wie im Abschnitt Shells beschrieben.

## 5.11. Geräte und Gerätedateien

Der Begriff Gerät wird meist in Verbindung mit Hardware wie Laufwerken, Druckern, Grafikkarten oder Tastaturen gebraucht. Der Großteil der Meldungen, die beim Booten von FreeBSD angezeigt werden, beziehen sich auf gefundene Geräte. Eine Kopie dieser Bootmeldungen wird in /var/run/dmesg.boot gespeichert.

Jedes Gerät verfügt über einen Gerätenamen und Gerätenummer. Zum Beispiel steht ada0 für die erste SATA Festplatte, während kbd0 die Tastatur repräsentiert.

Auf die meisten Geräte wird unter FreeBSD über spezielle Gerätedateien im /dev Verzeichnis zugegriffen.

## 5.12. Manualpages

### 5.12.1. Manualpages

Die umfassendste Dokumentation rund um FreeBSD gibt es in Form von Manualpages. Annähernd jedes Programm im System bringt eine kurze Referenzdokumentation mit, die die grundsätzliche Funktion und verschiedene Parameter erklärt. Diese Manuals können mit `man` eingesehen werden:

```
% man Kommando
```

*Kommando* ist der Name des Kommandos, über das man etwas erfahren will. Um beispielsweise mehr über das Kommando ls(1) zu erfahren, geben Sie ein:

```
% man ls
```

Die Manualpages sind in nummerierte Sektionen unterteilt, die jeweils ein Thema darstellen. In FreeBSD sind die folgenden Sektionen verfügbar:

1. Benutzerkommandos.
2. Systemaufrufe und Fehlernummern.
3. Funktionen der C Bibliothek.
4. Gerätetreiber.
5. Dateiformate.
6. Spiele und andere Unterhaltung.
7. Verschiedene Informationen.
8. Systemverwaltung und -Kommandos.
9. Kernel Schnittstellen.

In einigen Fällen kann dasselbe Thema in mehreren Sektionen auftauchen. Es gibt zum Beispiel ein `chmod` Benutzerkommando und einen `chmod()` Systemaufruf. Um man(1) mitzuteilen, aus welcher Sektion die Information angezeigt werden soll, kann die Sektionsnummer mit angeben werden:

```
% man 1 chmod
```

Dies wird Ihnen die Manualpage für das Benutzerkommando chmod(1) zeigen. Verweise auf eine Sektion der Manualpages werden traditionell in Klammern gesetzt. So bezieht sich chmod(1) auf das Benutzerkommando und chmod(2) auf den Systemaufruf.

Wenn das Kommando nicht bekannt ist, kann `man -k` benutzt werden, um nach Schlüsselbegriffen in den Kommandobeschreibungen zu suchen:

```
% man -k mail
```

Dieser Befehl zeigt eine Liste von Kommandos, deren Beschreibung das Schlüsselwort "mail" enthält. Die gleiche Funktionalität erhalten Sie auch, wenn Sie apropos(1) benutzen.

Um die Beschreibungen der Kommandos in /usr/bin zu lesen, geben Sie ein:

```
% cd /usr/bin
% man -f * | more
```

Dasselbe erreichen Sie durch Eingabe von:

```
% cd /usr/bin
% whatis * | more
```

### 5.12.2. GNU Info Dateien

FreeBSD enthält verschiedene Anwendungen und Utilities der Free Software Foundation (FSF). Zusätzlich zu den Manualpages können diese Programme Hypertext-Dokumente enthalten, die `info` -Seiten genannt werden. Diese Dokumente können mit info(1) ansehen kann. Wenn editors/emacs installiert ist, kann auch der info-Modus von emacs benutzt werden.

Um info(1) zu benutzen, geben Sie ein:

```
% info
```

Eine kurze Einführung gibt es mit `h`; eine Befehlsreferenz erhalten Sie durch Eingabe von: `?`.

# Kapitel 6. Installieren von Anwendungen: Pakete und Ports

## 6.1. Übersicht

FreeBSD enthält eine umfassende Sammlung von Systemwerkzeugen, die Teil des Basissystems sind. Darüber hinaus stellt FreeBSD zwei sich ergänzende Methoden zur Installation von Drittanbieter-Software zur Verfügung: Die Ports-Sammlung zur Installation aus dem Quellcode sowie Pakete zur Installation von vorkompilierten binären Softwarepaketen. Beide Methoden können benutzt werden, um Anwendungen von lokalen Medien oder über das Netzwerk zu installieren.

Dieses Kapitel behandelt die folgenden Themen:

- Den Unterschied zwischen binären Softwarepaketen und Ports.
- Wie man Drittanbieter-Software findet, die nach FreeBSD portiert wurde.
- Wie Binärpakete mit pkg verwaltet werden.
- Den Bau von Drittanbieter-Software aus dem Quellcode mithilfe der Ports-Sammlung.
- Wie man die Dateien findet, die zusammen mit der Anwendung installiert wurden.
- Was zu tun ist, wenn die Installation einer Software fehlschlägt.

## 6.2. Installation von Software

Die typischen Schritte zur Installation von Drittanbieter-Software auf einem UNIX® System sind:

1. Download der Software, die als Quelltext oder im Binärformat vorliegen kann.
2. Auspacken der Software. Dies ist typischerweise ein mit compress(1), gzip(1), bzip2(1) oder xz(1) komprimiertes Tar-Archiv.
3. Durchsuchen der Dokumentation, die sich in INSTALL, README oder mehreren Dateien im Verzeichnis doc/ befindet, nach Anweisungen, wie die Software zu installieren ist.
4. Kompilieren der Software, wenn sie als Quelltext vorliegt. Dazu muss vielleicht das Makefile angepasst, oder `configure` ausgeführt werden.
5. Testen und installieren der Software.

Ein FreeBSD-Port ist eine Sammlung von Dateien, die das Kompilieren der Quelltexte einer Anwendung automatisieren. Die Dateien, die ein Port umfasst enthalten alle notwendigen Informationen um die Anwendung herunterzuladen, zu extrahieren, anzupassen und zu installieren.

Wenn die Software nicht bereits für FreeBSD angepasst und getestet wurde, muss vielleicht sogar der Quelltext angepasst werden, damit die Software funktioniert.

Bislang wurden über 37000 Anwendungen von Drittanbietern nach FreeBSD portiert. Falls möglich,

werden diese Anwendungen als vorkompilierte *Pakete* zur Verfügung gestellt.

Pakete können mit FreeBSDs Paketverwaltungswerkzeugen manipuliert werden.

Pakete und Ports beachten Abhängigkeiten zwischen Anwendungen. Wenn ein Paket oder die Ports-Sammlung benutzt wird, um eine Anwendung zu installieren, dann werden fehlende Bibliotheken zuerst installiert, sofern sie nicht schon vorher installiert waren.

Ein FreeBSD-Paket enthält vorkompilierte Kopien aller Befehle für eine Anwendung, sowie zusätzliche Konfigurationsdateien und Dokumentation. Pakete können mit den pkg(8)-Befehlen, wie `pkg install`, manipuliert werden.

Obwohl beide Technologien gleichartig sind, so haben Pakete und Ports jeweils ihre eigenen Stärken. Welche Technologie eingesetzt wird, hängt letzten Endes von den Anforderungen ab, die an eine bestimmte Anwendung gestellt werden.

*Vorteile von Paketen*

- Das komprimierte Paket einer Anwendung ist normalerweise kleiner als das komprimierte Archiv der Quelltexte.
- Pakete müssen nicht mehr kompiliert werden. Dies ist ein Vorteil, wenn große Pakete wie Mozilla, KDE oder GNOME auf langsamen Maschinen installiert werden.
- Wenn Sie Pakete verwenden, brauchen Sie nicht zu verstehen, wie Software unter FreeBSD kompiliert wird.

*Vorteile von Ports*

- Da die Pakete auf möglichst vielen System laufen sollen, werden Optionen beim Übersetzen zurückhaltend gesetzt. Wird eine Anwendung über die Ports übersetzt, können die Optionen nach eigenen Bedürfnissen angepasst werden.
- Die Eigenschaften einiger Anwendungen werden über Optionen zum Zeitpunkt des Übersetzens festgelegt. Apache kann zum Beispiel über eine große Auswahl an eingebauten Optionen konfiguriert werden.

  Für einige Fälle existieren verschiedene Pakete einer Anwendung, die beim Übersetzen unterschiedlich konfiguriert wurden. Für Ghostscript gibt es ein ghostscript-Paket und ein ghostscript-nox11-Paket, die sich durch die Xorg Unterstützung unterscheiden. Das Erstellen von verschiedenen Paketen wird aber schnell unhandlich, wenn eine Anwendung mehr als ein oder zwei Optionen zum Zeitpunkt des Übersetzens besitzt.

- Die Lizenzbestimmungen mancher Software verbietet ein Verbreiten in binärer Form. Diese Software muss als Quelltext, der durch den Benutzer kompiliert werden muss, ausgeliefert werden.
- Einige Leute trauen binären Distributionen nicht, oder sie ziehen es vor den Quelltext zu lesen, um diesen nach möglichen Problemen zu durchsuchen.
- Der Quellcode wird benötigt, um individuelle Anpassungen anzuwenden.

Wenn Sie über aktualisierte Ports informiert sein wollen, lesen Sie die Mailinglisten FreeBSD ports und FreeBSD ports bugs.



Bevor Sie eine Anwendung installieren, informieren Sie sich auf der Seite https://vuxml.FreeBSD.org/ über mögliche Sicherheitsprobleme mit der Anwendung, oder führen Sie `pkg audit -F` aus, um alle installierten Pakete auf bekannte Sicherheitslücken zu überprüfen.

Der Rest dieses Kapitels beschreibt, wie man Software Dritter mit Paketen und Ports unter FreeBSD installiert und verwaltet.

## 6.3. Suchen einer Anwendung

Die Anzahl der nach FreeBSD portierten Anwendungen steigt ständig. Es gibt einige Wege, um nach Anwendungen zu suchen:

- Die FreeBSD-Webseite stellt unter https://www.FreeBSD.org/ports/ eine aktuelle und durchsuchbare Liste aller Anwendungen zur Verfügung. Die Ports können nach dem Namen den Anwendung, oder über die Software-Kategorie durchsucht werden.
- Dan Langille verwaltet FreshPorts.org, das eine umfassende Suchfunktion bietet und Änderungen an den Anwendungen in der Ports-Sammlung verfolgt. Registrierte Benutzer können eine Merkliste erstellen, um automatisch eine E-Mail zu erhalten, sobald ein Port von dieser Liste aktualisiert wurde.
- Wenn Sie bei der Suche nach einer bestimmten Anwendung nicht weiter kommen, versuchen Sie eine Webseite wie SourceForge.net oder GitHub.com. Schauen Sie dann auf der FreeBSD-Webseite nach, ob die Anwendung portiert wurde.
- Das Paket Repository nach einer Anwendung durchsuchen:

```
# pkg search subversion
git-subversion-1.9.2
java-subversion-1.8.8_2
p5-subversion-1.8.8_2
py27-hgsubversion-1.6
py27-subversion-1.8.8_2
ruby-subversion-1.8.8_2
subversion-1.8.8_2
subversion-book-4515
subversion-static-1.8.8_2
subversion16-1.6.23_4
subversion17-1.7.16_2
```

Die Paketnamen enthalten jeweils die Versionsnummer. Wenn ein Port von python abhängt, wird auch die Versionsnummer von python ausgegeben, mit der die Anwendung gebaut wurde. Für einige Ports stehen sogar mehrere Versionen zur Verfügung. Im Fall von Subversion gibt es drei verschiedene Versionen, mit unterschiedlichen Optionen. In diesem Fall wird die Version von Subversion statisch gelinkt. Wenn Sie ein Paket installieren, ist es am besten den Ursprung des Ports anzugeben, also den Pfad in der Ports-Sammlung. Wiederholen Sie `pkg search` mit `-o` um den Ursprung der Pakete anzuzeigen:

```
# pkg search -o subversion
devel/git-subversion
java/java-subversion
devel/p5-subversion
devel/py-hgsubversion
devel/py-subversion
devel/ruby-subversion
devel/subversion16
devel/subversion17
devel/subversion
devel/subversion-book
devel/subversion-static
```

Zudem unterstützt `pkg search` die Suche mit regulären Ausdrücken, nach exakten Treffern, nach der Beschreibung oder nach anderen Feldern in der Repository-Datenbank. Nach der Installation von ports-mgmt/pkg oder ports-mgmt/pkg-devel, finden Sie in pkg-search(8) weitere Details.

- Wenn die Ports-Sammlung bereits installiert ist, gibt es mehrere Methoden, um die lokale Version dieser Port-Sammlung abzufragen. Verwenden Sie `whereis Datei` um herauszufinden, in welcher Kategorie ein Port ist, wobei *Datei* der Name des Programms ist, das installiert werden soll:

  ```
  # whereis lsof
  lsof: /usr/ports/sysutils/lsof
  ```

  Alternativ kann der echo(1)-Befehl verwendet werden:

  ```
  # echo /usr/ports/*/*lsof*
  /usr/ports/sysutils/lsof
  ```

  Beachten Sie aber, dass dieser Befehl auch alle Dateien im Verzeichnis /usr/ports/distfiles findet, auf die der angegebene Suchbegriff passt.

- Ein weiterer Weg nach Software zu suchen besteht darin, die eingebaute Suchfunktion der Ports-Sammlung zu benutzen. Wechseln Sie dazu in das Verzeichnis /usr/ports, und rufen Sie `make search name=Anwendungsname` auf, wobei *Anwendungsname* der Name der Software ist. Um zum Beispiel nach `lsof` zu suchen:

  ```
  # cd /usr/ports
  # make search name=lsof
  Port:   lsof-4.88.d,8
  Path:   /usr/ports/sysutils/lsof
  Info:   Lists information about open files (similar to fstat(1))
  Maint:  ler@lerctr.org
  Index:  sysutils
  ```

```
B-deps:
R-deps:
```



Der integrierte Suchmechanismus verwendet eine Datei mit Index-Informationen. Erscheint eine Meldung, dass der INDEX benötigt wird, führen Sie `make fetchindex` aus, um die aktuelle Index-Datei herunterzuladen. Mit einem vorhandenen INDEX ist `make search` in der Lage, die gewünschte Suche durchzuführen.

Die "Path:"-Zeile zeigt an, wo der Port zu finden ist.

Um weniger Informationen zu erhalten, benutzen Sie die Funktion `quicksearch`:

```
# cd /usr/ports
# make quicksearch name=lsof
Port:   lsof-4.88.d,8
Path:   /usr/ports/sysutils/lsof
Info:   Lists information about open files (similar to fstat(1))
```

Erweiterte Suchen führen Sie mit `make search key=Text` oder `make quicksearch key=Text` aus. Damit werden Portnamen, Kommentare, Beschreibungen und Abhängigkeiten nach *Text* durchsucht. Dies kann sehr nützlich sein, wenn der Name des Programms nicht bekannt ist.

Bei der Verwendung von `search` und `quicksearch` wird Groß- und Kleinschreibung bei der Suche ignoriert. Die Suche nach "LSOF" wird dieselben Ergebnisse wie die Suche nach "lsof" liefern.

## 6.4. Benutzen von pkg zur Verwaltung von Binärpaketen

pkg ist der Nachfolger für die traditionellen Paketverwaltungswerkzeuge von FreeBSD. Es bietet viele Funktionen, die den Umgang mit Binärpaketen schneller und einfacher machen.

Wenn Sie lediglich vorgefertigte Binärpakete von den FreeBSD Spiegeln benutzen möchten, ist pkg für die Verwaltung von Paketen ausreichend.

Falls Sie jedoch die Software aus dem Quellcode bauen oder eigene Repositories verwenden, benötigen Sie ein separates Paketverwaltungswerkzeug.

pkg ist kein Ersatz für diese Werkzeuge. Während diese Werkzeuge Drittanbieter-Software sowohl aus Binärpaketen als auch aus der Ports-Sammlung installieren können, so installiert pkg ausschließlich Binärpakete.

### 6.4.1. Erste Schritte mit pkg

FreeBSD enthält ein Bootstrap-Programm, welches pkg zusammen mit den Manualpages installiert. pkg wurde für FreeBSD Versionen ab 10.*X* entwickelt.



Nicht alle FreeBSD Versionen unterstüzen den folgenden Bootstrap Prozess. Eine aktuelle Liste finden Sie unter https://pkg.FreeBSD.org/. Andernfalls muss pkg aus der Ports-Sammlung oder als Binärpaket installiert werden.

Um das Bootstrap Programm zu starten, geben Sie folgendes ein:

```
# /usr/sbin/pkg
```

Sie müssen eine Internetverbindung haben, damit der Bootstrap Prozess funktioniert.

Um den Port zu installieren, geben Sie stattdessen folgendes ein:

```
# cd /usr/ports/ports-mgmt/pkg
# make
# make install clean
```

Bei der Aktualisierung eines bestehenden Systems, welches ursprünglich die alten pkg_* Werkzeuge verwendet hat, muss die Datenbank in das neue Format konvertiert werden, damit die neuen Werkzeuge wissen, welche Pakete bereits installiert sind. Sobald pkg installiert ist, muss die Paketdatenbank mit dem folgenden Befehl vom traditionellen Format in das neue Format konvertiert werden:

```
# pkg2ng
```



Auf neu installieren Systemen, auf denen noch keine Software von Drittanbietern installiert wurde, kann dieser Schritt entfallen.



Die Konvertierung ist unwiderruflich. Sobald die Paketdatenbank in das Format von pkg umgewandelt wurde, sollten die traditionellen pkg_* Werkzeuge nicht mehr benutzt werden.



Bei der Konvertierung der Paketdatenbank können Fehler ausgegeben werden, wenn die Inhalte auf die neue Version umgewandelt werden. Im Allgemeinen können diese Fehler ignoriert werden. Wenn pkg2ng fertig ist, wird eine Liste von Software ausgegeben, die nicht erfolgreich konvertiert werden konnte. Diese Anwendungen müssen manuell neu installiert werden.

Um sicherzustellen, dass die Ports-Sammlung neue Pakete mit pkg und nicht mit den traditionellen Formaten registriert, muss in FreeBSD 10.*X* und früheren Versionen folgende Zeile in /etc/make.conf hinzugefügt werden:

```
WITH_PKGNG=     yes
```

In der Voreinstellung benutzt pkg die Pakete der FreeBSD-Spiegel (das *Repository*). Wenn Sie ein eigenes Paket-Repository erstellen möchten, lesen Sie Pakete mit Poudriere bauen

Weitere Konfigurationsoptionen für pkg sind in pkg.conf(5) beschrieben.

Informationen zur Bedienung von pkg ist in pkg(8) verfügbar. Alternativ kann `pkg` ohne zusätzliche Argumente aufgerufen werden.

Jedes Argument von pkg ist in seiner spezifischen Manualpage dokumentiert. Um beispielsweise die Manualpage von `pkg install` zu lesen, geben Sie einen der folgenden Befehle ein:

```
# pkg help install
```

```
# man pkg-install
```

Der Rest dieses Abschnitts beschreibt die typischen Verwaltungsaufgaben für Binärpakete, die mit pkg erledigt werden können. Jedes gezeigte Kommando verfügt über Optionen, um das Verhalten anzupassen. Details und weitere Beispiele finden Sie in den Manualpages der einzelnen Kommandos.

### 6.4.2. Die Port-Zweige *Quarterly* und *Latest*

Der vierteljährliche Zweig (Quarterly) bietet eine besser vorhersehbare und stabilere Erfahrung bei der Installation und Aktualisierung von Ports und Paketen. Dies wird im Wesentlichen dadurch erreicht, das nur Aktualisierungen zugelassen werden, die nicht zum Funktionsumfang gehören. Der vierteljährliche Zweig zielt darauf ab, Sicherheitskorrekturen (Aktualisierungen und Rückportierungen von Commits), Fehlerbehebungen und Port-Konformität oder Framework-Änderungen zu erhalten. Der vierteljährliche Zweig wird zu Beginn eines jeden Quartals im Januar, April, Juli und Oktober von HEAD abgetrennt. Die Zweige werden nach dem Jahr (YYYY) und dem Quartal (Q1 - Q4) benannt, in dem sie erstellt wurden. Zum Beispiel wird der Zweig, der im Januar 2016 erstellt wurde, 2016Q1 genannt. Der neueste Zweig (Latest) stellt die aktuellsten Versionen der Pakete zur Verfügung.

Um vom Quarterly auf Latest zu wechseln, führen Sie die folgenden Befehle aus:

```
# cp /etc/pkg/FreeBSD.conf /usr/local/etc/pkg/repos/FreeBSD.conf
```

Bearbeiten Sie die Datei /usr/local/etc/pkg/FreeBSD.conf und ändern Sie in der `url:`-Zeile die Zeichenkette *quarterly* in *latest*.

Das Ergebnis sollte wie folgt aussehen:

```
FreeBSD: {
  url: "pkg+http://pkg.FreeBSD.org/${ABI}/latest",
  mirror_type: "srv",
  signature_type: "fingerprints",
```

```
  fingerprints: "/usr/shared/keys/pkg",
  enabled: yes
}
```

Führen Sie zuletzt diesen Befehl aus, um die neuen Repository-Metadaten zu aktualisieren:

```
# pkg update -f
```

### 6.4.3. Informationen über installierte Pakete anzeigen

Informationen über bereits installierte Pakete können mit `pkg info` angezeigt werden. Dabei wird, wenn keine weiteren Optionen angegeben werden, die Version und die Beschreibung aller Pakete oder eines einzelnen Pakets ausgegeben.

Um zu ermitteln welche Version von pkg installiert ist, geben Sie folgendes ein:

```
# pkg info pkg
pkg-1.1.4_1
```

### 6.4.4. Installation und Deinstallation von Paketen

Ein Binärpaket installieren Sie mit dem folgenden Befehl, wobei *paketname* der Name des zu installierenden Pakets ist:

```
# pkg install paketname
```

Dieser Befehl verwendet Daten aus dem Repository um zu bestimmen, welche Version der Software und welche Abhängigkeiten installiert werden müssen. Um beispielsweise curl zu installieren:

```
# pkg install curl
Updating repository catalogue
/usr/local/tmp/All/curl-7.31.0_1.txz          100% of 1181 kB 1380 kBps 00m01s

/usr/local/tmp/All/ca_root_nss-3.15.1_1.txz   100% of  288 kB 1700 kBps 00m00s

Updating repository catalogue
The following 2 packages will be installed:

        Installing ca_root_nss: 3.15.1_1
        Installing curl: 7.31.0_1

The installation will require 3 MB more space

0 MB to be downloaded

Proceed with installing packages [y/N]: y
```

```
Checking integrity... done
[1/2] Installing ca_root_nss-3.15.1_1... done
[2/2] Installing curl-7.31.0_1... done
Cleaning up cache files...Done
```

Das neue Paket und jedes weitere Paket, das als Abhängigkeit installiert wurde, ist in der Liste der installierten Pakete zu sehen:

```
# pkg info
ca_root_nss-3.15.1_1     The root certificate bundle from the Mozilla Project
curl-7.31.0_1     Non-interactive tool to get files from FTP, GOPHER, HTTP(S) servers
pkg-1.1.4_6       New generation package manager
```

Wird ein Paket nicht mehr benötigt, kann es mit `pkg delete` entfernt werden. Zum Beispiel:

```
# pkg delete curl
The following packages will be deleted:

      curl-7.31.0_1

The deletion will free 3 MB

Proceed with deleting packages [y/N]: y
[1/1] Deleting curl-7.31.0_1... done
```

### 6.4.5. Installierte Pakete aktualisieren

Installierte Pakete können mit diesem Kommando auf die neuesten Versionen aktualisiert werden:

```
# pkg upgrade
```

Dieses Kommando vergleicht und aktualisiert die installierten Versionen der Pakete mit denen im Repository.

### 6.4.6. Installierte Pakete auditieren

Regelmäßig werden Sicherheitslücken in Drittanbieter-Software entdeckt. pkg besitzt einen eingebauten Auditing-Mechanismus. Um die auf dem System installierte Software auf Sicherheitslücken zu prüfen, geben Sie folgenden Befehl ein:

```
# pkg audit -F
```

### 6.4.7. Automatisches Entfernen unbenutzter Pakete

Das Entfernen eines Pakets kann möglicherweise Abhängigkeiten hinterlassen, die nicht mehr benötigt werden. Unnötige Pakete, die als Abhängigkeit von anderen Paketen installiert wurden, können automatisch erfasst und entfernt werden:

```
# pkg autoremove
Packages to be removed:
    ca_root_nss-3.15.1_1

The autoremoval will free 723 kB

Proceed with autoremoval of packages [y/N]: y
Deinstalling ca_root_nss-3.15.1_1... done
```

Pakete, die als Abhängigkeiten installiert werden, bezeichnet man als *automatische* Pakete. Nichtautomatische Pakete, also die Pakete, die explizit nicht als Abhängigkeit von einem anderen Paket installiert wurden, können wie folgt angezeigt werden:

```
# pkg prime-list
nginx
openvpn
sudo
```

`pkg prime-list` ist ein Alias-Befehl, der in /usr/local/etc/pkg.conf definiert ist. Es gibt noch weitere Befehle die Sie verwenden können, um die Paketdatenbank des Systems abzufragen. Beispielsweise kann der Befehl `pkg prime-origins` benutzt werden, um das ursprüngliche Portverzeichnis der oben gezeigten Liste zu erhalten:

```
# pkg prime-origins
www/nginx
security/openvpn
security/sudo
```

Diese Liste kann verwendet werden, um alle auf einem System installierten Pakete mit Hilfe von Werkzeugen wie ports-mgmt/poudriere oder ports-mgmt/synth neu zu erstellen.

Um ein bereits installiertes Paket als automatisches Paket zu kennzeichnen, können Sie folgenden Befehl benutzen:

```
# pkg set -A 1 devel/cmake
```

Sobald ein Paket nicht mehr genutzt wird und es als automatisch gekennzeichnet ist, wird es durch `pkg autoremove` erfasst.

Das kennzeichnen eines installierten Pakets als *nicht* automatisch kann wie folgt gemacht werden:

```
# pkg set -A 0 devel/cmake
```

### 6.4.8. Wiederherstellung der Paketdatenbank

Im Gegensatz zum alten Paketverwaltungssystem beinhaltet pkg einen eigenen Mechanismus zur Sicherung der Paketdatenbank. Diese Funktionalität ist standardmäßig aktiviert.



Um das Skript daran zu hindern, eine Sicherung der Paketdatenbank zu erstellen, muss in periodic.conf(5) `daily_backup_pkgdb_enable="NO"` gesetzt werden.

Um den Inhalt einer früheren Paketdatenbank wiederherzustellen, geben Sie folgendes Kommando ein und ersetzen Sie */path/to/pkg.sql* durch den Speicherort der gesicherten Datenbank:

```
# pkg backup -r /path/to/pkg.sql
```



Wenn Sie eine Sicherung wiederherstellen, die von einem `periodic` Skript erstellt wurde, müssen Sie diese zuerst dekomprimieren.

Um eine manuelle Sicherung der pkg Paketdatenbank zu erstellen, führen Sie den folgenden Befehl aus, und ersetzen Sie */path/to/pkg.sql* durch einen geeigneten Dateinamen:

```
# pkg backup -d /path/to/pkg.sql
```

### 6.4.9. Alte Pakete entfernen

Standardmäßig speichert pkg Pakete in einem Cache-Verzeichnis, welches in pkg.conf(5) in der Variablen `PKG_CACHEDIR` definiert wird. Nur Kopien der neusten installierten Pakete werden beibehalten. Ältere Versionen von pkg haben alle Pakete aufbewahrt. Um diese veralteten Pakete zu entfernen, geben Sie folgendes ein:

```
# pkg clean
```

Um alle Pakte aus dem Cache-Verzeichnis zu löschen, geben Sie ein:

```
# pkg clean -a
```

### 6.4.10. Manipulation der Paket-Metadaten

Bei Software aus der FreeBSD Ports-Sammlung kann es vorkommen, dass die Hauptversionsnummer geändert wird. Dafür hat pkg ein eingebautes Kommando, um die Quelle eines Pakets zu aktualisieren. Dies ist nützlich, wenn zum Beispiel lang/php5 zu lang/php53 umbenannt wurde, damit lang/php5 jetzt die Version `5.4` integrieren kann.

Um die Quelle des Pakets für das obige Beispiel zu ändern, geben Sie folgendes ein:

```
# pkg set -o lang/php5:lang/php53
```

Ein weiteres Beispiel: Um lang/ruby18 auf lang/ruby19 zu aktualisieren, geben Sie folgendes ein:

```
# pkg set -o lang/ruby18:lang/ruby19
```

In diesem letzten Beispiel wird die Quelle der Bibliotheken von libglut von graphics/libglut auf graphics/freeglut geändert:

```
# pkg set -o graphics/libglut:graphics/freeglut
```



Bei einem Wechsel der Paketquelle ist es notwendig, die Pakete neu zu installieren, welche von dem Paket abhängig sind, das seine Paketquelle geändert hat. Um eine Neuinstallation von abhängigen Paketen zu erzwingen, führen Sie folgenden Befehl aus:

```
# pkg install -Rf graphics/freeglut
```

## 6.5. Benutzen der Ports-Sammlung

Die Ports-Sammlung ist eine Reihe von Makefiles, Patches und Beschreibungen. Die Dateien für den Bau und die Installation von einzelnen Anwendungen unter FreeBSD werden als *Port* bezeichnet.

In der Voreinstellung wird die Ports-Sammlung im Verzeichnis /usr/ports gespeichert.

Bevor eine Anwendung aus den Ports erstellt werden kann, muss zuerst die Ports-Sammlung installiert werden. Wenn dies nicht bereits bei der Installation von FreeBSD geschehen ist, benutzen Sie eine der beiden Methoden um sie zu installieren:

**Procedure: Installation mit Portsnap**

FreeBSDs Basissystem enthält mit Portsnap ein schnelles und benutzerfreundliches Werkzeug zur Installation der Ports-Sammlung und die bevorzugte Wahl für die meisten Benutzer, die noch nicht FreeBSD-CURRENT benutzen. Dieses Programm stellt eine Verbindung zu einem FreeBSD-Server her, überprüft den gesicherten Schlüssel und lädt eine aktuelle Kopie der Ports-Sammlung herunter. Der Schlüssel wird benötigt, um die Integrität der heruntergeladenen Dateien zu untersuchen.

1. Laden Sie einen komprimierten Snapshot der Ports-Sammlung in /var/db/portsnap:

```
# portsnap fetch
```

2. Wenn Sie Portsnap das erste Mal verwenden, müssen Sie den Snapshot nach /usr/ports extrahieren:

```
# portsnap extract
```

3. Nach dem ersten Einsatz von Portsnap, kann /usr/ports wie folgt aktualisiert werden:

```
# portsnap fetch
# portsnap update
```

Bei der Verwendung von `fetch` können die `extract` oder `update` Operationen nacheinander ausgeführt werden, etwa so:

```
# portsnap fetch update
```

**Procedure: Installation mit Subversion**

Wird mehr Kontrolle über die Ports-Sammlung benötigt, oder wenn die lokalen Änderungen beibehalten werden sollen, oder Sie FreeBSD-CURRENT benutzen, kann Subversion benutzt werden, um die Ports-Sammlung zu laden. Lesen Sie den Subversion Primer für eine detaillierte Beschreibung von Subversion.

1. Subversion muss installiert sein, bevor die Ports-Sammlung geladen werden kann. Ist eine lokale Kopie der Ports-Sammlung bereits vorhanden, installieren Sie Subversion wie folgt:

```
# cd /usr/ports/devel/subversion
# make install clean
```

Wenn keine lokale Kopie der Ports-Sammlung vorhanden ist, oder pkg zur Verwaltung von Paketen benutzt wird, kann Subversion als Paket installiert werden:

```
# pkg install subversion
```

2. Laden Sie eine Kopie der Ports-Sammlung:

```
# svn checkout https://svn.FreeBSD.org/ports/head /usr/ports
```

3. Nach dem erstmaligen checkout mit Subversion kann /usr/ports wie folgt aktualisiert werden:

```
# svn update /usr/ports
```

Die Ports-Sammlung enthält eine Reihe von Verzeichnissen, die jeweils eine Softwarekategorie repräsentieren. Jede Kategorie hat für jede einzelne Anwendung ein weiteres Unterverzeichnis. Jedes Unterverzeichnis enthält Dateien, die FreeBSD sagen, wie ein Programm kompiliert und installiert werden muss. Diese Dateien werden auch Port-"Gerüst" genannt. Jedes Port-"Gerüst" beinhaltet die folgenden Dateien und Verzeichnisse:

- Makefile: enthält Anweisungen, die spezifizieren, wie die Anwendung kompiliert wird und wohin die Komponenten installiert werden sollten.
- distinfo: enthält die Namen und die Prüfsummen der Dateien, die heruntergeladen werden müssen, um den Port zu bauen.
- files: dieses Verzeichnis enthält Patches, welche das Übersetzen und Installieren der Anwendung unter FreeBSD ermöglichen. Zudem können noch weitere Dateien, die für die Übersetzung des Ports verwendet werden, enthalten sein.
- pkg-descr: enthält eine ausführlichere Beschreibung der Anwendung.
- pkg-plist: eine Liste aller Dateien, die durch diesen Port installiert werden. Außerdem sind hier Informationen enthalten, die zum Entfernen des Ports benötigt werden.

Einige Ports beinhalten noch pkg-message oder weitere Dateien, die vom Port-System benutzt werden, um spezielle Situationen zu handhaben. Wenn Sie mehr über diese Dateien oder das Port-System erfahren wollen, lesen Sie das FreeBSD Porter's Handbook.

Ein Port enthält nicht den eigentlichen Quellcode, der auch als "Distfile" bekannt ist. Der heruntergeladene Quellcode wird automatisch nach /usr/ports/distfiles extrahiert.

### 6.5.1. Ports installieren

Dieser Abschnitt beschreibt die grundlegende Benutzung der Ports-Sammlung, um Software zu installieren oder zu deinstallieren. Eine ausführliche Beschreibung der einzelnen `make`-Targets finden Sie in ports(7).



Stellen Sie sicher, dass die Ports-Sammlung aktuell ist, bevor Sie einen Port kompilieren. Informieren Sie sich vorher zusätzlich unter https://vuxml.FreeBSD.org/ über mögliche Sicherheitsprobleme des zu installierenden Ports. Alternativ können Sie `pkg audit -F` ausführen, bevor Sie einen neuen Port installieren. Die täglich laufende Sicherheitsprüfung des Systems aktualisiert ebenfalls die Datenbank und prüft installierte Anwendungen auf vorhandene Sicherheitsprobleme. Weitere Informationen finden Sie in pkg-audit(8) und periodic(8).

Die Benutzung der Ports-Sammlung setzt eine funktionierende Internetverbindung und Superuser-Rechte voraus.

Um einen Port zu installieren, wechseln Sie in das Verzeichnis des Ports, den Sie installieren

möchten. Geben Sie dann `make install` am Prompt ein:

```
# cd /usr/ports/sysutils/lsof
# make install
>> lsof_4.88D.freebsd.tar.gz doesn't seem to exist in /usr/ports/distfiles/.
>> Attempting to fetch from ftp://lsof.itap.purdue.edu/pub/tools/unix/lsof/.
===>  Extracting for lsof-4.88
...
[Ausgabe des Auspackens weggelassen]
...
>> Checksum OK for lsof_4.88D.freebsd.tar.gz.
===>  Patching for lsof-4.88.d,8
===>  Applying FreeBSD patches for lsof-4.88.d,8
===>  Configuring for lsof-4.88.d,8
...
[configure-Ausgabe weggelassen]
...
===>  Building for lsof-4.88.d,8
...
[Ausgabe der Übersetzung weggelassen]
...
===>  Installing for lsof-4.88.d,8
...
[Ausgabe der Installation weggelassen]
...
===>   Generating temporary packing list
===>   Compressing manual pages for lsof-4.57
===>   Registering installation for lsof-4.57
===>  SECURITY NOTE:
      This port has installed the following binaries which execute with
      increased privileges.
/usr/local/bin/lsof
#
```

Da `lsof` eine Anwendung ist, die mit erhöhten Rechten läuft, wird nach der Installation eine Sicherheitswarnung angezeigt. Sobald die Installation abgeschlossen ist, erscheint wieder der Prompt.

Um die Suche nach Kommandos zu beschleunigen, speichern einige Shells eine Liste der verfügbaren Kommandos in den durch die Umgebungsvariable `PATH` gegebenen Verzeichnissen. Benutzer der `tcsh` müssen eventuell `rehash` eintippen, um die neu installierten Kommandos benutzen zu können, ohne den vollständigen Pfad anzugeben. Benutzer der Shell `sh` müssen stattdessen `hash -r` eintippen. Weitere Informationen finden Sie in der Dokumentation der jeweiligen Shell.

Bei der Installation wird ein Arbeitsverzeichnis erstellt, das alle temporären Dateien enthält, die während des Bauvorgangs benötigt werden. Wenn dieses Verzeichnis nach der Installation entfernt wird, spart dies Plattenplatz und minimiert mögliche Probleme bei der Aktualisierung des Ports auf eine neuere Version:

```
# make clean
===>  Cleaning for lsof-4.88.d,8
#
```



Sie können zwei Schritte sparen, wenn Sie bei der Kompilierung des Ports gleich `make install clean` eingeben.

**Port Installation anpassen**

Einige Ports bieten Optionen, mit denen zusätzliche Funktionen oder Sicherheitsoptionen eingestellt werden können. Beispiele dafür sind www/firefox, security/gpgme und mail/sylpheed-claws. Wenn ein Port von anderen Ports abhängig ist und diese über zusätzliche Abhängigkeiten und Optionen verfügen, wird mehrmals ein Menü ausgegeben, wo der Benutzer verschiedene Optionen wählen kann. Um dies zu vermeiden und die Konfiguration in einem Stück zu erledigen, wechseln Sie in das Verzeichnis des Ports und geben Sie `make config-recursive` ein. Führen Sie danach `make install [clean]` aus, um den Port zu kompilieren und zu installieren.



Bei der Verwendung von `config-recursive` wird eine Liste von Ports, die konfiguriert werden, vom Target `all-depends-list` erstellt. Es wird empfohlen, `make config-recursive` so lange auszuführen, bis alle Optionen der abhängigen Ports definiert sind und keine Optionen und Menüs mehr erscheinen. Damit soll sichergestellt werden, dass alle Optionen konfiguriert wurden.

Es gibt diverse Möglichkeiten, dieses Menü nach dem Bau eines Ports erneut aufzurufen, um Optionen zu entfernen, hinzuzufügen oder anzupassen. Sie können beispielsweise mit `cd` in das Verzeichnis des Ports wechseln und dort `make config` eingeben. Eine andere Möglichkeit ist `make showconfig`. Eine weitere Alternative bietet `make rmconfig`, das alle ursprünglich gewählten Optionen zurücksetzt und es Ihnen dadurch ermöglicht, die Konfiguration erneut zu beginnen. Die eben erwähnten Optionen werden ausführlich in ports(7) beschrieben.

Die Ports-Sammlung benutzt zum Herunterladen von Dateien fetch(3), das diverse Umgebungsvariablen unterstützt. Die Variablen `FTP_PASSIVE_MODE`, `FTP_PROXY` und `FTP_PASSWORD` müssen unter Umständen gesetzt werden, wenn das FreeBSD-System hinter einer Firewall oder einem FTP/HTTP-Proxy arbeitet. Eine vollständige Liste der unterstützten Variablen finden Sie in fetch(1).

Benutzer ohne eine ständige Internet-Verbindung können `make fetch` im Verzeichnis /usr/ports ausführen, um die benötigten Dateien herunterzuladen. Es ist auch möglich, `make fetch` nur in einem Teil des Baums, wie /usr/ports/net, aufzurufen. Die Dateien von allen abhängigen Ports werden mit diesem Kommando allerdings nicht heruntergeladen. Wenn Sie diese Dateien ebenfalls herunterladen wollen, benutzen Sie stattdessen `make fetch-recursive`.

In einigen seltenen Fällen ist es erforderlich, die benötigten Dateien von einem anderen Ort als den im Port definierten `MASTER_SITES` herunterzuladen. Sie können `MASTER_SITES` mit dem folgenden Kommando überschreiben:

```
# cd /usr/ports/directory
```

```
# make MASTER_SITE_OVERRIDE= \
ftp://ftp.FreeBSD.org/pub/FreeBSD/ports/distfiles/ fetch
```

Die Variablen `WRKDIRPREFIX` und `PREFIX` überschreiben das voreingestellte Bau- und Zielverzeichnis. Zum Beispiel:

```
# make WRKDIRPREFIX=/usr/home/example/ports install
```

Dieses Kommando baut den Port unter /usr/home/example/ports und installiert ihn unter /usr/local.

Die Variable `PREFIX` legt das Installations-Verzeichnis fest:

```
# make PREFIX=/usr/home/example/local install
```

In diesem Beispiel wird der Port unter /usr/ports gebaut und nach /usr/home/example/local installiert.

Sie können beide Variablen auch zusammen benutzen:

```
# make WRKDIRPREFIX=../ports PREFIX=../local install
```

Alternativ können diese Variablen auch als Umgebungsvariablen gesetzt werden. In der Manualpage Ihrer Shell finden Sie Anweisungen, wie Umgebungsvariablen gesetzt werden.

### 6.5.2. Entfernen installierter Ports

Installierte Ports können mit `pkg delete` wieder deinstalliert werden. Beispiele für dieses Kommando finden Sie in pkg-delete(8).

Alternativ kann `make deinstall` im Verzeichnis des Ports aufgerufen werden:

```
# cd /usr/ports/sysutils/lsof
# make deinstall
===>  Deinstalling for sysutils/lsof
===>   Deinstalling
Deinstallation has been requested for the following 1 packages:

        lsof-4.88.d,8

Thee deinstallation will free 229 kB
[1/1] Deleting lsof-4.88.d,8... done
```

Es wird empfohlen die Nachrichten zu lesen, die ausgegeben werden, wenn ein Port deinstalliert wird. Wenn der Port noch Anwendungen hat, die von ihm abhängig sind, werdenn diese am Bildschirm angezeigt, aber die Deinstallation wird forgesetzt. In solchen Fällen ist es besser, die

Anwendung neu zu installieren, um fehlende Abhängigkeiten zu vermeiden.

### 6.5.3. Ports aktualisieren

Im Laufe der Zeit stehen neuere Versionen der Software in der Ports-Sammlung zur Verfügung. In diesem Abschnitt wird beschrieben, wie Sie bestimmen, welche Software aktualisiert werden kann und wie das Upgrade durchzuführen ist.

Um festzustellen, ob neuere Versionen der installierten Ports verfügbar sind, stellen Sie sicher, dass die neueste Version der Ports-Sammlung installiert ist. Dies wird in "Installation mit Portsnap" und "Installation mit Subversion" beschrieben. Führen Sie unter FreeBSD 10 und neueren Versionen, bzw. auf Systemen die bereits mit pkg arbeiten, den folgenden Befehl aus, um eine Liste der installierten Ports zu erhalten für die eine aktuelle Version existiert:

```
# pkg version -l "<"
```

Mit FreeBSD 9.*X* und älteren Versionen kann stattdessen dieser Befehl verwendet werden:

```
# pkg_version -l "<"
```



Lesen Sie zuerst /usr/ports/UPDATING, bevor Sie einen Port aktualisieren. In dieser Datei werden Probleme und zusätzlich durchzuführende Schritte bei der Aktualisierung einzelner Ports beschrieben. Dazu gehören solche Dinge wie geänderte Dateiformate, verschobene Konfigurationsdateien, aber auch Inkompatibilitäten zu einer Vorgängerversion. Notieren Sie sich alle Anweisungen der Ports, die aktualisiert werden müssen. Folgen Sie den Anweisungen, wenn Sie das Upgrade durchführen.

#### Werkzeuge für die Aktualisierung und Verwaltung von Ports

Die Ports-Sammlung enthält mehrere Werkzeuge, um die eigentliche Aktualisierung durchzuführen. Jedes hat seine Stärken und Schwächen.

Historisch gesehen verwenden die meisten Installationen entweder Portmaster oder Portupgrade. Synth ist eine neuere Alternative.



Es bleibt dem Systemadministrator überlassen, welches dieser Werkzeuge für ein bestimmtes System am besten geeignet ist. Es wird empfohlen, die Daten zu sichern, bevor Sie eines dieser Werkzeuge verwenden.

#### Ports mit Portmaster aktualisieren

ports-mgmt/portmaster ist ein sehr kleines Werkzeug zum Aktualisieren von Ports. Es wurde entwickelt, um mit den Werkzeugen aus dem FreeBSD Basissystem zu arbeiten, ohne dabei von anderen Ports oder Datenbanken abhängig zu sein. Sie können das Programm aus der Ports-Sammlung installieren:

```
# cd /usr/ports/ports-mgmt/portmaster
# make install clean
```

Portmaster teilt Ports in vier Kategorien ein:

- Root Port: hat keine Abhängigkeiten und andere Ports sind nicht von diesem Port abhängig.
- Trunk Port: hat keine Abhängigkeiten, aber andere Ports sind von diesem Port abhängig.
- Branch Port: hat Abhängigkeiten und andere Ports sind von diesem Port abhängig.
- Leaf Port: hat Abhängigkeiten, aber andere Ports sind nicht von diesem Port abhängig.

Um eine Liste der installierten Ports anzuzeigen und nach neueren Versionen zu suchen, verwenden Sie:

```
# portmaster -L
===>>> Root ports (No dependencies, not depended on)
===>>> ispell-3.2.06_18
===>>> screen-4.0.3
        ===>>> New version available: screen-4.0.3_1
===>>> tcpflow-0.21_1
===>>> 7 root ports
...
===>>> Branch ports (Have dependencies, are depended on)
===>>> apache22-2.2.3
        ===>>> New version available: apache22-2.2.8
...
===>>> Leaf ports (Have dependencies, not depended on)
===>>> automake-1.9.6_2
===>>> bash-3.1.17
        ===>>> New version available: bash-3.2.33
...
===>>> 32 leaf ports

===>>> 137 total installed ports
        ===>>> 83 have new versions available
```

Um alle installierten Ports zu aktualisieren, verwenden Sie folgenden Befehl:

```
# portmaster -a
```



In der Voreinstellung erzeugt Portmaster eine Sicherheitskopie, bevor ein installierter Port gelöscht wird. Ist die Installation der neuen Version erfolgreich, wird dieses Backup wieder gelöscht. Wollen Sie das Backup lieber manuell löschen, verwenden Sie die Option `-b` beim Aufruf von Portmaster. Durch die Verwendung von `-i` wird Portmaster im interaktiven Modus gestartet und fragt bei jedem zu aktualisierenden Port nach, wie weiter vorgegangen werden soll. Viele

> weitere Optionen stehen zur Verfügung. Lesen Sie die Manualpage von portmaster(8) für weitere Einzelheiten in Bezug auf ihre Nutzung.

Treten während der Aktualisierung Fehler auf, verwenden Sie die Option `-f`, um alle Ports zu aktualisieren beziehungsweise neu zu bauen:

```
# portmaster -af
```

Portmaster ist auch in der Lage, neue Ports zu installieren, wobei zuvor alle abhängigen Ports aktualisiert werden. Um diese Funktion zu nutzen, geben Sie den Pfad des Ports in der Ports-Sammlung an:

```
# portmaster shells/bash
```

Weitere Informationen über ports-mgmt/portmaster finden Sie in der Beschreibung pkg-descr.

#### Ports mit Portupgrade aktualisieren

ports-mgmt/portupgrade ist ein weiteres Werkzeug zur Aktualisierung von Ports. Es installiert eine Reihe von Anwendungen, die für die Verwaltung von Ports verwendet werden können. Das Programm ist jedoch von Ruby abhängig. Um den Port zu installieren, geben Sie ein:

```
# cd /usr/ports/ports-mgmt/portupgrade
# make install clean
```

Durchsuchen Sie vor jedem Update die Liste der installierten Ports mit `pkgdb -F` und beheben Sie alle gefundenen Probleme.

Benutzen Sie `portupgrade -a`, um automatisch alle veralteten Ports auf dem System zu aktualisieren. Verwenden Sie zusätzlich den Schalter `-i`, wenn Sie individuell entscheiden wollen, ob ein Port aktualisiert werden soll:

```
# portupgrade -ai
```

Um nur eine spezifische Anwendung zu aktualisieren, verwenden Sie `portupgrade Paketname`. Es ist wichtig den Schalter `-R` zu benutzen, um zuvor alle Ports zu aktualisieren, die von dem gegebenen Anwendung abhängen.

```
# portupgrade -R firefox
```

Um Pakete anstelle von Ports zu installieren, verwenden Sie den Schalter `-P`. Mit dieser Option durchsucht Portupgrade die in der Umgebungsvariablen `PKG_PATH` aufgeführten Verzeichnisse nach Paketen. Sind lokal keine Pakete vorhanden, versucht Portupgrade die Pakete über das Netz herunterzuladen. Gibt es die Pakete weder lokal noch auf entfernten Rechnern, werden die Ports

verwendet. Um die Nutzung von Ports gänzlich zu verhindern, benutzen Sie die Option `-PP`. Portupgrade würde dann abbrechen, falls keine Pakete zur Verfügung stehen.

```
# portupgrade -PP gnome3
```

Wenn Sie nur die Quelldateien des Ports, oder die Pakete mit `-P` herunterladen möchten, ohne die Anwendung zu bauen oder zu installieren, geben Sie den Schalter `-F` an. Weitere Informationen zu den verfügbaren Schaltern finden Sie in der Manualpage von portupgrade(1).

Weitere Informationen über ports-mgmt/portupgrade finden Sie in der Beschreibung pkg-descr.

### 6.5.4. Platzbedarf von Ports

Die Nutzung der Ports-Sammlung wird im Laufe der Zeit viel Plattenplatz verschlingen. Nach dem Bau und der Installation eines Ports, wird `make clean` die temporären Arbeitsverzeichnisse work aufräumen. Portmaster wird dieses Verzeichnis nach der Installation eines Ports automatisch entfernen (es sei denn, die Option `-K` wird verwendet). Wenn Portupgrade installiert ist, wird der folgende Befehl alle Arbeitsverzeichnisse der lokalen Ports-Sammlung entfernen:

```
# portsclean -C
```

Zusätzlich werden sich im Laufe der Zeit zahlreiche veraltete Distfiles in /usr/ports/distfiles ansammeln. Mit Portupgrade können alle Distfiles gelöscht werden, die vom keinem Port mehr benötigt werden:

```
# portsclean -D
```

Portupgrade kann alle Distfiles löschen, die von keinem derzeit installierten Port benötigt werden:

```
# portsclean -DD
```

Wenn Portmaster installiert ist, benutzen Sie diesen Befehl:

```
# portmaster --clean-distfiles
```

In der Voreinstellung arbeitet dieses Programm interaktiv und fragt den Benutzer um Bestätigung, bevor ein Distfile gelöscht wird.

Zusätzlich zu diesen Kommandos gibt es noch port-mgmt/pkg_cutleaves. Dieses Werkzeug automatisiert die Deinstallation von installierten Ports, die nicht weiter benötigt werden.

## 6.6. Pakete mit Poudriere bauen

Poudriere ist ein unter der BSD-Lizenz stehendes Werkzeug zum Erstellen und Testen von FreeBSD-Paketen. Dieses Programm nutzt FreeBSD Jails, um die Pakete in einer isolierten Umgebung zu bauen. Diese Jails können verwendet werden, um Pakete für andere Versionen von FreeBSD zu bauen, oder um auf einem amd64-System Pakete für i386 zu bauen. Sobald die Pakete gebaut sind, haben sie das gleiche Format wie auf den offiziellen Spiegeln. Die Pakete können dann mit pkg(8) oder anderen Paketverwaltungswerkzeugen benutzt werden.

Poudriere wird über das Paket oder den Port ports-mgmt/poudriere installiert. Die Installation beinhaltet eine Beispielkonfiguration in /usr/local/etc/poudriere.conf.sample. Kopieren Sie diese Datei nach /usr/local/etc/poudriere.conf. Bearbeiten Sie dann die kopierte Datei, um die Konfiguration anzupassen.

Obwohl ZFS für poudriere nicht zwingend erforderlich ist, so hat die Nutzung doch einige Vorteile. Wird ZFS eingesetzt, muss in /usr/local/etc/poudriere.conf die Variable `ZPOOL` definiert, und die Variable `FREEBSD_HOST` auf einen nahe gelegenen Spiegel gesetzt werden. Die Definition von `CCACHE_DIR` erlaubt die Verwendung von devel/ccache, um die Bauzeit für häufig kompilierten Code verkürzen. Es kann vorteilhaft sein, die poudriere-Datasets in einem separaten Verzeichnis auf /poudriere einzuhängen. Die Werte der anderen Konfigurationsvariablen sind in der Regel angemessen und brauchen nicht geändert werden.

Die Anzahl der Kerne im Prozessor wird verwendet um zu bestimmen, wie viele Bauprozesse parallel ausgeführt werden. Stellen Sie ausreichend virtuellen Speicher bereit, entweder in Form von RAM oder als Swap-Speicher. Ist der virtuelle Speicher aufgebraucht, bricht der Bauprozess ab und die Jails stürzen ab, was zu seltsamen Fehlermeldungen führt.

### 6.6.1. Jails und Ports-Sammlung initialisieren

Nach der Konfiguration muss poudriere initialisiert werden, damit es eine Jail mit der benötigten Ports-Sammlung startet. Geben Sie mit `-j` den Namen der Jail und mit `-v` die gewünschte FreeBSD-Version an. Auf FreeBSD/amd64-Systemen kann die Architektur mit dem Schalter `-a` und `i386` oder `amd64` gesetzt werden. Der voreingestellte Wert für die Architektur können Sie sich mit `uname` anzeigen lassen.

```
# poudriere jail -c -j 11amd64 -v 11.4-RELEASE
[00:00:00] Creating 11amd64 fs at /poudriere/jails/11amd64... done
[00:00:00] Using pre-distributed MANIFEST for FreeBSD 11.4-RELEASE amd64
[00:00:00] Fetching base for FreeBSD 11.4-RELEASE amd64
/poudriere/jails/11amd64/fromftp/base.txz              125 MB 4110 kBps    31s
[00:00:33] Extracting base... done
[00:00:54] Fetching src for FreeBSD 11.4-RELEASE amd64
/poudriere/jails/11amd64/fromftp/src.txz               154 MB 4178 kBps    38s
[00:01:33] Extracting src... done
[00:02:31] Fetching lib32 for FreeBSD 11.4-RELEASE amd64
/poudriere/jails/11amd64/fromftp/lib32.txz              24 MB 3969 kBps    06s
[00:02:38] Extracting lib32... done
[00:02:42] Cleaning up... done
[00:02:42] Recording filesystem state for clean... done
```

```
[00:02:42] Upgrading using ftp
/etc/resolv.conf -> /poudriere/jails/11amd64/etc/resolv.conf
Looking up update.FreeBSD.org mirrors... 3 mirrors found.
Fetching public key from update4.freebsd.org... done.
Fetching metadata signature for 11.4-RELEASE from update4.freebsd.org... done.
Fetching metadata index... done.
Fetching 2 metadata files... done.
Inspecting system... done.
Preparing to download files... done.
Fetching 124
patches.....10....20....30....40....50....60....70....80....90....100....110....120..
done.
Applying patches... done.
Fetching 6 files... done.
The following files will be added as part of updating to
11.4-RELEASE-p1:
/usr/src/contrib/unbound/.github
/usr/src/contrib/unbound/.github/FUNDING.yml
/usr/src/contrib/unbound/contrib/drop2rpz
/usr/src/contrib/unbound/contrib/unbound_portable.service.in
/usr/src/contrib/unbound/services/rpz.c
/usr/src/contrib/unbound/services/rpz.h
/usr/src/lib/libc/tests/gen/spawnp_enoexec.sh
The following files will be updated as part of updating to
11.4-RELEASE-p1:
[…]
Installing updates...Scanning //usr/shared/certs/blacklisted for certificates...
Scanning //usr/shared/certs/trusted for certificates...
 done.
11.4-RELEASE-p1
[00:04:06] Recording filesystem state for clean... done
[00:04:07] Jail 11amd64 11.4-RELEASE-p1 amd64 is ready to be used
```

```
# poudriere ports -c -p local -m svn+https
[00:00:00] Creating local fs at /poudriere/ports/local... done
[00:00:00] Checking out the ports tree... done
```

poudriere kann auf einem einzelnen Rechner Ports mit mehreren Konfigurationen bauen, in mehreren Jails und aus unterschiedlichen Ports-Sammlungen. Spezifische Konfigurationen für diese Kombinationen werden *Sets* genannt. Lesen Sie den Abschnitt CUSTOMIZATION in poudriere(8) für weitere Einzelheiten nach der Installation von port-mgmt/poudriere oder ports-mgmt/poudriere-devel.

Die hier gezeigte Konfiguration verwendet eine einzelne Jail-, Port- und Set-spezifische make.conf in /usr/local/etc/poudriere.d. Der verwendete Dateiname in diesem Beispiel wird aus einer Kombination von Jailnamen, Portnamen und Setnamen zusammen gesetzt: 11amd64-local-workstation-make.conf. Die make.conf des Systems und diese neue Datei werden verwendet, um die make.conf für die Jail zu erzeugen.

Die zu bauenden Pakete werden in 11amd64-local-workstation-pkglist eingetragen:

```
editors/emacs
devel/git
ports-mgmt/pkg
...
```

Die Optionen und Abhängigkeiten für die Ports werden wie folgt konfiguriert:

```
# poudriere options -j 11amd64 -p local -z workstation -f 11amd64-local-workstation-
pkglist
```

Schließlich werden die Pakete gebaut und ein Paket-Repository erstellt:

```
# poudriere bulk -j 11amd64 -p local -z workstation -f 11amd64-local-workstation-
pkglist
```

Während der Ausführung zeigt `Ctrl` + `t` den aktuellen Status des Baus an. Poudriere speichert zudem Dateien in /poudriere/logs/bulk/jailname. Diese Dateien kann ein Webserver nutzen, um Informationen über den Bau anzuzeigen.

Nach der Fertigstellung stehen die Pakete im poudriere Repository für die Installation zur Verfügung.

Weitere Informationen zu poudriere finden Sie in poudriere(8) und unter https://github.com/freebsd/poudriere/wiki.

### 6.6.2. Konfiguration des pkg-Clients für das Poudriere Repository

Obwohl es möglich ist ein eigenes Repository zusammen mit dem offiziellen Repository zu nutzen, ist es manchmal sinnvoll das offizielle Repository zu deaktivieren. Dazu wird eine Konfigurationsdatei erstellt, welche die offizielle Konfigurationsdatei überschreibt. Erzeugen Sie dazu /usr/local/etc/pkg/repos/FreeBSD.conf mit dem folgenden Inhalt:

```
FreeBSD: {
        enabled: no
}
```

Am einfachsten ist es, das poudriere Repository über HTTP zur Verfügung zu stellen. Setzen Sie einen Webserver auf, der die Dateien des Paketverzeichnisses ausliefert, zum Beispiel /usr/local/poudriere/data/packages/11amd64. 11amd64 bezeichnet dabei den Namen des Baus.

Wenn die URL des Paket Repositories `http://pkg.example.com/11amd64` ist, dann sollte die Konfiguration des Repositories in /usr/local/etc/pkg/repos/custom.conf wie folgt aussehen:

```
custom: {
    url: "http://pkg.example.com/11amd64",
    enabled: yes,
}
```

## 6.7. Nach der Installation

Unabhängig davon, ob die Software aus einem binären Paket oder aus einem Port installiert wird, benötigen die meisten Anwendungen von Drittanbietern ein gewisses Maß an Konfiguration, nachdem sie installiert wurden. Die folgenden Kommandos und Speicherorte helfen Ihnen dabei festzustellen, was mit der Anwendung zusammen installiert wurde.

- Die meisten Anwendungen installieren mindestens eine Konfigurationsdatei nach /usr/local/etc. Falls die Anwendung viele Konfigurationsdateien enthält, wird ein Unterverzeichnis erstellt um die Dateien zu speichern. Oft werden die Konfigurationsdateien mit einem Suffix wie beispielsweise .sample installiert. Die Konfigurationsdateien sollten überprüft und ggf. bearbeitet werden, um die Anforderungen des Systems zu erfüllen. Um eine Konfigurationsdatei zu bearbeiten, kopieren Sie diese zunächst ohne die Erweiterung .sample.
- Wenn die Anwendung Dokumentation zur Verfügung stellt, wird diese nach /usr/local/shared/doc installiert. Viele Anwendungen installieren auch Manualpages. Diese Dokumentation sollten Sie lesen, bevor Sie fortfahren.
- Einige Anwendungen laufen als Dienst und müssen vor dem ersten Start in /etc/rc.conf eingetragen werden. Diese Anwendungen installieren meist ein Skript in /usr/local/etc/rc.d. Weitere Informationen finden Sie im Start von Diensten.



In der Voreinstellung führen Anwendungen weder ihr Startskript bei der Installation aus, noch führen sie ihr Stopskript während der Deinstallation aus. Diese Entscheidung bleibt dem einzelnen Systemadministrator überlassen.

- Benutzer der csh(1) sollten `rehash` ausführen, um die neu installierten Programme nutzen zu können.
- Benutzen Sie `pkg info`, um die Dateien, Manualpages und Binaries zu ermitteln, die mit der Anwendung installiert wurden.

## 6.8. Kaputte Ports

Wenn sich ein Port nicht bauen oder installieren lässt, versuchen Sie folgendes:

1. Stellen Sie fest, ob die Datenbank mit den Problemberichten bereits einen Lösungsvorschlag enthält. Ist dies der Fall, kann die vorgeschlagene Lösung getestet werden.
2. Bitten Sie den Betreuer des Ports um Hilfe. Geben Sie dazu `make maintainer` ein oder lesen Sie das Makefile im Verzeichnis des Ports, um an die E-Mail-Adresse zu kommen. Vergessen Sie nicht die Zeile mit `$FreeBSD:` aus dem Makefile und die Ausgabe bis zur Fehlermeldung mitzuschicken.



Einige Ports werden nicht von einer Einzelperson, sondern von einer Mailingliste betreut. Viele (aber nicht alle) dieser Adressen haben die Form freebsd-NameDerListe@FreeBSD.org. Denken Sie daran, wenn Sie Ihre Fragen formulieren.

Dies gilt insbesondere für Ports, die von ports@FreeBSD.org betreut werden. Derartige Ports haben überhaupt keinen Betreuer. Korrekturen und Unterstützung kommen daher nur von Personen, die diese Mailingliste abonniert haben. Gerade in diesem Bereich werden jederzeit zusätzliche freiwillige Helfer benötigt!

Erhalten Sie auf Ihre Anfrage keine Antwort, benutzen Sie Bugzilla, um einen Problembericht zu erstellen. Bevor Sie einen solchen Bericht erstellen, lesen Sie den Artikel Writing FreeBSD Problem Reports.

3. Reparieren Sie ihn! Das FreeBSD Porter's Handbook enthält eine detaillierte Beschreibung des Portsystems. Damit sind Sie in der Lage, einen zeitweilig kaputten Port zu reparieren oder einen eigenen Port zu erstellen.
4. Installieren Sie das Paket anstelle des Ports. Anweisungen hierzu finden Sie in Benutzen von pkg zur Verwaltung von Binärpaketen.

# Kapitel 7. Das X-Window-System

## 7.1. Übersicht

Bei einer Installation von FreeBSD mit bsdinstall wird nicht automatisch eine grafische Benutzeroberfläche installiert. Dieses Kapitel beschreibt die Installation und Konfiguration von Xorg, das eine grafische Umgebung über das quelloffene X-Window-System zur Verfügung stellt. Weiterhin wird beschrieben, wie Sie eine Desktop-Umgebung oder einen Window Manager finden und installieren können.



Benutzer die eine Installationsmethode bevorzugen, welche automatisch Xorg konfiguriert, sollten sich FuryBSD, GhostBSD oder MidnightBSD ansehen.

Weitere Informationen über Video-Hardware, die von Xorg unterstützt wird, finden Sie auf der x.org Webseite.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- Die Komponenten des X-Window-Systems und ihr Zusammenspiel kennen.
- Wissen, wie Xorg installiert und konfiguriert wird.
- Wissen, wie verschiedene Window-Manager und Desktop-Umgebungen installiert und konfiguriert werden.
- Wissen, wie TrueType®-Schriftarten mit Xorg benutzt werden.
- Wissen, wie Sie die grafische Anmeldung (XDM) einrichten.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Wissen, wie Sie Software Dritter, wie in Installieren von Anwendungen: Pakete und Ports beschrieben, installieren.

## 7.2. Terminologie

Obwohl es nicht nötig ist, alle Details der verschiedenen Komponenten des X Window Systems und deren Zusammenspiel zu kennen, kann es trotzdem nützlich sein die Grundlagen dieser Komponenten zu verstehen:

**X-Server**

X wurde von Anfang an netzwerktransparent entworfen und verwendet ein "Client-Server-Modell". In diesem Modell läuft der "X-Server" auf dem Rechner, an dem die Tastatur, der Bildschirm und die Maus angeschlossen ist. Der Server ist für Dinge wie die Verwaltung des Bildschirms und die Verarbeitung von Tastatur- und Maus-Eingaben sowie anderer Ein- und Ausgabegeräte, wie beispielsweise ein Tablet oder ein Videoprojektor, verantwortlich. Dieses Modell verwirrt viele Leute, die erwarten, dass der "X-Server" der leistungsstarke Rechner im Maschinenraum und der "X-Client" ihr Arbeitsplatzrechner ist.

**X-Client**

Jede X-Anwendung, wie beispielsweise XTerm oder Firefox ist ein "X-Client". Der Client sendet dem Server Nachrichten wie "Zeichne an diesen Koordinaten ein Fenster" und der Server sendet dem Client Nachrichten der Art "Der Benutzer hat gerade den Ok-Knopf gedrückt".

In kleinen Umgebungen laufen der X-Server und die X-Clients auf demselben Rechner. Es ist auch möglich, den X-Server auf einem weniger leistungsfähigen Rechner laufen zu lassen und die X-Anwendungen auf einem leistungsfähigeren Rechner zu betreiben. In diesem Fall kommunizieren der X-Server und die X-Clients über das Netzwerk.

**Window-Manager**

X schreibt nicht vor, wie Fenster auf dem Bildschirm auszusehen haben, wie sie mit der Maus zu verschieben sind, welche Tastenkombinationen benutzt werden sollen um zwischen den Fenstern zu wechseln, wie die Fensterrahmen aussehen, oder ob diese Schaltflächen zum schließen haben. Stattdessen gibt X die Verantwortung für all diese Sachen an eine separate *Window-Manager* Anwendung ab. Es stehen zahlreiche Window-Manager zur Verfügung. Jeder Window-Manager bietet ein anderes Erscheinungsbild: einige unterstützen virtuelle Bildschirme, andere erlauben Tastenkombinationen zur Verwaltung des Bildschirms. Einige besitzen eine "Start" Schaltfläche und in manchen lässt sich das Aussehen und Verhalten der Anwendung über Themes beliebig einstellen. Window-Manager stehen in der Kategorie x11-wm der Ports-Sammlung zur Verfügung.

Jeder Window-Manager wird unterschiedlich konfiguriert. Einige erwarten eine manuell erstellte Konfigurationsdatei, während andere ein grafisches Werkzeug für die meisten Konfigurationsarbeiten anbieten.

**Desktop-Umgebungen**

KDE und GNOME werden als Desktop-Umgebungen bezeichnet, da sie eine ganze Reihe von Anwendungen für typische Desktop-Aufgaben enthalten. Dazu zählen beispielsweise Office-Pakete, Webbrowser und Spiele.

**Fokus**

Der Window-Manager ist für die Methode verantwortlich, mit der ein Fenster den Fokus bekommt. Jedes System, das Fenster verwendet muss entscheiden, wie ein Fenster aktiviert wird, damit es Eingaben empfangen kann. Das aktive Fenster sollte zudem sichtbar gekennzeichnet werden.

Eine Methode wird "click-to-focus" genannt. Ein Fenster wird aktiv, wenn es mit der Maus angeklickt wird. Eine weitere Methode ist "focus-follows-mouse". Hier hat liegt der Fokus auf dem Fenster, auf dem sich der Mauszeiger befindet. Wird der Mauszeiger in ein anderes Fenster bewegt, so erhält dieses Fenster den Fokus. Eine dritte Methode ist "sloppy-focus". Hier wechselt der Fokus nur dann, wenn sich der Mauszeiger in ein neues Fenster bewegt und nicht, wenn er das aktive Fenster verlässt. Ist der Mauszeiger auf der Desktop Oberfläche, so bleibt der Fokus auf dem zuletzt verwendeten Fenster. Bei der Methode "click-to-focus" wird das aktive Fenster durch einen Mausklick festgelegt. Dabei kann das Fenster vor alle anderen Fenster gesetzt werden. Alle Eingaben werden dann, unabhängig von der Position des Mauszeigers, dem aktiven Fenster zugeordnet.

Die verschiedenen Window-Manager unterstützen noch andere Methoden. Alle unterstützen jedoch "click-to-focus" und die meisten von ihnen auch die anderen Methoden. Lesen Sie die Dokumentation des Window-Managers um festzustellen, welche Methoden zur Verfügung stehen.

**Widgets**

*Widget* bezeichnet Objekte, die in irgendeiner Weise geklickt oder manipuliert werden können. Dazu gehören buttons (Schaltflächen), check buttons (Schaltfläche für Mehrfachauswahlen), radio buttions (Schaltfläche für Einfachauswahlen), Icons und Auswahllisten. Eine Widget-Sammlung ist eine Reihe von Widgets, die verwendet werden um grafische Anwendungen zu erstellen. Es gibt mehrere populäre Widget-Sammlungen, einschließlich Qt, das von KDE benutzt wird, und GTK+, das von GNOME benutzt wird. Als Folge dessen, haben Anwendungen einen bestimmten look and feel, je nachdem welche Widget-Sammlung benutzt wurde, um die Anwendung zu erstellen.

# 7.3. Xorg installieren

In FreeBSD kann Xorg als Paket oder Port installiert werden.

Die Installation des Pakets ist zwar schneller, dafür können weniger Optionen angepasst werden:

```
# pkg install xorg
```

Die nachstehenden Kommandos bauen und installieren Xorg aus der Ports-Sammlung:

```
# cd /usr/ports/x11/xorg
# make install clean
```

Bei beiden Vorgehensweisen wird ein vollständiges Xorg-System installiert. Für die meisten Anwender ist die Installation des Binärpakets die bessere Option.

Eine kleinere Version des Xorg-Systems für erfahrene Anwender ist mit x11/xorg-minimal verfügbar. Die meisten Dokumente, Bibliotheken und Anwendungen werden hierbei nicht installiert. Einige Anwendungen erfordern jedoch diese zusätzlichen Komponenten, um ordnungsgemäß zu funktionieren.

# 7.4. Xorg konfigurieren

## 7.4.1. Schnellstartanleitung

Xorg unterstützt die meisten gängigen Grafikkarten, Tastaturen und Zeigegeräte.



Grafikkarten, Monitore und Eingabegeräte werden automatisch erkannt und müssen nicht manuell konfiguriert werden. Erstellen Sie keine xorg.conf und führen Sie nicht `-configure` aus, es sei denn, die automatische Konfiguration

schlägt fehl.

1. Wenn Xorg bereits zuvor auf diesem Computer verwendet wurde, verschieben oder entfernen Sie alle vorhandenen Konfigurationsdateien:

```
# mv /etc/X11/xorg.conf ~/xorg.conf.etc
# mv /usr/local/etc/X11/xorg.conf ~/xorg.conf.localetc
```

2. Fügen Sie die Benutzer, die Xorg verwenden, zur Gruppe `video` oder `wheel` hinzu, um die 3D-Beschleunigung zu aktivieren. Um den Benutzer *jru* in eine der verfügbaren Gruppen hinzuzufügen:

```
# pw groupmod video -m jru || pw groupmod wheel -m jru
```

3. Der Window-Manager `twm` ist standardmäßig enthalten und wird auch gestartet, wenn Xorg startet:

```
% startx
```

4. Auf einigen älteren Versionen von FreeBSD muss die Systemkonsole auf vt(4) eingestellt sein, damit der Wechsel auf die Konsole ordnungsgemäß funktioniert. Informationen dazu finden Sie im Kernel Mode Setting (KMS).

### 7.4.2. Benutzergruppen für Grafikbeschleunigung

Um die 3D-Beschleunigung für Grafikkarten zu ermöglichen, ist der Zugriff auf /dev/dri notwendig. In der Regel ist es am einfachsten, die Benutzer zur Gruppe `video` oder `wheel` hinzuzufügen. In diesem Beispiel wird pw(8) verwendet, um den Benutzer *slurms* zu der Gruppe `video` hinzuzufügen, bzw. zur Gruppe `wheel`, falls die Gruppe `video` nicht existiert:

```
# pw groupmod video -m slurms || pw groupmod wheel -m slurms
```

### 7.4.3. Kernel Mode Setting (KMS)

Wenn der Computer die Anzeige von der Konsole auf eine höhere Bildschirmauflösung für X umstellt, muss der Videoausgabe-Modus eingestellt werden. Neuere Versionen von Xorg verwenden dazu ein System innerhalb des Kernels, um diesen Modus effizienter zu ändern. Ältere Versionen von FreeBSD verwenden dafür sc(4), welches jedoch nicht mit dem KMS-System umgehen kann. Das führt dazu, dass nach dem Schließen von X die Konsole leer bleibt, obwohl sie weiterhin funktioniert. Die neuere vt(4) Konsole vermeidet dieses Problem.

Fügen Sie diese Zeile in /boot/loader.conf ein um vt(4) zu aktivieren:

```
kern.vty=vt
```

### 7.4.4. Konfigurationsdateien

Eine manuelle Konfiguration ist in der Regel nicht erforderlich. Bitte erstellen Sie keine manuellen Konfigurationsdateien, es sei denn, die automatische Konfiguration funktioniert nicht.

#### Verzeichnis

Xorg sucht in verschiedenen Verzeichnissen nach Konfigurationsdateien. Unter FreeBSD ist /usr/local/etc/X11/ das bevorzugte Verzeichnis für diese Dateien. Die Verwendung dieses Verzeichnisses hilft dabei, Anwendungsdateien vom Betriebssystem getrennt zu halten.

Das Speichern von Konfigurationsdateien unter /etc/X11/ funktioniert immer noch, allerdings vermischt diese Methode Anwendungsdateien mit Dateien des Basissystems und wird daher nicht empfohlen.

#### Einzelne oder mehrere Dateien

Anstatt die traditionelle xorg.conf zu verwenden, ist es einfacher, mehrere Dateien, die jeweils eine bestimmte Einstellung konfigurieren, zu verwenden. Diese Dateien werden im Unterverzeichnis xorg.conf.d/ des Hauptverzeichnisses gespeichert. Der vollständige Pfad ist normalerweise /usr/local/etc/X11/xorg.conf.d/.

Beispiele für diese Dateien werden später in diesem Abschnitt vorgestellt.

Die traditionelle, einzelne xorg.conf funktioniert weiterhin, ist jedoch nicht so übersichtlich und flexibel wie die Verwendung von mehreren Dateien im Unterverzeichnis xorg.conf.d/.

### 7.4.5. Grafikkarten

Aufgrund von Änderungen in neueren Versionen von FreeBSD ist es nun möglich, Grafiktreiber zu benutzen, die aus der Ports-Sammlung oder als Pakete bereitgestellt werden. Die folgenden Treiber sind mit graphics/drm-kmod verfügbar:

**Intel KMS driver**

2D- und 3D-Beschleunigung wird auf den meisten Intel KMS driver Grafikkarten von Intel® unterstützt.

Name des Treibers: `i915kms`

2D- und 3D-Beschleunigung wird auf den meisten älteren Radeon KMS driver Grafikkarten von AMD® unterstützt.

Name des Treibers: `radeonkms`

2D- und 3D-Beschleunigung wird auf den meisten neueren AMD KMS driver Grafikkarten von AMD® unterstützt.

Name des Treibers: `amdgpu`

Eine Liste der unterstützten GPUs finden Sie unter https://en.wikipedia.org/wiki/List_of_Intel_graphics_processing_units und https://en.wikipedia.org/wiki/List_of_AMD_graphics_processing_units.

**Intel®**

3D-Beschleunigung wird von den meisten Intel®-Grafikkarten unterstützt, einschließlich Ivy Bridge (HD Graphics 2500, 4000 und P4000), Iron Lake (HD Graphics) und Sandy Bridge (HD Graphics 2000).

Treibername: `intel`

Weitere Informationen finden Sie unter https://en.wikipedia.org/wiki/List_of_Intel_graphics_processing_units.

**AMD® Radeon**

2D- und 3D-Beschleunigung wird von den meisten Radeon-Karten bis zur HD6000-Serie unterstützt.

Treibername: `radeon`

Weitere Informationen finden Sie unter https://en.wikipedia.org/wiki/List_of_AMD_graphics_processing_units.

**NVIDIA**

Verschiedene NVIDIA Treiber sind in der Kategorie x11 der Ports-Sammlung enthalten. Installieren Sie den Treiber, der für die Grafikkarte benötigt wird.

Weitere Informationen finden Sie unter https://en.wikipedia.org/wiki/List_of_Nvidia_graphics/processing_units.

**Hybride Kombinationen**

Einige Notebooks besitzen zusätzlich zum Chipsatz oder Prozessor einen Grafikprozessor. *Optimus* kombiniert Intel® und NVIDIA Hardware. *Umschaltbare Grafik* bzw. *Hybride Grafik* ist eine Kombination aus Intel®, oder AMD® Prozessor mit AMD® Radeon GPU.

Die Implementierungen dieser Hybrid-Grafik-Systeme variieren und Xorg in FreeBSD ist nicht in der Lage, alle Versionen der Hardware zu betreiben.

Einige Computer bieten jedoch eine BIOS-Option, um eine der beiden Grafikkarten zu deaktivieren oder den *diskreten* Modus einzuschalten. Zum Beispiel ist es manchmal möglich, die NVIDIA GPU in einem Optimus-System zu deaktivieren. Intel® Video kann dann mit einem Intel® Treiber verwendet werden.

Die BIOS-Einstellungen sind abhängig vom Modell des Computers. In manchen Situationen können beide GPUs aktiviert bleiben. Um solch ein System lauffähig zu machen genügt es bereits, nur die Haupt-GPU im Abschnitt `Device` der Konfigurationsdatei zu setzen.

### Andere Grafikkarten

Treiber für weniger gebräuchliche Grafikkarten finden Sie in der Kategorie x11-drivers der Ports-Sammlung.

Karten, die nicht durch einen speziellen Treiber unterstützt werden, sind vielleicht noch mit dem Treiber x11-drivers/xf86-video-vesa nutzbar. Dieser Treiber wird von x11/xorg installiert. Der Treiber kann auch manuell als x11-drivers/xf86-video-vesa installiert werden. Xorg versucht immer diesen Treiber zu verwenden, wenn für die Grafikkarte kein passender Treiber gefunden wird.

x11-drivers/xf86-video-scfb ist ein ähnlicher Treiber, der mit vielen UEFI und ARM® Computern funktioniert.

### Video-Treiber über eine Datei einstellen

Den Intel® Treiber in einer Konfigurationsdatei einstellen:

*Beispiel 15. Den Intel® Treiber über eine Datei auswählen*

/usr/local/etc/X11/xorg.conf.d/driver-intel.conf

```
Section "Device"
    Identifier  "Card0"
    Driver      "intel"
    # BusID     "PCI:1:0:0"
EndSection
```

Wenn mehr als eine Grafikkarte vorhanden ist, kann der Eintrag `BusID` verwendet werden, um die gewünschte Karte auszuwählen. Eine Liste der `BusID`s der Grafikkarten kann mit `pciconf -lv | grep -B3 display` ausgegeben werden.

Den Radeon Treiber in einer Konfigurationsdatei einstellen:

*Beispiel 16. Den Radeon Treiber über eine Datei auswählen*

/usr/local/etc/X11/xorg.conf.d/driver-radeon.conf

```
Section "Device"
    Identifier  "Card0"
    Driver      "radeon"
EndSection
```

Den VESA Treiber in einer Konfigurationsdatei einstellen:

*Beispiel 17. Den VESA Treiber über eine Datei auswählen*

/usr/local/etc/X11/xorg.conf.d/driver-vesa.conf

```
Section "Device"
    Identifier  "Card0"
    Driver      "vesa"
EndSection
```

Den Treiber `scfb` für UEFI- oder ARM®-Computer auswählen:

*Beispiel 18. Den `scfb` Treiber über eine Datei auswählen*

/usr/local/etc/X11/xorg.conf.d/driver-scfb.conf

```
Section "Device"
    Identifier  "Card0"
    Driver      "scfb"
EndSection
```

### 7.4.6. Monitore

Fast alle Monitore unterstützen den Extended Display Identification Data Standard (EDID). Xorg verwendet EDID um mit dem Monitor zu kommunizieren und die unterstützten Auflösungen und Bildwiederholfrequenzen zu erkennen. Xorg wählt dann die für den Monitor am besten geeignete Kombination von Einstellungen.

Weitere vom Monitor unterstützte Auflösungen, können in der Konfigurationsdatei, oder nach dem Start des X-Servers mit xrandr(1) gesetzt werden.

**xrandr(1) benutzen**

Führen Sie xrandr(1) ohne Parameter aus, um eine Liste von Video-Ausgängen und erkannten Monitor-Modi zu sehen:

```
% xrandr
Screen 0: minimum 320 x 200, current 3000 x 1920, maximum 8192 x 8192
DVI-0 connected primary 1920x1200+1080+0 (normal left inverted right x axis y axis)
495mm x 310mm
   1920x1200     59.95*+
   1600x1200     60.00
   1280x1024     85.02    75.02    60.02
   1280x960      60.00
   1152x864      75.00
   1024x768      85.00    75.08    70.07    60.00
   832x624       74.55
   800x600       75.00    60.32
   640x480       75.00    60.00
   720x400       70.08
DisplayPort-0 disconnected (normal left inverted right x axis y axis)
```

```
HDMI-0 disconnected (normal left inverted right x axis y axis)
```

Die Auflistung zeigt, dass der DVI-0 Ausgang benutzt wird, um eine Bildschirmauflösung von 1920x1200 bei einer Bildwiederholrate von 60 Hz anzuzeigen. An den Anschlüssen DisplayPort-0 und HDMI-0 sind keine Monitore angeschlossen.

Die anderen Anzeigemodi können mit xrandr(1) ausgewählt werden. Um beispielsweise auf 1280x1024 bei 60 Hz umzuschalten:

```
% xrandr --mode 1280x1024 --rate 60
```

Häufig wird für einen Videoprojektor der externe Videoausgang eines Notebooks verwendet.

Die Typen und Anzahl der Videoanschlüsse variiert zwischen den Geräten und auch die Ausgabe variiert von Treiber zu Treiber. Was für den einen Treiber HDMI-1 ist, nennt ein anderer Treiber vielleicht HDMI1. Führen Sie daher zunächst xrandr(1) aus, um alle verfügbaren Anschlüsse aufzulisten.

```
% xrandr
Screen 0: minimum 320 x 200, current 1366 x 768, maximum 8192 x 8192
LVDS1 connected 1366x768+0+0 (normal left inverted right x axis y axis) 344mm x
193mm
   1366x768      60.04*+
   1024x768      60.00
   800x600       60.32    56.25
   640x480       59.94
VGA1 connected (normal left inverted right x axis y axis)
   1280x1024     60.02 +  75.02
   1280x960      60.00
   1152x864      75.00
   1024x768      75.08    70.07    60.00
   832x624       74.55
   800x600       72.19    75.00    60.32    56.25
   640x480       75.00    72.81    66.67    60.00
   720x400       70.08
HDMI1 disconnected (normal left inverted right x axis y axis)
DP1 disconnected (normal left inverted right x axis y axis)
```

Vier Ausgänge wurden gefunden: das integrierte Panel LVDS1, sowie die externen Anschlüsse VGA1, HDMI1 und DP1.

Der Videoprojektor wurde am Ausgang VGA1 angeschlossen. xrandr(1) wird nun verwendet, um diese Ausgabe auf die native Auflösung des Projektors einzustellen und den zusätzlichen Platz auf der rechten Seite des Desktops hinzuzufügen:

```
% xrandr --output VGA1 --auto --right-of LVDS1
```

`--auto` wählt die Auflösung und Aktualisierungsrate die von EDID ermittelt wurden. Wenn die Auflösung nicht richtig ermittelt wurde, kann ein fester Wert mit `--mode` anstelle von `--auto` angegeben werden. Beispielsweise können die meisten Projektoren mit einer Auflösung von 1024x768 betrieben werden, die mit `--mode 1024x768` gesetzt wird.

xrandr(1) wird häufig aus .xinitrc ausgeführt, um den entsprechenden Modus zu setzen wenn X startet.

**Bildschirmauflösung über eine Datei einstellen**

Eine Bildschirmauflösung von 1024x768 in einer Konfigurationsdatei einstellen:

*Beispiel 19. Die Bildschirmauflösung in eine Datei schreiben*

/usr/local/etc/X11/xorg.conf.d/screen-resolution.conf

```
Section "Screen"
    Identifier "Screen0"
    Device     "Card0"
    SubSection "Display"
    Modes      "1024x768"
    EndSubSection
EndSection
```

Die wenigen Monitore, die EDID nicht beherrschen, können durch setzen von `HorizSync` und `VertRefresh` auf den Bereich der vom Monitor unterstützten Frequenzen konfiguriert werden.

*Beispiel 20. Manuelles Einstellen der Monitorfrequenzen*

/usr/local/etc/X11/xorg.conf.d/monitor0-freq.conf

```
Section "Monitor"
    Identifier   "Monitor0"
    HorizSync    30-83  # kHz
    VertRefresh  50-76  # Hz
EndSection
```

## 7.4.7. Eingabegeräte

**Tastaturen**

**Tastaturlayout**

Die standardisierte Position von Tasten auf einer Tastatur wird als *Layout* bezeichnet. Layouts und andere einstellbare Parameter werden in xkeyboard-config(7) beschrieben.

In der Voreinstellung ist ein US-amerikanisches Layout aktiv. Um ein alternatives Layout zu wählen, setzen Sie die Optionen `XkbLayout` und `XkbVariant` in der Klasse `InputClass`. Dies wird für

alle Eingabegeräte der entsprechenden Klasse angewendet werden.

Dieses Beispiel konfiguriert ein deutsches Tastaturlayout.

*Beispiel 21. Konfiguration eines Tastaturlayouts*

/usr/local/etc/X11/xorg.conf.d/keyboard-de.conf

```
Section "InputClass"
    Identifier  "KeyboardDefaults"
    MatchIsKeyboard "on"
    Option      "XkbLayout" "de"
EndSection
```

*Beispiel 22. Konfiguration mehrerer Tastaturlayouts*

Hier werden die Tastaturlayouts für Vereinigte Staaten, Spanien und Ukraine gesetzt. Mit `Alt` + `Shift` können Sie zwischen den einzelnen Layouts wechseln. Für eine verbesserte Steuerung des Layouts kann x11/xxkb oder x11/sbxkb benutzt werden.

/usr/local/etc/X11/xorg.conf.d/kbd-layout-multi.conf

```
Section "InputClass"
    Identifier  "All Keyboards"
    MatchIsKeyboard "yes"
    Option      "XkbLayout" "us,es,ua"
EndSection
```

### Xorg über die Tastatur beenden

X kann über eine Tastenkombination geschlossen werden. Standardmäßig ist die Tastenkombination jedoch nicht gesetzt, da sie mit Tastaturbefehlen für einige Anwendungen in Konflikt steht. Die Aktivierung dieser Option erfordert Änderungen in der Sektion `InputDevice` für die Tastatur:

*Beispiel 23. X über die Tastatur beenden*

/usr/local/etc/X11/xorg.conf.d/keyboard-zap.conf

```
Section "InputClass"
    Identifier  "KeyboardDefaults"
    MatchIsKeyboard "on"
    Option      "XkbOptions" "terminate:ctrl_alt_bksp"
EndSection
```

#### Mäuse und Zeigegeräte



Wenn Sie unter FreeBSD 12.1 das Paket xorg-server 1.20.8 oder eine neuere Version installiert haben, und Sie auch nicht den moused(8)-Daemon verwenden, fügen Sie `kern.evdev.rcpt_mask=12` in /etc/sysctl.conf ein.

Viele Parameter für die Maus können über Konfigurationseinstellungen eingestellt werden. mousedrv(4) enthält eine vollständige Liste.

#### Mauszeiger

Die Anzahl der Maustasten wird in xorg.conf im Abschnitt `InputDevice` für die Maus festgelegt. Um die Anzahl der Tasten auf 7 zu setzen:

*Beispiel 24. Die Anzahl der Maustasten festlegen*

/usr/local/X11/xorg.conf.d/mouse0-buttons.conf

```
Section "InputDevice"
    Identifier  "Mouse0"
    Option      "Buttons" "7"
EndSection
```

### 7.4.8. Manuelle Konfiguration

In einigen Fällen funktioniert die Autokonfiguration nicht mit bestimmter Hardware, oder es wird eine andere Konfiguration benötigt. Für diese Fälle kann eine benutzerdefinierte Konfigurationsdatei erstellt werden.



Erstellen Sie keine manuellen Konfigurationsdateien, sofern dies nicht erforderlich ist. Eine unnötige manuelle Konfiguration kann den ordnungsgemäßen Betrieb verhindern.

Eine Konfigurationsdatei kann, basierend auf der von Xorg erfassten Hardware erzeugt werden. Diese Konfigurationsdatei ist ein guter Ausgangspunkt für angepasste Konfigurationen.

Erzeugung einer xorg.conf:

```
# Xorg -configure
```

Die Konfigurationsdatei wird in /root/xorg.conf.new gespeichert. Machen Sie alle gewünschten Änderungen an dieser Datei. Danach testen Sie die Datei mit:

```
# Xorg -retro -config /root/xorg.conf.new
```

Nachdem die neue Konfiguration angepasst und getestet wurde, kann die Konfiguration in kleinere

Dateien unter /usr/local/etc/X11/xorg.conf.d/ aufgeteilt werden.

# 7.5. Schriftarten in Xorg benutzen

## 7.5.1. Type 1 Schriftarten

Die Schriftarten, die mit Xorg ausgeliefert werden, eignen sich ganz und gar nicht für Desktop-Publishing-Anwendungen. Große Schriftarten zeigen bei Präsentationen deutliche Treppenstufen und kleine Schriftarten sind fast unleserlich. Es gibt allerdings mehrere hochwertige Type 1 Schriftarten (PostScript®), die mit Xorg benutzt werden können. Beispielsweise enthalten die URW-Schriftarten (x11-fonts/urwfonts) hochwertige Versionen gängiger Type 1 Schriftarten (unter anderem Times Roman™, Helvetica™, Palatino™). Die Sammlung Freefonts (x11-fonts/freefonts) enthält viele weitere Schriftarten, doch sind diese für den Einsatz in Grafikprogrammen wie Gimp gedacht und nicht für den alltäglichen Gebrauch. Weiterhin kann Xorg mit einem Minimum an Aufwand konfiguriert werden, damit TrueType®-Schriftarten benutzt werden können. Mehr dazu erfahren Sie in der Manualpage X(7) und im TrueType®-Schriftarten.

Die Type 1 Schriftarten lassen sich als Paket wie folgt installieren:

```
# pkg install urwfonts
```

Alternativ können die Schriftarten aus der Ports-Sammlung gebaut und installiert werden:

```
# cd /usr/ports/x11-fonts/urwfonts
# make install clean
```

Analog lassen sich Freefont und andere Sammlungen installieren. Damit der X-Server diese Schriftarten erkennt, fügen Sie eine entsprechende Zeile in die Konfigurationsdatei des X-Servers (/etc/X11/xorg.conf) hinzu:

```
FontPath "/usr/local/shared/fonts/urwfonts/"
```

Alternativ kann in der X-Sitzung das folgende Kommando abgesetzt werden:

```
% xset fp+ /usr/local/shared/fonts/urwfonts
% xset fp rehash
```

Jetzt kennt der X-Server die neuen Schriftarten, jedoch nur bis zu Ende der Sitzung. Soll die Änderung dauerhaft sein, müssen die Befehle in ~/.xinitrc eingetragen werden, wenn X mittels `startx` gestartet wird, beziehungsweise in ~/.xsession, wenn ein grafischer Login-Manager, wie XDM verwendet wird. Eine dritte Möglichkeit besteht darin, /usr/local/etc/fonts/local.conf zu verwenden, was im Anti-aliasing demonstriert wird.

### 7.5.2. TrueType®-Schriftarten

Xorg besitzt eine eingebaute Unterstützung zur Darstellung von TrueType®-Schriftarten. Hierzu existieren zwei verschiedene Module, die diese Funktionalität aktivieren können. In diesem Beispiel wird das Freetype-Modul benutzt, da es besser mit anderen Werkzeugen, die TrueType®-Schriftarten darstellen, übereinstimmt. Um das Freetype-Modul zu aktivieren, muss die folgende Zeile zum Abschnitt `"Module"` in /etc/X11/xorg.conf hinzugefügt werden.

```
Load "freetype"
```

Erstellen Sie ein Verzeichnis für die TrueType®-Schriftarten (beispielsweise /usr/local/shared/fonts/TrueType) und kopieren Sie alle Schriftarten dorthin. Beachten Sie, dass die Schriftarten für Xorg im UNIX®/MS-DOS®/Windows®-Format vorliegen müssen und nicht direkt von einem Apple® Mac® übernommen werden können. Sobald die Dateien in das Verzeichnis kopiert wurden, verwenden Sie mkfontscale um fonts.dir zu erstellen, damit X weiß, dass diese neuen Dateien installiert wurden. `mkfontscale` kann als Paket installiert werden:

```
# pkg install mkfontscale
```

Erstellen Sie dann einen Index der Schriftarten für X:

```
# cd /usr/local/shared/fonts/TrueType
# mkfontscale
```

Geben Sie dem System das TrueType®-Verzeichnis, wie im Type 1 Schriftarten beschrieben, bekannt:

```
# xset fp+ /usr/local/shared/fonts/TrueType
# xset fp rehash
```

Oder fügen Sie eine `FontPath`-Zeile in xorg.conf ein.

Jetzt sollten Gimp, Apache OpenOffice und alle anderen X-Anwendungen die TrueType®-Schritarten erkennen. Extrem kleine Schriftarten (Webseiten, die mit hoher Auflösung betrachtet werden) und sehr große Schriftarten (in StarOffice™) werden jetzt viel besser aussehen.

### 7.5.3. Anti-aliasing

Alle Schriftarten in Xorg, die in den Verzeichnissen /usr/local/shared/fonts/ und ~/.fonts/ gefunden werden, werden automatisch für Anti-aliasing an Anwendungen zur Verfügung gestellt, die Xft beherrschen. Die meisten aktuellen Anwendungen beherrschen Xft, dazu gehören auch KDE, GNOME und Firefox.

In /usr/local/etc/fonts/local.conf werden die Schriftarten, die mit dem Anti-aliasing-Verfahren benutzt werden sollen und die Eigenschaften des Verfahrens festgelegt. In diesem Abschnitt wird

nur die grundlegende Konfiguration von Xft beschrieben. Weitere Details entnehmen Sie bitte der Hilfeseite fonts-conf(5).

Die Datei local.conf ist ein XML-Dokument. Achten Sie beim Editieren der Datei daher auf die richtige Groß- und Kleinschreibung und darauf, dass alle Tags geschlossen sind. Die Datei beginnt mit der üblichen XML-Deklaration gefolgt von einer DOCTYPE-Definition und dem `<fontconfig>` -Tag:

```
<?xml version="1.0"?>
        <!DOCTYPE fontconfig SYSTEM "fonts.dtd">
        <fontconfig>
```

Wie vorher erwähnt, stehen schon alle Schriftarten in /usr/local/shared/fonts/ und ~/.fonts/ für Anwendungen, die Xft unterstützen, zur Verfügung. Um ein Verzeichnis außerhalb dieser beiden Bäume zu benutzen, fügen Sie eine Zeile wie die nachstehende in /usr/local/etc/fonts/local.conf hinzu:

```
<dir>/path/to/my/fonts</dir>
```

Wenn Sie neue Schriftarten hinzugefügt haben, müssen Sie den Schriftarten-Cache neu aufbauen:

```
# fc-cache -f
```

Das Anti-aliasing-Verfahren zeichnet Ränder leicht unscharf, dadurch werden kleine Schriften besser lesbar und der Treppenstufen-Effekt bei wird großen Schriften vermieden. Auf normale Schriftgrößen sollte das Verfahren aber nicht angewendet werden, da dies die Augen zu sehr anstrengt. Um kleinere Schriftgrößen als 14 Punkt von dem Verfahren auszuschließen, fügen Sie in local.conf die nachstehenden Zeilen ein:

```
        <match target="font">
      <test name="size" compare="less">
          <double>14</double>
      </test>
      <edit name="antialias" mode="assign">
          <bool>false</bool>
      </edit>
    </match>
    <match target="font">
      <test name="pixelsize" compare="less" qual="any">
          <double>14</double>
      </test>
      <edit mode="assign" name="antialias">
          <bool>false</bool>
      </edit>
    </match>
```

Das Anti-aliasing-Verfahren kann die Abstände einiger Fixschriften falsch darstellen, dies fällt besonders unter KDE auf. Sie können das Problem umgehen, indem Sie die Abstände dieser Schriften auf den Wert `100` festsetzen. Fügen Sie die nachstehenden Zeilen hinzu:

```
        <match target="pattern" name="family">
           <test qual="any" name="family">
               <string>fixed</string>
           </test>
           <edit name="family" mode="assign">
               <string>mono</string>
           </edit>
        </match>
        <match target="pattern" name="family">
            <test qual="any" name="family">
                <string>console</string>
            </test>
            <edit name="family" mode="assign">
                <string>mono</string>
            </edit>
        </match>
```

Damit werden die Namen der gebräuchlichen Fixschriften auf `"mono"` abgebildet. Für diese Schriften setzen Sie dann den Abstand fest:

```
         <match target="pattern" name="family">
             <test qual="any" name="family">
                 <string>mono</string>
             </test>
             <edit name="spacing" mode="assign">
                 <int>100</int>
             </edit>
         </match>
```

Bestimmte Schriftarten, wie Helvetica, können Probleme mit dem Anti-Aliasing-Verfahren verursachen. In der Regel erscheinen diese Schriftarten dann vertikal halbiert. Im schlimmsten Fall stürzen Anwendungen als Folge davon ab. Sie vermeiden dies, indem Sie betroffene Schriftarten in local.conf von dem Verfahren ausnehmen:

```
         <match target="pattern" name="family">
             <test qual="any" name="family">
                 <string>Helvetica</string>
             </test>
             <edit name="family" mode="assign">
                 <string>sans-serif</string>
             </edit>
         </match>
```

Nachdem Sie local.conf editiert haben, müssen Sie sicherstellen, dass die Datei mit dem Tag `</fontconfig>` endet. Ist das nicht der Fall, werden die Änderungen nicht berücksichtigt.

Benutzer können personalisierte Einstellungen in ~/.fonts.conf vornehmen. Diese Datei verwendet die gleiche XML-Syntax wie im obigen Beispiel.

Mit einem LCD können Sie sub-pixel sampling anstelle von Anti-aliasing einsetzen. Dieses Verfahren behandelt die horizontal getrennten Rot-, Grün- und Blau-Komponenten eines Pixels gesondert und verbessert damit (teilweise sehr wirksam) die horizontale Auflösung. Die nachstehende Zeile in local.conf aktiviert diese Funktion:

```
        <match target="font">
            <test qual="all" name="rgba">
                <const>unknown</const>
            </test>
            <edit name="rgba" mode="assign">
                <const>rgb</const>
            </edit>
        </match>
```



Abhängig von der Art Ihres Bildschirms müssen Sie anstelle von `rgb` eines der folgenden verwenden: `bgr`, `vrgb` oder `vbgr`. Experimentieren Sie und vergleichen, was besser aussieht.

# 7.6. Der X-Display-Manager

Xorg enthält den X-Display-Manager XDM, um Sitzungen zu verwalten. XDM stellt eine graphische Anmeldemaske zur Verfügung, in der Sie den Server, auf dem eine Sitzung laufen soll, auswählen können und in der Sie die Autorisierungs-Informationen, wie Benutzername und Passwort, eingeben können.

Dieser Abschnitt zeigt, wie der X-Displaymanager konfiguriert wird. Einige grafische Oberflächen enthalten ihre eigenen graphischen Login-Manager. Eine Anleitung zur Konfiguration des GNOME Display-Managers finden Sie im GNOME. Eine Anleitung zur Konfiguration des KDE Display Managers finden Sie im KDE.

## 7.6.1. XDM einrichten

XDM kann über das Paket oder den Port x11/xdm installiert werden. Nach der Installation lässt sich XDM durch einen Eintrag in /etc/ttys bei jedem Start des Rechners aktivieren:

```
ttyv8   "/usr/local/bin/xdm -nodaemon"  xterm   off secure
```

Ändern Sie den Wert `off` zu `on` und speichern Sie die Datei. `ttyv8` zeigt an, dass XDM auf dem neunten virtuellen Terminal ausgeführt wird.

Die Konfigurationsdateien von XDM befinden sich in /usr/local/etc/X11/xdm. Dieses Verzeichnis

enthält einige Dateien, mit denen das Verhalten und Aussehen von XDM beeinflusst werden kann, sowie ein paar Skripte und Programme zur Einrichtung des Desktops. Eine Zusammenfassung der Aufgaben dieser Dateien beschreibt die Die Konfigurationsdateien von XDM. Die genaue Syntax und Verwendung wird in xdm(1) beschrieben.

*Tabelle 6. Die Konfigurationsdateien von XDM*

| Datei | Beschreibung |
|---|---|
| Xaccess | Verbindungen zu XDM werden über das "X Display Manager Connection Protocol" (XDMCP) hergestellt. Xaccess enthält die Client-Berechtigungen zur Steuerung der XDMCP-Verbindungen entfernter Maschinen. In der Voreinstellung erlaubt diese Datei keine Verbindungen von entfernten Maschinen. |
| Xresources | Diese Datei steuert das Erscheinungsbild der Bildschirmauswahl und Anmeldemasken von XDM. In der Voreinstellung erscheint ein rechteckiges Anmeldefenster, dass den Hostnamen und einen Anmeldeprompt mit "Login:" und "Password" anzeigt. Das Format dieser Datei entspricht den Dateien im Verzeichnis app-defaults, die in der Dokumentation von Xorg beschrieben sind. |
| Xservers | Diese Datei enthält eine Liste entfernter Rechner, die in der Bildschirmauswahl angeboten werden. |
| Xsession | Dieses Skript wird von XDM aufgerufen, nachdem sich ein Benutzer erfolgreich angemeldet hat. Es verweist auf ein angepasstes Skript in ~/.xsession. |
| Xsetup_* | Diese Skripten werden automatisch ausgeführt, bevor die Bildschirmauswahl oder die Anmeldemasken angezeigt werden. Für jeden lokalen Bildschirm gibt es ein Skript namens Xsetup_*, wobei * die lokale Bildschirmnummer ist. Normalerweise werden damit ein oder zwei Programme, wie `xconsole`, im Hindergrund gestartet. |
| xdm-config | Konfiguration für alle auf der Maschine verwalteten Bildschirme. |

| Datei | Beschreibung |
| --- | --- |
| xdm-errors | Enthält Fehler, die vom Server generiert werden. Wenn ein von XDM verwalteter Bildschirm hängen bleibt, suchen Sie in dieser Datei nach Fehlermeldungen. Für jede Sitzung werden die Meldungen auch in die Datei ~/.xsession-errors des Benutzers geschrieben. |
| xdm-pid | Die Prozess-ID des gerade laufenden XDM-Prozesses. |

### 7.6.2. Fernzugriff einrichten

In der Voreinstellung können sich nur Benutzer auf dem selben System über XDM anmelden. Um es Benutzern anderer Systeme zu ermöglichen, sich mit dem Bildschirm-Server zu verbinden, muss der Zugriffsregelsatz bearbeitet und der Listener aktiviert werden.

Um XDM so zu konfigurieren, dass jede Verbindung angenommen wird, kommentieren Sie die Zeile `DisplayManager.requestPort` in /usr/local/etc/X11/xdm/xdm-config aus, indem Sie der Zeile ein `!` voranstellen.

```
! SECURITY: do not listen for XDMCP or Chooser requests
! Comment out this line if you want to manage X terminals with xdm
DisplayManager.requestPort:     0
```

Speichern Sie die Änderungen und starten Sie XDM neu. Um den Fernzugriff zu beschränken, sehen Sie sich die Beispiele in /usr/local/etc/X11/xdm/Xaccess an. Zusätzliche Informationen finden Sie in xdm(1)

## 7.7. Grafische Oberflächen

Dieser Abschnitt beschreibt die Installation der drei beliebtesten grafischen Oberflächen unter FreeBSD. Eine Oberfläche kann alles von einem einfachen Window-Manager bis hin zu kompletten Anwendungen sein. Mehr als einhundert grafische Oberflächen stehen in der Kategorie x11-wm der Ports-Sammlung zur Verfügung.

### 7.7.1. GNOME

GNOME ist eine benutzerfreundliche Oberfläche. Es besitzt eine Leiste, mit der Anwendungen gestartet werden und die Statusinformationen anzeigen kann. Programme und Daten können auf der Oberfläche abgelegt werden und Standardwerkzeuge stehen zur Verfügung. Es gibt Konventionen, die es Anwendungen leicht machen, zusammenzuarbeiten und ein konsistentes Erscheinungsbild garantieren. Weitere Informationen zu GNOME unter FreeBSD finden Sie unter https://www.FreeBSD.org/gnome. Die Webseite enthält zusätzliche Informationen über die Installation, Konfiguration und Verwaltung von GNOME unter FreeBSD.

Diese grafische Oberfläche kann als Paket installiert werden:

```
# pkg install gnome3
```

Um GNOME stattdessen aus der Ports-Sammlung zu übersetzen, nutzen Sie das folgende Kommando. GNOME ist eine große Anwendung, die sogar auf einem schnellen Computer einige Zeit zum Übersetzten benötigt.

```
# cd /usr/ports/x11/gnome3
# make install clean
```

GNOME benötigt ein eingehängtes /proc Dateisystem. Fügen Sie daher die folgende Zeile in /etc/fstab ein, damit procfs(5) beim Systemstart automatisch eingehängt wird:

```
proc           /proc       procfs  rw  0   0
```

GNOME benötigt D-Bus und HAL für einen Nachrichtenbus und Hardware Abstraktion. Diese Anwendungen werden automatisch als Abhängigkeiten von GNOME installiert. Aktivieren Sie die Dienste in /etc/rc.conf, sodass sie automatisch gestartet werden wenn das System bootet:

```
dbus_enable="YES"
hald_enable="YES"
```

Nach der Installation weisen Sie Xorg an, GNOME zu starten. Der einfachste Weg, dies zu tun, ist über den GNOME Display Manager GDM, der als Teil des GNOME-Desktops installiert wird. Um GDM zu aktivieren, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
gdm_enable="YES"
```

In der Regel ist es ratsam, alle GNOME-Dienste zu starten. Um dies zu erreichen, fügen Sie die folgende Zeile in /etc/rc.conf ein:

```
gnome_enable="YES"
```

GDM wird nun automatisch gestartet, wenn das System hochfährt.

GNOME kann alternativ auch von der Kommandozeile gestartet werden, wenn eine entsprechend konfigurierte ~/.xinitrc vorliegt. Existiert diese Datei bereits, ersetzen Sie den Aufruf des Window-Managers durch /usr/local/bin/gnome-session. Wenn .xinitrc nicht existiert, erstellen Sie die Datei mit folgendem Befehl:

```
% echo "exec /usr/local/bin/gnome-session" > ~/.xinitrc
```

Eine dritte Methode ist, XDM als Display-Manager zu verwenden. In diesem Fall erstellen Sie eine

ausführbare ~/.xsession:

```
% echo "exec /usr/local/bin/gnome-session" > ~/.xsession
```

### 7.7.2. KDE

KDE ist eine weitere, leicht zu benutzende Desktop-Umgebung. Dieser Desktop bietet eine Sammlung von Anwendungen mit einheitlichem Erscheinungsbild (look and feel), einheitlichen Menüs, Werkzeugleisten, Tastenkombinationen, Farbschemata, Internationalisierung und einer zentralen, dialoggesteuerten Desktop-Konfiguration. Weitere Informationen zu KDE finden Sie unter http://www.kde.org/. Spezifische Informationen für FreeBSD finden Sie unter http://freebsd.kde.org.

Um KDE als Paket zu installieren, geben Sie ein:

```
# pkg install x11/kde5
```

Um KDE stattdessen aus dem Quellcode zu übersetzen, verwenden Sie das folgende Kommando. Bei der Installation wird ein Menü zur Auswahl der Komponenten angezeigt. KDE ist eine große Anwendung, die sogar auf einem schnellen Computer einige Zeit zum Übersetzen benötigt.

```
# cd /usr/ports/x11/kde5
# make install clean
```

KDE benötigt ein eingehängtes /proc. Fügen Sie diese Zeile in /etc/fstab ein, um das Dateisystem automatisch beim Systemstart einzuhängen:

```
proc           /proc       procfs  rw  0   0
```

KDE benötigt D-Bus und HAL für einen Nachrichtenbus und Hardware Abstraktion. Diese Anwendungen werden automatisch als Abhängigkeiten von KDE installiert. Aktivieren Sie die Dienste in /etc/rc.conf, sodass sie automatisch gestartet werden wenn das System bootet:

```
dbus_enable="YES"
hald_enable="YES"
```

Seit KDE Plasma 5 wird der KDE Display-Manager KDM nicht weiterentwickelt. Eine mögliche Alternative ist SDDM. Sie können das Paket wie folgt installieren:

```
# pkg install x11/sddm
```

Fügen Sie anschließend folgende Zeile in /etc/rc.conf ein:

```
sddm_enable="YES"
```

Eine zweite Möglichkeit KDE zu starten, ist `startx` in der Kommandozeile einzugeben. Damit dies funktioniert, wird folgende Zeile in ~/.xinitrc benötigt:

```
exec ck-launch-session startplasma-x11
```

Eine dritte Möglichkeit ist KDE über XDM zu starten. Um dies zu tun, erstellen Sie eine ausführbare ~/.xsession wie folgt:

```
% echo "exec ck-launch-session startkde" > ~/.xsession
```

Sobald KDE gestartet wird, finden Sie im integrierten Hilfesystem weitere Informationen zur Benutzung der verschiedenen Menüs und Anwendungen.

### 7.7.3. Xfce

Xfce ist eine Desktop-Umgebung, basierend auf den von GNOME verwendeten GTK+-Bibliotheken. Es hat einen geringeren Speicherbedarf und stellt dabei einen schlichten, effizienten und einfach zu benutzenden Desktop zur Verfügung. Xfce ist vollständig konfigurierbar, verfügt über eine Programmleiste mit Menüs, Applets und einen Programmstarter. Zudem sind ein Datei-Manager und ein Sound-Manager enthalten und das Programm ist über Themes anpassbar. Da es schnell, leicht und effizient ist, eignet sich Xfce ideal für ältere oder langsamere Rechner mit wenig Speicher. Weitere Informationen zu Xfce finden Sie unter http://www.xfce.org.

Um das Paket Xfce zu installieren, geben Sie folgendes ein:

```
# pkg install xfce
```

Um stattdessen den Port zu übersetzen:

```
# cd /usr/ports/x11-wm/xfce4
# make install clean
```

Xfce benutzt D-Bus als Nachrichtenbus. Die Komponente wird automatisch als Abhängigkeit von Xfce installiert. Um D-Bus beim Hochfahren des Systems zu starten, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
dbus_enable="YES"
```

Im Gegensatz zu GNOME oder KDE, besitzt Xfce keinen eigenen Login-Manager. Damit Xfce von der Kommandozeile mit `startx` gestartet werden kann, muss zunächst ~/.xinitrc mit diesem Befehl erstellt werden:

```
% echo ". /usr/local/etc/xdg/xfce4/xinitrc" > ~/.xinitrc
```

Alternativ dazu kann XDM verwendet werden. Um diese Methode zu konfigurieren, erstellen Sie eine ausführbare ~/.xsession:

```
% echo ". /usr/local/etc/xdg/xfce4/xinitrc" > ~/.xsession
```

# 7.8. Compiz Fusion installieren

Der Einsatz von hübschen 3D-Effekten ist eine Möglichkeit, die Benutzerfreundlichkeit eines Desktop-Rechners zu erhöhen.

Die Installation des Compiz Fusion Pakets ist einfach, aber bei der Konfiguration sind ein paar Schritte notwendig, die nicht in der Dokumentation des Ports beschrieben werden.

## 7.8.1. Konfiguration des FreeBSD nVidia-Treibers

Desktop-Effekte erzeugen eine hohe Last auf der Grafikkarte. Für nVidia-basierte Grafikkarten sind die proprietären Treiber für eine gute Leistung erforderlich. Benutzer anderer Grafikkarten können diesen Abschnitt überspringen und mit der Konfiguration von Xorg fortfahren.

Lesen Sie die FAQ zu diesem Thema, um herauszufinden, wie der richtige nVidia-Treiber ermittelt werden kann.

Nachdem der richtige Treiber für die Karte ermittelt wurde, kann er wie jedes andere Paket installiert werden.

Um beispielsweise den aktuellsten Treiber zu installieren:

```
# pkg install x11/nvidia-driver
```

Der Treiber erstellt ein Kernelmodul, welches beim Systemstart geladen werden muss. Fügen folgende Zeile in /boot/loader.conf ein:

```
nvidia_load="YES"
```



Um das Kernelmodul direkt in den laufenden Kernel zu laden, kann der Befehl `kldload nvidia` eingeben werden. Allerdings wurde festgestellt, dass einige Versionen von Xorg nicht richtig funktionieren, wenn der Treiber nicht beim Systemstart geladen wurde. Nach der Änderung in /boot/loader.conf wird daher ein Neustart des Systems empfohlen.

Wenn das Kernelmodul geladen ist, muss in der Regel nur noch eine einzige Zeile in xorg.conf geändert werden, um den proprietären Treiber zu aktivieren:

Suchen Sie folgende Zeile in /etc/X11/xorg.conf:

```
Driver      "nv"
```

und ändern Sie die Zeile zu:

```
Driver      "nvidia"
```

Wenn Sie nun die grafische Oberfläche starten, sollten Sie vom nVidia Startbildschirm begrüßt werden. Alles sollte wie gewohnt funktionieren.

### 7.8.2. Konfiguration von Desktop-Effekten in xorg.conf

Um Compiz Fusion zu aktivieren, muss /etc/X11/xorg.conf angepasst werden:

Fügen Sie diesen Abschnitt hinzu, um Composite-Effekte zu aktivieren:

```
Section "Extensions"
    Option         "Composite" "Enable"
EndSection
```

Suchen Sie den Abschnitt "Screen", der ähnlich wie hier gezeigt aussehen sollte:

```
Section "Screen"
    Identifier     "Screen0"
    Device         "Card0"
    Monitor        "Monitor0"
    ...
```

und fügen Sie die beiden folgenden Zeilen hinzu (z.B. nach "Monitor"):

```
DefaultDepth    24
Option         "AddARGBGLXVisuals" "True"
```

Suchen Sie den Abschnitt "Subsection", der sich auf die gewünschte Bildschirmauflösung bezieht. Wenn Sie z.B. 1280x1024 verwenden möchten, suchen Sie den folgenden Abschnitt. Sollte die gewünschte Auflösung nicht in allen Unterabschnitten vorhanden sein, können Sie den entsprechenden Eintrag manuell hinzufügen:

```
SubSection     "Display"
    Viewport    0 0
    Modes      "1280x1024"
EndSubSection
```

Für Composite-Effekte wird eine Farbtiefe von 24 Bit benötigt. Ändern Sie dazu den obigen Abschnitt wie folgt:

```
SubSection     "Display"
    Viewport    0 0
    Depth       24
    Modes      "1280x1024"
EndSubSection
```

Zuletzt muss noch sichergestellt werden, dass die Module "glx" und "extmod" im Abschnitt "Module" geladen werden:

```
Section "Module"
    Load           "extmod"
    Load           "glx"
    ...
```

Die vorangegangenen Einstellungen können automatisch mit x11/nvidia-xconfig erledigt werden, indem Sie folgende Kommandos als root ausführen:

```
# nvidia-xconfig --add-argb-glx-visuals
# nvidia-xconfig --composite
# nvidia-xconfig --depth=24
```

### 7.8.3. Installation und Konfiguration von Compiz Fusion

Die Installation von Compiz Fusion ist so einfach wie die Installation jedes anderen Pakets:

```
# pkg install x11-wm/compiz-fusion
```

Wenn die Installation abgeschlossen ist, starten Sie (als normaler Benutzer) den grafischen Desktop mit folgendem Befehl:

```
% compiz --replace --sm-disable --ignore-desktop-hints ccp &
% emerald --replace &
```

Der Bildschirm wird für einige Sekunden flackern, da der Window Manager (z.B. Metacity, wenn Sie GNOME benutzen) von Compiz Fusion ersetzt wird. Emerald kümmert sich um die Fensterdekoration (z.B. die Schatzflächenn schließen, minimieren und maximieren, Titelleisten, usw.).

Sie können dieses einfache Skript anpassen und es dann beim Start automatisch ausführen lassen (z.B. durch Hinzufügen von "Sessions" beim GNOME-Desktop):

```
#! /bin/sh
compiz --replace --sm-disable --ignore-desktop-hints ccp &
emerald --replace &
```

Speichern Sie die Datei in Ihrem Heimatverzeichnis, beispielsweise als start-compiz und machen Sie die Datei ausführbar:

```
% chmod +x ~/start-compiz
```

Benutzen Sie dann die grafische Oberfläche, um das Skript zu Autostart-Programme hinzuzufügen (beim GNOME-Desktop unter Systemwerkzeuge, Einstellungen, Sessions).

Um die gewünschten Effekte und Einstellungen zu konfigurieren, starten Sie (wieder als normaler Benutzer) den Compiz Config Einstellungs-Manager:

```
% ccsm
```



In GNOME finden Sie diese Einstellungen wieder im Menü unter Systemwerkzeuge, Einstellungen.

Wenn Sie "gconf support" während der Installation ausgewählt haben, können Sie diese Einstellungen auch im `gconf-editor` unter `apps/compiz` finden.

## 7.9. Fehlersuche

Wenn die Maus nicht funktioniert, müssen Sie diese zuerst konfigurieren. In neueren Versionen von Xorg werden die `InputDevice`-Abschnitte in xorg.conf ignoriert, um stattdessen die automatisch erkannten Geräte zu verwenden. Um das alte Verhalten wiederherzustellen, fügen Sie folgende Zeile zum Abschnitt `ServerLayout` oder `ServerFlags` dieser Datei hinzu:

```
Option "AutoAddDevices" "false"
```



Wie zuvor erwähnt, wird standardmäßig der hald-Dienst automatisch die Tastatur erkennen. Es kann jedoch passieren, dass das Tastaturlayout oder das Modell nicht korrekt erkannt wird. Grafische Oberflächen wie GNOME, KDE oder Xfce stellen Werkzeuge für die Konfiguration der Tastatur bereit. Es ist allerdings auch möglich, die Tastatureigenschaften direkt zu setzen, entweder mit Hilfe von setxkbmap(1) oder mit einer Konfigurationsregel von hald.

Wenn Sie zum Beispiel eine PC 102-Tasten Tastatur mit französischem Layout verwenden möchten, müssen sie eine Tastaturkonfigurationsdatei x11-input.fdi für hald im Verzeichnis /usr/local/etc/hal/fdi/policy anlegen. Diese Datei sollte die folgenden Zeilen enthalten:

```
<?xml version="1.0" encoding="utf-8"?>
<deviceinfo version="0.2">
  <device>
    <match key="info.capabilities" contains="input.keyboard">
      <merge key="input.x11_options.XkbModel"
type="string">pc102</merge>
      <merge key="input.x11_options.XkbLayout" type="string">fr</merge>
    </match>
  </device>
</deviceinfo>
```

Wenn diese Datei bereits existiert, kopieren Sie nur die Zeilen in die Datei, welche die Tastaturkonfiguration betreffen.

Sie müssen Ihren Computer neu starten, um hald zu zwingen, diese Datei einzulesen.

Es ist auch möglich, die gleiche Konfiguration von einem X-Terminal oder einem Skript über den folgenden Befehl heraus zu tätigen:

```
% setxkbmap -model pc102 -layout fr
```

/usr/local/shared/X11/xkb/rules/base.lst enthält die zur Verfügung stehenden Tastatur- und Layoutoptionen.

Die Konfigurationsdatei xorg.conf.new kann nun an bestimmte Bedürfnisse angepasst werden. Öffnen Sie die Datei in einem Editor, wie emacs(1) oder ee(1). Falls der Monitor ein älteres oder ungewöhnliches Modell ist und keine automatische Erkennung unterstützt, können die Synchronisationsfrequenzen im Abschnitt "Monitor" der xorg.conf.new eingetragen werden.

```
Section "Monitor"
      Identifier   "Monitor0"
      VendorName   "Monitor Vendor"
      ModelName    "Monitor Model"
      HorizSync    30-107
      VertRefresh  48-120
EndSection
```

Die meisten Monitore unterstützen die automatische Erkennung der Synchronisationsfrequenzen, so dass eine manuelle Eingabe der Werte nicht erforderlich ist. Für die wenigen Monitore, die keine automatische Erkennung unterstützen, sollten nur die vom Hersteller zur Verfügung gestellten Werte eingegeben werden, um einen möglichen Schaden zu vermeiden.

X unterstützt die Energiesparfunktionen (DPMS, Energy Star) für Monitore. Mit xset(1) können die Zeitlimits für die DPMS-Modi standby, suspend, off vorgeben, oder zwingend aktiviert werden. Die DPMS-Funktionen können mit der folgenden Zeile im Abschnitt "Monitor" aktiviert werden:

```
Option         "DPMS"
```

Die gewünschte Auflösung und Farbtiefe stellen sie im Abschnitt "Screen" ein:

```
Section "Screen"
      Identifier "Screen0"
      Device     "Card0"
      Monitor    "Monitor0"
      DefaultDepth 24
      SubSection "Display"
          Viewport  0 0
          Depth     24
          Modes     "1024x768"
      EndSubSection
EndSection
```

Mit DefaultDepth wird die standardmäßige Farbtiefe angegeben. Mit der Option -depth von Xorg(1) lässt sich die vorgegebene Farbtiefe überschreiben. Modes gibt die Auflösung für die angegebene Farbtiefe an. Die Farbtiefe im Beispiel beträgt 24 Bits pro Pixel, die zugehörige Auflösung ist 1024x768 Pixel. Beachten Sie, dass in der Voreinstellung nur Standard-VESA-Modi der Grafikkarte angegeben werden können.

Sichern Sie die Konfigurationsdatei. Testen Sie anschließend die Konfiguration, wie oben beschrieben.



Bei der Fehlersuche stehen Ihnen die Protokolldateien von Xorg zur Verfügung. Die Protokolle enthalten Informationen über alle Geräte, die mit dem Xorg-Server verbunden ist. Die Namen der Xorg-Protkolldateien haben das Format /var/log/Xorg.0.log. Der exakte Name der Datei variiert dabei von Xorg.0.log bis Xorg.8.log, und so weiter.

Wenn alles funktioniert, installieren Sie die Datei an einen Ort, an dem Xorg(1) sie finden kann. Typischerweise ist dies /etc/X11/xorg.conf oder /usr/local/etc/X11/xorg.conf.

```
# cp xorg.conf.new /etc/X11/xorg.conf
```

Damit ist die Konfiguration von Xorg abgeschlossesn. Xorg kann nun mit dem Programm startx(1) gestartet werden. Alternativ kann der Xorg-Server auch mithilfe von xdm(1) gestartet werden.

### 7.9.1. Konfiguration des Intel® i810 Graphics Chipsets

Der Intel® i810-Chipset benötigt den Treiber agpgart, die AGP-Schnittstelle für Xorg. Die Manualpage für den Treiber agp(4) enthält weitere Informationen.

Ab jetzt kann die Hardware wie jede andere Grafikkarte auch konfiguriert werden. Beachten Sie, dass der Treiber agp(4) nicht nachträglich in einen laufenden Kernel geladen werden kann. Er

muss entweder fest im Kernel eingebunden sein, oder beim Systemstart über /boot/loader.conf geladen werden.

### 7.9.2. Einen Widescreen-Monitor einsetzen

Dieser Abschnitt geht über die normalen Konfigurationsarbeiten hinaus und setzt ein wenig Vorwissen voraus. Selbst wenn die Standardwerkzeuge zur X-Konfiguration bei diesen Geräten nicht zum Erfolg führen, gibt es in den Protokolldateien genug Informationen, mit denen Sie letztlich doch einen funktionierenden X-Server konfigurieren können. Alles, was Sie dazu benötigen, ist ein Texteditor.

Aktuelle Widescreen-Formate (wie WSXGA, WSXGA+, WUXGA, WXGA, WXGA+, und andere mehr) unterstützen Seitenverhältnisse wie 16:10 oder 10:9, die unter X Probleme verursachen können. Bei einem Seitenverhältnis von 16:10 sind beispielsweise folgende Auflösungen möglich:

- 2560x1600
- 1920x1200
- 1680x1050
- 1440x900
- 1280x800

Irgendwann wird die Konfiguration vereinfacht werden, dass nur noch die Auflösung als `Mode` in `Section "Screen"` eingetragen wird, so wie hier:

```
Section "Screen"
Identifier  "Screen 0"
Device      "Card 0"
Monitor     "Monitor0"
Default Depth 24
SubSection  "Display"
    ViewPort  0 0
    Depth     24
    Modes     "1680x1050"
EndSubSection
EndSection
```

Xorg ist intelligent genug, um die Informationen zu den Auflösungen über I2C/DDC zu beziehen, und weiß daher, welche Auflösungen und Frequenzen der Widescreen-Monitor unterstützt.

Wenn diese `ModeLines` in den Treiberdateien nicht vorhanden sind, kann es sein, dass Sie Xorg beim Finden der korrekten Werte unterstützen müssen. Dazu extrahieren Sie die benötigten Informationen aus /var/log/Xorg.0.log und erzeugen daraus eine funktionierende `ModeLine`. Suchen Sie nach Zeilen ähnlich den folgenden:

```
(II) MGA(0): Supported additional Video Mode:
(II) MGA(0): clock: 146.2 MHz   Image Size:  433 x 271 mm
(II) MGA(0): h_active: 1680  h_sync: 1784  h_sync_end 1960 h_blank_end 2240 h_border:
```

```
0
(II) MGA(0): v_active: 1050  v_sync: 1053  v_sync_end 1059 v_blanking: 1089 v_border:
0
(II) MGA(0): Ranges: V min: 48  V max: 85 Hz, H min: 30  H max: 94 kHz, PixClock max
170 MHz
```

Diese Informationen werden auch als EDID-Informationen bezeichnet. Um daraus eine funktionierende `ModeLine` zu erzeugen, müssen lediglich die Zahlen in die korrekte Reihenfolge gebracht werden:

```
ModeLine <name> <clock> <4 horiz. timings> <4 vert. timings>
```

Die korrekte `ModeLine` in `Section "Monitor"` würde für dieses Beispiel folgendermaßen aussehen:

```
Section "Monitor"
Identifier      "Monitor1"
VendorName      "Bigname"
ModelName       "BestModel"
ModeLine        "1680x1050" 146.2 1680 1784 1960 2240 1050 1053 1059 1089
Option          "DPMS"
EndSection
```

Nachdem diese Äderungen durchgeführt sind, sollte X auch auf Ihrem neuen Widescreen-Monitor starten.

### 7.9.3. Fehersuche in Compiz Fusion

**Ich habe Compiz Fusion installiert und anschließend die hier erwähnten Kommandos eingegeben. Nun fehlen den Fenstern die Titelleisten und Schaltflächen. Was kann ich tun?**

Wahrscheinlich fehlt eine Einstellung in /etc/X11/xorg.conf. Überprüfen Sie diese Datei gründlich, und überprüfen Sie insbesondere die Richtlinien `DefaultDepth` und `AddARGBGLXVisuals`.

**Wenn ich Compiz Fusion starte, bringt dass den X-Server zum Absturz. Was kann ich tun?**

Wenn Sie /var/log/Xorg.0.log durchsuchen, finden Sie wahrscheinlich Fehlermeldungen, die während des Starts von X ausgegeben werden. Die häufigste Meldung ist:

```
(EE) NVIDIA(0):     Failed to initialize the GLX module; please check in your X
(EE) NVIDIA(0):     log file that the GLX module has been loaded in your X
(EE) NVIDIA(0):     server, and that the module is the NVIDIA GLX module.  If
(EE) NVIDIA(0):     you continue to encounter problems, Please try
(EE) NVIDIA(0):     reinstalling the NVIDIA driver.
```

Dies ist für gewöhnlich der Fall, wenn Sie Xorg aktualisieren. Sie müssen das Paket x11/nvidia-driver neu installieren, damit GLX neu gebaut wird.

params: path: "/books/handbook/partii/" :leveloffset: +1

# Teil II: Desktop-Anwendungen

# Kapitel 8. Übersicht

Obwohl FreeBSD wegen seiner Leistung und Stabilität vor allem auf Serversystemen sehr beliebt ist, so ist es auch für den täglichen Einsatz als Desktop geeignet. Mit über 37000 Anwendungen, die als Pakete oder Ports vorliegen, ist es leicht einen individuellen Desktop zu bauen, auf dem eine Vielzahl von Desktop-Anwendungen laufen. Dieses Kapitel zeigt, wie Sie die zahlreichen Desktop-Anwendungen, wie Web-Browser, Office-Pakete, Dokumentbetrachter und Finanzsoftware, installieren können.



Benutzer die es vorziehen eine vorkonfigurierte Desktop-Version von FreeBSD zu installieren, anstatt das System von Grund auf zu konfigurieren, sollten sich FuryBSD, GhostBSD oder MidnightBSD ansehen.

Bevor Sie dieses Kapitel lesen, sollten Sie wissen:

- wie zusätzliche Anwendungen als Paket oder aus der Ports-Sammlung installiert werden. Dies wird in Installieren von Anwendungen: Pakete und Ports beschrieben.
- wie X und ein Window-Manager installiert wird. Dies wird in Das X-Window-System beschrieben.

Informationen zur Konfiguration von Multimedia-Anwendungen finden Sie in Multimedia.

# Kapitel 9. Browser

FreeBSD besitzt keinen vorinstallierten Browser, stattdessen enthält das www-Verzeichnis der Ports-Sammlung viele Browser, die als Paket oder aus der Ports-Sammlung installiert werden können.

Die Desktop-Umgebungen KDE und GNOME verfügen über eigene HTML-Browser. Weitere Informationen zur Einrichtung dieser Umgebungen finden Sie in “Grafische Oberflächen”.

Besonders schlanke Browser sind www/dillo2, www/links und www/w3m.

Dieser Abschnitt demonstriert, wie die folgenden gängigen Webbrowser installiert werden, sowie den Ressourcenbedarf, den Installationsaufwand beim Übersetzen des Ports, oder ob die Anwendung wichtige Abhängigkeiten benötigt.

| Anwendung | Ressourcenbedarf | Installationsaufwand aus den Ports | Anmerkungen |
|---|---|---|---|
| Firefox | mittel | hoch | FreeBSD, Linux® und lokalisierte Versionen sind verfügbar |
| Konqueror | mittel | hoch | Benötigt KDE-Biliotheken |
| Chromium | mittel | hoch | Benötigt Gtk+ |

## 9.1. Firefox

Firefox ist ein Open-Source Browser. Er bietet eine dem HTML-Standard konforme Anzeige, Browserfenster als Tabs, Blockierung von Pop-up-Fenstern, Erweiterungen, verbesserte Sicherheit und mehr. Firefox basiert auf der Mozilla Codebasis.

Installieren Sie das Paket der aktuellen Release-Version von Firefox:

```
# pkg install firefox
```

Um stattdessen die Extended Support Release (ESR) Version zu installieren, benutzen Sie:

```
# pkg install firefox-esr
```

Alternativ kann auch die Ports-Sammlung verwendet werden, um die gewünschte Version von Firefox aus dem Quellcode zu installieren. Dieses Beispiel baut www/firefox, wobei sich `firefox` durch die ESR oder die lokalisierte Version ersetzen lässt.

```
# cd /usr/ports/www/firefox
# make install clean
```

## 9.2. Konqueror

Konqueror ist mehr als nur ein Webbrowser, da es ebenfalls Dateimanager und Multimedia-Betrachter ist. Es unterstützt sowohl WebKit als auch sein eigenes KHTML. WebKit wird von vielen modernen Browsern verwendet, einschließlich Chromium.

Das Konqueror-Paket wird wie folgt installiert:

```
# pkg install konqueror
```

Alternativ können Sie den Port installieren:

```
# cd /usr/ports/www/konqueror
# make install clean
```

## 9.3. Chromium

Chromium ist ein quelloffenes Browserprojekt mit dem Ziel ein sicheres, schnelleres und stabileres Surferlebnis im Web zu ermöglichen. Chromium ermöglicht surfen mit Tabs, Blockieren von Pop-Ups, Erweiterungen und vieles mehr. Chromium ist das Open Source Projekt, welches auf dem Google Chrome Webbrowser basiert.

Chromium kann als Paket durch die Eingabe des folgenden Befehls installiert werden:

```
# pkg install chromium
```

Als Alternative kann Chromium aus dem Quellcode durch die Ports Collection übersetzt werden:

```
# cd /usr/ports/www/chromium
# make install clean
```



Die ausführbare Datei für Chromium ist /usr/local/bin/chrome und nicht /usr/local/bin/chromium.

# Kapitel 10. Büroanwendungen

Neue Benutzer suchen oft ein komplettes Office-Paket oder eine leicht zu bedienende Textverarbeitung. Einige graphische Oberflächen wie KDE enthalten zwar ein Office-Paket, diese werden unter FreeBSD jedoch nicht standardmäßig installiert. Unabhängig von der installierten graphischen Oberfläche können diverse Office-Pakete jederzeit installiert werden.

Dieser Abschnitt demonstriert, wie die folgenden gängigen Büroanwendungen installiert werden, sowie den Ressourcenbedarf, den Installationsaufwand beim Übersetzen des Ports, oder ob die Anwendung wichtige Abhängigkeiten benötigt.

| **Anwendung** | **Ressourcenbedarf** | **Installationsaufwand aus den Ports** | **wichtige Abhängigkeiten** |
|---|---|---|---|
| Calligra | niedrig | hoch | KDE |
| AbiWord | niedrig | niedrig | Gtk+ oder GNOME |
| The Gimp | niedrig | hoch | Gtk+ |
| Apache OpenOffice | hoch | enorm | JDK™ und Mozilla |
| LibreOffice | etwas hoch | enorm | Gtk+, KDE/ GNOME oder JDK™ |

## 10.1. Calligra

Die KDE-Gemeinschaft stellt ein Office-Paket bereit, das auch separat von KDE eingesetzt werden kann. Calligra umfasst Standardkomponenten, die auch in anderen Office-Paketen enthalten sind. Words ist die Textverarbeitung, Sheets die Tabellenkalkulation, mit Stage werden Präsentationen erstellt und Karbon ist ein Zeichenprogramm.

In FreeBSD kann editors/calligra als Paket oder Port installiert werden. Um das Paket zu installieren, geben Sie folgendes ein:

```
# pkg install calligra
```

Wenn das Paket nicht verfügbar ist, benutzen Sie stattdessen die Ports-Sammlung:

```
# cd /usr/ports/editors/calligra
# make install clean
```

## 10.2. AbiWord

AbiWord ist eine freie Textverarbeitung, die dem Erscheinungsbild von Microsoft® Word ähnlich ist. Das Programm ist schnell, besitzt viele Funktionen und ist benutzerfreundlich.

AbiWord kann viele Dateiformate importieren oder exportieren, unter anderem auch propietäre

wie Microsoft® .rtf.

Das AbiWord-Paket installieren Sie wie folgt:

```
# pkg install abiword
```

Sollte das Paket nicht zur Verfügung stehen, kann es über die Ports-Sammlung installiert werden:

```
# cd /usr/ports/editors/abiword
# make install clean
```

## 10.3. The GIMP

The GIMP ist ein ausgereiftes Bildverarbeitungsprogramm mit dem Bilder erstellt oder retuschiert werden können. Es kann sowohl als einfaches Zeichenprogramm oder zum retuschieren von Fotografien benutzt werden. Das Programm besitzt eine eingebaute Skriptsprache und es existieren sehr viele Plugins. The GIMP kann zahlreiche Formate lesen und speichern und stellt Schnittstellen zu Scannern und Tablets zur Verfügung.

Um das Paket zu installieren, geben Sie ein:

```
# pkg install gimp
```

Benutzen Sie alternativ die Ports-Sammlung:

```
# cd /usr/ports/graphics/gimp
# make install clean
```

Die Kategorie *graphics* (freebsd.org/ports/graphics.html) der Ports-Sammlung enthält für The Gimp verschiedene Plugins, Hilfedateien und Handbücher.

## 10.4. Apache OpenOffice

Apache OpenOffice ist eine Open Source Büroanwendung, die unter Leitung der Apache Software Foundation weiterentwickelt wird. Es enthält die typischen Anwendungen eines Office-Pakets: Textverarbeitung, Tabellenkalkulation, Präsentation und ein Zeichenprogramm. Die Bedienung gleicht anderen Office-Paketen und das Programm kann zahlreiche Dateiformate importieren und exportieren. Es gibt lokalisierte Versionen mit angepassten Menüs, Rechtschreibkontrollen und Wörterbüchern.

Die Textverarbeitung von Apache OpenOffice speichert Dateien im XML-Format. Dadurch wird die Verwendbarkeit der Dateien auf anderen Systemen erhöht und die Handhabung der Daten vereinfacht. Die Tabellenkalkulation besitzt eine Makrosprache und eine Schnittstelle zu Datenbanken. Apache OpenOffice läuft stabil auf Windows®, Solaris™, Linux®, FreeBSD und Mac

OS® X. Weitere Informationen über Apache OpenOffice finden Sie auf openoffice.org. Spezifische Informationen für FreeBSD finden Sie auf porting.openoffice.org/freebsd/.

Apache OpenOffice installieren Sie wie folgt:

```
# pkg install apache-openoffice
```

Nachdem das Paket installiert ist, geben Sie folgenden ein, um Apache OpenOffice zu starten:

```
% openoffice-X.Y.Z
```

wobei *X.Y.Z* die Versionsnummer von Apache OpenOffice darstellt. Nach dem ersten Start werden einige Fragen gestellt. Außerdem wird im Heimatverzeichnis des Benutzers ein Verzeichnis .openoffice.org angelegt.

Falls das gewünschte Apache OpenOffice-Paket nicht verfügbar ist, kann immer noch der Port übersetzt werden. Es erfordert jedoch eine Menge Plattenplatz und ziemlich viel Zeit um die Quellen zu übersetzten.

```
# cd /usr/ports/editors/openoffice-4
# make install clean
```



Um eine lokalisierte Version zu bauen, ersetzen Sie den letzten Befehl durch:

```
# make LOCALIZED_LANG=Ihre_Sprache install clean
```

Ersetzen Sie *Ihre_Sprache* durch den korrekten ISO-Code. Eine Liste der unterstützten Codes steht in files/Makefile.localized, die sich im Portsverzeichnis befindet.

## 10.5. LibreOffice

LibreOffice ist ein frei verfügbares Office-Paket, welches von documentfoundation.org entwickelt wird. Es mit anderen großen Office-Paketen kompatibel und für eine Vielzahl von Plattformen erhältlich. Es ist ein Fork von Apache OpenOffice unter neuem Namen, das alle Anwendungen in einem kompletten Office-Paket enthält: Textverarbeitung, Tabellenkalkulation, Präsentationsmanager, Zeichenprogramm, Datenbankmanagementprogramm und ein Werkzeug zum Erstellen und Bearbeiten von mathematischen Formeln. Das Programm steht in verschiedenen Sprachen zur Verfügung, und die Internationalisierung wurde auf die Oberfläche, Rechtschreibkorrektur und die Wörterbücher ausgeweitet.

Das Textverarbeitungsprogramm von LibreOffice benutzt ein natives XML-Dateiformat für erhöhte Portabilität und Flexibilität. Die Tabellenkalkulation enthält eine Makrosprache und kann mit externen Datenbanken Verbindungen herstellen. LibreOffice ist stabil und läuft nativ auf

Windows®, Linux®, FreeBSD und Mac OS® X. Weitere Informationen zu LibreOffice finden Sie unter libreoffice.org.

Um die englische Version von LibreOffice als Paket zu installieren, geben Sie folgenden Befehl ein:

```
# pkg install libreoffice
```

Die Kategorie *editors* (freebsd.org/ports/) der Ports-Sammlung enthält viele Lokalisierungen für LibreOffice. Wenn Sie ein lokalisiertes Paket installieren, ersetzen Sie `libreoffice` durch den Namen des lokalisierten Pakets.

Wenn das Paket installiert ist, geben Sie folgendes Kommando ein, um LibreOffice zu starten:

```
% libreoffice
```

Während des ersten Starts werden einige Fragen gestellt. Außerdem wird im Heimatverzeichinis des Benutzers ein Verzeichnis .libreoffice angelegt.

Falls das gewünschte LibreOffice-Paket nicht verfügbar ist, kann immer noch der Port übersetzt werden. Es erfordert jedoch eine Menge Plattenplatz und ziemlich viel Zeit um die Quellen zu übersetzten. Dieses Beispiel übersetzt die englische Version:

```
# cd /usr/ports/editors/libreoffice
# make install clean
```



Um eine lokalisierte Version zu bauen, wechseln Sie mit `cd` in das Portverzeichnis der gewünschten Sprache. Unterstützte Sprachen finden Sie in der Kategorie *editors* ( freebsd.org/ports/) der Ports-Sammlung.

# Kapitel 11. Anzeigen von Dokumenten

Einige neuere Dokumentformate, die sich aktuell großer Beliebtheit erfreuen, können Sie sich mit den im Basissystem enthaltenen Programmen möglicherweise nicht ansehen. Dieser Abschnitt zeigt, wie Sie die folgenden Dokumentbetrachter installieren können:

Die nachstehenden Anwendungen werden behandelt:

| Anwendung | Ressourcenbedarf | Installationsaufwand aus den Ports | wichtige Abhängigkeiten |
|---|---|---|---|
| Xpdf | niedrig | niedrig | FreeType |
| gv | niedrig | niedrig | Xaw3d |
| Geeqie | niedrig | niedrig | Gtk+ oder GNOME |
| ePDFView | niedrig | niedrig | Gtk+ |
| Okular | niedrig | hoch | KDE |

## 11.1. Xpdf

Für Benutzer, die einen schnellen PDF-Betrachter bevorzugen, bietet Xpdf eine schlanke und effiziente Alternative, die wenig Ressourcen benötigt. Da das Programm die Standard X-Zeichensätze benutzt, ist es nicht auf andere Toolkits angewiesen.

Um das Xpdf-Paket zu installieren, geben Sie folgendes ein:

```
# pkg install xpdf
```

Wenn das Paket nicht verfügbar ist, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/graphics/xpdf
# make install clean
```

Starten Sie nach der Installation `xpdf` und aktivieren Sie das Menü mit der rechten Maustaste.

## 11.2. gv

gv kann PostScript®- und PDF-Dokumente anzeigen. Es stammt von ghostview ab, hat aber wegen der Xaw3d-Bibliothek eine schönere Benutzeroberfläche. gv besitzt viele konfigurierbare Funktionen, wie z. B. Ausrichtung, Papiergröße, Skalierung und Kantenglättung (Anti-Aliasing). Fast jede Operation kann sowohl mit der Tastatur als auch mit der Maus durchgeführt werden.

Installieren Sie das gv-Paket wie folgt:

```
# pkg install gv
```

Benutzen Sie die Ports-Sammlung, wenn das Paket nicht zur Verfügung steht:

```
# cd /usr/ports/print/gv
# make install clean
```

## 11.3. Geeqie

Geeqie ist ein Fork des nicht mehr betreuten GQview Projekts, mit dem Ziel die Entwicklung weiter voranzutreiben und bestehende Fehlerkorrekturen zu integrieren. Mit Geeqie lassen sich Bilder verwalten. Es kann unter anderem Bilder anzeigen, einen externen Editor starten und eine Vorschau (thumbnail) erzeugen. Zudem beherrscht Geeqie einen Diashow-Modus und einige grundlegende Dateioperationen, was die Verwaltung von Bildern und das auffinden von doppelten Dateien erleichtert. Geeqie unterstützt Vollbild-Ansicht und Internationalisierung.

Um das Geeqie-Paket zu installieren, geben Sie folgendes ein:

```
# pkg install geeqie
```

Wenn das Paket nicht verfügbar ist, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/graphics/geeqie
# make install clean
```

## 11.4. ePDFView

ePDFView ist ein leichtgewichtiger PDF-Betrachter, der nur die Gtk+- und Poppler-Bibliotheken benötigt. Es befindet sich derzeit noch in Entwicklung, kann aber bereits die meisten PDF-Dateien (auch verschlüsselte) öffnen, speichern und über CUPS drucken.

Um das Paket ePDFView zu installieren, geben Sie folgendes ein:

```
# pkg install epdfview
```

Benutzen Sie die Ports-Sammlung, falls das Paket nicht verfügbar ist:

```
# cd /usr/ports/graphics/epdfview
# make install clean
```

## 11.5. Okular

Okular ist ein universeller Dokumentbetrachter der auf KPDF für KDE basiert. Es kann die meisten Formate öffnen, einschließlich PDF, PostScript®, DjVu, CHM, XPS und ePub.

Um das Paket Okular zu installieren, geben Sie folgendes ein:

```
# pkg install okular
```

Benutzen Sie die Ports-Sammlung, falls das Paket nicht verfügbar ist:

```
# cd /usr/ports/graphics/okular
# make install clean
```

# Kapitel 12. Finanzsoftware

Zur Verwaltung der persönlichen Finanzen können einige leistungsfähige und einfach zu bedienende Anwendungen installiert werden. Einige von ihnen unterstützen verbreitete Formate, darunter Dateiformate, die von Quicken und Excel verwendet werden.

Dieser Abschnitt behandelt die folgenden Anwendungen:

| Anwendung | Ressourcenbedarf | Installationsaufwand aus den Ports | wichtige Abhängigkeiten |
|---|---|---|---|
| GnuCash | niedrig | hoch | GNOME |
| Gnumeric | niedrig | hoch | GNOME |
| KMyMoney | niedrig | hoch | KDE |

## 12.1. GnuCash

GnuCash ist Teil des GNOME-Projekts, mit dem Ziel, leicht zu bedienende und leistungsfähige Anwendungen bereitzustellen. Mit GnuCash können Einnahmen und Ausgaben, Bankkonten und Wertpapiere verwaltet werden. Das Programm ist leicht zu bedienen und genügt dennoch hohen Ansprüchen.

GnuCash stellt ein Register, ähnlich dem in einem Scheckheft und ein hierarchisches System von Konten zur Verfügung. Eine Transaktion kann in einzelne Teile aufgespaltet werden. GnuCash kann Quicken-Dateien (QIF) importieren und einbinden. Weiterhin unterstützt das Programm die meisten internationalen Formate für Zeitangaben und Währungen. Die Bedienung des Programms kann durch zahlreiche Tastenkombinationen und dem automatischen Vervollständigen von Eingaben beschleunigt werden.

Das GnuCash-Paket installieren Sie wie folgt:

```
# pkg install gnucash
```

Wenn das Paket nicht zur Verfügung steht, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/finance/gnucash
# make install clean
```

## 12.2. Gnumeric

Gnumeric ist eine Tabellenkalkulation, die von der GNOME-Gemeinschaft entwickelt wird. Das Programm kann Eingaben anhand des Zellenformats oder einer Folge von Eingaben vervollständigen. Dateien verbreiteter Formate, wie die von Excel, Lotus 1-2-3 oder Quattro Pro lassen sich importieren. Es besitzt viele eingebaute Funktionen und Zellenformate, darunter die üblichen wie Zahl, Währung, Datum, Zeit, und viele weitere.

Installieren Sie das Gnumeric-Paket mit folgendem Kommando:

```
# pkg install gnumeric
```

Wenn das Paket nicht zur Verfügung steht, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/math/gnumeric
# make install clean
```

## 12.3. KMyMoney

KMyMoney ist ein Programm zur Verwaltung der persönlichen Finanzen, das von der KDE-Gemeinschaft entwickelt wird. KMyMoney hat das Ziel, wichtige Funktionen zu bieten, die auch von kommerziellen Programmen zur Verwaltung der persönlichen Finanzen unterstützt werden. Zudem zählen eine einfache Bedienung sowie korrekte doppelte Buchführung zu den herausragenden Fähigkeiten dieses Programms. KMyMoney unterstützt den Import von Datendateien im Format Quicken (QIF), kann Investionen verfolgen, unterstützt verschiedene Währungen und bietet umfangreiche Reportmöglichkeiten.

Um das Paket KMyMoney zu installieren, geben Sie folgendes ein:

```
# pkg install kmymoney-kde4
```

Sollte das Paket nicht verfügbar sein, benutzen Sie die Ports-Sammlung:

```
# cd /usr/ports/finance/kmymoney2-kde4
# make install clean
```

# Kapitel 13. Multimedia

## 13.1. Übersicht

FreeBSD unterstützt viele unterschiedliche Soundkarten, die Benutzern den Genuss von Highfidelity-Klängen auf dem Computer ermöglichen. Dazu gehört unter anderem die Möglichkeit, Tonquellen in den Formaten MPEG Audio Layer 3 (MP3), Waveform Audio File (WAV), Ogg Vorbis und vielen weiteren Formaten aufzunehmen und wiederzugeben. Darüber hinaus enthält die FreeBSD Ports-Sammlung Anwendungen, die das Bearbeiten von aufgenommenen Tonspuren, das Hinzufügen von Klangeffekten und die Kontrolle der angeschlossenen MIDI-Geräte erlauben.

FreeBSD unterstützt auch die Wiedergabe von Videos und DVDs. Die FreeBSD Ports-Sammlung enthält Anwendungen, um verschiedene Video-Medien wiederzugeben, zu kodieren und zu konvertieren.

Dieses Kapitel beschreibt die Einrichtung von Soundkarten, Video-Wiedergabe, TV-Tuner Karten und Scannern unter FreeBSD. Es werden auch einige Anwendungen beschrieben, die für die Verwendung dieser Geräte zur Verfügung stehen.

Dieses Kapitel behandelt die folgenden Punkte:

- Konfiguration einer Soundkarte in FreeBSD.
- Fehlersuche bei Sound Einstellungen.
- Wiedergabe und Kodierung von MP3s und anderen Audio-Formaten.
- Vorbereitung des Systems für die Wiedergabe von Videos.
- Wiedergabe von DVDs, .mpg- und .avi-Dateien.
- Rippen von CDs und DVDs.
- Konfiguration von TV-Karten.
- Installation und Konfiguration von MythTV.
- Konfiguration von Scannern
- Konfiguration von Bluetooth-Kopfhörern

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Wissen, wie Sie Anwendungen installieren (Installieren von Anwendungen: Pakete und Ports).

## 13.2. Soundkarten einrichten

Bevor Sie die Konfiguration beginnen, sollten Sie in Erfahrung bringen welches Soundkartenmodell und welcher Chip benutzt wird. FreeBSD unterstützt eine Reihe Soundkarten. Die Hardware-Notes zählen alle unterstützten Karten und deren Treiber für FreeBSD auf.

Um die Soundkarte benutzen zu können, muss der richtige Gerätetreiber geladen werden. Am einfachsten ist es, das Kernelmodul für die Soundkarte mit kldload(8) zu laden. Dieses Beispiel lädt

den Treiber für einen integrierten Chipsatz, basierend auf der Intel Spezifikation:

```
# kldload snd_hda
```

Um den Treiber automatisch beim Systemstart zu laden, fügen Sie folgende Zeile in /boot/loader.conf ein:

```
snd_hda_load="YES"
```

Weitere ladbare Soundmodule sind in /boot/defaults/loader.conf aufgeführt. Wenn Sie nicht sicher sind, welchen Gerätetreiber Sie laden müssen, laden Sie das Modul snd_driver:

```
# kldload snd_driver
```

Der Treiber snd_driver ist ein Meta-Treiber, der alle gebräuchlichen Treiber lädt und die Suche nach dem richtigen Treiber vereinfacht. Durch Hinzufügen des Meta-Treibers in /boot/loader.conf können alternativ alle Audio-Treiber geladen werden.

Um zu ermitteln, welcher Treiber für die Soundkarte vom Meta-Treiber snd_driver geladen wurde, geben Sie `cat /dev/sndstat` ein.

### 13.2.1. Soundkarten in der Kernelkonfiguration einrichten

Die Unterstützung für die Soundkarte kann auch direkt in den Kernel kompiliert werden. Weitere Informationen über den Bau eines Kernels finden Sie im Konfiguration des FreeBSD-Kernels.

Bei der Verwendung eines eigenen Kernels müssen Sie sicherstellen, dass der Treiber für das Audio-Framework in der Kernelkonfigurationsdatei vorhanden ist:

```
device sound
```

Als Nächstes muss die Unterstützung für die Soundkarte hinzugefügt werden. Um das Beispiel mit dem integrierten Intel Audio-Chipsatz aus dem vorherigen Abschnitt fortzusetzen, verwenden Sie die folgende Zeile in der Kernelkonfigurationsdatei:

```
device snd_hda
```

Lesen Sie die Manualpage des Treibers, um den entsprechenden Gerätenamen herauszufinden.

Nicht PnP-fähige ISA-Soundkarten benötigen eventuell Einstellungen, wie IRQ und I/O-Port in /boot/device.hints. Während des Systemstarts liest der loader(8) diese Datei und reicht die Einstellungen an den Kernel weiter. Für eine alte Creative SoundBlaster® 16 ISA-Karte, die sowohl den snd_sbc(4)- als auch den `snd_sb16`-Treiber benötigt, müssen die folgenden Zeilen in die Kernelkonfigurationsdatei eingetragen werden:

```
device snd_sbc
device snd_sb16
```

Wenn die Karte den I/O-Port `0x220` und IRQ `5` benutzt, müssen folgende Zeilen zusätzlich in /boot/device.hints hinzugefügt werden:

```
hint.sbc.0.at="isa"
hint.sbc.0.port="0x220"
hint.sbc.0.irq="5"
hint.sbc.0.drq="1"
hint.sbc.0.flags="0x15"
```

Die Syntax für /boot/device.hints wird in sound(4), sowie in der Manualpage des jeweiligen Treibers beschrieben.

Das Beispiel verwendet die vorgegebenen Werte. Falls die Karteneinstellungen andere Werte vorgeben, müssen die Werte in der Kernelkonfiguration angepasst werden. Weitere Informationen zu dieser Soundkarte finden Sie in snd_sbc(4).

### 13.2.2. Die Soundkarte testen

Nachdem Sie den neuen Kernel gestartet oder das erforderliche Modul geladen haben, sollte die Soundkarte erkannt werden. Führen Sie `dmesg | grep pcm` aus, um dies zu überprüfen. Diese Ausgabe stammt von einem System mit einem integrierten Conexant CX20590 Chipsatz:

```
pcm0: <NVIDIA (0x001c) (HDMI/DP 8ch)> at nid 5 on hdaa0
pcm1: <NVIDIA (0x001c) (HDMI/DP 8ch)> at nid 6 on hdaa0
pcm2: <Conexant CX20590 (Analog 2.0+HP/2.0)> at nid 31,25 and 35,27 on hdaa1
```

Der Status der Karte kann auch mit diesem Kommando geprüft werden:

```
# cat /dev/sndstat
FreeBSD Audio Driver (newpcm: 64bit 2009061500/amd64)
Installed devices:
pcm0: <NVIDIA (0x001c) (HDMI/DP 8ch)> (play)
pcm1: <NVIDIA (0x001c) (HDMI/DP 8ch)> (play)
pcm2: <Conexant CX20590 (Analog 2.0+HP/2.0)> (play/rec) default
```

Die Ausgabe kann für jede Soundkarte anders aussehen. Wenn das Gerät pcm nicht erscheint, prüfen Sie die Kernelkonfigurationsdatei und stellen Sie sicher, dass der richtige Treiber geladen oder in den Kernel kompiliert wurde. Im nächsten Abschnitt werden häufig auftretende Probleme sowie deren Lösungen besprochen.

Jetzt sollte die Soundkarte unter FreeBSD funktionieren. Wenn ein CD- oder DVD-Laufwerk an die Soundkarte angeschlossen ist, können Sie jetzt mit cdcontrol(1) eine CD abspielen:

```
% cdcontrol -f /dev/acd0 play 1
```



Audio CDs besitzen eine spezielle Kodierung. Daher sollten sie nicht mit mount(8) in das Dateisystem eingehangen werden.

Es gibt viele Anwendungen, wie audio/workman, die eine bessere Benutzerschnittstelle besitzen. Zur Wiedergabe von MP3-Audiodateien kann audio/mpg123 installiert werden.

Eine weitere schnelle Möglichkeit die Karte zu prüfen, ist es, Daten an das Gerät /dev/dsp zu senden:

```
% cat Datei > /dev/dsp
```

Für Datei kann eine beliebige Datei verwendet werden. Wenn Sie einige Geräusche hören, funktioniert die Soundkarte.



Die Gerätedateien /dev/dsp* werden automatisch erzeugt, wenn sie das erste Mal benötigt werden. Werden sie nicht verwendet, sind sie hingegen nicht vorhanden und tauchen daher auch nicht in der Ausgabe von ls(1) auf.

### 13.2.3. Konfiguration von Bluetooth-Soundgeräten

Die Verbindung zu einem Bluetooth-Gerät wird in diesem Abschnitt nicht erläutert. Dazu finden Sie weitere Informationen in “Bluetooth”.

Damit Bluetooth zusammen mit dem Soundsystem von FreeBSD funktioniert, müssen Benutzer zuerst audio/virtual_oss installieren:

```
# pkg install virtual_oss
```

audio/virtual_oss setzt voraus, dass cuse in den Kernel geladen wird:

```
# kldload cuse
```

Führen Sie folgenden Befehl aus, damit cuse beim Systemstart automatisch geladen wird:

```
# sysrc -f /boot/loader.conf cuse_load=yes
```

Um Kopfhörer mit audio/virtual_oss zu benutzten, muss nach der Verbindung mit einem Bluetooth-Audiogerät ein virtuelles Gerät erstellt werden:

```
# virtual_oss -C 2 -c 2 -r 48000 -b 16 -s 768 -R /dev/null -P
```

```
/dev/bluetooth/headphones -d dsp
```



`headphones` ist in diesem Beispiel ein Hostname aus /etc/bluetooth/hosts. Stattdessen kann auch `BT_ADDR` verwendet werden.

Weitere Informationen finden Sie in virtual_oss(8).

### 13.2.4. Fehlerbehebung

Typische Fehlermeldungen zeigt typische Fehlermeldungen sowie deren Lösungen:

*Tabelle 7. Typische Fehlermeldungen*

| Fehler | Lösung |
|---|---|
| `sb_dspwr(XX) timed out` | Der I/O-Port ist nicht korrekt angegeben. |
| `bad irq XX` | Der IRQ ist falsch angegeben. Stellen Sie sicher, dass der angegebene IRQ mit dem Sound IRQ übereinstimmt. |
| `xxx: gus pcm not attached, out of memory` | Es ist nicht genug Speicher verfügbar, um das Gerät zu betreiben. |
| `xxx: can't open /dev/dsp!` | Überprüfen Sie mit `fstat \| grep dsp` ob eine andere Anwendung das Gerät geöffnet hat. Häufige Störenfriede sind esound oder die Sound-Unterstützung von KDE. |

Moderne Grafikkarten beinhalten oft auch ihre eigenen Soundtreiber, um HDMI zu verwenden. Diese Audiogeräte werden manchmal vor der eigentlichen, separaten Soundkarte aufgeführt und dadurch nicht als das Standardgerät zum Abspielen von Tönen benutzt. Um zu prüfen, ob das der Fall ist, führen Sie dmesg aus und suchen Sie nach der Zeichenfolge `pcm`. Die Ausgabe sieht in etwa so aus:

```
...
hdac0: HDA Driver Revision: 20100226_0142
hdac1: HDA Driver Revision: 20100226_0142
hdac0: HDA Codec #0: NVidia (Unknown)
hdac0: HDA Codec #1: NVidia (Unknown)
hdac0: HDA Codec #2: NVidia (Unknown)
hdac0: HDA Codec #3: NVidia (Unknown)
pcm0: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 0 nid 1 on hdac0
pcm1: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 1 nid 1 on hdac0
pcm2: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 2 nid 1 on hdac0
pcm3: <HDA NVidia (Unknown) PCM #0 DisplayPort> at cad 3 nid 1 on hdac0
hdac1: HDA Codec #2: Realtek ALC889
pcm4: <HDA Realtek ALC889 PCM #0 Analog> at cad 2 nid 1 on hdac1
pcm5: <HDA Realtek ALC889 PCM #1 Analog> at cad 2 nid 1 on hdac1
pcm6: <HDA Realtek ALC889 PCM #2 Digital> at cad 2 nid 1 on hdac1
pcm7: <HDA Realtek ALC889 PCM #3 Digital> at cad 2 nid 1 on hdac1
...
```

In diesem Beispiel wurde die Grafikkarte (`NVidia`) vor der Soundkarte (`Realtek ALC889`) aufgeführt. Um die Soundkarte als Standardabspielgerät einzusetzen, ändern Sie `hw.snd.default_unit` auf die Einheit, welche für das Abspielen benutzt werden soll:

```
# sysctl hw.snd.default_unit=n
```

Hier repräsentiert `n` die Nummer der Soundkarte, die verwendet werden soll, in diesem Beispiel also `4`. Sie können diese Änderung dauerhaft machen, indem Sie die folgende Zeile in /etc/sysctl.conf hinzufügen:

```
hw.snd.default_unit=4
```

### 13.2.5. Mehrere Tonquellen abspielen

Oft sollen mehrere Tonquellen gleichzeitig abgespielt werden. FreeBSD verwendet dazu *virtuelle Tonkanäle*. Virtuelle Kanäle mischen die Tonquellen im Kernel, sodass mehrere Kanäle benutzt werden können, als von der Hardware unterstützt werden.

Drei sysctl(8) Optionen stehen zur Konfiguration der virtuellen Kanäle zur Verfügung:

```
# sysctl dev.pcm.0.play.vchans=4
# sysctl dev.pcm.0.rec.vchans=4
# sysctl hw.snd.maxautovchans=4
```

Im Beispiel werden vier virtuelle Kanäle eingerichtet, eine im Normalfall ausreichende Anzahl. Sowohl `dev.pcm.0.play.vchans=4` und `dev.pcm.0.rec.vchans=4` sind die Anzahl der virtuellen Kanäle des Geräts pcm0, die fürs Abspielen und Aufnehmen verwendet werden und sie können konfiguriert werden, sobald das Gerät existiert. Da das Modul pcm unabhängig von den Hardware-Treibern geladen werden kann, gibt `hw.snd.maxautovchans` die Anzahl der virtuellen Kanäle an, die später eingerichtete Audiogeräte erhalten. Lesen Sie pcm(4) für weitere Informationen.



Die Anzahl der virtuellen Kanäle kann nicht geändert werden, solange das Gerät genutzt wird. Schließen Sie daher zuerst alle Programme wie Musikabspielprogramme oder Sound-Daemonen, die auf dieses Gerät zugreifen.

Die korrekte pcm-Gerätedatei wird automatisch zugeteilt, wenn ein Programm das Gerät /dev/dsp0 anfordert.

### 13.2.6. Den Mixer einstellen

Die Voreinstellungen des Mixers sind im Treiber pcm(4) fest kodiert. Es gibt zwar viele Anwendungen und Dienste, die den Mixer einstellen können und die eingestellten Werte bei jedem Start wieder setzen, am einfachsten ist es allerdings, die Standardwerte für den Mixer direkt im Treiber einzustellen. Der Mixer kann mit den entsprechenden Werten in /boot/device.hints eingestellt werden:

```
hint.pcm.0.vol="50"
```

Die Zeile setzt die Lautstärke des Mixers beim Laden des Moduls pcm(4) auf den Wert 50.

# 13.3. MP3-Audio

Dieser Abschnitt beschreibt einige unter FreeBSD verfügbare MP3-Player. Zudem wird beschrieben, wie Audio-CDs gerippt und MP3s kodiert und dekodiert werden.

## 13.3.1. MP3-Player

Ein beliebter graphischer MP3-Player ist Audacious, welcher WinAmp-Skins und zusätzliche Plugins unterstützt. Die Benutzerschnittstelle ist leicht zu erlernen und enthält eine Playlist, einen graphischen Equalizer und vieles mehr. Diejenigen, die bereits mit WinAmp vertraut sind, werden Audacious sehr leicht zu benutzen finden. Unter FreeBSD kann Audacious als Port oder Paket multimedia/audacious installiert werden. Audacious ist ein Ableger von XMMS.

Das Paket audio/mpg123 ist ein alternativer, kommandozeilenorientierter MP3-Player. Nach der Installation kann die abzuspielende MP3-Datei auf der Kommandozeile angegeben werden. Geben Sie auch das entsprechende Soundkarte an, falls das System über mehrere Audiogeräte verfügt:

```
# mpg123 -a /dev/dsp1.0 Foobar-GreatestHits.mp3
High Performance MPEG 1.0/2.0/2.5 Audio Player for Layer 1, 2 and 3
        version 1.18.1; written and copyright by Michael Hipp and others
        free software (LGPL) without any warranty but with best wishes

Playing MPEG stream from Foobar-GreatestHits.mp3 ...
MPEG 1.0 layer III, 128 kbit/s, 44100 Hz joint-stereo
```

Weitere MP3-Player stehen in der FreeBSD Ports-Sammlung zur Verfügung.

## 13.3.2. CD-Audio Tracks rippen

Bevor eine ganze CD oder einen CD-Track in das MP3-Format umgewandelt werden kann, müssen die Audiodaten von der CD auf die Festplatte gerippt werden. Dabei werden die CDDA (CD Digital Audio) Rohdaten in WAV-Dateien kopiert.

Die Anwendung `cdda2wav`, die im sysutils/cdrtools Paket enthalten ist, kann zum Rippen der Audiodaten von CDs genutzt werden.

Wenn die Audio CD in dem Laufwerk liegt, kann der folgende Befehl als `root` ausgeführt werden, um eine ganze CD in einzelne WAV-Dateien zu rippen:

```
# cdda2wav -D 0,1,0 -B
```

In diesem Beispiel bezieht sich der Schalter `-D 0,1,0` auf das SCSI-Gerät 0,1,0, das die zu rippende

CD enthält. Benutzen Sie `cdrecord -scanbus` um die richtigen Geräteparameter für das System zu bestimmen.

Um einzelne Tracks zu rippen, benutzen Sie `-t` wie folgt:

```
# cdda2wav -D 0,1,0 -t 7
```

Um mehrere Tracks zu rippen, zum Beispiel die Tracks eins bis sieben, können Sie wie folgt einen Bereich angeben:

```
# cdda2wav -D 0,1,0 -t 1+7
```

Wenn Sie von einem ATAPI (IDE) CD-ROM-Laufwerk rippen, geben Sie den Gerätenamen anstelle der SCSI-Gerätenummer an. Dieses Beispiel rippt Track 7 von einem IDE-Laufwerk:

```
# cdda2wav -D /dev/acd0 -t 7
```

Alternativ können mit `dd` ebenfalls Audio-Stücke von ATAPI-Laufwerken kopiert werden. Dies wird in “Kopieren von Audio-CDs” erläutert.

### 13.3.3. MP3-Dateien kodieren und dekodieren

Lame ist ein weitverbreiteter MP3-Encoder, der als Port audio/lame installiert werden kann. Wegen Patentproblemen ist kein Paket verfügbar.

Der folgende Befehl konvertiert die gerippte WAV-Datei audio01.wav in audio01.mp3 um:

```
# lame -h -b 128 --tt "Foo Liedtietel" --ta "FooBar Künstler" --tl "FooBar Album" \
--ty "2014" --tc "Gerippt und kodiert von Foo" --tg "Musikrichtung" audio01.wav
audio01.mp3
```

128 kbits ist die gewöhnliche MP3-Bitrate, wohingegen die Bitraten 160 und 192 kbits eine höhere Qualität bieten. Je höher die Bitrate ist, desto mehr Speicherplatz benötigt die resultierende MP3-Datei. Die Option `-h` verwendet den "higher quality but a little slower" (höhere Qualität, aber etwas langsamer) Modus. Die Schalter, die mit `--t` beginnen, sind ID3-Tags, die in der Regel Informationen über das Lied enthalten und in die MP3-Datei eingebettet sind. Weitere Optionen können in der Manualpage von lame nachgelesen werden.

Um aus MP3-Dateien eine Audio CD zu erstellen, müssen diese zuerst in ein nicht komprimiertes Format umgewandelt werden. Verwenden Sie XMMS um die Datei im WAV-Format zu schreiben und mpg123, um die MP3-Datei in rohe PCM-Audiodaten umzuwandeln.

Um audio01.mp3 mit mpg123 umzuwandeln, geben Sie den Namen der PCM-Datei an:

```
# mpg123 -s audio01.mp3 > audio01.pcm
```

So verwenden Sie XMMS um eine MP3-Datei in das WAV-Format zu konvertieren:

**Procedure: Mit XMMS in das WAV-Format konvertieren** . Starten Sie XMMS. . Klicken Sie mit der rechten Maustaste, um das XMMS-Menu zu öffnen. . Wählen Sie `Preferences` im Untermenü `Options`. . Ändern Sie das Output-Plugin in "Disk Writer Plugin". . Drücken Sie `Configure`. . Geben Sie ein Verzeichnis ein, in das Sie die unkomprimierte Datei schreiben wollen. . Laden Sie die MP3-Datei wie gewohnt in XMMS mit einer Lautstärke von 100% und einem abgeschalteten EQ. . Drücken Sie `Play` und es wird so aussehen, als spiele XMMS die MP3-Datei ab, aber keine Musik ist zu hören. Der Player überspielt die MP3-Datei in eine Datei. . Vergessen Sie nicht, das Output-Plugin wieder in den Ausgangszustand zurückzusetzen um wieder MP3-Dateien anhören zu können.

cdrecord kann mit beiden Formaten Audio-CDs erstellen. Der Dateikopf von WAV-Dateien erzeugt am Anfang des Stücks ein Knacken. Der Dateikopf mit dem Port oder Paket audio/sox entfernt werden:

```
% sox -t wav -r 44100 -s -w -c 2 track.wav track.raw
```

Lesen Sie “Erstellen und Verwenden von CDs”, um mehr Informationen zur Benutzung von CD-Brennern mit FreeBSD zu erhalten.

## 13.4. Videos wiedergeben

Bevor Sie beginnen, sollten Sie das Modell und den benutzten Chip der Videokarte kennen. Obwohl Xorg viele Videokarten unterstützt, können nicht alle Karten Videos schnell genug wiedergeben. Eine Liste der Erweiterungen, die der Xorg-Server für eine Videokarte unterstützt, erhalten Sie unter laufendem Xorg mit `xdpyinfo`.

Halten Sie eine kurze MPEG-Datei bereit, mit der Sie Wiedergabeprogramme und deren Optionen testen können. Da einige DVD-Spieler in der Voreinstellung das DVD-Gerät mit /dev/dvd ansprechen oder diesen Namen fest einkodiert haben, ist es vielleicht hilfreich symbolische Links auf die richtigen Geräte anzulegen:

```
# ln -sf /dev/acd0 /dev/dvd
```

Aufgrund der Beschaffenheit devfs(5) gehen gesondert angelegte Links wie diese bei einem Neustart des Systems verloren. Damit die symbolischen Links automatisch beim Neustart des Systems angelegt werden, fügen Sie die folgende Zeile in /etc/devfs.conf ein:

```
link acd0 dvd
```

Das Entschlüsseln von DVDs erfordert den Aufruf bestimmter Funktionen, sowie Schreibzugriff auf das DVD-Gerät.

Xorg benutzt Shared-Memory und es wird empfohlen, die nachstehenden sysctl(8)-Variablen auf die gezeigten Werte zu erhöhen:

```
kern.ipc.shmmax=67108864
kern.ipc.shmall=32768
```

### 13.4.1. Video-Schnittstellen

Es gibt einige Möglichkeiten, Videos unter Xorg abzuspielen. Welche Möglichkeit funktioniert, hängt stark von der verwendeten Hardware ab.

Gebräuchliche Video-Schnittstellen sind:

1. Xorg: normale Ausgabe über Shared-Memory.
2. XVideo: Eine Erweiterung der Xorg-Schnittstelle, die Videos in jedem X11-Drawable anzeigen kann. Diese Erweiterung bietet auch auf leistungsschwachen Maschinen eine gute Qualität der Wiedergabe. Der nächste Abschnitt beschreibt, wie Sie feststellen, ob diese Erweiterung ausgeführt wird.
3. SDL: Simple DirectMedia Layer ist eine portable Schnittstelle für verschiedene Betriebssysteme, mit denen Anwendungen plattformunabhängig und effizient Ton und Grafik benutzen können. SDL bietet eine hardwarenahe Schnittstelle, die manchmal schneller ist als die Xorg-Schnittstelle. Unter FreeBSD kann SDL über das Paket oder den Port devel/sdl20 installiert werden.
4. DGA: Direct Graphics Access ist eine Xorg-Erweiterung die es Anwendungen erlaubt, am Xorg-Server vorbei direkt in den Framebuffer zu schreiben. Da die Anwendung und der Xorg-Server auf gemeinsame Speicherbereiche zugreifen, müssen die Anwendungen unter dem Benutzer `root` laufen. Die DGA-Erweiterung kann mit dga(1) getestet werden. Wenn DGA ausgeführt wird, ändert sich die Farbe des Bildschrims, wenn eine Taste gedrückt wird. Drücken Sie zum Beenden `q`.
5. SVGAlib: Eine Schnittstelle zur Grafikausgabe auf der Konsole.

#### XVideo

Ob die Erweiterung läuft, entnehmen Sie der Ausgabe von `xvinfo`:

```
% xvinfo
```

XVideo wird untertsützt, wenn die Ausgabe in etwa wie folgt aussieht:

```
X-Video Extension version 2.2
screen #0
  Adaptor #0: "Savage Streams Engine"
    number of ports: 1
    port base: 43
    operations supported: PutImage
```

```
    supported visuals:
      depth 16, visualID 0x22
      depth 16, visualID 0x23
    number of attributes: 5
      "XV_COLORKEY" (range 0 to 16777215)
              client settable attribute
              client gettable attribute (current value is 2110)
      "XV_BRIGHTNESS" (range -128 to 127)
              client settable attribute
              client gettable attribute (current value is 0)
      "XV_CONTRAST" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_SATURATION" (range 0 to 255)
              client settable attribute
              client gettable attribute (current value is 128)
      "XV_HUE" (range -180 to 180)
              client settable attribute
              client gettable attribute (current value is 0)
    maximum XvImage size: 1024 x 1024
    Number of image formats: 7
      id: 0x32595559 (YUY2)
        guid: 59555932-0000-0010-8000-00aa00389b71
        bits per pixel: 16
        number of planes: 1
        type: YUV (packed)
      id: 0x32315659 (YV12)
        guid: 59563132-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x30323449 (I420)
        guid: 49343230-0000-0010-8000-00aa00389b71
        bits per pixel: 12
        number of planes: 3
        type: YUV (planar)
      id: 0x36315652 (RV16)
        guid: 52563135-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x3e0, 0x7c00
      id: 0x35315652 (RV15)
        guid: 52563136-0000-0000-0000-000000000000
        bits per pixel: 16
        number of planes: 1
        type: RGB (packed)
        depth: 0
        red, green, blue masks: 0x1f, 0x7e0, 0xf800
      id: 0x31313259 (Y211)
```

```
        guid: 59323131-0000-0010-8000-00aa00389b71
        bits per pixel: 6
        number of planes: 3
        type: YUV (packed)
      id: 0x0
        guid: 00000000-0000-0000-0000-000000000000
        bits per pixel: 0
        number of planes: 0
        type: RGB (packed)
        depth: 1
        red, green, blue masks: 0x0, 0x0, 0x0
```

Einige der aufgeführten Formate, wie YUV2 oder YUV12 existieren in machen XVideo-Implementierungen nicht. Dies kann zu Problemen mit einigen Spielern führen.

XVideo wird wahrscheinlich von der Karte nicht unterstützt, wenn die Ausgabe wie folgt aussieht:

```
X-Video Extension version 2.2
screen #0
no adaptors present
```

Wenn die XVideo-Erweiterung auf der Karte nicht läuft, wird es nur etwas schwieriger, die Anforderungen für die Wiedergabe von Videos zu erfüllen.

### 13.4.2. Video-Anwendungen

Dieser Abschnitt behandelt Anwendungen aus der FreeBSD-Ports-Sammlung, die für die Wiedergabe von Videos genutzt werden können.

#### MPlayer und MEncoder

MPlayer ist ein auf Geschwindigkeit und Flexibilität ausgelegter Video-Spieler für die Kommandozeile mit optionaler graphischer Oberfläche. Weitere graphische Oberflächen für MPlayer stehen in der FreeBSD Ports-Sammlung zur Verfügung.

MPlayer kann als Paket oder Port multimedia/mplayer installiert werden. Der Bau von MPlayer berücksichtigt die vorhandene Hardware und es können zahlreiche Optionen ausgewählt werden. Aus diesen Gründen ziehen es manche Benutzer vor, den Port zu übersetzen, anstatt das Paket zu installieren.

Die Optionen sollten beim Bau des Ports überprüft werden, um dem Umfang der Unterstützung, mit dem der Port gebaut wird, zu bestimmen. Wenn eine Option nicht ausgewählt wird, ist MPlayer nicht in der Lage, diese Art von Video-Format wiederzugeben. Mit den Pfeiltasten und der Leertaste können die erforderlichen Formate ausgewählt werden. Wenn Sie fertig sind, drücken Sie `Enter`, um den Bau und die Installation fortzusetzen.

In der Voreinstellung wird das Paket oder der Port das `mplayer`-Kommandozeilenprogramm und das graphische Programm `gmplayer` bauen. Um Videos zu dekodieren, installieren Sie den Port multimedia/mencoder. Aus lizenzrechtlichen Gründen steht ein Paket von MEncoder nicht zur

Verfügung.

MPlayer erstellt beim ersten Start ~/.mplayer im Heimatverzeichnis des Benutzers. Dieses Verzeichnis enthält die voreingestellten Konfigurationseinstellungen für den Benutzer.

Dieser Abschnitt beschreibt nur ein paar wenige Anwendungsmöglichkeiten. Eine vollständige Beschreibung der zahlreichen Möglichkeiten finden Sie in der Manualpage von mplayer(1).

Um die Datei testfile.avi abzuspielen, geben Sie die Video-Schnittstelle mit `-vo` an:

```
% mplayer -vo xv testfile.avi
```

```
% mplayer -vo sdl testfile.avi
```

```
% mplayer -vo x11 testfile.avi
```

```
# mplayer -vo dga testfile.avi
```

```
# mplayer -vo 'sdl:dga' testfile.avi
```

Es lohnt sich, alle Option zu testen. Die erzielte Geschwindigkeit hängt von vielen Faktoren ab und variiert beträchtlich je nach eingesetzter Hardware.

Wenn Sie eine DVD abspielen wollen, ersetzen Sie testfile.avi durch `-dvd://N Gerät`. `N` ist die Nummer des Stücks, das Sie abspielen wollen und Gerät gibt den Gerätenamen der DVD an. Das nachstehende Kommando spielt das dritte Stück von /dev/dvd:

```
# mplayer -vo dga -dvd://3 /dev/dvd
```



Das standardmäßig verwendete DVD-Laufwerk kann beim Bau des MPlayer-Ports mit der Option `WITH_DVD_DEVICE=/pfad/zum/gerät` festgelegt werden. Die Voreinstellung verwendet das Gerät /dev/cd0. Weitere Details finden Sie in Makefile.options des Ports.

Die Tastenkombinationen zum Abbrechen, Anhalten und Weiterführen der Wiedergabe entnehmen Sie der Ausgabe von `mplayer -h` oder der mplayer(1) Manualpage.

Weitere nützliche Optionen für die Wiedergabe sind `-fs -zoom` zur Wiedergabe im Vollbild-Modus und `-framedrop` zur Steigerung der Geschwindigkeit.

Jeder Benutzer kann häufig verwendete Optionen in seine ~/.mplayer/config eintragen:

```
vo=xv
fs=yes
zoom=yes
```

`mplayer` kann verwendet werden, um DVD-Stücke in .vob-Dateien zu rippen. Das zweite Stück einer DVD wandeln Sie wie folgt in eine Datei um:

```
# mplayer -dumpstream -dumpfile out.vob -dvd://2 /dev/dvd
```

Die Ausgabedatei out.vob wird im MPEG-Format abgespeichert.

Jeder Benutzer, der mehr Informationen über Video unter UNIX® sammeln möchte, sollte mplayerhq.hu/DOCS konsultieren, da es technisch sehr informativ ist. Diese Dokumentation sollte ebenfalls studiert werden, bevor Fehlerberichte eingereicht werden.

Vor der Verwendung von `mencoder` ist es hilfreich, sich mit den auf mplayerhq.hu/DOCS/HTML/en/mencoder.html beschriebenen Optionen vertraut zu machen. Es gibt unzählige Möglichkeiten die Qualität zu verbessern, die Bitrate zu verringern und Formate zu konvertieren. Einige davon haben erhebliche Auswirkungen auf die Geschwindigkeit. Falsche Kombinationen von Kommandozeilenparametern ergeben eventuell Dateien, die selbst `mplayer` nicht mehr wiedergeben kann.

Hier ist ein Beispiel für eine einfache Kopie:

```
% mencoder input.avi -oac copy -ovc copy -o output.avi
```

Wenn Sie in eine Datei rippen, benutzen Sie die Option `-dumpfile` von `mplayer`.

Um input.avi nach MPEG4 mit MPEG3 für den Ton zu konvertieren, muss zunächst der Port audio/lame installiert werden. Aus lizenzrechtlichen Gründen ist ein Paket nicht verfügbar. Wenn der Port installiert ist, geben Sie ein:

```
% mencoder input.avi -oac mp3lame -lameopts br=192 \
      -ovc lavc -lavcopts vcodec=mpeg4:vhq -o output.avi
```

Die Ausgabedatei lässt sich mit Anwendungen wie `mplayer` oder `xine` abspielen.

input.avi kann durch `-dvd://1 /dev/dvd` ersetzt und das Kommando als `root` ausgeführt werden, um ein DVD-Stück direkt zu konvertieren. Da vielleicht ein paar Versuche nötig sind, um das gewünschte Ergebnis zu erhalten, empfiehlt es sich das Stück zuerst in eine Datei zu schreiben und anschließend die Datei weiter zu bearbeiten.

#### Der Video-Spieler xine

xine ist ein Video-Spieler mit einer wiederverwendbaren Bibliothek und ein Programm, das durch Plugins erweitert werden kann. Es kann als Paket oder Port multimedia/xine installiert werden.

Für einen reibungslosen Betrieb benötigt xine entweder eine schnelle CPU mit einer schnellen Grafikkarte, oder die XVideo-Erweiterung. Am schnellsten läuft xine mit der XVideo-Erweiterung.

In der Voreinstellung startet xine eine grafische Benutzeroberfläche. Über die Menüs können dann bestimmte Dateien geöffnet werden.

Alternativ kann xine auch über die Kommandozeile aufgerufen werden, um Dateien direkt wiederzugeben:

```
% xine -g -p mymovie.avi
```

Weitere Informationen und Tipps zur Fehlerbehebung finden Sie unter xine-project.org/faq.

**Die Transcode-Werkzeuge**

Transcode ist eine Sammlung von Werkzeugen zur Umwandlung von Video- und Audio-Dateien. Transcode mischt Video-Dateien und kann kaputte Video-Dateien reparieren. Die Werkzeuge werden als Filter verwendet, das heißt die Ein- und Ausgaben verwenden stdin/stdout.

Unter FreeBSD kann Transcode als Paket oder Port multimedia/transcode installiert werden. Viele Benutzer bevorzugen es den Port zu bauen, da er ein Menü bereitstellt, wo die entsprechenden Formate für den Bau ausgewählt werden können. Mit den Pfeiltasten und der Leertaste können die erforderlichen Formate ausgewählt werden. Wenn Sie fertig sind, drücken Sie `Enter`, um den Bau und die Installation fortzusetzen.

Dieses Beispiel zeigt, wie eine DivX-Datei in eine PAL MPEG-1-Datei konvertiert wird:

```
% transcode -i input.avi -V --export_prof vcd-pal -o output_vcd
% mplex -f 1 -o output_vcd.mpg output_vcd.m1v output_vcd.mpa
```

Die daraus resultierende MPEG-Datei, output_vcd.mpg, kann beispielsweise mit MPlayer abgespielt werden. Die Datei kann auch mit einem Programm wie multimedia/vcdimager oder sysutils/cdrdao als Video-CD auf eine CD-R gebrannt werden.

Zusätzlich zu der Manualpage von `transcode`, sollten Sie auch die Informationen und Beispiele im transcoding.org/cgi-bin/transcode lesen.

## 13.5. TV-Karten

Mit TV-Karten können Sie mit dem Rechner über Kabel oder Antenne fernsehen. Die meisten Karten besitzen einen RCA- oder S-Video-Eingang. Einige Karten haben auch einen FM-Radio-Empfänger.

Der bktr(4)-Treiber von FreeBSD unterstützt PCI-TV-Karten mit einem Brooktree Bt848/849/878/879 Chip. Dieser Teiber unterstützt die meisten Pinnacle PCTV Karten. Die Karte sollte einen der unterstützten Empfänger besitzen, die in bktr(4) aufgeführt sind.

### 13.5.1. Den Treiber laden

Um die Karte benutzen zu können, muss der bktr(4)-Treiber geladen werden. Damit dies beim Systemstart automatisch erfolgt, muss die folgende Zeile in /boot/loader.conf hinzugefügt werden:

```
bktr_load="YES"
```

Alternativ kann der Treiber für die TV-Karte auch fest in den Kernel kompiliert werden. In diesem Fall müssen folgende Zeilen in die Kernelkonfigurationsdatei aufgenommen werden:

```
device   bktr
device  iicbus
device  iicbb
device  smbus
```

Die zusätzlichen Treiber werden benötigt, da die Komponenten der Karte über einen I2C-Bus verbunden sind. Bauen und installieren Sie dann den neuen Kernel.

Um den Treiber zu testen, muss das System neu gestartet werden. Während des Neustarts sollte die TV-Karte erkannt werden:

```
bktr0: <BrookTree 848A> mem 0xd7000000-0xd7000fff irq 10 at device 10.0 on pci0
iicbb0: <I2C bit-banging driver> on bti2c0
iicbus0: <Philips I2C bus> on iicbb0 master-only
iicbus1: <Philips I2C bus> on iicbb0 master-only
smbus0: <System Management Bus> on bti2c0
bktr0: Pinnacle/Miro TV, Philips SECAM tuner.
```

Abhängig von der verwendeten Hardware können die Meldungen natürlich anders aussehen. Die entdeckten Geräte lassen sich mit sysctl(8) oder in der Kernelkonfigurationsdatei überschreiben. Wenn Sie beispielsweise einen Philips-SECAM-Empfänger erzwingen wollen, fügen Sie die folgende Zeile zur Kernelkonfigurationsdatei hinzu:

```
options OVERRIDE_TUNER=6
```

Alternativ können Sie sysctl(8) benutzen:

```
# sysctl hw.bt848.tuner=6
```

Weitere Informationen zu den verschiedenen Kerneloptionen und sysctl(8)-Parametern finden Sie in bktr(4).

### 13.5.2. Nützliche Anwendungen

Um die TV-Karte zu benutzen, installieren Sie eine der nachstehenden Anwendungen:

- multimedia/fxtv lässt das Fernsehprogramm in einem Fenster laufen und kann Bilder, Audio und Video aufzeichnen.
- multimedia/xawtv eine weitere TV-Anwendung mit vergleichbaren Funktionen.
- Mit audio/xmradio lässt sich der FM-Radio-Empfänger, der sich auf TV-Karten befindet, benutzen.

Weitere Anwendungen finden Sie in der FreeBSD Ports-Sammlung.

### 13.5.3. Fehlersuche

Wenn Sie Probleme mit der TV-Karte haben, prüfen Sie zuerst, ob der Video-Capture-Chip und der Empfänger vom bktr(4)-Treiber unterstützt werden und ob Sie die richtigen Optionen verwenden. Weitere Hilfe zu unterstützten TV-Karten finden Sie auf der Mailingliste FreeBSD multimedia.

## 13.6. MythTV

MythTV ist eine beliebte Open Source PVR-Anwendung. Dieser Abschnitt beschreibt die Installation und Konfiguration von MythTV unter FreeBSD. Weitere Informationen zur Benutzung von MythTV finden Sie unter mythtv.org/wiki.

MythTV benötigt ein Frontend und ein Backend. Diese Komponenten können entweder auf dem gleichen System, oder auf unterschiedlichen Maschinen installiert werden.

Das Frontend kann unter FreeBSD über den Port oder das Paket multimedia/mythtv-frontend installiert werden. Zudem muss Xorg, wie in Das X-Window-System beschrieben, installiert und konfiguriert sein. Idealerweise besitzt das System auch eine Videokarte, die X-Video Motion Compensation (XvMC) unterstützt, sowie optional eine LIRC-kompatible Fernbedienung.

Benutzen Sie multimedia/mythtv, um sowohl das Frontend als auch das Backend zu installieren. Ein MySQL™ Datenbank-Server ist ebenfalls erforderlich und sollte automatisch als Abhängigkeit installiert werden. Optional sollte das System einen Empfänger und ausreichend Speicherplatz haben, um die aufgezeichneten Daten speichern zu können.

### 13.6.1. Hardware

MythTV verwendet V4L um auf Videoeingabegeräte, wie Kodierer und Empfänger zuzugreifen. Unter FreeBSD funktioniert MythTV am besten mit USB DVB-S/C/T Karten, die von multimedia/webcamd unterstützt werden, da dies eine V4L-Anwendung zur Verfügung stellt, die als Benutzerprogramm läuft. Jede DVB-Karte, die von webcamd unterstützt wird, sollte mit MythTV funktionieren, jedoch gibt es eine Liste von Karten, die unter wiki.freebsd.org/WebcamCompat abgerufen werden kann. Es existieren auch Treiber für Hauppauge-Karten in den folgenden Paketen: multimedia/pvr250 und multimedia/pvrxxx, allerdings liefern diese nur eine Treiberschnittstelle, die nicht dem Standard entspricht und die nicht mit MythTV-Versionen grösser als 0.23 funktionieren. Aus lizenzrechtlichen Gründen ist ein Paket nicht verfügbar, sodass die

beiden Ports übersetzt werden müssen.

Die wiki.freebsd.org/HTPC enthält eine Liste von allen verfügbaren DVB-Treibern.

### 13.6.2. MythTV Backend einrichten

Geben Sie folgendes ein, um MythTV als Binärpaket zu installieren:

```
# pkg install mythtv
```

Alternativ können Sie den Port installieren:

```
# cd /usr/ports/multimedia/mythtv
# make install
```

Richten Sie anschließend die MythTV-Datenbank ein:

```
# mysql -uroot -p < /usr/local/shared/mythtv/database/mc.sql
```

Konfigurieren Sie dann das Backend:

```
# mythtv-setup
```

Zum Schluss starten Sie das Backend:

```
# sysrc mythbackend_enable=yes
# service mythbackend start
```

## 13.7. Scanner

Unter FreeBSD stellt SANE (Scanner Access Now Easy) aus der Ports-Sammlung eine einheitliche Schnittstelle (API) für den Zugriff auf Scanner bereit. SANE wiederum greift auf Scanner mithilfe einiger FreeBSD-Treiber zu.

FreeBSD unterstützt sowohl SCSI- als auch USB-Scanner. Abhängig von der Schnittstelle des Scanners, werden unterschiedliche Treiber benötigt. Prüfen Sie vor der Konfiguration mithilfe der Liste der unterstützten Geräte ob der Scanner von SANE unterstützt wird.

Dieses Kapitel beschreibt, wie Sie feststellen können, ob der Scanner von FreeBSD erkannt wurde. Zudem enthält es einen Überblick über die Konfiguration und Verwendung von SANE unter FreeBSD.

### 13.7.1. Den Scanner überprüfen

Im GENERIC-Kernel sind schon alle, für einen USB-Scanner notwendigen Treiber enthalten. Benutzer mit einem angepassten Kernel sollten sicherstellen, dass die Kernelkonfiguration die nachstehenden Zeilen enthält:

```
device usb
device uhci
device ohci
device ehci
device xhci
```

Um zu überprüfen ob der Scanner erkannt wird, schließen Sie den USB-Scanner an. Prüfen Sie dann mit dmesg(8), ob der Scanner in den Systemmeldungen erscheint:

```
ugen0.2: <EPSON> at usbus0
```

In diesem Beispiel wurde ein EPSON Perfection® 1650 USB-Scanner an /dev/ugen0.2 erkannt.

Wenn der Scanner eine SCSI-Schnittstelle besitzt, ist die Kernelkonfiguration abhängig vom verwendeten SCSI-Controller. Der GENERIC-Kernel unterstützt die gebräuchlichen SCSI-Controller. Den richtigen Treiber finden Sie in /usr/src/sys/conf/NOTES. Neben dem SCSI-Treiber muss die Kernelkonfiguration noch die nachstehenden Zeilen enthalten:

```
device scbus
device pass
```

Nachdem Sie einen Kernel gebaut und installiert haben, sollte der Scanner beim Neustart in den Systemmeldungen erscheinen:

```
pass2 at aic0 bus 0 target 2 lun 0
pass2: <AGFA SNAPSCAN 600 1.10> Fixed Scanner SCSI-2 device
pass2: 3.300MB/s transfers
```

Wenn der Scanner während des Systemstarts ausgeschaltet war, können Sie die Geräteerkennung erzwingen, indem Sie den SCSI-Bus erneut absuchen. Verwenden Sie dazu `camcontrol`:

```
# camcontrol rescan all
Re-scan of bus 0 was successful
Re-scan of bus 1 was successful
Re-scan of bus 2 was successful
Re-scan of bus 3 was successful
```

Der Scanner sollte jetzt in der SCSI-Geräteliste erscheinen:

```
# camcontrol devlist
<IBM DDRS-34560 S97B>              at scbus0 target 5 lun 0 (pass0,da0)
<IBM DDRS-34560 S97B>              at scbus0 target 6 lun 0 (pass1,da1)
<AGFA SNAPSCAN 600 1.10>           at scbus1 target 2 lun 0 (pass3)
<PHILIPS CDD3610 CD-R/RW 1.00>     at scbus2 target 0 lun 0 (pass2,cd0)
```

Weitere Informationen über SCSI-Geräte unter FreeBSD finden Sie in scsi(4) und camcontrol(8).

### 13.7.2. SANE konfigurieren

Das SANE-System ermöglicht den Zugriff auf den Scanner über Backends (graphics/sane-backends). Lesen Sie http://www.sane-project.org/sane-supported-devices.html um herauszufinden, welches Backend welchen Scanner unterstützt. Eine graphische Oberfläche wird über Anwendungen von Drittanbietern wie Kooka (graphics/kooka) oder XSane (graphics/xsane) bereitgestellt. Die Backends von SANE reichen aus, um den Scanner zu testen.

Installieren Sie die Backends als Paket:

```
# pkg install sane-backends
```

Alternativ können Sie die Backends aus der Ports-Sammlung installieren:

```
# cd /usr/ports/graphics/sane-backends
# make install clean
```

Nachdem Sie den Port oder das Paket graphics/sane-backends installiert haben, können Sie mit dem Befehl `sane-find-scanner` prüfen, ob SANE den Scanner erkennt:

```
# sane-find-scanner -q
found SCSI scanner "AGFA SNAPSCAN 600 1.10" at /dev/pass3
```

Die Ausgabe zeigt die Schnittstelle und die verwendete Gerätedatei des Scanners. Der Hersteller und das Modell können in der Ausgabe fehlen.



Bei einigen USB-Scannern muss die Firmware geladen werden. Lesen Sie sane-find-scanner(1) und sane(7) für weitere Details.

Als nächstes müssen Sie prüfen, ob der Scanner vom Frontend erkannt wird. Die SANE-Backends werden mit dem Kommandozeilenwerkzeug `scanimage` geliefert. Mit diesem Werkzeug können Sie sich Scanner anzeigen lassen und den Scan-Prozess von der Kommandozeile starten. Die Option `-L` zeigt die Scanner an. Das erste Beispiel ist für einen SCSI-Scanner, das zweite ist für einen USB-Scanner:

```
# scanimage -L
```

```
device `snapscan:/dev/pass3' is a AGFA SNAPSCAN 600 flatbed scanner
# scanimage -L
device 'epson2:libusb:000:002' is a Epson GT-8200 flatbed scanner
```

Im zweiten Beispiel ist `epson2` der Backend-Name. `libusb:000:002` bedeutet, dass /dev/ugen0.2 die vom Scanner verwendete Gerätedatei ist.

Wenn `scanimage` den Scanner nicht erkennen kann, erscheint folgende Meldung:

```
# scanimage -L

No scanners were identified. If you were expecting something different,
check that the scanner is plugged in, turned on and detected by the
sane-find-scanner tool (if appropriate). Please read the documentation
which came with this software (README, FAQ, manpages).
```

Wenn das passiert, müssen Sie in der Konfigurationsdatei des Backends unterhalb von /usr/local/etc/sane.d/ den verwendeten Scanner eintragen. Wenn der Scanner EPSON Perfection® 1650, der das Backend `epson2` benutzt, nicht erkannt wurde, muss /usr/local/etc/sane.d/epson2.conf angepasst werden. Fügen Sie eine Zeile mit der Schnittstelle und dem Gerätenamen in die Datei ein. In diesem Beispiel wurde die nachstehende Zeile eingefügt:

```
usb /dev/ugen0.2
```

Speichern Sie die Änderungen und prüfen Sie, ob der Scanner mit dem richtigen Backend und Gerätenamen erkannt wird:

```
# scanimage -L
device 'epson2:libusb:000:002' is a Epson GT-8200 flatbed scanner
```

Wenn `scanimage -L` den Scanner erkannt hat, ist der Scanner eingerichtet und bereit, zu scannen.

Obwohl `scanimage` von der Kommandozeile scannen kann, ist eine graphische Anwendung zum Scannen besser geeignet. Bekannte Programme sind Koka oder XSane. Diese Frontends besitzten erweiterte Funktionen wie den Scan-Modus, Farbkorrektur und Batch-Scans. XSane lässt sich auch als GIMP-Plugin verwenden.

### 13.7.3. Berechtigungen für den Scanner

Wenn andere Benutzer den Scanner benutzen sollen, müssen sie Lese- und Schreibrechte auf die Gerätedatei des Scanners besitzen. Im vorherigen Beispiel wird die Datei /dev/ugen0.2 verwendet, die faktisch nur ein Symlink auf die echte Gerätedatei, /dev/usb/0.2.0 genannt, darstellt. Sowohl der Symlink als auch die Gerätedatei sind jeweils im Besitz der Gruppen `wheel` und `operator`. Damit ein Benutzer den Scanner benutzen kann, muss er Mitglied in einer der beiden Gruppen sein. Allerdings sollte aus Sicherheitsgründen genau überlegt werden, welche Benutzer zu welcher Gruppe hinzugefügt werden, besonders bei der Gruppe `wheel`. Eine bessere Lösung ist es, eine

spezielle Gruppe für den Zugriff anzulegen und den Scanner für Mitglieder dieser Gruppe zugänglich zu machen.

Dieses Beispiel erstellt eine Gruppe namens `usb`:

```
# pw groupadd usb
```

Anschließend muss der /dev/ugen0.2-Symlink und der Gerätename /dev/usb/0.2.0 für die Gruppe `usb` mit den Schreibrechten `0660` oder `0664` ausgestattet werden. All dies kann durch das Hinzufügen der folgenden Zeilen in /etc/devfs.rules erreicht werden:

```
[system=5]
add path ugen0.2 mode 0660 group usb
add path usb/0.2.0 mode 0666 group usb
```



Es kommt vor, dass sich der Gerätename mit dem Hinzufügen oder Entfernen von Geräten ändert, so dass man stattdessen vielleicht allen USB-Geräten mit diesem Regelsatz Zugriff gewähren möchte:

```
[system=5]
add path 'ugen*' mode 0660 group usb
add path 'usb/*' mode 0666 group usb
```

Weitere Informationen zu dieser Datei finden Sie in devfs.rules(5).

Als nächstes aktivieren Sie den Regelsatz in /etc/rc.conf:

```
devfs_system_ruleset="system"
```

Starten Sie anschließend das devfs(8)-System neu:

```
# service devfs restart
```

Jetzt müssen nur noch Benutzer zur Gruppe `usb` hinzugefügt werden, um ihnen den Zugriff auf den Scanner zu erlauben:

```
# pw groupmod usb -m joe
```

Weitere Details finden Sie in pw(8).

# Kapitel 14. Konfiguration des FreeBSD-Kernels

## 14.1. Übersicht

Der Kernel ist das Herz des FreeBSD-Betriebssystems. Er ist verantwortlich für die Speicherverwaltung, das Durchsetzen von Sicherheitsdirektiven, Netzwerkfähigkeit, Festplattenzugriffen und vieles mehr. Obwohl FreeBSD es ermöglicht, dynamisch konfiguriert zu werden, ist es ab und an notwendig, einen angepassten Kernel zu konfigurieren und zu kompilieren.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie Folgendes wissen:

- Wann Sie einen angepassten Kernel kompilieren sollten.
- Wie Sie eine Hardware-Inventur durchführen.
- Wie Sie eine Kernelkonfigurationsdatei verändern.
- Wie Sie mit der Konfigurationsdatei einen neuen Kernel kompilieren.
- Wie Sie den neuen Kernel installieren.
- Was zu tun ist, falls etwas schiefgeht.

Alle Kommandos, aus den Beispielen dieses Kapitels, müssen mit `root`-Rechten ausgeführt werden.

## 14.2. Wieso einen eigenen Kernel bauen?

Traditionell besaß FreeBSD einen monolithischen Kernel. Der Kernel war ein einziges großes Programm, das eine bestimmte Auswahl an Hardware unterstützte. Um das Kernelverhalten zu ändern, musste man einen neuen Kernel kompilieren und dann den neuen Kernel booten.

Heutzutage befinden sich die meisten Funktionen des FreeBSD-Kernels in Modulen, die je nach Bedarf dynamisch geladen und entladen werden können. Dies erlaubt es, einen laufenden Kernel anzupassen, um sofort neue Hardware und neue Funktionen zu unterstützen. Dies ist als modularer Kernel bekannt.

Gelegentlich ist es noch notwendig, eine statische Kernelkonfigurationen durchzuführen. In einigen Fällen ist die Funktion zu systemnah, um durch ein Modul realisiert zu werden. Andere Umgebungen verhindern vielleicht das Laden und Entladen von Kernelmodulen und erfordern, dass nur die benötigte Funktionalität statisch in den Kernel kompiliert wird.

Das Erstellen eines angepassten Kernels ist eines der Rituale für erfahrene BSD-Benutzer. Obwohl dieser Prozess recht viel Zeit in Anspruch nimmt, kann er doch viele Vorteile für das FreeBSD-System bringen. Im Gegensatz zum GENERIC-Kernel, der eine Vielzahl von Hardware unterstützen muss, kann ein angepasster Kernel so eingeschränkt werden, dass er nur noch die Hardware des Rechners unterstützt. Dies hat einige Vorteile:

- Schnellerer Bootvorgang. Da der Kernel nur nach der Hardware des Systems sucht, kann sich

die Zeit für einen Systemstart verkürzen.

- Geringerer Speicherbedarf. Ein eigener Kernel benötigt in der Regel weniger Speicher als ein GENERIC-Kernel durch das Entfernen von Funktionen und Gerätetreibern. Das ist vorteilhaft, denn der Kernel verweilt immer im RAM und verhindert dadurch, dass dieser Speicher von Anwendungen genutzt wird. Deshalb ist ein angepasster Kernel auf einem System mit wenig RAM sinnvoll.
- Zusätzliche Hardwareunterstützung. Ein angepasster Kernel kann Unterstützung für Geräte bieten, die im GENERIC-Kernel nicht enthalten sind.

Bevor Sie einen angepassten Kernel erstellen, überlegen Sie sich bitte, warum Sie dies tun wollen. Wenn Sie lediglich eine bestimmte Hardwareunterstützung benötigen, existiert diese vielleicht schon als Kernelmodul.

Kernelmodule existieren in /boot/kernel und können mit kldload(8) dynamisch in den laufenden Kernel geladen werden. Die meisten Kerneltreiber verfügen über ein ladbares Modul und eine Manualpage. Der drahtlose Ethernet-Treiber ath(4) hat die folgenden Informationen in seiner Manualpage:

```
Alternatively, to load the driver as a module at boot time, place the
following line in loader.conf(5):

    if_ath_load="YES"
```

Durch das Hinzufügen von `if_ath_load="YES"` in /boot/loader.conf wird das Modul dynamisch beim Systemstart geladen.

In manchen Fällen gibt es kein entsprechendes Modul in /boot/kernel. Dies gilt insbesondere für bestimmte Subsysteme.

# 14.3. Informationen über die vorhandene Hardware beschaffen

Bevor die Kernelkonfigurationsdatei bearbeitet wird, ist es empfehlenswert eine Bestandsaufnahme der Hardware des Systems durchzuführen. Auf einem Dual-Boot-System können diese Informationen aus dem anderen Betriebssystem ermittelt werden. Microsoft®s Gerätemanager enthält beispielsweise Informationen über die installierte Hardware.



Einige Versionen von Microsoft® Windows® verfügen über ein System-Icon auf dem Desktop, über das Sie den Gerätemanager direkt aufrufen können.

Wenn FreeBSD das einzige installierte Betriebssystem ist, dann listet dmesg(8) die Hardware auf, die während des Systemstarts gefunden wurde. Die meisten FreeBSD-Gerätetreiber haben eine eigene Manualpage, die Informationen über die unterstützte Hardware enthält. Die folgenden Zeilen zeigen beispielsweise an, dass der psm(4)-Treiber eine angeschlossene Maus gefunden hat:

```
psm0: <PS/2 Mouse> irq 12 on atkbdc0
```

```
psm0: [GIANT-LOCKED]
psm0: [ITHREAD]
psm0: model Generic PS/2 mouse, device ID 0
```

Da diese Hardware vorhanden ist, sollte dieser Treiber nicht aus einer angepassten Kernelkonfigurationsdatei entfernt werden.

Wenn dmesg keine Informationen zur gefundenen Hardware anzeigt, können diese Informationen auch aus /var/run/dmesg.boot entnommen werden.

Ein weiteres Werkzeug für die Suche nach Hardware ist pciconf(8), das ausführliche Informationen bereitstellt. Ein Beispiel:

```
% pciconf -lv
ath0@pci0:3:0:0:        class=0x020000 card=0x058a1014 chip=0x1014168c rev=0x01 hdr
=0x00
    vendor     = 'Atheros Communications Inc.'
    device     = 'AR5212 Atheros AR5212 802.11abg wireless'
    class      = network
    subclass   = ethernet
```

Die Ausgabe zeigt, dass der Treiber ath eine drahtlose Ethernetkarte gefunden hat.

Die Option -k von man(1) kann verwendet werden, um nützliche Informationen zu erhalten. Um beispielsweise eine Liste von Manualpages zu erhalten, welche ein spezifisches Wort enthalten:

```
# man -k Atheros
ath(4)                   - Atheros IEEE 802.11 wireless network driver
ath_hal(4)               - Atheros Hardware Access Layer (HAL)
```

Mit einer Inventarliste der Hardware können Sie dann sicherstellen, dass Sie die Treiber der installierten Hardware nicht versehentlich entfernen, wenn Sie die Kernelkonfigurationsdatei bearbeiten.

## 14.4. Die Kernelkonfigurationsdatei

Bevor eine angepasste Kernelkonfigurationsdatei erstellt werden kann, muss zuerst der vollständige FreeBSD Quellcodebaum installiert werden.

Falls /usr/src/ nicht existiert oder leer ist, sind die Kernelquellen nicht installiert. Die Quellen können mit Subversion und der Anleitung im “Benutzen von Subversion” installiert werden.

Sobald die Quellen installiert sind, können Sie sich einen Überblick über /usr/src/sys verschaffen. Dieses Verzeichnis enthält eine Reihe von Unterverzeichnissen, einschließlich Verzeichnisse für die unterstützten Architekturen amd64, i386, powerpc und sparc64. Alles in diesen Verzeichnissen ist nur für die jeweilige Architektur relevant. Der Rest des Codes ist maschinenunabhängig und für alle Architekturen gleich. Jede unterstützte Architektur hat ein Unterverzeichnis conf, das die

GENERIC Kernelkonfigurationsdatei für diese Architektur enthält.

Bearbeiten Sie GENERIC nicht direkt. Kopieren Sie stattdessen die Datei unter einem anderen Namen und machen dann die Änderungen an dieser Kopie. Traditionell besteht der Name des Kernels immer aus Großbuchstaben. Wenn Sie mehrere FreeBSD-Maschinen mit unterschiedlicher Hardware betreuen, ist es eine gute Idee, die Konfigurationsdatei nach den Hostnamen der Maschinen zu benennen. In diesem Beispiel wird eine Kopie der GENERIC Kernelkonfigurationsdatei, namens MYKERNEL, für die amd64-Architektur erstellt:

```
# cd /usr/src/sys/amd64/conf
# cp GENERIC MYKERNEL
```

MYKERNEL kann jetzt mit einem Texteditor bearbeitet werden. Der Standard-Editor ist vi, jedoch steht mit ee ein weiterer, einfach zu bedienender Editor bereit.

Das Format der Konfigurationsdatei ist einfach. Jede Zeile enthält ein Schlüsselwort, das ein Gerät oder ein Subsystem repräsentiert, ein Argument und eine kurze Beschreibung. Jeder Text, der hinter einem `steht, gilt als Kommentar und wird ignoriert. Um die Kernel-Unterstützung für ein Gerät oder Subsystem zu entfernen, muss ein` an den Anfang der Zeile, die dieses Gerät oder Subsystem repräsentiert, gesetzt werden. Verändern Sie keine Zeilen, die Sie nicht genau verstehen.

Neben den Kurzbeschreibungen in dieser Datei, finden Sie zusätzliche Erklärungen in NOTES, die sich in demselben Verzeichnis wie GENERIC für die jeweilige Architektur befindet. Von der Architektur unabhängige Optionen sind in /usr/src/sys/conf/NOTES aufgeführt.



Wenn Sie die Kernelkonfigurationsdatei fertig bearbeitet haben, sollten Sie eine Sicherungskopie außerhalb von /usr/src speichern

Alternativ kann die Kernelkonfigurationsdatei an anderer Stelle gespeichert, und ein symbolischer Link auf die Datei erstellt werden:

```
# cd /usr/src/sys/amd64/conf
# mkdir /root/kernels
# cp GENERIC /root/kernels/MYKERNEL
# ln -s /root/kernels/MYKERNEL
```

Es ist möglich, eine `include`-Anweisung in die Kernelkonfigurationsdatei aufzunehmen. Diese erlaubt das lokale Einfügen von anderen Konfigurationsdateien in die aktuelle, was es einfacher macht, kleinere Änderungen an einer existierenden Datei zu vollziehen. Wenn Sie einen GENERIC -Kernel mit nur einer kleinen Anzahl von zusätzlichen Optionen und Treibern benötigen, brauchen Sie mit den folgenden Zeilen nur ein kleines Delta im Vergleich zu GENERIC anpassen, wie in diesem Beispiel zu sehen:

```
include GENERIC
ident MYKERNEL

options         IPFIREWALL
```

```
options         DUMMYNET
options         IPFIREWALL_DEFAULT_TO_ACCEPT
options         IPDIVERT
```

Diese Methode zeigt die Unterschiede der lokalen Konfigurationsdatei zu einem GENERIC-Kernel an. Sobald Aktualisierungen durchgeführt werden, können neue Eigenschaften, die zu GENERIC hinzugefügt werden, auch dem lokalen Kernel angehängt werden, es sei denn, es wird durch `nooptions` oder `nodevice` unterbunden. Eine umfassende Liste von Konfigurationseinstellungen und deren Beschreibungen finden Sie in config(5).



Um einen Kernel mit allen möglichen Optionen zu bauen, führen Sie als `root` die folgenden Befehle aus:

```
# cd /usr/src/sys/arch/conf && make LINT
```

## 14.5. Einen angepassten Kernel bauen und installieren

Nachdem die Änderungen an der angepassten Kernelkonfigurationsdatei gespeichert sind, kann der Quellcode für den Kernel mit den folgenden Schritten übersetzt werden:

**Procedure: Einen Kernel bauen**

1. Wechseln Sie das Verzeichnis:

   ```
   # cd /usr/src
   ```

2. Bauen Sie den Kernel, indem Sie den Namen der Kernelkonfigurationsdatei angeben:

   ```
   # make buildkernel KERNCONF=MYKERNEL
   ```

3. Installieren Sie den neuen Kernel. Dieser Befehl wird den neuen Kernel nach /boot/kernel/kernel kopieren, und den alten Kernel nach /boot/kernel.old/kernel speichern:

   ```
   # make installkernel KERNCONF=MYKERNEL
   ```

4. Fahren Sie das System herunter und starten Sie den neuen Kernel. Wenn etwas nicht funktioniert, lesen Sie Der Kernel bootet nicht.

In der Voreinstellung werden beim Bau eines angepassten Kernels stets alle Kernelmodule neu gebaut. Um einen Kernel schneller zu bauen, oder um nur bestimmte Module zu bauen, bearbeiten Sie /etc/make.conf, bevor Sie den Kernel neu bauen.

In diesem Beispiel werden über eine Variable nur die Kernelmodule definiert, die auch tatsächlich

gebaut werden sollen. In der Voreinstellung werden alle Module gebaut:

```
MODULES_OVERRIDE = linux acpi
```

Alternativ kann auch eine Variable verwendet werden, die bestimmte Kernelmodule vom Bauprozess ausschließt:

```
WITHOUT_MODULES = linux acpi sound
```

Weitere Variablen und deren Beschreibung finden Sie in make.conf(5).

## 14.6. Wenn etwas schiefgeht

Es gibt vier Hauptfehlerquellen beim Erstellen eines angepassten Kernels:

**`config` verursacht Fehler**

Wenn `config` fehlschlägt, zeigt es die Nummer der Zeile an, die das Problem verursacht. Bei der folgenden Fehlermeldung sollten Sie die angegebene Zeile mit GENERIC oder NOTES vergleichen und sicherstellen, dass das Schlüsselwort in Zeile 17 richtig geschrieben ist:

```
config: line 17: syntax error
```

**`make` verursacht Fehler**

Wenn `make` fehlschlägt, liegen meistens Fehler in der Konfigurationsdatei vor, die aber nicht schwerwiegend genug für `config` waren. Überprüfen Sie die Konfiguration und wenn Sie keinen Fehler entdecken können, schicken Sie eine E-Mail mit der Kernelkonfigurationsdatei an die Mailingliste Fragen und Antworten zu FreeBSD <de-bsd-questions@de.FreeBSD.org>.

**Der Kernel bootet nicht**

Wenn der neue Kernel nicht bootet oder die Geräte nicht erkannt werden, ist das noch kein Grund zur Panik. Glücklicherweise besitzt FreeBSD einen exzellenten Mechanismus zur Wiederherstellung nach dem Einsatz inkompatibler Kernel. Wählen Sie einfach den zu bootenden Kernel im FreeBSD Bootloader aus. Dazu wählen Sie im Bootmenü die Option "Escape to a loader prompt". Danach geben Sie am Prompt `boot kernel.old` oder den Namen eines anderen Kernels ein, der sauber bootet.

Nun kann die Konfiguration noch einmal überprüft und der Kernel neu kompiliert werden. Dazu ist /var/log/messages sehr nützlich, da hier sämtliche Kernelmeldungen von jedem erfolgreichen Bootvorgang gespeichert werden. dmesg(8) gibt die Kernelmeldungen vom letzten Bootvorgang aus.



Wenn Sie Probleme beim Kernelbau bekommen, heben Sie sich immer eine Kopie von GENERIC oder einen anderen Kernel, der garantiert bootet, auf. Dies ist sehr wichtig, weil jedes Mal, wenn ein neuer Kernel installiert wird, kernel.old mit dem zuletzt installierten Kernel überschrieben wird und dieser

möglicherweise nicht bootfähig ist. Verschieben Sie daher den funktionierenden Kernel so schnell wie möglich, indem Sie das Verzeichnis mit dem funktionierenden Kernel umbenennen:

```
# mv /boot/kernel /boot/kernel.bad
# mv /boot/kernel.good /boot/kernel
```

**Der Kernel funktioniert, aber `ps` nicht**

Wenn Sie eine andere Version des Kernels installiert haben als die, mit der Ihre Systemwerkzeuge gebaut wurden, beispielsweise einen Kernel aus den -CURRENT-Quellen auf einem -RELEASE-System, werden Programme wie ps(1) und vmstat(8) nicht mehr funktionieren. Um dies zu beheben, sollten Sie das komplette System neu bauen und installieren. Achten Sie darauf, dass die Quellen, aus denen das System gebaut wird, zum installierten Kernel passt. Man sollte niemals einen Kernel benutzen, der nicht zur Systemversion passt.

# Kapitel 15. Drucken

Trotz vieler Versuche es zu vermeiden, ist der Druck von Informationen auf Papier immer noch eine wichtige Funktion. Drucken hat zwei grundlegende Komponenten. Die Daten müssen an den Drucker gesendet werden, und zwar in einer Form, die der Drucker verstehen kann.

## 15.1. Schnellstart

Die grundlegende Druckfunktion kann schnell eingerichtet werden. Der Drucker muss lediglich fähig sein, normalen ASCII-Text zu drucken. Informationen zum Druck von anderen Dateien finden Sie in Filter.

1. Erstellen Sie ein Verzeichnis zur Speicherung der Druckaufträge:

```
# mkdir -p /var/spool/lpd/lp
# chown daemon:daemon /var/spool/lpd/lp
# chmod 770 /var/spool/lpd/lp
```

2. Erstellen Sie als `root` die Datei /etc/printcap mit folgendem Inhalt:

```
lp:\
    :lp=/dev/unlpt0:\  ①
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/lp:\
    :lf=/var/log/lpd-errs:
```

① Diese Zeile ist für einen Drucker, der an einem USB-Port angeschlossen ist.Für einen Drucker, der am parallelen oder "Drucker"-Port angeschlossen ist, verwenden Sie:Für einen Netzwerkdrucker verwenden Sie:Ersetzen Sie *network-printer-name* durch den DNS-Namen des Netzwerkdruckers.

3. Aktivieren Sie `lpd` beim Systemstart, indem Sie folgende Zeile in /etc/rc.conf hinzufügen:

```
lpd_enable="YES"
```

Starten Sie den Dienst:

```
# service lpd start
Starting lpd.
```

Drucken Sie eine Testseite:

```
# printf "1. Der Drucker kann drucken.\n2. Dies ist die zweite Zeile.\n" |
```

```
lpr
```



Wenn die beiden Zeilen nicht am linken Rand starten und Sie einen "Treppeneffekt" beobachten, lesen Sie Den Treppeneffekt verhindern.

Mit `lpr` können nun Textdateien gedruckt werden. Geben Sie den Dateinamen auf der Kommandozeile an oder lassen Sie `lpr` von einer Pipe lesen.

```
% lpr textfile.txt
% ls -lh | lpr
```

## 15.2. Druckerverbindungen

Es gibt eine Vielzahl von Möglichkeiten, einen Drucker mit einem Rechner zu verbinden. Kleine Desktop-Drucker werden in der Regel mit dem USB-Anschluss verbunden, ältere Modelle nutzen oft die parallele Schnittstelle. Einige Drucker sind direkt mit einem Netzwerk verbunden, damit sie leichter von mehreren Rechnern benutzt werden können. Nur noch wenige Drucker verwenden einen seriellen Anschluss.

FreeBSD unterstützt die folgenden Arten von Druckern:

**USB**

USB-Drucker können mit einem freien USB-Anschluss des Rechners verbunden werden.

Wenn FreeBSD einen USB-Drucker erkennt, werden zwei Gerätenamen erstellt: /dev/ulpt0 und /dev/unlpt0. Beide Geräte leiten die Daten an den Drucker weiter. Nach jedem Druckauftrag wird der USB-Anschluss von ultp0 zurückgesetzt. Das Zurücksetzen kann bei einigen Druckern Probleme verursachen, daher wird in der Regel stattdessen unlpt0 verwendet, das den Anschluss nicht zurücksetzt.

**Prallel (IEEE-1284)**

Die parallele Schnittstelle ist /dev/lpt0. Der Gerätename erscheint unabhängig davon, ob ein Drucker angeschlossen ist oder nicht. Eine automatische Erkennung findet nicht statt.

Die Hersteller haben sich weitgehend von diesem älteren Anschluss verabschiedet und auch viele Rechner haben keine parallele Schnittstelle mehr. Es existieren jedoch Adapter, um einen parallelen Drucker an einem USB-Port anzuschließen. Der Drucker wird dann wie ein USB-Drucker behandelt. Es können auch *Printserver* verwendet werden, um parallele Drucker direkt mit einem Netzwerk zu verbinden.

**Seriell (RS-232)**

Serielle Anschlüsse sind veraltet und werden außer in Nischenanwendungen nur noch selten verwendet. Die Kabel, Stecker und die erforderliche Verkabelung sind oft sehr unterschiedlich.

Der Gerätename für einen seriellen Anschlüsse ist /dev/cuau0 oder /dev/cuau1. Es können auch USB-Adapter verwendet werden. Diese erscheinen als /dev/cuaU0.

Damit mit dem Drucker kommuniziert werden kann, müssen einige Kommunikationsparameter bekannt sein. Zu den wichtigsten zählen die *Baudrate* (BPS - Bits pro Sekunde) und die *Parität*. Diese Werte variieren, aber typische serielle Drucker verwenden eine Baudrate von 9600 und keine Parität.

**Netzwerk**

Netzwerkdrucker werden direkt mit dem lokalen Netzwerk verbunden.

Der DNS-Name des Druckers muss bekannt sein. Wenn dem Drucker eine dynamische Adresse per DHCP zugeteilt wird, sollte das DNS automatisch aktualisiert werden, so dass der Drucker immer die richtige IP-Adresse hat. Um dieses Problem zu vermeiden, werden Netzwerkdruckern häufig statische IP-Adressen zugeteilt.

Die meisten Netzwerkdrucker verstehen Druckaufträge, die über das LPD-Protokoll empfangen werden. Sie können auch den Namen der Druckwarteschlange angeben. Einige Drucker verarbeiten die Daten unterschiedlich, je nachdem welche Warteschlange verwendet wird. Zum Beispiel druckt eine `Raw`-Warteschlange die Daten unverändert, während eine `Text` -Warteschlange den Text um Wagenrückläufe ergänzt.

Viele Netzwerkdrucker können auch Daten drucken, die direkt an Port 9100 gesendet werden.

### 15.2.1. Zusammenfassung

Verkabelte Netzwerkdrucker drucken in der Regel am schnellsten und sind einfach einzurichten. Für den direkten Anschluss am Rechner wird USB wegen seiner Geschwindigkeit und Einfachheit bevorzugt. Parallele Verbindungen funktionieren, haben jedoch ihre Begrenzung in Bezug auf Kabellänge und Geschwindigkeit. Serielle Verbindungen sind schwieriger zu konfigurieren und die Verdrahtung unterscheidet sich zwischen den Modellen. Zudem müssen Baudrate und Parität bekannt sein. Glücklicherweise sind serielle Drucker selten geworden.

## 15.3. Gebräuchliche Seitenbeschreibungssprachen

Daten, die an einen Drucker gesendet werden, müssen in einer Sprache verfasst sein, die der Drucker verstehen kann. Diese Sprachen werden Seitenbeschreibungssprachen oder Page Description Languages (PDL) genannt.

**ASCII**

Schlichter ASCII-Text ist die einfachste Möglichkeit, um Daten an einen Drucker zu senden. Die Zeichen werden eins zu eins gedruckt: ein `A` in den Daten erscheint beim Druck als `A` auf dem Papier. Eine Formatierung ist nur bedingt verfügbar und es gibt keine Möglichkeit, eine Schriftart oder eine bestimmte Laufweite zu wählen. Die Einfachheit von schlichtem ASCII-Text bedeutet, dass Text ohne bzw. wenig Codierung oder Übersetzung gedruckt werden kann. Die gedruckte Ausgabe entspricht dem, was an den Drucker gesendet wurde.

Einige kostengünstige Drucker können keinen einfachen ASCII-Text drucken. Das macht sie in der Regel schwieriger einzurichten.

**PostScript®**

PostScript® ist fast das Gegenteil von ASCII. Anstelle von einfachem Text, besteht ein PostScript®-Programm aus einer Reihe von Anweisungen, die das endgültige Dokument generieren. Es können auch verschiedene Schriften und Grafiken benutzt werden. Diese Fähigkeiten haben jedoch ihren Preis. Das Programm, das die Seite generiert, muss zunächst erzeugt werden. Normalerweise wird dieses Programm durch die Anwendung erzeugt, so dass der Prozess für den Benutzer transparent bleibt.

Kostengünstige Drucker sind manchmal nicht kompatibel mit PostScript®.

**PCL (Printer Command Language)**

PCL ist eine Erweiterung von ASCII. Es enthält Escape-Sequenzen für die Formatierung, Schriftauswahl und das Drucken von Grafiken. Viele Drucker bieten Unterstützung für PCL5, einige unterstützen auch das neuere PCL6 oder PCLXL. Die neueren Versionen sind Kombinationen von PCL5 und bieten eine schnellere Druckgeschwindigkeit.

**Host-basiert**

Hersteller können die Kosten eines Druckers reduzieren, indem sie einen einfachen Prozessor und etwas Speicher verbauen. Diese Drucker sind nicht in der Lage normalen Text zu drucken. Stattdessen werden die Texte und Grafiken von einem Treiber auf dem Host-Rechner generiert und dann an den Drucker gesendet. Diese Drucker werden Host-basierte Drucker genannt.

Die Kommunikation zwischen dem Treiber und dem Drucker wird oft durch proprietäre oder nicht dokumentierte Protokolle realisiert, weshalb sie nur mit den gängigsten Betriebssystemen funktionieren.

### 15.3.1. PostScript® in eine andere Sprache konvertieren

Viele Anwendungen aus der Ports-Sammlung und FreeBSD Werkzeuge können PostScript® erzeugen. Die folgende Tabelle listet die verfügbaren Programme, um PostScript® in andere PDLs zu konvertieren:

*Tabelle 8. Ausgabe PDLs*

| Ausgabe PDL | Generiert von | Hinweis |
|---|---|---|
| PCL oder PCL5 | print/ghostscript9-base | `-sDEVICE=ljet4` für Schwarzweiß, `-sDEVICE=cljet5` für Farbe |
| PCLXL oder PCL6 | print/ghostscript9-base | `-sDEVICE=pxlmono` für Schwarzweiß, `-sDEVICE=pxlcolor` für Farbe |
| ESC/P2 | print/ghostscript9-base | `-sDEVICE=uniprint` |
| XQX | print/foo2zjs | |

### 15.3.2. Zusammenfassung

Um die Konfiguration einfach zu halten, wählen Sie einen Drucker, der PostScript® oder auch PCL unterstützt. Mit print/ghostscript9-base können diese Drucker PostScript® nativ verstehen. Wenn

der Drucker PostScript® oder PCL direkt unterstützt, können Sie auch sofort einfache ASCII-Textdateien drucken.

Zeilenbasierte Drucker wie Tintenstrahldrucker unterstützen in der Regel kein PostScript® oder PCL. Dennoch können Sie ASCII-Textdateien drucken. print/ghostscript9-base unterstützt die Sprachen dieser Drucker. Jedoch ist der Druck von Grafiken auf diesen Druckern oft sehr langsam, da aufgrund der großen Menge an Daten übertragen und ausgedruckt werden müssen.

Host-basierte Drucker sind oft schwieriger einzurichten. Einige Drucker können überhaupt nicht benutzt werden, da sie proprieräte PDLs verwerden. Solche Drucker sollten Sie nach Möglichkeit vermeiden.

Die Beschreibungen vieler PDLs finden Sie auf http://www.undocprint.org/formats/page_description_languages. Spezielle PDLs, die von einigen Druckern verwendet werden finden Sie auf http://www.openprinting.org/printers.

## 15.4. Direktes Drucken

Für den gelegentlichen Druck können die Dateien auch direkt, ohne zusätzliche Einstellungen, an den Drucker gesendet werden. Zum Beispiel kann die Datei sample.txt direkt an einen USB-Drucker gesendet werden:

```
# cp sample.txt /dev/unlpt0
```

Ob Sie direkt auf einen Netzwerkdrucker drucken können, hängt von den Fähigkeiten des Druckers ab. Die meisten akzeptieren jedoch Druckaufträge auf Port 9100, die Sie mit nc(1) an den Drucker senden können. So drucken Sie die gleiche Datei auf einem Drucker mit dem DNS-Namen *netlaser*:

```
# nc netlaser 9100 < sample.txt
```

## 15.5. LPD (Line Printer Daemon)

Drucken im Hintergrund wird *Spooling* genannt. Ein Spooler (Warteschlange) ermöglicht es dem Benutzer die Programme auf dem Rechner fortzusetzen, ohne warten zu müssen bis der Druckauftrag abgeschlossen ist.

FreeBSD enthält den Spooler namens lpd(8). Druckaufträge werden mit lpr(1) übermittelt.

### 15.5.1. Konfiguration

Erstellen Sie ein Verzeichnis zur Speicherung der Druckaufträge und setzen Sie die Berechtigungen auf diesem Verzeichnis, damit der Inhalt der Druckaufträge nicht von anderen Benutzern eingesehen werden kann:

```
# mkdir -p /var/spool/lpd/lp
# chown daemon:daemon /var/spool/lpd/lp
```

```
# chmod 770 /var/spool/lpd/lp
```

Drucker werden in /etc/printcap angelegt. Ein Eintrag für einen Drucker enthält dessen Name, Anschluss sowie weitere Einstellungen. Erstellen Sie /etc/printcap mit folgendem Inhalt:

```
lp:\                ①
    :lp=/dev/unlpt0:\   ②
    :sh:\           ③
    :mx#0:\         ④
    :sd=/var/spool/lpd/lp:\ ⑤
    :lf=/var/log/lpd-errs:  ⑥
```

① Der Name des Druckers. lpr(1) sendet Druckaufträge an den Drucker `lp`, es sei denn, ein anderer Drucker wird mit der Option `-P` angegeben. Der Standarddrucker sollte also `lp` genannt werden.

② Der Anschluss, über den der Drucker verbunden ist. Ersetzen Sie diese Zeile mit dem entsprechenden, hier aufgeführten Verbindungstyp.

③ Unterdrückt das Drucken eines Deckblattes zu Beginn des Druckauftrags.

④ Die maximale Größe des Druckauftrags wird nicht begrenzt.

⑤ Das Verzeichnis zur Speicherung der Druckdaten. Jeder Drucker verwendet ein eigenes Verzeichnis.

⑥ Die Logdatei, in welche die Fehler des Druckers geschrieben werden.

Nachdem Sie /etc/printcap erstellt haben, verwenden Sie chkprintcap(8) um die Datei auf Fehler zu testen:

```
# chkprintcap
```

Beheben Sie alle gemeldeten Fehler, bevor Sie fortfahren.

Aktivieren Sie lpd(8) in /etc/rc.conf:

```
lpd_enable="YES"
```

Starten Sie den Dienst:

```
# service lpd start
```

### 15.5.2. Drucken mit lpr(1)

Mit `lpr` werden Dokumente an den Drucker geschickt. Die Datei können Sie auf der Kommandozeile angeben, oder über eine Pipe an `lpr` schicken. Die beiden folgenden Kommandos sind gleichwertig, sie schicken den Inhalt von doc.txt an den Standarddrucker:

```
% lpr doc.txt
% cat doc.txt | lpr
```

Drucker können auch mit `-P` ausgewählt werden. Um auf einen Drucker namens *laser* zu drucken:

```
% lpr -Plaser doc.txt
```

### 15.5.3. Filter

In den bisher gezeigten Beispielen wurde lediglich eine Textdatei an den Drucker gesendet. Solange der Drucker den Inhalt dieser Dateien versteht, wird die Ausgabe korrekt gedruckt werden.

Einige Drucker sind nicht in der Lage einfachen Text zu drucken. Es kann sogar sein, das die Eingabedatei gar keinen Text enthält.

Mit Hilfe von *Filtern* können Dateien übersetzt oder verarbeitet werden. Ein typischer Anwendungsfall ist die Umwandlung der Eingabedaten in ein Format, das der Drucker verstehen kann, wie bspw. PostScript® oder PCL. Filter können auch verwendet werden um zusätzliche Funktionen hinzuzufügen, wie bspw. Seitenzahlen oder das Hervorheben von Quellcode, um die Lesbarkeit zu verbessern.

Die hier beschriebenen Filter werden *Eingabefilter* oder auch *Textfilter* genannt. Diese Filter übersetzen die eingehende Datei in verschiedene Formen. Werden Sie mit su(1) zu `root`, bevor Sie die Dateien erstellen.

Filter werden in /etc/printcap mit der Kennung `if=` festgelegt. Um /usr/local/libexec/lf2crlf als Filter einzusetzen, bearbeiten Sie /etc/printcap wie folgt:

```
lp:\
    :lp=/dev/unlpt0:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/lp:\
    :if=/usr/local/libexec/lf2crlf:\   ①
    :lf=/var/log/lpd-errs:
```

① `if=` identifiziert den Eingangsfilter, der auf den eingehenden Text angewendet werden soll.



Der Backslash am Ende der Zeilen zeigt an, das ein Eintrag für einen Drucker wirklich nur eine Zeile ist, in der die einzelnen Einträge durch einen Doppelpunkt getrennt sind. Das Beispiel hätte man auch wie folgt schreiben können:

```
lp:lp=/dev/unlpt0:sh:mx#0:sd=/var/spool/lpd/lp:if=/usr/local/libexec/lf
2crlf:lf=/var/log/lpd-errs:
```

#### Den Treppeneffekt verhindern

Typische Textdateien enthalten einen Zeilenvorschub am Ende jeder Zeile. Diese Zeilen erzeugen auf dem Drucker einen "Treppeneffekt":

```
A printed file looks
                   like the steps of a staircase
                                                scattered by the wind
```

Ein Filter kann Zeilenumbrüche in Wagenrückläufe und Zeilenumbrüche konvertieren. Erstellen Sie /usr/local/libexec/lf2crlf mit folgendem Inhalt:

```
#!/bin/sh
CR=$'\r'
/usr/bin/sed -e "s/$/${CR}/g"
```

Setzen Sie die Berechtigungen und machen Sie die Datei ausführbar:

```
# chmod 555 /usr/local/libexec/lf2crlf
```

Passen Sie /etc/printcap an, so dass der neue Filter verwendet wird:

```
:if=/usr/local/libexec/lf2crlf:\
```

Drucken Sie nochmal die gleiche Datei, um den Filter zu testen.

#### Mit print/enscript normalen Text auf PostScript®-Druckern drucken

GNUEnscript wandelt Textdateien in formatiertes PostScript® um, die dann auf PostScript®-Druckern gedruckt werden können. Das Programm fügt auch Seitenzahlen und Zeilenumbrüche hinzu und stellt andere Funktionen bereit, um gedruckte Textdateien besser lesbar zu machen. Abhängig vom Papierformat können Sie entweder print/enscript-letter oder print/enscript-a4 aus der Ports-Sammlung installieren.

Erstellen Sie /usr/local/libexec/enscript mit diesem Inhalt:

```
#!/bin/sh
/usr/local/bin/enscript -o -
```

Setzen Sie die Berechtigungen und machen Sie die Datei ausführbar:

```
# chmod 555 /usr/local/libexec/enscript
```

Bearbeiten Sie /etc/printcap um den neuen Filter zu verwenden:

```
:if=/usr/local/libexec/enscript:\
```

Testen Sie den Filter, indem Sie eine einfache Textdatei drucken.

#### PostScript® auf PCL-Druckern drucken

Viele Programme erzeugen PostScript®-Dokumente. Allerdings können kostengünstige Drucker oft nur Textdateien oder PCL verstehen. Dieser Filter wandelt PostScript®-Dateien in PCL um, bevor die Datei an den Drucker geschickt wird. Installieren Sie den Ghostscript PostScript® Interpreter print/ghostscript9-base aus der Ports-Sammlung.

Erstellen Sie /usr/local/libexec/ps2pcl mit diesem Inhalt:

```
#!/bin/sh
/usr/local/bin/gs -dSAFER -dNOPAUSE -dBATCH -q -sDEVICE=ljet4 -sOutputFile=- -
```

Setzen Sie die Berechtigungen und machen Sie die Datei ausführbar:

```
# chmod 555 /usr/local/libexec/ps2pcl
```

Die PostScript®-Eingabe wird von dem Skript erst in PCL umgewandelt, bevor es an den Drucker geschickt wird.

Bearbeiten Sie /etc/printcap um den neuen Filter zu verwenden:

```
:if=/usr/local/libexec/ps2pcl:\
```

Testen Sie den Filter mit einem kleinen PostScript®-Programm.

```
% printf "%%\!PS \n /Helvetica findfont 18 scalefont setfont \
72 432 moveto (PostScript printing successful.) show showpage \004" | lpr
```

#### Intelligente Filter

Ein Filter kann sehr nützlich sein, wenn er die Eingabe erkennt und sie automatisch in ein für den Drucker verständliches Format umwandelt. Die ersten beiden Zeichen in einer PostScript®-Datei sind in der Regel `%!`. Ein Filter ist in der Lage diese beiden Zeichen zu erkennen. PostScript®-Dateien können unverändert an einen PostScript®-Drucker geschickt werden. Textdateien können, wie eben gezeigt, mit Enscript in PostScript® umgewandelt werden. Erstellen Sie /usr/local/libexec/psif mit diesem Inhalt:

```
#!/bin/sh
#
#  psif - Print PostScript or plain text on a PostScript printer
```

```
#
IFS="" read -r first_line
first_two_chars=`expr "$first_line" : '\(..\)'`

case "$first_two_chars" in
%!)
    # %! : PostScript job, print it.
    echo "$first_line" && cat && exit 0
    exit 2
    ;;
*)
    # otherwise, format with enscript
    ( echo "$first_line"; cat ) | /usr/local/bin/enscript -o - && exit 0
    exit 2
    ;;
esac
```

Setzen Sie die Berechtigungen und machen Sie die Datei ausführbar:

```
# chmod 555 /usr/local/libexec/psif
```

Bearbeiten Sie /etc/printcap um den neuen Filter zu verwenden:

```
:if=/usr/local/libexec/psif:\
```

Um den Filter zu testen, drucken Sie PostScript®- und einfache Textdateien.

### 15.5.4. Mehrere Warteschlangen

Die Einträge in /etc/printcap sind nichts anderes als Definitionen von Warteschlangen. Für jeden Drucker können eine oder mehrere Warteschlangen definiert werden. Kombiniert mit Filtern bieten mehrere Warteschlangen eine bessere Kontrolle über die Druckaufträge.

Als Beispiel dient ein vernetzter PostScript®-Laserdrucker in einem Büro. Die meisten Benutzer möchten einfache Textdateien drucken, aber ein paar fortgeschrittene Anwender sollen in der Lage sein, PostScript®-Dateien direkt zu drucken. Hierfür werden zwei Einträge für den Drucker in /etc/printcap erstellt:

```
textprinter:\
    :lp=9100@officelaser:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/textprinter:\
    :if=/usr/local/libexec/enscript:\
    :lf=/var/log/lpd-errs:

psprinter:\
```

```
    :lp=9100@officelaser:\
    :sh:\
    :mx#0:\
    :sd=/var/spool/lpd/psprinter:\
    :lf=/var/log/lpd-errs:
```

Dokumente, die zum `textprinter` geschickt werden, werden wie im vorherigen Beispiel durch den Filter /usr/local/libexec/enscript formatiert. Fortgeschrittene Anwender können PostScript®-Dateien direkt auf dem Drucker `psprinter` drucken, wo keine Filterung stattfindet.

Mit mehreren Warteschlangen können Sie einen direkten Zugriff auf alle Arten von Druckerfunktionen zur Verfügung stellen. Ein Duplex-Drucker könnte zwei Warteschlangen verwenden, eine für den gewöhnlichen Druck und eine für den Duplexdruck.

### 15.5.5. Druckaufträge steuern und überwachen

Es stehen verschiedene Programme zur Verfügung um Druckaufträge zu überwachen und den Druckbetrieb zu steuern.

#### lpq(1)

lpq(1) zeigt den Status der Druckaufträge des Benutzers an. Druckaufträge anderer Benutzer werden nicht angezeigt.

Dieser Befehl zeigt die anstehenden Druckaufträge eines Benutzers für einen Drucker an:

```
% lpq -Plp
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     0    (standard input)                      12792 bytes
```

Der folgende Befehl zeigt die anstehenden Druckaufträge eines Benutzers für alle Drucker an:

```
% lpq -a
lp:
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     1    (standard input)                      27320 bytes

laser:
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     287  (standard input)                      22443 bytes
```

#### lprm(1)

Mit lprm(1) können Druckaufträge gelöscht werden. Normale Benutzer dürfen lediglich ihre eigenen Aufträge löschen. `root` kann hingegen jeden beliebigen Auftrag löschen.

Dieser Befehl löscht alle anstehenden Druckaufträge eines Druckers:

```
# lprm -Plp -
dfA002smithy dequeued
cfA002smithy dequeued
dfA003smithy dequeued
cfA003smithy dequeued
dfA004smithy dequeued
cfA004smithy dequeued
```

Mit dem folgenden Befehl löschen Sie einen bestimmten Druckauftrag. Benutzen Sie lpq(1), um die Nummer des Auftrags zu finden.

```
% lpq
Rank   Owner      Job  Files                                 Total Size
1st    jsmith     5    (standard input)                      12188 bytes
% lprm -Plp 5
dfA005smithy dequeued
cfA005smithy dequeued
```

### lpc(8)

Mit lpc(8) kann der Druckerstatus überprüft und verändert werden. `lpc` wird zusammen mit einem Kommando und optional mit einem Druckernamen aufgerufen. Mit `all` können alle Drucker angesprochen werden, auf denen das Kommando ausgeführt werden soll. Normale Benutzer können sich den Status mit lpc(8) ansehen. Nur `root` darf Kommandos ausführen, die den Status des Druckers verändern.

Dieser Befehl zeigt den Status von allen Druckern an:

```
% lpc status all
lp:
    queuing is enabled
    printing is enabled
    1 entry in spool area
    printer idle
laser:
    queuing is enabled
    printing is enabled
    1 entry in spool area
    waiting for laser to come up
```

Der Drucker kann die Annahme neuer Druckaufträge verweigern. Anschließend sollen Aufträge wieder akzeptiert werden:

```
# lpc stop lp
lp:
    printing disabled
```

```
# lpc start lp
lp:
    printing enabled
    daemon started
```

Starten Sie den Drucker nach einem Fehler neu:

```
# lpc restart lp
lp:
    no daemon to abort
    printing enabled
    daemon restarted
```

Schalten Sie die Warteschlange aus und deaktivieren Sie den Druck. Sie können den Benutzern gleichzeitig eine Nachricht hinterlassen:

```
# lpc down lp Ersatzteile werden am Montag ankommen
lp:
    printer and queuing disabled
    status message is now: Ersatzteile werden am Montag ankommen
```

Reaktivieren Sie den Drucker:

```
# lpc up lp
lp:
    printing enabled
    daemon started
```

Weitere Kommandos und Optionen finden Sie in lpc(8).

### 15.5.6. Gemeinsam genutzte Drucker

In Unternehmen und Schulen werden Drucker häufig von mehreren Benutzern genutzt. Es werden zusätzliche Funktionen angeboten, um die gemeinsame Nutzung von Druckern zu erleichtern.

#### Aliase

Der Druckername wird in der ersten Zeile von /etc/printcap festgelegt. Weitere Namen oder *Aliase* können nach dem Druckernamen hinzugefügt werden. Aliase werden vom Namen durch das Pipe-Zeichen | getrennt:

```
lp|repairsprinter|salesprinter:\
```

Anstelle des Druckernamens können Aliase verwendet werden. Zum Beispiel können Mitarbeiter der Verkaufsabteilung wie folgt auf ihren Drucker drucken:

```
% lpr -Psalesprinter sales-report.txt
```

Mitarbeiter der Reparaturabteilung drucken auf dem Drucker mit:

```
% lpr -Prepairsprinter repairs-report.txt
```

Alle Dokumente werden auf diesem einen Drucker gedruckt. Wenn die Verkaufsabteilung größer wird und die Abteilung einen eigenen Drucker benötigt, kann der Alias entfernt und für einen neuen Drucker verwendet werden. Die Mitarbeiter in beiden Abteilungen benutzen zum Drucken weiterhin die gleichen Befehle, nur dass die Aufträge der Verkaufsabteilung jetzt zum neuen Drucker gesendet werden.

**Deckblätter**

Bei einem viel benutzten Drucker kann es für die Anwender schwierig sein, ihre Dokumente in einem großen Papierstapel wiederzufinden. Um dieses Problem zu lösen, können *Deckblätter* verwendet werden. Dabei wird vor jedem Druckauftrag ein Deckblatt mit dem Benutzernamen und dem Dokumentnamen gedruckt. Deckblätter werden manchmal auch als *Banner* oder *Trennseite* bezeichnet.

Das Aktivieren der Deckblätter hängt davon ab, ob der Drucker direkt über ein USB, paralleles oder serielles Kabel, oder über ein Netzwerk mit dem Rechner verbunden ist.

Wenn der Drucker direkt verbunden ist, aktivieren Sie die Deckblätter durch Entfernen der Zeile `:sh:\` (Supress Header) in /etc/printcap. Diese Deckblätter verwenden lediglich einen Zeilenvorschub für neue Zeilen. Einige Drucker benötigen den Filter /usr/shared/examples/printing/hpif um den Treppeneffekt zu vermeiden. Der Filter konfiguriert PCL-Drucker so, dass sowohl Zeilenumbrüche als auch Zeilenvorschübe verwendet werden, wenn ein Zeilenvorschub empfangen wird.

Für Netzwerkdrucker müssen Deckblätter auf dem Drucker selbst konfiguriert werden, da Einträge für Deckblätter in /etc/printcap ignoriert werden. Die Einstellungen sind über einen Webbrowser zugänglich und stehen in der Regel auf der Hauptseite der Konfigurations-Webseite zur Verfügung.

### 15.5.7. Referenzen

Beispieldateien: /usr/shared/examples/printing/.

Das *4.3BSD Line Printer Spooler Manual*, /usr/shared/doc/smm/07.lpd/paper.ascii.gz.

Manualpages: printcap(5), lpd(8), lpr(1), lpc(8), lprm(1), lpq(1).

## 15.6. Andere Drucksysteme

Neben dem in FreeBSD enthaltenen lpd(8) existieren noch weitere Drucksysteme. Diese Systeme bieten zusätzliche Funktionen und Unterstützung für andere Protokolle.

### 15.6.1. CUPS (Common UNIX® Printing System)

CUPS ist ein beliebtes Drucksystem, das für viele Betriebssysteme erhältlich ist. CUPS unter FreeBSD wird in einem separaten Artikel beschrieben: CUPS on FreeBSD.

### 15.6.2. HPLIP

Hewlett Packard stellt ein Drucksystem zur Verfügung, das viele ihrer Drucker unterstützt. Der Port heißt print/hplip. Die Webseite befindet sich unter http://hplipopensource.com/hplip-web/index.html. Der FreeBSD-Port kümmert sich um alle Details während der Installation. Informationen zur Konfiguration finden Sie unter http://hplipopensource.com/hplip-web/install/manual/hp_setup.html.

### 15.6.3. LPRng

LPRng wurde als eine verbesserte Alternative zu lpd(8) entwickelt. Der Port heißt sysutils/LPRng. Weitere Informationen und Dokumentation finden Sie unter https://lprng.sourceforge.net/.

# Kapitel 16. Linux®-Binärkompatibilität

## 16.1. Übersicht

FreeBSD bietet Binärkompatibilität zu Linux®, so dass Benutzer Linux® Anwendungen auf einem FreeBSD-System installieren und ausführen können, ohne die Binärdatei ändern zu müssen. Es wurde sogar berichtet, dass in einigen Situationen Linux® Anwendungen auf FreeBSD besser laufen als unter Linux®.

Allerdings werden einige Linux®-spezifischen Merkmale nicht von FreeBSD unterstützt. Linux®-Anwendungen, die i386™-spezifische Aufrufe, wie bspw. die Aktivierung des virtuellen 8086-Modus verwenden, werden derzeit nicht unterstützt.

Die Unterstützung für 64-Bit-Binärkompatibilität für Linux® wurde in FreeBSD 10.3 hinzugefügt.

Nach dem Lesen dieses Kapitels werden Sie wissen:

- Wie Sie die Linux®-Binärkompatibilität aktivieren.
- Wie zusätzliche Linux®-Systembibliotheken installiert werden.
- Wie Sie Linux®-Anwendungen unter FreeBSD installieren.
- Wie die Linux®-Binärkompatibilität unter FreeBSD implementiert ist.

Bevor Sie dieses Kapitel lesen, sollten Sie wissen:

- Wie Sie Software von Drittanbietern installieren.

## 16.2. Konfiguration der Linux®-Binärkompatibilität

Die Linux®-Binärkompatibilität ist per Voreinstellung nicht aktiviert und auch Linux®-Bibliotheken werden nicht installiert. Linux®-Bibliotheken können entweder manuell, oder aus der FreeBSD Ports-Sammlung installiert werden.

Bevor Sie versuchen den Port zu bauen, laden Sie das Linux®-Kernelmodul, da ansonsten der Bau fehlschlägt:

```
# kldload linux
```

Für 64-Bit Kompatibilität:

```
# kldload linux64
```

Prüfen Sie, ob das Modul geladen wurde:

```
% kldstat
Id Refs Address    Size     Name
```

```
 1    2 0xc0100000 16bdb8   kernel
 7    1 0xc24db000 d000     linux.ko
```

Der einfachste Weg um einen Basissatz von Linux®-Bibliotheken und Binärdateien auf einem FreeBSD-System zu installieren, ist über den Port oder das Paket emulators/linux_base-c7. So installieren Sie das Paket:

```
# pkg install emulators/linux_base-c7
```

Wollen Sie die Linux®-Binärkompatibilität beim Systemstart aktivieren, fügen Sie folgende Zeile in /etc/rc.conf hinzu:

```
linux_enable="YES"
```

Auf 64-Bit Maschinen wird das Modul für die 64-Bit Emulation automatisch von /etc/rc.d/abi geladen.

Seitdem die Linux®-Binärkompatibilität Unterstützung für die Ausführung von 32- und 64-Bit-Linux®-Binärdateien erhalten hat, ist es nicht mehr möglich, die Emulationsfähigkeit in einen angepassten Kernel zu integrieren.

### 16.2.1. Manuelle Installation zusätzlicher Bibliotheken

Wenn sich eine Linux®-Anwendung über fehlende Bibliotheken beschwert nachdem die Linux®-Binärkompatibilität installiert wurde, finden Sie heraus welche Bibliothken die Anwendung benötigt und installieren Sie diese manuell.

Mit `ldd` können Sie unter Linux® bestimmen, welche gemeinsam benutzten Bibliotheken eine Anwendung benötigt. Wenn Sie herausfinden wollen, welche Bibliotheken `linuxdoom` benötigt, können Sie folgenden Befehl auf einem Linux®-System ausführen, welches Doom installiert hat:

```
% ldd linuxdoom
libXt.so.3 (DLL Jump 3.1) => /usr/X11/lib/libXt.so.3.1.0
libX11.so.3 (DLL Jump 3.1) => /usr/X11/lib/libX11.so.3.1.0
libc.so.4 (DLL Jump 4.5pl26) => /lib/libc.so.4.6.29
```

Kopieren Sie alle Dateien aus der letzten Spalte der Ausgabe von einem Linux®-System auf das FreeBSD-System in das Verzeichnis /compat/linux. Nach dem Kopieren erstellen Sie symbolische Links auf die Namen in der ersten Spalte. In diesem Beispiel werden folgende Dateien auf dem FreeBSD-System installiert:

```
/compat/linux/usr/X11/lib/libXt.so.3.1.0
/compat/linux/usr/X11/lib/libXt.so.3 -> libXt.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3.1.0
/compat/linux/usr/X11/lib/libX11.so.3 -> libX11.so.3.1.0
```

```
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

Wenn Sie bereits eine Linux®-Bibliothek einer zur ersten Spalte passenden Hauptversionsnummer besitzen, muss sie nicht mehr kopiert werden, da die bereits vorhandene Version funktionieren sollte. Hat die Bibliothek jedoch eine neuere Versionsnummer, sollten Sie sie dennoch kopieren. Sie können die alte Version löschen, solange Sie einen symbolischen Link auf die neue Version anlegen.

Folgende Bibliotheken existieren bereits auf dem FreeBSD-System:

```
/compat/linux/lib/libc.so.4.6.27$
/compat/linux/lib/libc.so.4 -> libc.so.4.6.27
```

`ldd` zeigt an, dass eine Anwendung eine neuere Version benötigt:

```
libc.so.4 (DLL Jump 4.5pl26) -> libc.so.4.6.29
```

Wenn diese Bibliotheken sich nur um ein oder zwei Stellen in der Unterversionsnummer unterscheiden, sollte das Programm dennoch mit der älteren Version funktionieren. Wenn Sie wollen, können Sie die bestehende libc.so durch die neuere Version ersetzen:

```
/compat/linux/lib/libc.so.4.6.29
/compat/linux/lib/libc.so.4 -> libc.so.4.6.29
```

Der Mechanismus der symbolischen Links wird nur für Linux®-Binärdateien benötigt. Nach einer Weile wird es eine ausreichende Menge an Linux®-Bibliotheken auf dem System geben, sodass Sie neu installierte Linux®-Anwendungen ohne zusätzlichen Aufwand auf dem System laufen lassen können.

### 16.2.2. Linux® ELF-Binärdateien installieren

ELF-Binärdateien benötigen manchmal eine zusätzliche "Kennzeichnung". Wenn Sie versuchen, eine nicht gekennzeichnete ELF-Binärdatei auszuführen, werden Sie eine Fehlermeldung ähnlich der folgenden erhalten:

```
% ./my-linux-elf-binary
ELF binary type not known
Abort
```

Damit der FreeBSD-Kernel eine Linux®-ELF-Datei von einer FreeBSD-ELF-Datei unterscheiden kann, gibt es das Werkzeug brandelf(1).

```
% brandelf -t Linux my-linux-elf-binary
```

Die GNU Werkzeuge schreiben nun automatisch die passende Kennzeichnungsinformation in die ELF-Binärdateien, so dass Sie diesen Schritt in Zukunft nur noch selten benötigen.

### 16.2.3. Installieren einer RPM-basierten Linux®-Anwendung

Wenn Sie eine Linux® RPM-basierte Anwendung installieren möchten, installieren Sie zunächst den Port oder das Paket archivers/rpm4. Anschließend kann der Superuser das folgende Kommando benutzen, um ein .rpm zu installieren:

```
# cd /compat/linux
# rpm2cpio < /pfad/zum/linux.archiv.rpm | cpio -id
```

Fall notwendig, benutzen Sie `brandelf` auf den installierten ELF-Binärdateien. Beachten Sie, dass dies eine saubere Deinstallation verhindert.

### 16.2.4. Namensauflösung konfigurieren

Wenn DNS nicht funktioniert, oder die folgende Fehlermeldung erscheint:

```
resolv+: "bind" is an invalid keyword resolv+:
"hosts" is an invalid keyword
```

müssen Sie /compat/linux/etc/host.conf wie folgt bearbeiten:

```
order hosts, bind
multi on
```

Diese Reihenfolge legt fest, dass zuerst /etc/hosts und anschließend DNS durchsucht werden. Wenn /compat/linux/etc/host.conf nicht vorhanden ist, nutzen Linux®-Anwendungen /etc/host.conf und beschweren sich über die inkompatible FreeBSD-Syntax. Wenn Sie in /etc/resolv.conf keinen Nameserver konfiguriert haben, sollten Sie den Eintrag `bind` entfernen.

## 16.3. Weiterführende Themen

Dieser Abschnitt beschreibt wie die Linux®-Binärkompatibilität funktioniert. Die folgenden Informationen stammen aus einer E-Mail, die von Terry Lambert (tlambert@primenet.com) an FreeBSD chat geschrieben wurde (Message ID: <199906020108.SAA07001@usr09.primenet.com>).

FreeBSD verfügt über eine "execution class loader" genannte Abstraktion. Dabei handelt es sich um einen Eingriff in den execve(2) Systemaufruf.

Historisch gesehen untersuchte der einzige, auf UNIX®-Plattformen vorhandene Lader die "magische Zahl" (in der Regel die ersten 4 oder 8 Bytes der Datei), um festzustellen, ob der Binärtyp dem System bekannt war. War dies der Fall, wurde der Binärlader aufgerufen.

Wenn es sich nicht um den zum System gehörigen Binärtyp handelte, gab execve(2) einen Fehler

zurück, und die Shell versuchte stattdessen, die Datei als Shell-Befehl auszuführen. Dabei wurde als Standardeinstellung "was auch immer die aktuelle Shell ist" festgelegt.

Später wurde ein Hack in sh(1) eingefügt, der die zwei ersten Zeichen untersuchte. Wenn diese `:\n` entsprachen, wurde stattdessen die csh(1)-Shell aufgerufen.

FreeBSD verfügt über eine Liste von Ladern, anstelle eines einzigen, auf `#!` zurückgreifenden Laders, um Shell-Interpreter oder Shell-Skripte auszuführen.

Für die Linux® ABI-Unterstützung erkennt FreeBSD die magische Zahl als ELF-Binärdatei. Der ELF-Lader sucht nach einer speziellen *Kennzeichnung*, die aus einem Kommentarabschnitt in der ELF-Datei besteht, und die in SVR4/Solaris™ ELF Binärdateien nicht vorhanden ist.

Damit Linux®-Binärdateien unter FreeBSD funktionieren, müssen sie mit brandelf(1) als `Linux` *gekennzeichnet* werden:

```
# brandelf -t Linux file
```

Wenn der ELF-Lader die `Linux`-Kennzeichnung sieht, wird ein Zeiger in der `proc`-Struktur ersetzt. Alle Systemaufrufe werden durch diesen Zeiger indiziert. Der Prozess wird weiterhin speziell gekennzeichnet, so dass der Trap-vector im Signal-trampoline-code eine spezielle Behandlung erfährt und das Linux®-Kernelmodul verschiedene kleinere Korrekturen vornehmen kann.

Der Linux®-Systemaufrufvektor enthält neben anderen Dingen eine Liste der `sysent[]`-Einträge, deren Adressen sich im Kernelmodul befinden.

Wenn ein Linux®-Programm einen Systemaufruf ausführt, dereferenziert die Trap-Behandlungsroutine den Zeiger für den Systemaufruf aus der `proc`-Struktur und erhält damit die Linux®-Eintrittspunkte für den Systemaufruf.

Zusätzlich *verändert* der Linux®-Modus die Systempfade dynamisch; genauso, wie dies die Option `union` beim Einbinden von Dateisystemen macht. Zuerst wird die Datei im Verzeichnis /compat/linux/Originalpfad gesucht, wenn sie dort nicht gefunden wurde, wird sie im Verzeichnis /Originalpfad gesucht. Dadurch wird sichergestellt, dass Binärdateien, die zur Ausführung andere Binärdateien benötigen, ausgeführt werden können (so dass alle Linux®-Werkzeuge unter der ABI laufen). Dies bedeutet auch, dass Linux®-Binärdateien FreeBSD-Binärdateien laden und ausführen können, wenn keine passenden Linux®-Binärdateien vorhanden sind. Ein in /compat/linux plaziertes uname(1) kann damit Linux®-Programmen vorgaukeln, dass sie auf einem Linux®-System laufen.

Im Endeffekt gibt es einen Linux®-Kernel innerhalb des FreeBSD-Kernels. Die Sprungtabellen für Linux®- beziehungsweise FreeBSD-Systemaufrufe verweisen allerdings auf dieselben Funktionen, die Kerneldienste wie Dateisystemoperationen, Operationen für den virtuellen Speicher, Signalübermittlung und System V IPC bereitstellen. Der einzige Unterschied ist, dass Binärdateien unter FreeBSD FreeBSD-*glue*-Funktionen verwendet werden. Linux®-Binärdateien hingegen verwenden die Linux®-*glue*-Funktionen. FreeBSD-*glue*-Funktionen sind statisch in den Kernel gelinkt, Linux®-*glue*-Funktionen sind statisch gelinkt oder können über ein ladbares Kernelmodul eingebunden werden.

Technisch gesehen ist dies nicht wirklich eine Emulation, sondern eine ABI-Implementation. Es wird manchmal "Linux® Emulation" genannt, da es zu einer Zeit implementiert wurde, in der es kein anderes Wort gab, das beschrieb, was vor sich ging. Es war falsch zu behaupten, FreeBSD würde Linux®-Binärprogramme ausführen, da der Code nicht unter FreeBSD übersetzt wurde.

params: path: "/books/handbook/partiii/" :leveloffset: +1

# Teil III: Konfiguration und Tuning

# Kapitel 17. Übersicht

Die richtige Systemkonfiguration ist einer der wichtigsten Aspekte unter FreeBSD. Dieses Kapitel beschreibt die Konfiguration von FreeBSD sowie Maßnahmen zur Leistungssteigerung von FreeBSD-Systemen.

Nachdem Sie dieses Kapitel durchgearbeitet haben, werden Sie Folgendes wissen:

- Die Grundlagen der Konfiguration von rc.conf und die Skripte zum Starten von Anwendungen in /usr/local/etc/rc.d.
- Wie Sie Netzwerkkarten konfigurieren und testen.
- Wie Sie virtuelle Hosts und Netzwerkgeräte konfigurieren.
- Wie Sie die verschiedenen Konfigurationsdateien in /etc benutzen.
- Wie Sie mit FreeBSD mit sysctl(8)-Variablen einstellen können.
- Wie Sie die Platten-Performance einstellen und Kernel-Parameter modifizieren können.

Bevor Sie dieses Kapitel lesen, sollten Sie

- die Grundlagen von UNIX® und FreeBSD (Grundlagen des FreeBSD Betriebssystems) verstehen.
- Damit vertraut sein, wie Sie einen Kernel konfigurieren und kompilieren (Konfiguration des FreeBSD-Kernels).

# Kapitel 18. Start von Diensten

Viele Benutzer installieren Software Dritter auf FreeBSD mithilfe der Ports-Sammlung. Häufig soll die Software bei einem Systemstart mitgestartet werden. Beispielsweise sollen die Dienste mail/postfix oder www/apache22 nach einem Systemstart laufen. Dieser Abschnitt stellt die Startprozeduren für Software Dritter vor.

Unter FreeBSD werden die meisten der im System enthaltenen Dienste wie cron(8) mithilfe von Systemskripten gestartet.

## 18.1. Dienste über das rc.d-System starten

Mit rc.d lässt sich der Start von Anwendungen besser steuern und es sind mehr Funktionen verfügbar. Mit den in Dienste unter FreeBSD verwalten besprochenen Schlüsselwörtern können Anwendungen in einer bestimmten Reihenfolge gestartet werden und Optionen können in rc.conf statt fest im Startskript der Anwendung festgelegt werden. Ein einfaches Startskript sieht wie folgt aus:

```
#!/bin/sh
#
# PROVIDE: utility
# REQUIRE: DAEMON
# KEYWORD: shutdown

. /etc/rc.subr

name=utility
rcvar=utility_enable

command="/usr/local/sbin/utility"

load_rc_config $name

#
# DO NOT CHANGE THESE DEFAULT VALUES HERE
# SET THEM IN THE /etc/rc.conf FILE
#
utility_enable=${utility_enable-"NO"}
pidfile=${utility_pidfile-"/var/run/utility.pid"}

run_rc_command "$1"
```

Dieses Skript stellt sicher, dass `utility` nach den `DAEMON`-Pseudodiensten gestartet wird. Es stellt auch eine Methode bereit, die Prozess-ID (PID) der Anwendung in einer Datei zu speichern.

In /etc/rc.conf könnte für diese Anwendung die folgende Zeile stehen:

```
utility_enable="YES"
```

Die Methode erleichtert den Umgang mit Kommandozeilenargumenten, bindet Funktionen aus /etc/rc.subr ein, ist kompatibel zu rcorder(8) und lässt sich über rc.conf leichter konfigurieren.

## 18.2. Andere Arten, um Dienste zu starten

Andere Dienste können über inetd(8) gestartet werden. Die Konfiguration von inetd(8) wird in "Der inetd Super-Server" ausführlich beschrieben.

Systemdienste können auch mit cron(8) gestartet werden. Dieser Ansatz hat einige Vorteile; nicht zuletzt, weil cron(8) die Prozesse unter dem Eigentümer der crontab startet, ist es möglich, dass Dienste von normalen Benutzern gestartet und gepflegt werden können.

Für die Zeitangabe in cron(8) kann `@reboot` eingesetzt werden. Damit wird das Kommando gestartet, wenn cron(8) kurz nach dem Systemboot gestartet wird.

# Kapitel 19. cron(8) konfigurieren

Ein sehr nützliches Werkzeug von FreeBSD ist cron. Dieses Programm läuft im Hintergrund und überprüft fortlaufend /etc/crontab und /var/cron/tabs. In diesen Dateien wird festgelegt, welche Programme zu welchem Zeitpunkt von cron ausgeführt werden sollen. Jede Zeile in diesen Dateien definiert eine auszuführende Aufgabe, die auch als *Cronjob* bezeichnet wird.

Das Werkzeug verwendet zwei verschiedene Konfigurationsdateien: die System-crontab, welche nicht verändert werden sollte und die Benutzer-crontabs, die nach Bedarf erstellt und geändert werden können. Das Format, dass von diesen beiden Dateien verwendet wird, ist in crontab(5) dokumentiert. Das Format der System-crontab in /etc/crontab enthält das Feld `who`, das in der Benutzer-crontab nicht existiert. Dieses Feld gibt den Benutzer an, mit dem die Aufgabe ausgeführt wird. Die Aufgaben in den Benutzer-crontabs laufen unter dem Benutzer, der die crontab erstellt hat.

Benutzer-crontabs erlauben es den Benutzern, ihre eigenen Aufgaben zu planen. Der Benutzer `root` kann auch seine eigene Benutzer-crontab haben, um Aufgaben zu planen, die nicht in der System-crontab existieren.

Hier ist ein Beispieleintrag aus der System-crontab, /etc/crontab:

```
# /etc/crontab - root's crontab for FreeBSD
#
# $FreeBSD$
①
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin ②
#
#minute hour    mday    month   wday    who command ③
#
*/5 *   *   *   *   root    /usr/libexec/atrun ④
```

① Das Zeichen `#` am Zeilenanfang leitet einen Kommentar ein. Benutzen Sie Kommentare, um die Funktion eines Eintrags zu erläutern. Kommentare müssen in einer extra Zeile stehen. Sie können nicht in derselben Zeile wie ein Kommando stehen, da sie sonst Teil des Kommandos wären. Leerzeilen in dieser Datei werden ignoriert.

② Umgebungsvariablen werden mit dem Gleichheits-Zeichen (`=`) festgelegt. Im Beispiel werden die Variablen `SHELL`, `PATH` und `HOME` definiert. Wenn die Variable `SHELL` nicht definiert wird, benutzt cron die Bourne Shell. Wird die Variable `PATH` nicht gesetzt, müssen alle Pfadangaben absolut sein, da es keinen Vorgabewert für `PATH` gibt.

③ In dieser Zeile werden sieben Felder der System-crontab beschrieben: `minute`, `hour`, `mday`, `month`, `wday`, `who` und `command`. Das Feld `minute` legt die Minute fest in der die Aufgabe ausgeführt wird, das Feld `hour` die Stunde, das Feld `mday` den Tag des Monats. Im Feld `month` wird der Monat und im Feld `wday` der Wochentag festgelegt. Alle Felder müssen numerische Werte enthalten und die Zeitangaben sind im 24-Stunden-Format. Das Zeichen `*` repräsentiert dabei alle möglichen Werte für dieses Feld. Das Feld `who` gibt es nur in der System-crontab und gibt den Account an, unter dem das Kommando laufen soll. Im letzten Feld wird schließlich das auszuführende Kommando

angegeben.

④ Diese Zeile definiert die Werte für den Cronjob. Die Zeichenfolge `*/5` gefolgt von mehreren `*`-Zeichen bedeutet, dass `/usr/libexec/atrun` von `root` alle fünf Minuten aufgerufen wird.Bei den Kommandos können beliebig viele Optionen angegeben werden. Wenn das Kommando zu lang ist und auf der nächsten Zeile fortgesetzt werden soll, muss am Ende der Zeile das Fortsetzungszeichen (`\`) angegeben werden.

## 19.1. Eine Benutzer-crontab erstellen

Rufen Sie `crontab` im Editor-Modus auf, um eine Benutzer-crontab zu erstellen:

```
% crontab -e
```

Dies wird die crontab des Benutzers mit dem voreingestellten Editor öffnen. Wenn der Benutzer diesen Befehl zum ersten Mal ausführt, wird eine leere Datei geöffnet. Nachdem der Benutzer eine crontab erstellt hat, wird die Datei mit diesem Kommando zur Bearbeitung geöffnet.

Es empfiehlt sich, die folgenden Zeilen an den Anfang der crontab-Datei hinzuzufügen, um die Umgebungsvariablen zu setzen und die einzelnen Felder zu beschreiben:

```
SHELL=/bin/sh
PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin
# Order of crontab fields
# minute    hour    mday    month   wday    command
```

Fügen Sie dann für jedes Kommando oder Skript eine Zeile hinzu, mit der Angabe wann das Kommando ausgeführt werden soll. In diesem Beispiel wird ein Bourne Shell Skript täglich um 14:00 Uhr ausgeführt. Da der Pfad zum Skript nicht in `PATH` enthalten ist, wird der vollständige Pfad zum Skript angegeben:

```
0   14  *   *   *   /usr/home/dru/bin/mycustomscript.sh
```



Bevor Sie ein eigenes Skript verwenden, stellen Sie sicher, dass es ausführbar ist und dass es mit den wenigen Umgebungsvariablen von cron funktioniert. Um die Umgebung nachzubilden, die der obige cron-Eintrag bei der Ausführung verwenden würde, benutzen Sie dieses Kommando:

```
% env -i SHELL=/bin/sh PATH=/etc:/bin:/sbin:/usr/bin:/usr/sbin HOME
=/home/dru LOGNAME=dru /usr/home/dru/bin/mycustomscript.sh
```

Die Umgebung von cron wird in crontab(5) beschrieben. Es ist wichtig, dass sichergestellt wird, dass die Skripte in der Umgebung von cron korrekt arbeiten, besonders wenn Befehle enthalten sind, welche Dateien mit Wildcards löschen.

Wenn Sie mit der Bearbeitung der crontab fertig sind, speichern Sie die Datei. Sie wird automatisch installiert und cron wird die darin enthalten Cronjobs zu den angegebenen Zeiten ausführen. Um die Cronjobs in einer crontab aufzulisten, verwenden Sie diesen Befehl:

```
% crontab -l
0   14  *   *   *   /usr/home/dru/bin/mycustomscript.sh
```

Um alle Cronjobs einer Benutzer-crontab zu löschen, verwenden Sie diesen Befehl:

```
% crontab -r
remove crontab for dru? y
```

# Kapitel 20. Dienste unter FreeBSD verwalten

FreeBSD verwendet die vom rc(8)-System bereit gestellten Startskripten beim Systemstart und für die Verwaltung von Diensten. Die Skripte sind in /etc/rc.d abgelegt und bieten grundlegende Dienste an, die über die Optionen `start`, `stop` und `restart` des service(8) Kommandos kontrolliert werden können. Beispielsweise kann sshd(8) mit dem nachstehenden Kommando neu gestartet werden:

```
# service sshd restart
```

Analog können Sie andere Dienste starten und stoppen. Normalerweise werden die Dienste beim Systemstart über Einträge in der Datei rc.conf(5) automatisch gestartet. natd(8) wird zum Beispiel mit dem folgenden Eintrag in /etc/rc.conf aktiviert:

```
natd_enable="YES"
```

Wenn dort bereits die Zeile `natd_enable="NO"` existiert, ändern Sie `NO` in `YES`. Die rc(8)-Skripten starten, wie unten beschrieben, auch abhängige Dienste.

Da das rc(8)-System primär zum automatischen Starten und Stoppen von Systemdiensten dient, funktionieren die Optionen `start`, `stop` und `restart` nur, wenn die entsprechenden Variablen in /etc/rc.conf gesetzt sind. Beispielsweise funktioniert `sshd restart` nur dann, wenn in /etc/rc.conf die Variable `sshd_enable` auf `YES` gesetzt wurde. Wenn Sie die Optionen `start`, `stop` oder `restart` unabhängig von den Einstellungen in /etc/rc.conf benutzen wollen, müssen Sie den Optionen mit dem Präfix "one" verwenden. Um beispielsweise `sshd` unabhängig von den Einstellungen in /etc/rc.conf neu zu starten, benutzen Sie das nachstehende Kommando:

```
# service sshd onerestart
```

Ob ein Dienst in /etc/rc.conf aktiviert ist, können Sie herausfinden, indem Sie das entsprechende rc(8)-Skript mit der Option `rcvar` aufrufen. Dieses Beispiel prüft, ob der `sshd`-Dienst in /etc/rc.conf aktiviert ist:

```
# service sshd rcvar
# sshd
#
sshd_enable="YES"
#   (default: "")
```



Die Zeile `# sshd` wird von dem Kommando ausgegeben; sie kennzeichnet nicht die Eingabeaufforderung von `root`.

Ob ein Dienst läuft, kann mit `status` abgefragt werden. Das folgende Kommando überprüft, ob `sshd` auch wirklich gestartet wurde:

```
# service sshd status
sshd is running as pid 433.
```

Einige Dienste können über die Option `reload` neu initialisiert werden. Dazu wird dem Dienst über ein Signal mitgeteilt, dass er seine Konfigurationsdateien neu einlesen soll. Oft wird dazu das Signal `SIGHUP` verwendet. Beachten Sie aber, dass nicht alle Dienste diese Option unterstützen.

Die meisten Systemdienste werden beim Systemstart vom rc(8)-System gestartet. Zum Beispiel aktiviert das Skript /etc/rc.d/bgfsck die Prüfung von Dateisystemen im Hintergrund. Das Skript gibt die folgende Meldung aus, wenn es gestartet wird:

```
Starting background file system checks in 60 seconds.
```

Dieses Skript wird während des Systemstarts ausgeführt und führt eine Überprüfung der Dateisysteme im Hintergrund durch.

Viele Systemdienste hängen von anderen Diensten ab. yp(8) und andere RPC-basierende Systeme hängen beispielsweise von dem `rpcbind`-Dienst ab. Im Kopf der Startskripten befinden sich die Informationen über Abhängigkeiten von anderen Diensten und weitere Metadaten. Mithilfe dieser Daten bestimmt das Programm rcorder(8) beim Systemstart die Startreihenfolge der Dienste.

Folgende Schlüsselwörter müssen im Kopf aller Startskripten verwendet werden, da sie von rc.subr(8) zum "Aktivieren" des Startskripts benötigt werden:

- `PROVIDE`: Gibt die Namen der Dienste an, die mit dieser Datei zur Verfügung gestellt werden.

Die folgenden Schlüsselwörter können im Kopf des Startskripts angegeben werden. Sie sind zwar nicht unbedingt notwendig, sind aber hilfreich beim Umgang mit rcorder(8):

- `REQUIRE`: Gibt die Namen der Dienste an, von denen dieser Dienst abhängt. Ein Skript, das dieses Schlüsselwort enthält wird *nach* den angegebenen Diensten ausgeführt.
- `BEFORE`: Zählt Dienste auf, die auf diesen Dienst angewiesen sind. Ein Skript, dass dieses Schlüsselwort enthält wird *vor* den angegebenen Diensten ausgeführt.

Durch das Verwenden dieser Schlüsselwörter kann ein Administrator die Startreihenfolge von Systemdiensten feingranuliert steuern, ohne mit den Schwierigkeiten des "runlevel"-Systems anderer UNIX® Systeme kämpfen zu müssen.

Weitere Informationen über das rc(8)-System finden Sie in rc(8) und rc.subr(8). Wenn Sie eigene rc.d-Skripte schreiben wollen, sollten Sie diesen Artikel lesen.

## 20.1. Systemspezifische Konfiguration

Informationen zur Systemkonfiguration sind hauptsächlich in /etc/rc.conf, die meist beim Start des Systems verwendet wird, abgelegt. Sie enthält die Konfigurationen für die rc* Dateien.

In rc.conf werden die Vorgabewerte aus /etc/defaults/rc.conf überschrieben. Die Vorgabedatei sollte

nicht editiert werden. Stattdessen sollten alle systemspezifischen Änderungen in rc.conf vorgenommen werden.

Um den administrativen Aufwand gering zu halten, existieren in geclusterten Anwendungen mehrere Strategien, globale Konfigurationen von systemspezifischen Konfigurationen zu trennen. Der empfohlene Weg hält die globale Konfiguration in einer separaten Datei z.B. /etc/rc.conf.local. Zum Beispiel so:

- /etc/rc.conf:

```
sshd_enable="YES"
keyrate="fast"
defaultrouter="10.1.1.254"
```

- /etc/rc.conf.local:

```
hostname="node1.example.org"
ifconfig_fxp0="inet 10.1.1.1/8"
```

/etc/rc.conf kann dann auf jedes System mit rsync oder puppet verteilt werden, während /etc/rc.conf.local dabei systemspezifisch bleibt.

Bei einem Upgrade des Systems wird /etc/rc.conf nicht überschrieben, so dass die Systemkonfiguration erhalten bleibt.



/etc/rc.conf und /etc/rc.conf.local werden von sh(1) gelesen. Dies erlaubt es dem Systemadministrator, komplexe Konfigurationsszenarien zu erstellen. Lesen Sie rc.conf(5), um weitere Informationen zu diesem Thema zu erhalten.

# Kapitel 21. Einrichten von Netzwerkkarten

Die Konfiguration einer Netzwerkkarte gehört zu den alltäglichen Aufgaben eines FreeBSD Administrators.

## 21.1. Bestimmen des richtigen Treibers

Ermitteln Sie zunächst das Modell der Netzwerkkarte und den darin verwendeten Chip. FreeBSD unterstützt eine Vielzahl von Netzwerkkarten. Prüfen Sie die Hardware-Kompatibilitätsliste für das FreeBSD Release, um zu sehen ob die Karte unterstützt wird.

Wenn die Karte unterstützt wird, müssen Sie den Treiber für die Karte bestimmen. /usr/src/sys/conf/NOTES und /usr/src/sys/arch/conf/NOTES enthalten eine Liste der verfügbaren Treiber mit Informationen zu den unterstützten Chipsätzen. Wenn Sie sich nicht sicher sind, ob Sie den richtigen Treiber ausgewählt haben, lesen Sie die Hilfeseite des Treibers. Sie enthält weitere Informationen über die unterstützten Geräte und bekannte Einschränkungen des Treibers.

Die Treiber für gebräuchliche Netzwerkkarten sind schon im GENERIC-Kernel enthalten, so dass die Karte während des Systemstarts erkannt werden sollte. Die Systemmeldungen können Sie sich mit `more /var/run/dmesg.boot` ansehen. Mit der Leertaste können Sie durch den Text blättern. In diesem Beispiel findet das System zwei Karten, die den dc(4)-Treiber benutzen:

```
dc0: <82c169 PNIC 10/100BaseTX> port 0xa000-0xa0ff mem 0xd3800000-0xd38
000ff irq 15 at device 11.0 on pci0
miibus0: <MII bus> on dc0
bmtphy0: <BCM5201 10/100baseTX PHY> PHY 1 on miibus0
bmtphy0:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc0: Ethernet address: 00:a0:cc:da:da:da
dc0: [ITHREAD]
dc1: <82c169 PNIC 10/100BaseTX> port 0x9800-0x98ff mem 0xd3000000-0xd30
000ff irq 11 at device 12.0 on pci0
miibus1: <MII bus> on dc1
bmtphy1: <BCM5201 10/100baseTX PHY> PHY 1 on miibus1
bmtphy1:  10baseT, 10baseT-FDX, 100baseTX, 100baseTX-FDX, auto
dc1: Ethernet address: 00:a0:cc:da:da:db
dc1: [ITHREAD]
```

Ist der Treiber für die Netzwerkkarte nicht in GENERIC enthalten, muss zunächst ein Treiber geladen werden, um die Karte konfigurieren und benutzen zu können. Dafür gibt es zwei Methoden:

- Am einfachsten ist es, das Kernelmodul für die Karte mit kldload(8) zu laden. Um den Treiber automatisch beim Systemstart zu laden, fügen Sie die entsprechende Zeile in /boot/loader.conf ein. Es gibt nicht für alle Karten Kernelmodule.
- Alternativ kann der Treiber für die Karte fest in den Kernel eingebunden werden. Lesen Sie dazu /usr/src/sys/conf/NOTES, /usr/src/sys/arch/conf/NOTES und die Hilfeseite des Treibers, den Sie in den Kernel einbinden möchten, an. Die Übersetzung des Kernels wird in Konfiguration

des FreeBSD-Kernels beschrieben. Wenn die Karte während des Systemstarts vom Kernel erkannt wurde, muss der Kernel nicht neu übersetzt werden.

### 21.1.1. Windows®-NDIS-Treiber einsetzen

Leider stellen nach wie vor viele Unternehmen die Spezifikationen ihrer Treiber der Open Source Gemeinde nicht zur Verfügung, weil sie diese Informationen als Geschäftsgeheimnisse betrachten. Daher haben die Entwickler von FreeBSD und anderen Betriebssystemen nur zwei Möglichkeiten. Entweder versuchen sie in einem aufwändigen Prozess den Treiber durch Reverse Engineering nachzubauen, oder sie versuchen, die vorhandenen Binärtreiber der Microsoft® Windows®-Plattform zu verwenden.

FreeBSD bietet "native" Unterstützung für die Network Driver Interface Specification (NDIS). ndisgen(8) wird benutzt, um einen Windows® XP-Treiber in ein Format zu konvertieren, das von FreeBSD verwendet werden kann. Da der ndis(4)-Treiber einen Windows® XP-Binärtreiber nutzt, kann er nur auf i386™- und amd64-Systemen verwendet werden. Unterstützt werden PCI, CardBus, PCMCIA und USB-Geräte.

Um den NDISulator zu verwenden, benötigen Sie drei Dinge:

1. Die FreeBSD Kernelquellen
2. Den Windows® XP-Binärtreiber mit der Erweiterung .SYS
3. Die Konfigurationsdatei des Windows® XP-Treibers mit der Erweiterung .INF

Laden Sie die .SYS- und .INF-Dateien für die Karte. Diese befinden sich meistens auf einer beigelegten CD-ROM, oder können von der Internetseite des Herstellers heruntergeladen werden. In den folgenden Beispielen werden die Dateien W32DRIVER.SYS und W32DRIVER.INF verwendet.

Die Architektur des Treibers muss zur jeweiligen Version von FreeBSD passen. Benutzen Sie einen Windows® 32-bit Treiber für FreeBSD/i386. Für FreeBSD/amd64 wird ein Windows® 64-bit Treiber benötigt.

Als Nächstes kompilieren Sie den binären Treiber, um ein Kernelmodul zu erzeugen. Dazu rufen Sie als `root` ndisgen(8) auf:

```
# ndisgen /path/to/W32DRIVER.INF /path/to/W32DRIVER.SYS
```

Dieses Kommando arbeitet interaktiv, benötigt es weitere Informationen, so fragt es Sie danach. Das Ergebnis ist ein neu erzeugtes Kernelmodul im aktuellen Verzeichnis. Benutzen Sie kldload(8) um das neue Modul zu laden:

```
# kldload ./W32DRIVER.ko
```

Neben dem erzeugten Kernelmodul müssen auch die Kernelmodule ndis.ko und if_ndis.ko geladen werden. Dies passiert automatisch, wenn Sie ein von ndis(4) abhängiges Modul laden. Andernfalls können die Module mit den folgenden Kommandos manuell geladen werden:

```
# kldload ndis
# kldload if_ndis
```

Der erste Befehl lädt den ndis(4)-Miniport-Treiber, der zweite das tatsächliche Netzwerkgerät.

Überprüfen Sie die Ausgabe von dmesg(8) auf eventuelle Fehler während des Ladevorgangs. Gab es dabei keine Probleme, sollte die Ausgabe wie folgt aussehen:

```
ndis0: <Wireless-G PCI Adapter> mem 0xf4100000-0xf4101fff irq 3 at device 8.0 on pci1
ndis0: NDIS API version: 5.0
ndis0: Ethernet address: 0a:b1:2c:d3:4e:f5
ndis0: 11b rates: 1Mbps 2Mbps 5.5Mbps 11Mbps
ndis0: 11g rates: 6Mbps 9Mbps 12Mbps 18Mbps 36Mbps 48Mbps 54Mbps
```

Ab jetzt kann das Gerät ndis0 wie jede andere Netzwerkkarte konfiguriert werden.

Um die ndis(4)-Module automatisch beim Systemstart zu laden, kopieren Sie das erzeugte Modul W32DRIVER_SYS.ko nach /boot/modules. Danach fügen Sie die folgende Zeile in /boot/loader.conf ein:

```
W32DRIVER_SYS_load="YES"
```

## 21.2. Konfiguration von Netzwerkkarten

Nachdem der richtige Treiber für die Karte geladen ist, muss die Karte konfiguriert werden. Unter Umständen ist die Karte schon während der Installation mit bsdinstall(8) konfiguriert worden.

Das nachstehende Kommando zeigt die Konfiguration der Netzwerkkarten an:

```
% ifconfig
dc0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:da
        inet 192.168.1.3 netmask 0xffffff00 broadcast 192.168.1.255
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
dc1: flags=8802<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:db
        inet 10.0.0.1 netmask 0xffffff00 broadcast 10.0.0.255
        media: Ethernet 10baseT/UTP
        status: no carrier
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x4
        inet6 ::1 prefixlen 128
```

```
        inet 127.0.0.1 netmask 0xff000000
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
```

Im Beispiel werden Informationen zu den folgenden Geräten angezeigt:

- dc0: Der erste Ethernet-Adapter.
- dc1: Der zweite Ethernet-Adapter.
- lo0: Das Loopback-Gerät.

Der Name der Netzwerkkarte wird aus dem Namen des Treibers und einer Zahl zusammengesetzt. Die Zahl gibt die Reihenfolge an, in der die Geräte beim Systemstart erkannt wurden. Die dritte Karte, die den sis(4) Treiber benutzt, würde beispielsweise sis2 heißen.

Der Adapter dc0 aus dem Beispiel ist aktiv. Sie erkennen das an den folgenden Hinweisen:

1. `UP` bedeutet, dass die Karte konfiguriert und aktiv ist.
2. Der Karte wurde die Internet-Adresse (`inet`) `192.168.1.3` zugewiesen.
3. Die Subnetzmaske ist richtig (`0xffffff00` entspricht `255.255.255.0`).
4. Die Broadcast-Adresse `192.168.1.255` ist richtig.
5. Die MAC-Adresse der Karte (`ether`) lautet `00:a0:cc:da:da:da`.
6. Die automatische Medienerkennung ist aktiviert (`media: Ethernet autoselect (100baseTX <full-duplex>)`). Der Adapter dc1 benutzt das Medium `10baseT/UTP`. Weitere Informationen über die einstellbaren Medien entnehmen Sie der Hilfeseite des Treibers.
7. Der Verbindungsstatus (`status`) ist `active`, das heißt es wurde ein Trägersignal entdeckt. Für dc1 wird `status: no carrier` angezeigt. Das ist normal, wenn kein Kabel an der Karte angeschlossen ist.

Wäre die Karte nicht konfiguriert, würde die Ausgabe von ifconfig(8) so aussehen:

```
dc0: flags=8843<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=80008<VLAN_MTU,LINKSTATE>
        ether 00:a0:cc:da:da:da
        media: Ethernet autoselect (100baseTX <full-duplex>)
        status: active
```

Die Karte muss als Benutzer `root` konfiguriert werden. Die Konfiguration kann auf der Kommandozeile mit ifconfig(8) erfolgen. Allerdings gehen diese Informationen bei einem Neustart verloren. Tragen Sie stattdessen die Konfiguration in /etc/rc.conf ein. Wenn es im LAN einen DHCP-Server gibt, fügen Sie einfach folgende Zeile hinzu:

```
ifconfig_dc0="DHCP"
```

Ersetzen Sie >*dc0* durch die richtigen Werte für das System.

Nachdem Sie die Zeile hinzugefügt haben, folgen Sie den Anweisungen in Test und Fehlersuche.



Wenn das Netzwerk während der Installation konfiguriert wurde, existieren vielleicht schon Einträge für die Netzwerkkarte(n). Überprüfen Sie /etc/rc.conf bevor Sie weitere Zeilen hinzufügen.

Falls kein DHCP-Server zur Verfügung steht, müssen die Netzwerkkarten manuell konfiguriert werden. Fügen Sie für jede Karte im System eine Zeile hinzu, wie in diesem Beispiel zu sehen:

```
ifconfig_dc0="inet 192.168.1.3 netmask 255.255.255.0"
ifconfig_dc1="inet 10.0.0.1 netmask 255.255.255.0 media 10baseT/UTP"
```

Ersetzen Sie dc0 und dc1 und die IP-Adressen durch die richtigen Werte für das System. Die Manualpages des Treibers, ifconfig(8) und rc.conf(5) enthalten weitere Einzelheiten über verfügbare Optionen und die Syntax von /etc/rc.conf.

Wenn das Netzwerk kein DNS benutzt, können Sie in /etc/hosts die Namen und IP-Adressen der Rechner des LANs eintragen. Weitere Informationen entnehmen Sie hosts(5) und /usr/shared/examples/etc/hosts.



Falls kein DHCP-Server zur Verfügung steht, Sie aber Zugang zum Internet benötigen, müssen Sie das Standard-Gateway und die Nameserver manuell konfigurieren:

```
# echo 'defaultrouter="Ihr_Default_Gateway"' >> /etc/rc.conf
# echo 'nameserver Ihr_DNS_Server' >> /etc/resolv.conf
```

## 21.3. Test und Fehlersuche

Nachdem die notwendigen Änderungen in /etc/rc.conf gespeichert wurden, kann das System neu gestartet werden, um die Konfiguration zu testen und zu überprüfen, ob das System ohne Fehler neu gestartet wurde. Alternativ können Sie mit folgenden Befehl die Netzwerkeinstellungen neu initialisieren:

```
# service netif restart
```



Falls in /etc/rc.conf ein Default-Gateway definiert wurde, müssen Sie auch den folgenden Befehl ausführen:

```
# service routing restart
```

Wenn das System gestartet ist, sollten Sie die Netzwerkkarten testen.

### 21.3.1. Test der Ethernet-Karte

Um zu prüfen, ob die Ethernet-Karte richtig konfiguriert ist, testen Sie zunächst mit ping(8) den Adapter selbst und sprechen Sie dann eine andere Maschine im LAN an.

Zuerst, der Test des Adapters:

```
% ping -c5 192.168.1.3
PING 192.168.1.3 (192.168.1.3): 56 data bytes
64 bytes from 192.168.1.3: icmp_seq=0 ttl=64 time=0.082 ms
64 bytes from 192.168.1.3: icmp_seq=1 ttl=64 time=0.074 ms
64 bytes from 192.168.1.3: icmp_seq=2 ttl=64 time=0.076 ms
64 bytes from 192.168.1.3: icmp_seq=3 ttl=64 time=0.108 ms
64 bytes from 192.168.1.3: icmp_seq=4 ttl=64 time=0.076 ms

--- 192.168.1.3 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.074/0.083/0.108/0.013 ms
```

```
% ping -c5 192.168.1.2
PING 192.168.1.2 (192.168.1.2): 56 data bytes
64 bytes from 192.168.1.2: icmp_seq=0 ttl=64 time=0.726 ms
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.766 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.700 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=64 time=0.747 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=64 time=0.704 ms

--- 192.168.1.2 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.700/0.729/0.766/0.025 ms
```

Um die Namensauflösung zu testen, verwenden Sie den Namen der Maschine anstelle der IP-Adresse. Wenn kein DNS-Server im Netzwerk vorhanden ist, muss /etc/hosts entsprechend eingerichtet sein. Fügen Sie dazu die Namen und IP-Adressen der Rechner im LAN in /etc/hosts hinzu, falls sie nicht bereits vorhanden sind. Weitere Informationen finden Sie in hosts(5) und /usr/shared/examples/etc/hosts.

### 21.3.2. Fehlersuche

Fehler zu beheben, ist immer sehr mühsam. Indem Sie die einfachen Sachen zuerst prüfen, erleichtern Sie sich die Aufgabe. Steckt das Netzwerkkabel? Sind die Netzwerkdienste richtig konfiguriert? Funktioniert die Firewall? Wird die Netzwerkkarte von FreeBSD unterstützt? Lesen Sie immer die Hardware-Informationen des Releases, bevor Sie einen Fehlerbericht einsenden. Aktualisieren Sie die FreeBSD-Version auf die neueste -STABLE Version. Suchen Sie in den Archiven der Mailinglisten und im Internet nach bekannten Lösungen.

Wenn die Karte funktioniert, die Verbindungen aber zu langsam sind, sollten Sie tuning(7) lesen. Prüfen Sie auch die Netzwerkkonfiguration, da falsche Einstellungen die Ursache für langsame

Verbindungen sein können.

Wenn Sie viele `device timeout` Meldungen in den Systemprotokollen finden, prüfen Sie, dass es keinen Konflikt zwischen der Netzwerkkarte und anderen Geräten des Systems gibt. Überprüfen Sie nochmals die Verkabelung. Unter Umständen benötigen Sie eine andere Netzwerkkarte.

Bei `watchdog timeout` Fehlermeldungen, kontrollieren Sie zuerst die Verkabelung. Überprüfen Sie dann, ob der PCI-Steckplatz der Karte Bus Mastering unterstützt. Auf einigen älteren Motherboards ist das nur für einen Steckplatz (meistens Steckplatz 0) der Fall. Lesen Sie in der Dokumentation der Karte und des Motherboards nach, ob das vielleicht die Ursache des Problems sein könnte.

Die Meldung `No route to host` erscheint, wenn das System ein Paket nicht zustellen kann. Das kann vorkommen weil beispielsweise keine Default-Route gesetzt wurde oder das Netzwerkkabel nicht richtig steckt. Schauen Sie in der Ausgabe von `netstat -rn` nach, ob eine gültige Route zu dem Zielsystem existiert. Wenn nicht, lesen Sie “Gateways und Routen”.

Die Meldung `ping: sendto: Permission denied` wird oft von einer falsch konfigurierten Firewall verursacht. Wenn keine Regeln definiert wurden, blockiert eine aktivierte Firewall alle Pakete, selbst einfache ping(8)-Pakete. Weitere Informationen erhalten Sie in Firewalls.

Falls die Leistung der Karte schlecht ist, setzen Sie die Medienerkennung von `autoselect` (automatisch) auf das richtige Medium. In vielen Fällen löst diese Maßnahme Leistungsprobleme. Wenn nicht, prüfen Sie nochmal die Netzwerkeinstellungen und lesen Sie tuning(7).

# Kapitel 22. Virtual Hosts

Ein gebräuchlicher Zweck von FreeBSD ist das virtuelle Hosting, bei dem ein Server im Netzwerk wie mehrere Server aussieht. Dies wird dadurch erreicht, dass einem Netzwerkinterface mehrere Netzwerk-Adressen zugewiesen werden.

Ein Netzwerkinterface hat eine "echte" Adresse und kann beliebig viele "alias" Adressen haben. Die Aliase werden durch entsprechende alias Einträge in /etc/rc.conf festgelegt, wie in diesem Beispiel zu sehen ist:

```
ifconfig_fxp0_alias0="inet xxx.xxx.xxx.xxx netmask xxx.xxx.xxx.xxx"
```

Beachten Sie, dass die Alias-Einträge mit `alias_0_` anfangen müssen und weiter hochgezählt werden, das heißt `alias1`, `alias2`, und so weiter. Die Konfiguration der Aliase hört bei der ersten fehlenden Zahl auf.

Die Berechnung der Alias-Netzwerkmasken ist wichtig. Für jedes Interface muss es eine Adresse geben, die die Netzwerkmaske des Netzwerkes richtig beschreibt. Alle anderen Adressen in diesem Netzwerk haben dann eine Netzwerkmaske, die mit `1` gefüllt ist, also `255.255.255.255` oder hexadezimal `0xffffffff`.

Als Beispiel betrachten wir den Fall, in dem fxp0 mit zwei Netzwerken verbunden ist: dem Netzwerk `10.1.1.0` mit der Netzwerkmaske `255.255.255.0` und dem Netzwerk `202.0.75.16` mit der Netzwerkmaske `255.255.255.240`. Das System soll die Adressen `10.1.1.1` bis `10.1.1.5` und `202.0.75.17` bis `202.0.75.20` belegen. Nur die erste Adresse in einem Netzwerk sollte die richtige Netzwerkmaske haben. Alle anderen Adressen (`10.1.1.2` bis `10.1.1.5` und `202.0.75.18` bis `202.0.75.20`) müssen die Maske `255.255.255.255` erhalten.

Die folgenden Einträge in /etc/rc.conf konfigurieren den Adapter entsprechend dem Beispiel:

```
ifconfig_fxp0="inet 10.1.1.1 netmask 255.255.255.0"
ifconfig_fxp0_alias0="inet 10.1.1.2 netmask 255.255.255.255"
ifconfig_fxp0_alias1="inet 10.1.1.3 netmask 255.255.255.255"
ifconfig_fxp0_alias2="inet 10.1.1.4 netmask 255.255.255.255"
ifconfig_fxp0_alias3="inet 10.1.1.5 netmask 255.255.255.255"
ifconfig_fxp0_alias4="inet 202.0.75.17 netmask 255.255.255.240"
ifconfig_fxp0_alias5="inet 202.0.75.18 netmask 255.255.255.255"
ifconfig_fxp0_alias6="inet 202.0.75.19 netmask 255.255.255.255"
ifconfig_fxp0_alias7="inet 202.0.75.20 netmask 255.255.255.255"
```

Dies kann mit einer durch Leerzeichen getrennten Liste von IP-Adressbereichen auch einfacher ausgedrückt werden. Die erste Adresse hat wieder die angegebene Netzwerkmaske und die zusätzlichen Adressen haben die Netzwerkmaske `255.255.255.255`.

```
ifconfig_fxp0_aliases="inet 10.1.1.1-5/24 inet 202.0.75.17-20/28"
```

# Kapitel 23. Konfiguration der Systemprotokollierung

Die Aufzeichnung und Kontrolle von Log-Meldungen ist ein wichtiger Aspekt der Systemadministration. Die Informationen werden nicht nur verwendet um Hard- und Softwarefehler ausfindig zu machen, auch zur Überwachung der Sicherheit und der Reaktion bei einem Zwischenfall spielen diese Aufzeichnungen eine wichtige Rolle. Die meisten Systemdienste und Anwendungen erzeugen Log-Meldungen.

FreeBSD stellt mit syslogd ein Werkzeug zur Verwaltung von Protokollen bereit. In der Voreinstellung wird syslogd beim Booten automatisch gestartet. Dieses Verhalten wird über die Variable `syslogd_enable` in /etc/rc.conf gesteuert. Dazu gibt es noch zahlreiche Argumente, die in der Variable `syslogd_flags` in /etc/rc.conf gesetzt werden können. Lesen Sie syslogd(8) für weitere Informationen über die verfügbaren Argumente.

Dieser Abschnitt beschreibt die Konfiguration und Verwendung des FreeBSD Protokollservers, und diskutiert auch die Log-Rotation und das Management von Logdateien.

## 23.1. Konfiguration der lokalen Protokollierung

Die Konfigurationsdatei /etc/syslog.conf steuert, was syslogd mit Log-Meldungen macht, sobald sie empfangen werden. Es gibt verschiedene Parameter, die das Verhalten bei eingehenden Ereignissen kontrollieren. facility beschreibt das Subsystem, welches das Ereignis generiert hat. Beispielsweise der Kernel, oder ein Daemon. level hingegen beschreibt den Schweregrad des aufgetretenen Ereignisses. Dies macht es möglich, Meldungen in verschiedenen Logdateien zu protokollieren, oder Meldungen zu verwerfen, je nach Konfiguration von facility und level. Ebenfalls besteht die Möglichkeit auf Meldungen zu reagieren, die von einer bestimmten Anwendung stammen, oder von einem spezifischen Host erzeugt wurden.

Die Konfigurationsdatei von syslogd(8) enthält für jede Aktion eine Zeile. Die Syntax besteht aus einem Auswahlfeld, gefolgt von einem Aktionsfeld. Die Syntax für das Auswahlfeld ist *facility.level*. Dies entspricht Log-Meldungen von *facility* mit einem Level von *level* oder höher. Um noch präziser festzulegen was protokolliert wird, kann dem Level optional ein Vergleichsflag vorangestellt werden. Mehrere Auswahlen können, durch Semikolon (`;`) getrennt, für die gleiche Aktion verwendet werden. `*` wählt dabei alles aus. Das Aktionsfeld definiert, wohin die Log-Meldungen gesendet werden, beispielsweise in eine Datei oder zu einem entfernten Log-Server. Als Beispiel dient hier /etc/syslog.conf aus FreeBSD:

```
# $FreeBSD$
#
#       Spaces ARE valid field separators in this file. However,
#       other *nix-like systems still insist on using tabs as field
#       separators. If you are sharing this file between systems, you$
#       may want to use only tabs as field separators here.
#       Consult the syslog.conf(5) manpage.
*.err;kern.warning;auth.notice;mail.crit                /dev/console
*.notice;authpriv.none;kern.debug;lpr.info;mail.crit;news.err   /var/log/messages
```

```
security.*                                      /var/log/security
auth.info;authpriv.info                         /var/log/auth.log
mail.info                                       /var/log/maillog
lpr.info                                        /var/log/lpd-errs
ftp.info                                        /var/log/xferlog
cron.*                                          /var/log/cron
!-devd
*.=debug                                        /var/log/debug.log
*.emerg                                         *
# uncomment this to log all writes to /dev/console to /var/log/console.log
#console.info                                   /var/log/console.log
# uncomment this to enable logging of all log messages to /var/log/all.log
# touch /var/log/all.log and chmod it to mode 600 before it will work
#*.*                                            /var/log/all.log
# uncomment this to enable logging to a remote loghost named loghost
#*.*                                            @loghost
# uncomment these if you're running inn
# news.crit                                     /var/log/news/news.crit
# news.err                                      /var/log/news/news.err
# news.notice                                   /var/log/news/news.notice
# Uncomment this if you wish to see messages produced by devd
# !devd
# *.>=info
!ppp
*.*                                             /var/log/ppp.log
!*
```

In diesem Beispiel:

- Zeile 8 selektiert alle Meldungen vom Level `err`, sowie `kern.warning`, `auth.notice` und `mail.crit` und schickt diese zur Konsole (/dev/console).
- Zeile 12 selektiert alle Meldungen von `mail` ab dem Level `info` oder höher und schreibt diese in /var/log/maillog.
- Zeile 17 benutzt ein Vergleichsflag (`=`), um nur Meldungen vom Level `debug` zu selektieren und schreibt diese in /var/log/debug.log.
- Zeile 33 zeigt ein Beispiel für die Nutzung einer Programmspezifikation. Die nachfolgenden Regeln sind dann nur für Programme gültig, welche der Programmspezifikation stehen. In diesem Fall werden alle Meldungen von ppp (und keinem anderen Programm) in /var/log/ppp.log geschrieben.

Die verfügbaren level, beginnend mit den höchst kritischen, hin zu den weniger kritischen, sind: `emerg`, `alert`, `crit`, `err`, `warning`, `notice`, `info` und `debug`.

Die facilities, in beliebiger Reihenfolge, sind: `auth`, `authpriv`, `console`, `cron`, `daemon`, `ftp`, `kern`, `lpr`, `mail`, `mark`, `news`, `security`, `syslog`, `user`, `uucp`, sowie `local0` bis `local7`. Beachten Sie, dass andere Betriebssysteme hiervon abweichende facilities haben können.

Um alle Meldungen vom Level `notice` und höher in /var/log/daemon.log zu protokollieren, fügen Sie folgenden Eintrag hinzu:

```
daemon.notice                                   /var/log/daemon.log
```

Für weitere Informationen zu verschiedenen Level und faclilities, lesen Sie syslog(3) und syslogd(8). Weitere Informationen zu /etc/syslog.conf, dessen Syntax und erweiterten Anwendungsbeispielen, finden Sie in syslog.conf(5).

## 23.2. Management und Rotation von Logdateien

Logdateien können schnell wachsen und viel Speicherplatz belegen, was es schwieriger macht, nützliche Informationen zu finden. Log-Management versucht, diesen Effekt zu mildern. FreeBSD verwendet newsyslog für die Verwaltung von Logdateien. Dieses in FreeBSD integrierte Programm rotiert und komprimiert in regelmäßigen Abständen Logdateien. Optional kann es auch fehlende Logdateien erstellen und Programme benachrichtigen, wenn Logdateien verschoben wurden. Die Logdateien können von syslogd oder einem anderen Programm generiert werden. Obwohl newsyslog normalerweise von cron(8) aufgerufen wird, ist es kein Systemdämon. In der Standardkonfiguration wird dieser Job jede Stunde ausgeführt.

Um zu wissen, welche Maßnahmen zu ergreifen sind, liest newsyslog seine Konfigurationsdatei /etc/newsyslog.conf. Diese Konfigurationsdatei enthält eine Zeile für jede Datei, die von newsyslog verwaltet wird. Jede Zeile enthält Informationen über den Besitzer der Datei, die Dateiberechtigungen, wann die Datei rotiert wird, optionale Flags, welche die Log-Rotation beeinflussen (bspw. Komprimierung) und Programme, denen ein Signal geschickt wird, wenn Logdateien rotiert werden. Hier folgt die Standardkonfiguration in FreeBSD:

```
# configuration file for newsyslog
# $FreeBSD$
#
# Entries which do not specify the '/pid_file' field will cause the
# syslogd process to be signalled when that log file is rotated.  This
# action is only appropriate for log files which are written to by the
# syslogd process (ie, files listed in /etc/syslog.conf).  If there
# is no process which needs to be signalled when a given log file is
# rotated, then the entry for that file should include the 'N' flag.
#
# The 'flags' field is one or more of the letters: BCDGJNUXZ or a '-'.
#
# Note: some sites will want to select more restrictive protections than the
# defaults.  In particular, it may be desirable to switch many of the 644
# entries to 640 or 600.  For example, some sites will consider the
# contents of maillog, messages, and lpd-errs to be confidential.  In the
# future, these defaults may change to more conservative ones.
#
# logfilename          [owner:group]    mode count size when  flags [/pid_file]
[sig_num]
/var/log/all.log                        600  7     *    @T00  J
/var/log/amd.log                        644  7     100  *     J
/var/log/auth.log                       600  7     100  @0101T JC
/var/log/console.log                    600  5     100  *     J
```

```
/var/log/cron                           600  3     100  *     JC
/var/log/daily.log                      640  7     *    @T00  JN
/var/log/debug.log                      600  7     100  *     JC
/var/log/kerberos.log                   600  7     100  *     J
/var/log/lpd-errs                       644  7     100  *     JC
/var/log/maillog                        640  7     *    @T00  JC
/var/log/messages                       644  5     100  @0101T JC
/var/log/monthly.log                    640  12    *    $M1D0 JN
/var/log/pflog                          600  3     100  *     JB
/var/run/pflogd.pid
/var/log/ppp.log        root:network    640  3     100  *     JC
/var/log/devd.log                       644  3     100  *     JC
/var/log/security                       600  10    100  *     JC
/var/log/sendmail.st                    640  10    *    168   B
/var/log/utx.log                        644  3     *    @01T05 B
/var/log/weekly.log                     640  5     1    $W6D0 JN
/var/log/xferlog                        600  7     100  *     JC
```

Jede Zeile beginnt mit dem Namen der Protokolldatei, die rotiert werden soll, optional gefolgt von Besitzer und Gruppe für rotierende, als auch für neu erstellte Dateien. Das Feld `mode` definiert die Zugriffsrechte der Datei. `count` gibt an, wie viele rotierte Dateien aufbewahrt werden sollen. Anhand der `size`- und `when`-Flags erkennt newsyslog, wann die Datei rotiert werden muss. Eine Logdatei wird rotiert, wenn ihre Größe den Wert von `size` überschreitet, oder wenn die Zeit im `when` -Feld abgelaufen ist. Ein `*` bedeutet, dass dieses Feld ignoriert wird. Das *flags*-Feld gibt newsyslog weitere Instruktionen, zum Beispiel wie eine Datei zu rotieren ist, oder eine Datei zu erstellen falls diese nicht existiert. Die letzten beiden Felder sind optional und bestimmen die PID-Datei und wann die Datei rotiert wird.

Weitere Informationen zu allen Feldern, gültigen Flags und wie Sie die Rotationszeit angeben können, finden Sie in newsyslog.conf(5). Denken Sie daran, dass newsyslog von cron(8) aufgerufen wird und somit Dateien auch nur dann rotiert, wenn es von cron(8) aufgerufen wird, und nicht häufiger.

# 23.3. Protokollierung von anderen Hosts

Die Überwachung der Protokolldateien kann bei steigender Anzahl von Rechnern sehr unhandlich werden. Eine zentrale Protokollierung kann manche administrativen Belastungen bei der Verwaltung von Protokolldateien reduzieren.

Die Aggregation, Zusammenführung und Rotation von Protokolldateien kann in FreeBSD mit syslogd und newsyslog konfiguriert werden. In der folgenden Beispielkonfiguration sammelt Host `A`, genannt `logserv.example.com`, Protokollinformationen für das lokale Netzwerk. Host `B`, genannt `logclient.example.com` wird seine Protokollinformationen an den Server weiterleiten.

## 23.3.1. Konfiguration des Protokollservers

Ein Protokollserver ist ein System, welches Protokollinformationen von anderen Hosts akzeptiert. Bevor Sie diesen Server konfigurieren, prüfen Sie folgendes:

- Falls eine Firewall zwischen dem Protokollserver und den -Clients steht, muss das Regelwerk der Firewall UDP auf Port 514 sowohl auf Client- als auch auf Serverseite freigegeben werden.
- Der `syslogd`-Server und alle Clientrechner müssen gültige Einträge für sowohl Vorwärts- als auch Umkehr-DNS besitzen. Falls im Netzwerk kein DNS-Server vorhanden ist, muss auf jedem System die Datei /etc/hosts mit den richtigen Einträgen gepflegt werden. Eine funktionierende Namensauflösung ist zwingend erforderlich, ansonsten würde der Server die Protokollnachrichten ablehnen.

Bearbeiten Sie /etc/syslog.conf auf dem Server. Tragen Sie den Namen des Clients ein, den Verbindungsweg und den Namen der Protokolldatei. Dieses Beispiel verwendet den Rechnernamen `B`, alle Verbindungswege, und die Protokolle werden in /var/log/logclient.log gespeichert.

*Beispiel 25. Einfache Server Konfiguration*

```
+logclient.example.com
*.*     /var/log/logclient.log
```

Fügen Sie für jeden Client zwei Zeilen hinzu, falls Sie mehrere Clients in diese Datei aufnehmen. Weitere Informationen über die verfügbaren Verbindungswege finden Sie in syslog.conf(5).

Konfigurieren Sie als nächstes /etc/rc.conf:

```
syslogd_enable="YES"
syslogd_flags="-a logclient.example.com -v -v"
```

Der erste Eintrag startet `syslogd` während des Bootens. Der zweite Eintrag erlaubt es, Daten von dem spezifizierten Client auf diesem Server zu akzeptieren. Die Verwendung von `-v -v` erhöht die Anzahl von Protokollnachrichten. Dies ist hilfreich für die Feineinstellung der Verbindungswege, da Administratoren auf diese Weise erkennen, welche Arten von Nachrichten von welchen Verbindungswegen protokolliert werden.

Mehrere `-a`-Optionen können angegeben werden, um die Protokollierung von mehreren Clients zu erlauben. IP-Adressen und ganze Netzblöcke können ebenfalls spezifiziert werden. Eine vollständige Liste der Optionen finden Sie in syslogd(8).

Zum Schluss muss die Protokolldatei erstellt werden:

```
# touch /var/log/logclient.log
```

Zu diesem Zeitpunkt sollte `syslogd` neu gestartet und überprüft werden:

```
# service syslogd restart
# pgrep syslog
```

Wenn eine PID zurückgegeben wird, wurde der Server erfolgreich neu gestartet und die Clientkonfiguration kann beginnen. Wenn der Server nicht neu gestartet wurde, suchen Sie in /var/log/messages nach dem Fehler.

### 23.3.2. Konfiguration des Protokollclients

Ein Protokollclient sendet Protokollinformationen an einen Protokollserver. Zusätzlich behält er eine lokale Kopie seiner eigenen Protokolle.

Sobald der Server konfiguriert ist, bearbeiten Sie /etc/rc.conf auf dem Client:

```
syslogd_enable="YES"
syslogd_flags="-s -v -v"
```

Der erste Eintrag aktiviert den `syslogd`-Dienst während des Systemstarts. Der zweite Eintrag erhöht die Anzahl der Protokollnachrichten. Die Option `-s` verhindert, dass dieser Client Protokolle von anderen Hosts akzeptiert.

Als nächstes muss der Protokollserver in der /etc/syslog.conf des Clients eingetragen werden. In diesem Beispiel wird das `@`-Symbol benutzt, um sämtliche Protokolldaten an einen bestimmten Server zu senden:

```
*.*             @logserv.example.com
```

Nachdem die Änderungs gespeichert wurden, muss `syslogd` neu gestartet werden, damit die Änderungen wirksam werden:

```
# service syslogd restart
```

Um zu testen, ob Protokollnachrichten über das Netzwerk gesendet werden, kann logger(1) auf dem Client benutzt werden, um eine Nachricht an syslogd zu schicken:

```
# logger "Test message from logclient"
```

Diese Nachricht sollte jetzt sowohl in /var/log/messages auf dem Client, als auch in /var/log/logclient.log auf dem Server vorhanden sein.

### 23.3.3. Fehlerbehebung beim Protokollserver

Wenn der Server keine Nachrichten empfängt, ist die Ursache wahrscheinlich ein Netzwerkproblem, ein Problem bei der Namensauflösung oder ein Tippfehler in einer Konfigurationsdatei. Um die Ursache zu isolieren, müssen Sie sicherstellen, dass sich Server und Client über den in /etc/rc.conf konfigurierten Hostnamen mit `ping` erreichen lässt. Falls dies nicht gelingt sollten Sie die Netzwerkverkabelung überprüfen, außerdem Firewallregeln sowie die Einträge für Hosts im DNS und /etc/hosts. Überprüfen Sie diese Dinge auf dem Server und dem

Client, bis der `ping` von beiden Hosts erfolgreich ist.

Wenn sich die Hosts gegenseitig mit `ping` erreichen können, der Server aber immer noch keine Nachrichten empfängt, können Sie vorübergehend die Ausführlichkeit der Protokollierung erhöhen, um die Ursache für das Problem weiter einzugrenzen. In dem folgenden Beispiel ist auf dem Server die Datei /var/log/logclient.log leer und in der Datei /var/log/messages auf dem Client ist keine Ursache für das Problem erkennbar. Um nun die Ausführlichkeit der Protokollierung zu erhöhen, passen Sie auf dem Server den Eintrag `syslogd_flags` an. Anschließend starten Sie den Dienst neu:

```
syslogd_flags="-d -a logclient.example.com -v -v"
```

```
# service syslogd restart
```

Informationen wie diese werden sofort nach dem Neustart auf der Konsole erscheinen:

```
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is
/boot/kernel/kernel
Logging to FILE /var/log/messages
syslogd: kernel boot file is /boot/kernel/kernel
cvthname(192.168.1.10)
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
rejected in rule 0 due to name mismatch.
```

In diesem Beispiel werden die Nachrichten aufgrund eines fehlerhaften Namens abgewiesen. Der Hostname sollte `logclient` und nicht `logclien` sein. Beheben Sie den Tippfehler, starten Sie den Dienst neu und überprüfen Sie das Ergebnis:

```
# service syslogd restart
logmsg: pri 56, flags 4, from logserv.example.com, msg syslogd: restart
syslogd: restarted
logmsg: pri 6, flags 4, from logserv.example.com, msg syslogd: kernel boot file is
/boot/kernel/kernel
syslogd: kernel boot file is /boot/kernel/kernel
logmsg: pri 166, flags 17, from logserv.example.com,
msg Dec 10 20:55:02 <syslog.err> logserv.example.com syslogd: exiting on signal 2
cvthname(192.168.1.10)
validate: dgram from IP 192.168.1.10, port 514, name logclient.example.com;
accepted in rule 0.
logmsg: pri 15, flags 0, from logclient.example.com, msg Dec 11 02:01:28 trhodes: Test
message 2
Logging to FILE /var/log/logclient.log
Logging to FILE /var/log/messages
```

Zu diesem Zeitpunkt werden die Nachrichten korrekt empfangen und in die richtige Datei geschrieben.

### 23.3.4. Sicherheitsbedenken

Wie mit jedem Netzwerkdienst, müssen Sicherheitsanforderungen in Betracht gezogen werden, bevor ein Protokollserver eingesetzt wird. Manchmal enthalten Protokolldateien sensitive Daten über aktivierte Dienste auf dem lokalen Rechner, Benutzerkonten und Konfigurationsdaten. Daten, die vom Client an den Server geschickt werden, sind weder verschlüsselt noch mit einem Passwort geschützt. Wenn ein Bedarf für Verschlüsselung besteht, ist es möglich security/stunnel zu verwenden, welches die Protokolldateien über einen verschlüsselten Tunnel versendet.

Lokale Sicherheit ist ebenfalls ein Thema. Protokolldateien sind während der Verwendung oder nach ihrer Rotation nicht verschlüsselt. Lokale Benutzer versuchen vielleicht, auf diese Dateien zuzugreifen, um zusätzliche Einsichten in die Systemkonfiguration zu erlangen. Es ist absolut notwendig, die richtigen Berechtigungen auf diesen Dateien zu setzen. Das Werkzeug newsyslog unterstützt das Setzen von Berechtigungen auf gerade erstellte oder rotierte Protokolldateien. Protokolldateien mit Zugriffsmodus `600` sollten verhindern, dass lokale Benutzer darin herumschnüffeln. Zusätzliche Informationen finden Sie in newsyslog.conf(5).

# Kapitel 24. Konfigurationsdateien

## 24.1. /etc Layout

Konfigurationsdateien finden sich in einigen Verzeichnissen unter anderem in:

| | |
|---|---|
| /etc | Enthält generelle systemspezifische Konfigurationsinformationen. |
| /etc/defaults | Default Versionen der Konfigurationsdateien. |
| /etc/mail | Enthält die sendmail(8) Konfiguration und weitere MTA Konfigurationsdateien. |
| /etc/ppp | Hier findet sich die Konfiguration für die User- und Kernel-ppp Programme. |
| /usr/local/etc | Installierte Anwendungen legen hier ihre Konfigurationsdateien ab. Dieses Verzeichnis kann Unterverzeichnisse für bestimmte Anwendungen enthalten. |
| /usr/local/etc/rc.d | rc(8)-Skripten installierter Anwendungen. |
| /var/db | Automatisch generierte systemspezifische Datenbanken, wie die Paket-Datenbank oder die locate(1)-Datenbank. |

## 24.2. Hostnamen

### 24.2.1. /etc/resolv.conf

Wie ein FreeBSD-System auf das Internet Domain Name System (DNS) zugreift, wird in /etc/resolv.conf festgelegt.

Die gebräuchlichsten Einträge in /etc/resolv.conf sind:

| | |
|---|---|
| `nameserver` | Die IP-Adresse eines Nameservers, den der Resolver abfragen soll. Bis zu drei Server werden in der Reihenfolge, in der sie aufgezählt sind, abgefragt. |
| `search` | Suchliste mit Domain-Namen zum Auflösen von Hostnamen. Die Liste wird normalerweise durch den Domain-Teil des lokalen Hostnamens festgelegt. |
| `domain` | Der lokale Domain-Name. |

Beispiel für eine typische /etc/resolv.conf:

```
search example.com
nameserver 147.11.1.11
nameserver 147.11.100.30
```



Nur eine der Anweisungen `search` oder `domain` sollte benutzt werden.

Wenn Sie DHCP benutzen, überschreibt dhclient(8) für gewöhnlich /etc/resolv.conf mit den Informationen vom DHCP-Server.

### 24.2.2. /etc/hosts

/etc/hosts ist eine einfache textbasierte Datenbank. Zusammen mit DNS und NIS stellt sie eine Abbildung zwischen Namen und IP-Adressen zur Verfügung. Anstatt named(8) zu konfigurieren, können hier lokale Rechner, die über ein LAN verbunden sind, eingetragen werden. Lokale Einträge für gebräuchliche Internet-Adressen in /etc/hosts verhindern die Abfrage eines externen Servers und beschleunigen die Namensauflösung.

```
# $FreeBSD$
#
#
# Host Database
#
# This file should contain the addresses and aliases for local hosts that
# share this file.  Replace 'my.domain' below with the domainname of your
# machine.
#
# In the presence of the domain name service or NIS, this file may
# not be consulted at all; see /etc/nsswitch.conf for the resolution order.
#
#
::1         localhost localhost.my.domain
127.0.0.1       localhost localhost.my.domain
#
# Imaginary network.
#10.0.0.2       myname.my.domain myname
#10.0.0.3       myfriend.my.domain myfriend
#
# According to RFC 1918, you can use the following IP networks for
# private nets which will never be connected to the Internet:
#
#   10.0.0.0    -   10.255.255.255
#   172.16.0.0  -   172.31.255.255
#   192.168.0.0 -   192.168.255.255
#
# In case you want to be able to connect to the Internet, you need
# real official assigned numbers.  Do not try to invent your own network
# numbers but instead get one from your network provider (if any) or
# from your regional registry (ARIN, APNIC, LACNIC, RIPE NCC, or AfriNIC.)
```

```
#
```

/etc/hosts hat das folgende Format:

```
[Internet Adresse] [Offizieller Hostname] [Alias1] [Alias2] ...
```

Zum Beispiel:

```
10.0.0.1 myRealHostname.example.com myRealHostname foobar1 foobar2
```

Weitere Informationen entnehmen Sie bitte hosts(5).

# Kapitel 25. Einstellungen mit sysctl(8)

Mit sysctl(8) können Sie Änderungen an einem laufenden FreeBSD-System vornehmen. Unter anderem können Optionen des TCP/IP-Stacks oder des virtuellen Speichermanagements verändert werden. Unter der Hand eines erfahrenen Systemadministrators kann dies die Systemperformance erheblich verbessern. Über 500 Variablen können mit sysctl(8) gelesen und gesetzt werden.

Der Hauptzweck von sysctl(8) besteht darin, Systemeinstellungen zu lesen und zu verändern.

Alle auslesbaren Variablen werden wie folgt angezeigt:

```
% sysctl -a
```

Um eine spezielle Variable zu lesen, geben Sie den Namen an:

```
% sysctl kern.maxproc
kern.maxproc: 1044
```

Um eine Variable zu setzen, benutzen Sie die Syntax *Variable= Wert*:

```
# sysctl kern.maxfiles=5000
kern.maxfiles: 2088 -> 5000
```

Mit sysctl können Strings, Zahlen oder Boolean-Werte gesetzt werden. Bei Boolean-Werten steht `1` für wahr und `0` für falsch.

Um die Variablen automatisch während des Systemstarts zu setzen, fügen Sie sie in /etc/sysctl.conf ein. Weitere Informationen finden Sie in der Hilfeseite sysctl.conf(5) und in sysctl.conf.

## 25.1. sysctl.conf

/etc/sysctl.conf sieht ähnlich wie /etc/rc.conf aus. Werte werden in der Form `Variable=Wert` gesetzt. Die angegebenen Werte werden gesetzt, nachdem sich das System bereits im Mehrbenutzermodus befindet. Allerdings lassen sich im Mehrbenutzermodus nicht alle Werte setzen.

Um das Protokollieren von fatalen Signalen abzustellen und Benutzer daran zu hindern, von anderen Benutzern gestartete Prozesse zu sehen, können Sie in /etc/sysctl.conf die folgenden Variablen setzen:

```
# Do not log fatal signal exits (e.g. sig 11)
kern.logsigexit=0

# Prevent users from seeing information about processes that
# are being run under another UID.
security.bsd.see_other_uids=0
```

## 25.2. Schreibgeschützte Variablen

Wenn schreibgeschützte sysctl(8)-Variablen verändert werden, ist ein Neustart des Systems erforderlich.

Beispielsweise hat cardbus(4) auf einigen Laptops Schwierigkeiten, Speicherbereiche zu erkennen. Es treten dann Fehlermeldungen wie die folgende auf:

```
cbb0: Could not map register memory
device_probe_and_attach: cbb0 attach returned 12
```

Um dieses Problem zu lösen, muss eine schreibgeschützte sysctl(8)-Variable verändert werden. Fügen Sie `hw.pci.allow_unsupported_io_range=1` in /boot/loader.conf hinzu und starten Sie das System neu. Danach sollte cardbus(4) fehlerfrei funktionieren.

# Kapitel 26. Tuning von Laufwerken

Der folgende Abschnitt beschreibt die verschiedenen Methoden zur Feinabstimmung der Laufwerke. Oft sind mechanische Teile in Laufwerken, wie SCSI-Laufwerke, verbaut. Diese können einen Flaschenhals bei der Gesamtleistung des Systems darstellen. Sie können zwar auch ein Laufwerk ohne mechanische Teile einbauen, wie z.B. ein Solid-State-Drive, aber Laufwerke mit mechanischen Teilen werden auch in naher Zukunft nicht vom Markt verschwinden. Bei der Feinabstimmung ist es ratsam, die Funktionen von iostat(8) zu verwenden, um verschiedene Änderungen zu testen und um nützliche IO-Informationen des Systems zu erhalten.

## 26.1. Sysctl Variablen

### 26.1.1. `vfs.vmiodirenable`

Die sysctl(8)-Variable `vfs.vmiodirenable` besitzt in der Voreinstellung den Wert `1`. Die Variable kann auf den Wert `0` (deaktiviert) oder `1` (aktiviert) gesetzt werden. Sie steuert, wie Verzeichnisse vom System zwischengespeichert werden. Die meisten Verzeichnisse sind klein und benutzen nur ein einzelnes Fragment, typischerweise 1 kB, im Dateisystem und 512 Bytes im Buffer-Cache. Ist die Variable deaktiviert, wird der Buffer-Cache nur eine limitierte Anzahl Verzeichnisse zwischenspeichern, auch wenn das System über sehr viel Speicher verfügt. Ist die Variable aktiviert, kann der Buffer-Cache den VM-Page-Cache benutzen, um Verzeichnisse zwischenzuspeichern. Der ganze Speicher steht damit zum Zwischenspeichern von Verzeichnissen zur Verfügung. Der Nachteil bei dieser Vorgehensweise ist, dass zum Zwischenspeichern eines Verzeichnisses mindestens eine physikalische Seite im Speicher, die normalerweise 4 kB groß ist, anstelle von 512 Bytes gebraucht wird. Es wird empfohlen, diese Option aktiviert zu lassen, wenn Sie Dienste zur Verfügung stellen, die viele Dateien manipulieren. Beispiele für solche Dienste sind Web-Caches, große Mail-Systeme oder Netnews. Die aktivierte Variable vermindert, trotz des verschwendeten Speichers, in aller Regel nicht die Leistung des Systems, obwohl Sie das nachprüfen sollten.

### 26.1.2. `vfs.write_behind`

In der Voreinstellung besitzt die sysctl(8)-Variable `vfs.write_behind` den Wert `1` (aktiviert). Mit dieser Einstellung schreibt das Dateisystem anfallende vollständige Cluster, die besonders beim sequentiellen Schreiben großer Dateien auftreten, direkt auf das Medium aus. Dies verhindert, dass sich im Buffer-Cache veränderte Puffer (dirty buffers) ansammeln, die die I/O-Verarbeitung nicht mehr beschleunigen würden. Unter bestimmten Umständen blockiert diese Funktion allerdings Prozesse. Setzen Sie in diesem Fall die Variable `vfs.write_behind` auf den Wert `0`.

### 26.1.3. `vfs.hirunningspace`

Die sysctl(8)-Variable `vfs.hirunningspace` bestimmt systemweit die Menge ausstehender Schreiboperationen, die dem Platten-Controller zu jedem beliebigen Zeitpunkt übergeben werden können. Normalerweise können Sie den Vorgabewert verwenden. Auf Systemen mit vielen Platten kann der Wert aber auf 4 bis 5 *Megabyte* erhöht werden. Ein zu hoher Wert (größer als der Schreib-Schwellwert des Buffer-Caches) kann zu Leistungsverlusten führen. Setzen Sie den Wert daher nicht zu hoch! Hohe Werte können auch Leseoperationen verzögern, die gleichzeitig mit

Schreiboperationen ausgeführt werden.

Es gibt weitere sysctl(8)-Variablen, mit denen Sie den Buffer-Cache und den VM-Page-Cache beeinflussen können. Es wird nicht empfohlen, diese Variablen zu verändern, da das VM-System den virtuellen Speicher selbst sehr gut verwaltet.

### 26.1.4. vm.swap_idle_enabled

Die sysctl(8)-Variable vm.swap_idle_enabled ist für große Mehrbenutzer-Systeme gedacht, auf denen sich viele Benutzer an- und abmelden und auf denen es viele Prozesse im Leerlauf (idle) gibt. Solche Systeme fragen kontinuierlich freien Speicher an. Wenn Sie die Variable vm.swap_idle_enabled aktivieren, können Sie die Auslagerungs-Hysterese von Seiten mit den Variablen vm.swap_idle_threshold1 und vm.swap_idle_threshold2 einstellen. Die Schwellwerte beider Variablen geben die Zeit in Sekunden an, in denen sich ein Prozess im Leerlauf befinden muss. Wenn die Werte so eingestellt sind, dass Seiten früher als nach dem normalen Algorithmus ausgelagert werden, verschafft das dem Auslagerungs-Prozess mehr Luft. Aktivieren Sie diese Funktion nur, wenn Sie sie wirklich benötigen: Die Speicherseiten werden eher früher als später ausgelagert. Der Platz im Swap-Bereich wird dadurch schneller verbraucht und die Plattenaktivitäten steigen an. Auf kleinen Systemen hat diese Funktion spürbare Auswirkungen. Auf großen Systemen, die sowieso schon Seiten auslagern müssen, können ganze Prozesse leichter in den Speicher geladen oder ausgelagert werden.

### 26.1.5. hw.ata.wc

Obwohl das Abstellen des IDE-Schreib-Zwischenspeichers die Bandbreite zum Schreiben auf die IDE-Festplatte verringert, kann es aus Gründen der Datenkonsistenz als notwendig angesehen werden. Das Problem ist, dass IDE-Platten keine zuverlässige Aussage über das Ende eines Schreibvorgangs treffen. Wenn der Schreib-Zwischenspeicher aktiviert ist, werden die Daten nicht in der Reihenfolge ihres Eintreffens geschrieben. Es kann sogar passieren, dass das Schreiben mancher Blöcke im Fall von starker Plattenaktivität auf unbefristete Zeit verzögert wird. Ein Absturz oder Stromausfall zu dieser Zeit kann die Dateisysteme erheblich beschädigen. Sie sollten den Wert der sysctl(8)-Variable hw.ata.wc auf dem System überprüfen. Wenn der Schreib-Zwischenspeicher abgestellt ist, können Sie ihn beim Systemstart aktivieren, indem Sie die Variable in /boot/loader.conf auf den Wert 1 setzen.

Weitere Informationen finden Sie in ata(4).

### 26.1.6. SCSI_DELAY (kern.cam.scsi_delay)

Mit der Kerneloption SCSI_DELAY kann die Dauer des Systemstarts verringert werden. Der Vorgabewert ist recht hoch und er verzögert den Systemstart um 15 oder mehr Sekunden. Normalerweise kann dieser Wert, insbesondere mit modernen Laufwerken, mit der sysctl(8) -Variable kern.cam.scsi_delay auf 5 Sekunden heruntergesetzt werden. Die Variable sowie die Kerneloption verwenden für die Zeitangabe Millisekunden und *nicht* Sekunden.

## 26.2. Soft Updates

Mit tunefs(8) lassen sich Feineinstellungen an Dateisystemen vornehmen. Das Programm hat

verschiedene Optionen. Soft Updates werden wie folgt ein- und ausgeschaltet:

```
# tunefs -n enable /filesystem
# tunefs -n disable /filesystem
```

Ein eingehängtes Dateisystem kann nicht mit tunefs(8) modifiziert werden. Soft Updates werden am besten im Single-User Modus aktiviert, bevor Partitionen eingehangen sind.

Durch Einsatz eines Zwischenspeichers wird die Performance im Bereich der Metadaten, vorwiegend beim Anlegen und Löschen von Dateien, gesteigert. Es wird empfohlen, Soft Updates auf allen UFS-Dateisystemen zu aktivieren. Allerdings sollten Sie sich über die zwei Nachteile von Soft Updates bewusst sein: Erstens garantieren Soft Updates zwar die Konsistenz der Daten im Fall eines Absturzes, aber es kann passieren, dass das Dateisystem über mehrere Sekunden oder gar eine Minute nicht synchronisiert wurde. Nicht geschriebene Daten gehen dann vielleicht verloren. Zweitens verzögern Soft Updates die Freigabe von Datenblöcken. Eine größere Aktualisierung eines fast vollen Dateisystems, wie dem Root-Dateisystem, z.B. während eines `make installworld`, kann das Dateisystem vollaufen lassen. Dadurch würde die Aktualisierung fehlschlagen.

### 26.2.1. Details über Soft Updates

Bei einem Metadaten-Update werden die Inodes und Verzeichniseinträge aktualisiert auf die Platte zurückgeschrieben. Es gibt zwei klassische Ansätze, um die Metadaten des Dateisystems auf die Platte zu schreiben.

Das historisch übliche Verfahren waren synchrone Updates der Metadaten, d. h. wenn eine Änderung an einem Verzeichnis nötig war, wurde anschließend gewartet, bis diese Änderung tatsächlich auf die Platte zurückgeschrieben worden war. Der *Inhalt* der Dateien wurde im "Buffer Cache" zwischengespeichert und später asynchron auf die Platte geschrieben. Der Vorteil dieser Implementierung ist, dass sie sicher funktioniert. Wenn während eines Updates ein Ausfall erfolgt, haben die Metadaten immer einen konsistenten Zustand. Eine Datei ist entweder komplett angelegt oder gar nicht. Wenn die Datenblöcke einer Datei im Fall eines Absturzes noch nicht den Weg aus dem "Buffer Cache" auf die Platte gefunden haben, kann fsck(8) das Dateisystem reparieren, indem es die Dateilänge einfach auf `0` setzt. Außerdem ist die Implementierung einfach und überschaubar. Der Nachteil ist, dass Änderungen der Metadaten sehr langsam vor sich gehen. Ein `rm -r` beispielsweise fasst alle Dateien eines Verzeichnisses der Reihe nach an, aber jede dieser Änderungen am Verzeichnis (Löschen einer Datei) wird einzeln synchron auf die Platte geschrieben. Gleiches beim Auspacken großer Hierarchien mit `tar -x`.

Der zweite Ansatz sind asynchrone Metadaten-Updates. Das ist der Standard, wenn UFS-Dateisysteme mit `mount -o async` eingehängt werden. Man schickt die Updates der Metadaten einfach auch noch über den "Buffer Cache", sie werden also zwischen die Updates der normalen Daten eingeschoben. Vorteil ist, dass man nun nicht mehr auf jeden Update warten muss, Operationen, die zahlreiche Metadaten ändern, werden also viel schneller. Auch hier ist die Implementierung sehr einfach und wenig anfällig für Fehler. Nachteil ist, dass keinerlei Konsistenz des Dateisystems mehr gesichert ist. Wenn mitten in einer Operation, die viele Metadaten ändert, ein Ausfall erfolgt (Stromausfall, drücken des Reset-Schalters), dann ist das Dateisystem anschließend in einem unbestimmten Zustand. Niemand kann genau sagen, was noch geschrieben worden ist und was nicht mehr; die Datenblöcke einer Datei können schon auf der Platte stehen,

während die inode Tabelle oder das zugehörige Verzeichnis nicht mehr aktualisiert worden ist. Man kann praktisch kein fsck(8) mehr implementieren, das diesen Zustand wieder reparieren kann, da die dazu nötigen Informationen einfach auf der Platte fehlen. Wenn ein Dateisystem irreparabel beschädigt wurde, hat man nur noch die Möglichkeit es neu zu erzeugen und die Daten vom Backup zurückspielen.

Der Ausweg aus diesem Dilemma ist ein *dirty region logging*, was auch als *Journalling* bezeichnet wird. Man schreibt die Metadaten-Updates zwar synchron, aber nur in einen kleinen Plattenbereich, die *logging area*. Von da aus werden sie dann asynchron auf ihre eigentlichen Bereiche verteilt. Da die *logging area* ein kleines zusammenhängendes Stückchen ist, haben die Schreibköpfe der Platte bei massiven Operationen auf Metadaten keine allzu großen Wege zurückzulegen, so dass alles ein ganzes Stück schneller geht als bei klassischen synchronen Updates. Die Komplexität der Implementierung hält sich ebenfalls in Grenzen, somit auch die Anfälligkeit für Fehler. Als Nachteil ergibt sich, dass Metadaten zweimal auf die Platte geschrieben werden müssen (einmal in die *logging area*, einmal an die richtige Stelle), so dass das im Falle regulärer Arbeit (also keine gehäuften Metadatenoperationen) eine "Pessimisierung" des Falls der synchronen Updates eintritt, es wird alles langsamer. Dafür hat man als Vorteil, dass im Falle eines Absturzes der konsistente Zustand dadurch erzielbar ist, dass die angefangenen Operationen aus dem *dirty region log* entweder zu Ende ausgeführt oder komplett verworfen werden, wodurch das Dateisystem schnell wieder zur Verfügung steht.

Die Lösung von Kirk McKusick, dem Schöpfer von Berkeley FFS, waren *Soft Updates*: die notwendigen Updates der Metadaten werden im Speicher gehalten und dann sortiert auf die Platte geschrieben ("ordered metadata updates"). Dadurch hat man den Effekt, dass im Falle massiver Metadaten-Änderungen spätere Operationen die vorhergehenden, noch nicht auf die Platte geschriebenen Updates desselben Elements im Speicher "einholen". Alle Operationen, auf ein Verzeichnis beispielsweise, werden also in der Regel noch im Speicher abgewickelt, bevor der Update überhaupt auf die Platte geschrieben wird (die dazugehörigen Datenblöcke werden natürlich auch so sortiert, dass sie nicht vor ihren Metadaten auf der Platte sind). Im Fall eines Absturzes hat man ein implizites "log rewind": alle Operationen, die noch nicht den Weg auf die Platte gefunden haben, sehen danach so aus, als hätten sie nie stattgefunden. Man hat so also den konsistenten Zustand von ca. 30 bis 60 Sekunden früher sichergestellt. Der verwendete Algorithmus garantiert dabei, dass alle tatsächlich benutzten Ressourcen auch in den entsprechenden Bitmaps (Block- und inode Tabellen) als belegt markiert sind. Der einzige Fehler, der auftreten kann, ist, dass Ressourcen noch als "belegt" markiert sind, die tatsächlich "frei" sind. fsck(8) erkennt dies und korrigiert diese nicht mehr belegten Ressourcen. Die Notwendigkeit eines Dateisystem-Checks darf aus diesem Grunde auch ignoriert und das Dateisystem mittels `mount -f` zwangsweise eingebunden werden. Um noch allozierte Ressourcen freizugeben muss später ein fsck(8) nachgeholt werden. Das ist dann auch die Idee des *background fsck*: beim Starten des Systems wird lediglich ein *Schnappschuss* des Dateisystems gemacht, mit dem fsck(8) dann später arbeiten kann. Alle Dateisysteme dürfen "unsauber" eingebunden werden und das System kann sofort in den Multiuser-Modus gehen. Danach wird ein Hintergrund-fsck(8) für die Dateisysteme gestartet, die dies benötigen, um möglicherweise irrtümlich belegte Ressourcen freizugeben. Dateisysteme ohne *Soft Updates* benötigen natürlich immer noch den üblichen Vordergrund-fsck(8), bevor sie eingebunden werden können.

Der Vorteil ist, dass die Metadaten-Operationen beinahe so schnell ablaufen wie im asynchronen Fall, also auch schneller als beim *logging*, das die Metadaten immer zweimal schreiben muss. Als

Nachteil stehen dem die Komplexität des Codes, ein erhöhter Speicherverbrauch und einige spezielle Eigenheiten entgegen. Nach einem Absturz ist ein etwas "älterer" Stand auf der Platte - statt einer leeren, aber bereits angelegten Datei, wie nach einem herkömmlichen fsck(8) Lauf, ist auf einem Dateisystem mit *Soft Updates* keine Spur der entsprechenden Datei mehr zu sehen, da weder die Metadaten noch der Dateiinhalt je auf die Platte geschrieben wurden. Weiterhin kann der Platz nach einem rm(1) nicht sofort wieder als verfügbar markiert werden, sondern erst dann, wenn der Update auch auf die Platte vermittelt worden ist. Dies kann besonders dann Probleme bereiten, wenn große Datenmengen in einem Dateisystem installiert werden, das nicht genügend Platz hat, um alle Dateien zweimal unterzubringen.

# Kapitel 27. Einstellungen von Kernel Limits

## 27.1. Datei und Prozeß Limits

### 27.1.1. `kern.maxfiles`

Abhängig von den Anforderungen an das System kann die sysctl(8)-Variable `kern.maxfiles` erhöht oder gesenkt werden. Die Variable legt die maximale Anzahl von Dateideskriptoren auf dem System fest. Wenn die Dateideskriptoren aufgebraucht sind, werden Sie die Meldung `file: table is full` wiederholt im Puffer für Systemmeldungen sehen. Den Inhalt des Puffers können Sie sich mit dmesg(8) anzeigen lassen.

Jede offene Datei, jedes Socket und jede FIFO verbraucht einen Dateideskriptor. Auf "dicken" Produktionsservern können leicht Tausende Dateideskriptoren benötigt werden, abhängig von der Art und Anzahl der gleichzeitig laufenden Dienste.

In älteren FreeBSD-Versionen wurde die Voreinstellung von `kern.maxfile` aus der Kernelkonfigurationsoption `maxusers` bestimmt. `kern.maxfiles` wächst proportional mit dem Wert von `maxusers`. Wenn Sie einen angepassten Kernel kompilieren, empfiehlt es sich diese Option entsprechend der maximalen Benutzerzahl des Systems einzustellen. Obwohl auf einer Produktionsmaschine vielleicht nicht 256 Benutzer gleichzeitig angemeldet sind, können die benötigten Ressourcen ähnlich hoch wie bei einem großen Webserver sein.

Die nur lesbare sysctl(8)-Variable `kern.maxusers` wird beim Systemstart automatisch aus dem zur Verfügung stehenden Hauptspeicher bestimmt. Im laufenden Betrieb kann dieser Wert aus `kern.maxusers` ermittelt werden. Einige Systeme benötigen für diese Variable einen anderen Wert, wobei `64`, `128` und `256` gewöhnliche Werte darstellen. Es wird nicht empfohlen, die Anzahl der Dateideskriptoren auf einen Wert größer `256` zu setzen, es sei denn, Sie benötigen wirklich eine riesige Anzahl von ihnen. Viele der von `kern.maxusers` auf einen Standardwert gesetzten Parameter können beim Systemstart oder im laufenden Betrieb in /boot/loader.conf angepasst werden. In loader.conf(5) und /boot/defaults/loader.conf finden Sie weitere Details und Hinweise.

Ältere FreeBSD-Versionen setzen diesen Wert selbst, wenn Sie in der Konfigurationsdatei den Wert `0` angeben. Wenn Sie den Wert selbst bestimmen wollen, sollten Sie `maxusers` mindestens auf `4` setzen. Dies gilt insbesondere dann, wenn Sie beabsichtigen, Xorg zu benutzen oder Software zu kompilieren. Der wichtigste Wert, der durch `maxusers` bestimmt wird, die maximale Anzahl an Prozessen ist, die auf `20 + 16 * maxusers` gesetzt wird. Wird `maxusers` auf `1` setzen, können gleichzeitig nur `36` Prozesse laufen, von denen ungefähr `18` schon beim Booten des Systems gestartet werden. Dazu kommen nochmals etwa `15` Prozesse beim Start von Xorg. Selbst eine einfache Aufgabe wie das Lesen einer Manualpage benötigt neun Prozesse zum Filtern, Dekomprimieren und Betrachten der Datei. Für die meisten Benutzer sollte es ausreichen, `maxusers` auf `64` zu setzen, womit `1044` gleichzeitige Prozesse zur Verfügung stehen. Wenn Sie allerdings den Fehler beim Start eines Programms oder auf einem Server mit einer großen Benutzerzahl sehen, dann sollten Sie den Wert nochmals erhöhen und den Kernel neu bauen.



Die Anzahl der Benutzer, die sich auf einem Rechner anmelden kann, wird durch `maxusers` *nicht* begrenzt. Der Wert dieser Variablen legt neben der möglichen

Anzahl der Prozesse eines Benutzers weitere sinnvolle Größen für bestimmte Systemtabellen fest.

### 27.1.2. kern.ipc.soacceptqueue

Die sysctl(8)-Variable kern.ipc.soacceptqueue beschränkt die Größe der Warteschlange (Listen-Queue) für neue TCP-Verbindungen. Der Vorgabewert von 128 ist normalerweise zu klein, um neue Verbindungen auf einem stark ausgelasteten Webserver zuverlässig zu handhaben. Auf solchen Servern sollte der Wert auf 1024 oder höher gesetzt werden. Dienste wie sendmail(8) oder Apache können die Größe der Queue selbst einschränken. Oft gibt es die Möglichkeit, die Größe der Listen-Queue in einer Konfigurationsdatei einzustellen. Eine große Listen-Queue übersteht vielleicht auch einen Denial of Service Angriff ().

## 27.2. Netzwerk Limits

Die Kerneloption NMBCLUSTERS schreibt die Anzahl der Netzwerkpuffer (Mbufs) fest, die das System besitzt. Eine zu geringe Anzahl Mbufs auf einem Server mit viel Netzwerkverkehr verringert die Leistung von FreeBSD. Jeder Mbuf-Cluster nimmt ungefähr 2 kB Speicher in Anspruch, so dass ein Wert von 1024 insgesamt 2 Megabyte Speicher für Netzwerkpuffer im System reserviert. Wie viele Cluster benötigt werden, lässt sich durch eine einfache Berechnung herausfinden. Ein Webserver, der maximal 1000 gleichzeitige Verbindungen servieren soll, wobei jede der Verbindungen einen 6 kB großen Sendepuffer und einen 16 kB großen Empfangspuffer benötigt, braucht ungefähr 32 MB Speicher für Netzwerkpuffer. Als Daumenregel verdoppeln Sie diese Zahl, so dass sich für NMBCLUSTERS der Wert 2x32 MB / 2 kB= 64 MB / 2 kB= 32768 ergibt. Für Maschinen mit viel Speicher werden Werte zwischen 4096 und 32768 empfohlen. Unter keinen Umständen sollten Sie diesen Wert willkürlich erhöhen, da dies zu einem Absturz beim Systemstart führen kann. Verwenden Sie netstat(1) mit -m um den Gebrauch der Netzwerkpuffer zu kontrollieren.

Die Netzwerkpuffer können beim Systemstart mit der Loader-Variablen kern.ipc.nmbclusters eingestellt werden. Nur auf älteren FreeBSD-Systemen müssen Sie die Kerneloption NMBCLUSTERS verwenden.

Die Anzahl der sendfile(2) Puffer muss auf ausgelasteten Servern, die den Systemaufruf sendfile(2) oft verwenden, vielleicht erhöht werden. Dazu können Sie die Kerneloption NSFBUFS verwenden oder die Anzahl der Puffer in /boot/loader.conf (siehe loader(8)) setzen. Die Puffer sollten erhöht werden, wenn Sie Prozesse im Zustand sfbufa sehen. Die schreibgeschützte sysctl(8)-Variable kern.ipc.nsfbufs zeigt die Anzahl eingerichteten Puffer im Kernel. Der Wert dieser Variablen wird normalerweise von kern.maxusers bestimmt. Manchmal muss die Pufferanzahl jedoch manuell eingestellt werden.



Auch wenn ein Socket nicht blockierend angelegt wurde, kann der Aufruf von sendfile(2) blockieren, um auf freie struct sf_buf Puffer zu warten.

### 27.2.1. net.inet.ip.portrange.*

Die sysctl(8)-Variable net.inet.ip.portrange.* legt die Portnummern für TCP- und UDP-Sockets fest. Es gibt drei Bereiche: den niedrigen Bereich, den normalen Bereich und den hohen Bereich. Die

meisten Netzprogramme benutzen den normalen Bereich. Dieser Bereich umfasst in der Voreinstellung die Portnummern 1024 bis 5000 und wird durch die Variablen net.inet.ip.portrange.first und net.inet.ip.portrange.last festgelegt. Die festgelegten Bereiche für Portnummern werden von ausgehenden Verbindungen benutzt. Unter bestimmten Umständen, beispielsweise auf stark ausgelasteten Proxy-Servern, sind alle Portnummern für ausgehende Verbindungen belegt. Bereiche für Portnummern spielen auf Servern keine Rolle, die hauptsächlich eingehende Verbindungen verarbeiten (wie ein normaler Webserver) oder nur eine begrenzte Anzahl ausgehender Verbindungen öffnen (beispielsweise ein Mail-Relay). Wenn keine freien Portnummern mehr vorhanden sind, sollte die Variable net.inet.ip.portrange.last langsam erhöht werden. Ein Wert von 10000, 20000 oder 30000 ist angemessen. Beachten Sie auch eine vorhandene Firewall, wenn Sie die Bereiche für Portnummern ändern. Einige Firewalls sperren große Bereiche (normalerweise aus den kleinen Portnummern) und erwarten, dass hohe Portnummern für ausgehende Verbindungen verwendet werden. Daher kann es erforderlich sein, den Wert von net.inet.ip.portrange.first zu erhöhen.

### 27.2.2. TCP Bandwidth Delay Product Begrenzung

Die TCP Bandwidth Delay Product Begrenzung wird aktiviert, indem die sysctl(8)-Variable net.inet.tcp.inflight.enable auf den Wert 1 gesetzt wird. Das System wird dadurch angewiesen, für jede Verbindung, das Produkt aus der Übertragungsrate und der Verzögerungszeit zu bestimmen. Dieses Produkt begrenzt die Datenmenge, die für einen optimalen Durchsatz zwischengespeichert werden muss.

Diese Begrenzung ist nützlich, wenn Sie Daten über Verbindungen mit einem hohen Produkt aus Übertragungsrate und Verzögerungszeit wie Modems, Gigabit-Ethernet oder schnellen WANs, zur Verfügung stellen. Insbesondere wirkt sich die Begrenzung aus, wenn die Verbindung die Option Window-scaling verwendet oder große Sende-Fenster (send window) benutzt. Schalten Sie die Debug-Meldungen aus, wenn Sie die Begrenzung aktiviert haben. Dazu setzen Sie die Variable net.inet.tcp.inflight.debug auf 0. Auf Produktions-Systemen sollten Sie zudem die Variable net.inet.tcp.inflight.min mindestens auf den Wert 6144 setzen. Allerdings kann ein zu hoher Wert, abhängig von der Verbindung, die Begrenzungsfunktion unwirksam machen. Die Begrenzung reduziert die Datenmenge in den Queues von Routern und Switches, sowie die Datenmenge in der Queue der lokalen Netzwerkkarte. Die Verzögerungszeit (Round Trip Time) für interaktive Anwendungen sinkt, da weniger Pakete zwischengespeichert werden. Dies gilt besonders für Verbindungen über langsame Modems. Die Begrenzung wirkt sich allerdings nur auf das Versenden von Daten aus (Uploads, Server). Auf den Empfang von Daten (Downloads) hat die Begrenzung keine Auswirkungen.

Die Variable net.inet.tcp.inflight.stab sollte *nicht* angepasst werden. Der Vorgabewert der Variablen beträgt 20, das heißt es werden maximal zwei Pakete zu dem Produkt aus Übertragungsrate und Verzögerungszeit addiert. Dies stabilisiert den Algorithmus und verbessert die Reaktionszeit auf Veränderungen. Bei langsamen Verbindungen können sich aber die Laufzeiten der Pakete erhöhen (ohne diesen Algorithmus wären sie allerdings noch höher). In solchen Fällen können Sie versuchen, den Wert der Variablen auf 15, 10 oder 5 herabzusetzen. Gleichzeitig müssen Sie vielleicht auch net.inet.tcp.inflight.min auf einen kleineren Wert (beispielsweise 3500) setzen. Ändern Sie diese Variablen nur ab, wenn Sie keine anderen Möglichkeiten mehr haben.

## 27.3. Virtueller Speicher (Virtual Memory)

### 27.3.1. kern.maxvnodes

Ein vnode ist die interne Darstellung einer Datei oder eines Verzeichnisses. Die Erhöhung der Anzahl der für das Betriebssystem verfügbaren vnodes verringert also die Schreib- und Lesezugriffe auf der Festplatte. vnodes werden im Normalfall vom Betriebssystem automatisch vergeben und müssen nicht manuell angepasst werden. In einigen Fällen stellt der Zugriff auf eine Platte allerdings einen Flaschenhals dar, daher sollten Sie in diesem Fall die Anzahl der möglichen vnodes erhöhen, um dieses Problem zu beheben. Beachten Sie dabei aber die Größe des inaktiven und freien Hauptspeichers.

Um die Anzahl der derzeit verwendeten vnodes zu sehen, geben Sie Folgendes ein:

```
# sysctl vfs.numvnodes
vfs.numvnodes: 91349
```

Die maximal mögliche Anzahl der vnodes erhalten Sie durch die Eingabe von:

```
# sysctl kern.maxvnodes
kern.maxvnodes: 100000
```

Wenn sich die Anzahl der genutzten vnodes dem maximal möglichen Wert nähert, sollten Sie den Wert `kern.maxvnodes` zuerst um etwa `1000` erhöhen. Beobachten Sie danach die Anzahl der vom System genutzten `vfs.numvnodes`. Nähert sich der Wert wiederum dem definierten Maximum, müssen Sie `kern.maxvnodes` nochmals erhöhen. Sie sollten nun eine Änderung des Speicherverbrauches über top(1) registrieren können und über mehr aktiven Speicher verfügen.

# Kapitel 28. Hinzufügen von Swap-Bereichen

Manchmal benötigt ein System mehr Swap-Bereiche. Dieser Abschnitt beschreibt zwei Methoden, um Swap-Bereiche hinzuzufügen: auf einer bestehenden Partition oder auf einem neuen Laufwerk, und das Hinzufügen einer Swap-Datei auf einer existierenden Partition.

Für Informationen zur Verschlüsselung von Swap-Partitionen, zu den dabei möglichen Optionen sowie zu den Gründen für eine Verschlüsselung des Auslagerungsspeichers lesen Sie “Den Auslagerungsspeicher verschlüsseln”.

## 28.1. Swap auf einer neuen Festplatte oder einer existierenden Partition

Das Hinzufügen einer neuen Festplatte für den Swap-Bereich bietet eine bessere Leistung, als die Verwendung einer Partition auf einem vorhandenem Laufwerk. Die Einrichtung von Partitionen und Laufwerken wird in “Hinzufügen von Laufwerken“ beschrieben. “Ein Partitionslayout entwerfen“ diskutiert Aspekte über die Anordnung und Größe von Swap-Bereichen.

Benutzen Sie `swapon` um eine Swap-Partition zum System hinzuzufügen. Zum Beispiel:

```
# swapon /dev/ada1s1b
```



Sie können jede Partition verwenden, sofern sie nicht schon eingehangen ist. Das gilt auch dann, wenn die Partition bereits Daten enthält. Wird `swapon` auf einer Partition ausgeführt die Daten enthält, werden die vorhandenen Daten überschrieben und sind unweigerlich verloren. Stellen Sie sicher, dass die Partition, die Sie als Swap-Bereich hinzufügen möchten, wirklich die gewünschte Partition ist, bevor Sie `swapon` ausführen.

Um diese Swap-Partition automatisch beim Systemstart hinzuzufügen, fügen Sie einen Eintrag in /etc/fstab hinzu:

```
/dev/ada1s1b    none    swap    sw  0   0
```

Die einzelnen Einträge von /etc/fstab werden in fstab(5) erläutert. Weitere Informationen zu `swapon` finden Sie in swapon(8).

## 28.2. Swap-Dateien erstellen

Anstatt eine Partition zu verwenden, erstellen diese Beispiele eine 512 MB große Swap-Datei mit dem Namen /usr/swap0.

Die Verwendung von Swap-Dateien macht es erforderlich, dass das Modul md(4) entweder im Kernel vorhanden oder geladen wird, bevor Swap aktiviert ist. Konfiguration des FreeBSD-Kernels enthält Informationen zum Bau eines angepassten Kernels.

*Beispiel 26. Erstellen einer Swap-Datei*

1. Erstellen Sie die Swap-Datei:

```
# dd if=/dev/zero of=/usr/swap0 bs=1024k count=512
```

2. Setzen Sie die richtigen Berechtigungen für die neue Datei:

```
# chmod 0600 /usr/swap0
```

3. Fügen Sie einen Eintrag in /etc/fstab hinzu:

```
md99    none    swap    sw,file=/usr/swap0,late 0   0
```

   Das md(4) Gerät md99 wird verwendet, damit die niedrigeren Gerätenummer für die interaktive Benutzung frei bleiben.

4. Der Swap-Speicher wird nun automatisch beim Systemstart hinzugefügt. Benutzen Sie swapon(8) um den Swap-Speicher direkt zu aktivieren:

```
# swapon -aL
```

# Kapitel 29. Energie- und Ressourcenverwaltung

Es ist wichtig, Hardware effizient einzusetzen. Energie- und Ressourcenverwaltung ermöglicht es dem System auf verschiedene Ereignisse, beispielsweise einen unerwarteten Temperaturanstieg, reagieren zu können. Eine frühe Spezifikation für die Energieverwaltung war das Advanced Power Management (APM). APM steuert den Energieverbrauch eines Systems auf Basis der Systemaktivität. Ursprünglich konnten Stromverbrauch und Wärmeabgabe eines Systems nur schlecht von Betriebssystemen gesteuert werden. Die Hardware wurde vom BIOS gesteuert, was die Kontrolle der Energieverwaltung für den Anwender erschwerte. Das APM-BIOS wird von dem Hersteller des Systems zur Verfügung gestellt und ist auf die spezielle Hardware angepasst. Der APM-Treiber des Betriebssystems greift auf das *APM Software Interface* zu, das den Energieverbrauch regelt.

APM hat hauptsächlich vier Probleme. Erstens läuft die Energieverwaltung unabhängig vom Betriebssystem in einem herstellerspezifischen BIOS. Beispielsweise kann das APM-BIOS die Festplatten nach einer konfigurierbaren Zeit ohne die Zustimmung des Betriebssystems herunterfahren. Zweitens befindet sich die ganze APM-Logik im BIOS; das Betriebssystem hat gar keine APM-Komponenten. Bei Problemen mit dem APM-BIOS muss das Flash-ROM aktualisiert werden. Diese Prozedur ist gefährlich, da sie im Fehlerfall das System unbrauchbar machen kann. Zum Dritten ist APM eine Technik, die herstellerspezifisch ist und nicht koordiniert wird. Fehler im BIOS eines Herstellers werden nicht unbedingt im BIOS anderer Hersteller korrigiert. Das letzte Problem ist, dass im APM-BIOS nicht genügend Platz vorhanden ist, um eine durchdachte oder eine auf den Zweck der Maschine zugeschnittene Energieverwaltung zu implementieren.

Das *Plug and Play BIOS (PNPBIOS)* war in vielen Situationen ebenfalls unzureichend. Das PNPBIOS verwendet eine 16-Bit-Technik. Damit das Betriebssystem das PNPBIOS ansprechen kann, muss es in einer 16-Bit-Emulation laufen. FreeBSD stellt einen APM-Treiber zur Verfügung, welcher für Systeme benutzt werden sollte, die vor dem Jahr 2000 hergestellt wurden. Der Treiber wird in apm(4) beschrieben.

Der Nachfolger von APM ist das *Advanced Configuration and Power Interface* (ACPI). ACPI ist ein Standard verschiedener Hersteller, welcher die Verwaltung von Hardware und Energiesparfunktionen festlegt. Die ACPI-Funktionen, die mehr Kontrolle und Flexibilität bieten, können vom Betriebssystem gesteuert werden.

Dieser Abschnitt zeigt die Konfiguration von ACPI unter FreeBSD. Zudem werden einige Tipps zur Fehlersuche vorgestellt und wie Sie Problemberichte einreichen können, sodass Entwickler ACPI-Probleme erfassen und beheben können.

## 29.1. Konfiguration des ACPI

Der acpi(4)-Treiber wird standardmäßig beim Systemstart vom loader(8) geladen und sollte daher *nicht* fest in den Kernel eingebunden werden. Der Treiber kann im laufenden Betrieb nicht entfernt werden, da er zur Kommunikation mit der Hardware verwendet wird. Falls jedoch Probleme auftreten, kann ACPI auch komplett deaktiviert werden. Dazu muss `hint.acpi.0.disabled="1"` in /boot/loader.conf gesetzt und anschließend das System neu gestartet werden. Alternativ können Sie

diese Variable auch am loader(8)-Prompt eingeben, wie in "Phase Drei" beschrieben.



ACPI und APM können nicht zusammen verwendet werden. Das zuletzt geladene Modul beendet sich, sobald es bemerkt, dass das andere Modul geladen ist.

Mit `acpiconf` können Sie das System in einen Ruhemodus (sleep mode) versetzen. Es gibt verschiedene Modi (von `1` bis `5`), die Sie auf der Kommandozeile mit `-s` angeben können. Für die meisten Anwender sind die Modi `1` und `3` völlig ausreichend. Der Modus `5` schaltet das System aus (Soft-off) und entspricht dem Befehl `halt -p`.

Verschiedene Optionen können mit `sysctl` gesetzt werden. Lesen Sie dazu acpi(4) sowie acpiconf(8).

# 29.2. Häufige Probleme

ACPI gibt es in allen modernen Rechnern der ia32- (x86) und amd64- (AMD) Architektur. Der vollständige Standard bietet Funktionen zur Steuerung und Verwaltung der CPU-Leistung, der Stromversorgung, von Wärmebereichen, Batterien, eingebetteten Controllern und Bussen. Auf den meisten Systemen wird nicht der vollständige Standard implementiert. Arbeitsplatzrechner besitzen meist nur Funktionen zur Verwaltung der Busse, während Notebooks Funktionen zur Temperaturkontrolle und Ruhezustände besitzen.

Ein ACPI konformes System besitzt verschiedene Komponenten. Die BIOS- und Chipsatz-Hersteller stellen mehrere statische Tabellen bereit, zum Beispiel die Fixed-ACPI-Description-Table (FADT). Die Tabellen enthalten beispielsweise die mit SMP-Systemen benutzte APIC-Map, Konfigurationsregister und einfache Konfigurationen. Zusätzlich gibt es die *Differentiated-System-Description-Table* (DSDT), die Bytecode enthält. Die Tabelle ordnet Geräte und Methoden in einem baumartigen Namensraum an.

Ein ACPI-Treiber muss die statischen Tabellen einlesen, einen Interpreter für den Bytecode bereitstellen und die Gerätetreiber im Kernel so modifizieren, dass sie mit dem ACPI-Subsystem kommunizieren. Für FreeBSD, Linux® und NetBSD hat Intel® den Interpreter ACPI-CA, zur Verfügung gestellt. Der Quelltext zu ACPI-CA befindet sich im Verzeichnis src/sys/contrib/dev/acpica. Die Schnittstelle von ACPI-CA zu FreeBSD befindet sich unter src/sys/dev/acpica/Osd. Treiber, die verschiedene ACPI-Geräte implementieren, befinden sich im Verzeichnis src/sys/dev/acpica.

Damit ACPI richtig funktioniert, müssen alle Teile funktionieren. Im Folgenden finden Sie eine Liste mit Problemen und möglichen Abhilfen oder Korrekturen. Die Liste ist nach der Häufigkeit, mit der die Probleme auftreten, sortiert. Wenn eine Korrektur das Problem nicht behebt, finden Sie in Abrufen und Einreichen von Informationen zur Fehlersuche Anweisungen, wie Sie einen Problembericht einreichen können.

## 29.2.1. Mausprobleme

Es kann vorkommen, dass die Maus nicht mehr funktioniert, wenn Sie nach einem Suspend weiterarbeiten wollen. Ist dies bei Ihnen der Fall, reicht es meistens aus, den Eintrag `hint.psm.0.flags="0x3000"` in /boot/loader.conf aufzunehmen.

### 29.2.2. Suspend/Resume

ACPI kennt drei Suspend-to-RAM-Zustände (STR), S1-S3 sowie einen Suspend-to-Disk-Zustand (STD) S4. STD kann auf zwei Arten implementiert werden: S4BIOS und S4OS. Im ersten Fall wird der Suspend-to-Disk-Zustand durch das BIOS hergestellt im zweiten Fall alleine durch das Betriebssystem. Der Zustand S5 wird "Soft off" genannt. In diesem Zustand befindet sich ein Rechner, wenn die Stromversorgung angeschlossen ist, der Rechner aber nicht hochgefahren ist.

Benutzen Sie `sysctl hw.acpi` um die Suspend-Zustände zu ermitteln. Diese Beispielausgabe stammt von einem Thinkpad:

```
hw.acpi.supported_sleep_state: S3 S4 S5
hw.acpi.s4bios: 0
```

Diese Ausgabe besagt, dass mit dem Befehl `acpiconf -s` die Zustände S3, S4 und S5 eingestellt werden können. Hätte `s4bios` den Wert 1, gäbe es den Zustand S4BIOS anstelle von S4.

Wenn Sie die Suspend- und Resume-Funktionen testen, fangen Sie mit dem S1-Zustand an, wenn er angeboten wird. Dieser Zustand wird am ehesten funktionieren, da der Zustand wenig Treiber-Unterstützung benötigt. Der Zustand S2 ist ähnlich wie S1, allerdings hat ihn noch niemand implementiert. Als nächstes sollten Sie den Zustand S3 ausprobieren. Dies ist der tiefste STR-Schlafzustand. Dieser Zustand ist auf massive Treiber-Unterstützung angewiesen, um die Geräte wieder richtig zu initialisieren.

Ein häufiges Problem mit Suspend/Resume ist, dass viele Gerätetreiber ihre Firmware, Register und Gerätespeicher nicht korrekt speichern, wiederherstellen und/oder reinitialisieren. Um dieses Problem zu lösen, sollten Sie zuerst die folgenden Befehle ausführen:

```
# sysctl debug.bootverbose=1
# sysctl debug.acpi.suspend_bounce=1
# acpiconf -s 3
```

Dieser Test emuliert einen Suspend/Resume-Zyklus für alle Geräte (ohne dass diese dabei wirklich in den Status S3 wechseln). In vielen Fällen reicht dies bereits aus, um Probleme (beispielsweise verlorener Firmware-Status, Timeouts, hängende Geräte) zu entdecken. Beachten Sie dabei, dass das Gerät bei diesem Test nicht wirklich in den Status S3 wechseln. Es kann also vorkommen, dass manche Geräte weiterhin mit Strom versorgt werden (dies wäre bei einem wirklichen Wechsel in den Status S3 NICHT möglich. Andere Geräte werden normal weiterarbeiten, weil sie über keine Suspend/Resume-Funktionen verfügen.

Schwierigere Fälle können den Einsatz zusätzlicher Hardware (beispielsweise serielle Ports/Kabel für die Verbindung über eine serielle Konsole oder Firewire-Ports/Kabel für dcons(4)) sowie Kenntnisse im Bereich Kerneldebugging erforderlich machen.

Um das Problem einzugrenzen, entladen Sie soviele Treiber wie möglich. Wenn das funktioniert, laden Sie einen Treiber nach dem anderen, bis der Fehler wieder auftritt. Typischerweise verursachen binäre Treiber wie nvidia.ko, Grafiktreiber und USB-Treiber die meisten Fehler,

hingegen laufen Ethernet-Treiber für gewöhnlich sehr zuverlässig. Wenn ein Treiber zuverlässig geladen und entfernt werden kann, können Sie den Vorgang automatisieren, indem Sie die entsprechenden Kommandos in /etc/rc.suspend und /etc/rc.resume einfügen. In den Dateien finden Sie ein deaktiviertes Beispiel, das einen Treiber lädt und wieder entfernt. Ist die Bildschirmanzeige bei der Wiederaufnahme des Betriebs gestört, setzen Sie die Variable `hw.acpi.reset_video` auf `1`. Versuchen Sie auch, die Variable `hw.acpi.sleep_delay` auf kürzere Zeitspannen zu setzen.

Die Suspend- und Resume-Funktionen können Sie auch auf einer neuen Linux®-Distribution mit ACPI testen. Wenn es mit Linux® funktioniert, liegt das Problem wahrscheinlich bei einem FreeBSD-Treiber. Es hilft uns, das Problem zu lösen, wenn Sie feststellen können, welcher Treiber das Problem verursacht. Beachten Sie bitte, dass die ACPI-Entwickler normalerweise keine anderen Treiber pflegen (beispielsweise Sound- oder ATA-Treiber). Es ist wohl das beste, die Ergebnisse der Fehlersuche an die Mailingliste FreeBSD-CURRENT und den Entwickler des Treibers zu schicken. Erfahrene Benutzer können versuchen, den Fehler in der Resume-Funktion zu finden, indem sie einige printf(3)-Anweisungen in den Code des fehlerhaften Treibers einfügen.

Schließlich können Sie ACPI noch abschalten und stattdessen APM verwenden. Wenn die Suspend- und Resume-Funktionen mit APM funktionieren, sollten Sie besser APM verwenden (insbesondere mit alter Hardware von vor dem Jahr 2000). Die Hersteller benötigten einige Zeit, um ACPI korrekt zu implementieren, daher gibt es mit älterer Hardware oft ACPI-Probleme.

### 29.2.3. Systemhänger

Die meisten Systemhänger entstehen durch verlorene Interrupts oder einen Interrupt-Sturm. Probleme werden verursacht durch die Art, in der das BIOS Interrupts vor dem Systemstart konfiguriert, durch eine fehlerhafte APIC-Tabelle und durch die Zustellung des System-Control-Interrupts (SCI).

Anhand der Ausgabe des Befehls `vmstat -i` können Sie verlorene Interrupts von einem Interrupt-Sturm unterscheiden. Untersuchen Sie die Ausgabezeile, die `acpi0` enthält. Ein Interrupt-Sturm liegt vor, wenn der Zähler öfter als ein paar Mal pro Sekunde hochgezählt wird. Wenn sich das System aufgehangen hat, versuchen Sie mit der Tastenkombination `Ctrl` + `Alt` + `Esc` in den Debugger DDB zu gelangen. Geben Sie dort den Befehl `show interrupts` ein.

Wenn Sie Interrupt-Probleme haben, ist es vorerst wohl am besten, APIC zu deaktivieren. Tragen Sie dazu die Zeile `hint.apic.0.disabled="1"` in /boot/loader.conf ein.

### 29.2.4. Abstürze (Panics)

Panics werden so schnell wie möglich behoben; mit ACPI kommt es aber selten dazu. Zuerst sollten Sie die Panic reproduzieren und dann versuchen einen backtrace (eine Rückverfolgung der Funktionsaufrufe) zu erstellen. Richten Sie dazu den DDB über die serielle Schnittstelle (siehe “DDB Debugger über die serielle Schnittstelle”) oder eine gesonderte dump(8)-Partition ein. In DDB können Sie den backtrace mit dem Kommando `tr` erstellen. Falls Sie den backtrace vom Bildschirm abschreiben müssen, schreiben Sie bitte mindestens die fünf ersten und die fünf letzten Zeile der Ausgabe auf.

Versuchen Sie anschließend, das Problem durch einen Neustart ohne ACPI zu beseitigen. Wenn das funktioniert hat, können Sie versuchen, das verantwortliche ACPI-Subsystem durch Setzen der

Variablen debug.acpi.disable herauszufinden. Die Hilfeseite acpi(4) enthält dazu einige Beispiele.

### 29.2.5. Nach einem Suspend oder einem Stopp startet das System wieder

Setzen Sie zuerst hw.acpi.disable_on_poweroff="0" in /boot/loader.conf. Damit wird verhindert, dass ACPI während des Systemabschlusses die Bearbeitung verschiedener Ereignisse deaktiviert. Auf manchen Systemen muss die Variable den Wert 1 besitzen (die Voreinstellung). Normalerweise wird der unerwünschte Neustart des Systems durch Setzen dieser Variablen behoben.

### 29.2.6. BIOS mit fehlerhaftem Bytecode

Einige BIOS-Hersteller liefern einen fehlerhaften Bytecode aus. Dies erkennen Sie an Kernelmeldungen wie diesen:

```
ACPI-1287: *** Error: Method execution failed [\\_SB_.PCI0.LPC0.FIGD._STA] \\
(Node 0xc3f6d160), AE_NOT_FOUND
```

Oft können Sie das Problem dadurch lösen, dass Sie eine aktuelle BIOS-Version einspielen. Die meisten Meldungen auf der Konsole sind harmlos, wenn aber beispielsweise der Batteriestatus falsch angezeigt wird, können Sie in den Meldungen nach Problemen suchen.

## 29.3. Die voreingestellte ASL überschreiben

Der BIOS-Bytecode, bekannt als ACPI Maschine Language (AML) wird aus der Sprache namens ACPI Source Language (ASL) übersetzt. Die AML ist in einer Tabelle, bekannt als Differentiated System Description Table (DSDT), abgelegt.

Es ist das Ziel von FreeBSD, dass ACPI ohne Eingriffe des Benutzers läuft. Zurzeit werden allerdings noch Abhilfen für Fehler der BIOS-Hersteller entwickelt. Der Microsoft®-Interpreter (acpi.sys und acpiec.sys) prüft die ASL nicht streng gegen den Standard. Daher reparieren BIOS-Hersteller, die ACPI nur unter Windows® testen, ihre ASL nicht. Die FreeBSD Entwickler hoffen, dass sie das vom Standard abweichende Verhalten des Microsoft®-Interpreters dokumentieren und in FreeBSD replizieren können. Dadurch müssen Benutzer ihre ASL nicht selbst reparieren.

Um bei der Fehlersuche zu helfen und das Problem möglicherweise zu beheben, kann eine Kopie der ASL gemacht werden. Dazu nutzen Sie acpidump zusammen mit -t, um den Inhalt der Tabelle anzuzeigen und -d, um die AML zu zerlegen:

```
# acpidump -td > my.asl
```

Einige AMLs gehen davon aus, dass der Anwender eine Windows®-Versionen benutzt. Versuchen Sie das Betriebssystem, das Sie in der ASL finden, in /boot/loader.conf anzugeben: hw.acpi.osname="Windows 2009".

Manche Abhilfen erfordern eine Anpassung von my.asl. Wenn diese Datei bearbeitet wird, erstellen Sie die neue ASL mit dem folgenden Befehl. Warnung können meistens ignoriert werden, aber Fehler verhindern die ordnungsgemäße Funktion von ACPI.

```
# iasl -f my.asl
```

Die Option `-f` erzwingt das Erstellen der AML auch dann, wenn während der Übersetzung Fehler auftreten. Einige Fehler, wie fehlende Return-Anweisungen, werden automatisch vom FreeBSD Interpreter umgangen.

Die voreingestellte Ausgabedatei von `iasl` ist DSDT.aml. Wenn Sie diese Datei anstelle der fehlerhaften Kopie des BIOS laden wollen, editieren Sie /boot/loader.conf wie folgt:

```
acpi_dsdt_load="YES"
acpi_dsdt_name="/boot/DSDT.aml"
```

Stellen Sie bitte sicher, dass sich DSDT.aml in /boot befindet und starten Sie das System neu. Wenn dadurch das Problem behoben wird, schicken Sie einen diff(1) der alten und der neuen ASL an FreeBSD ACPI, damit die Entwickler das Problem in acpica umgehen können.

## 29.4. Abrufen und Einreichen von Informationen zur Fehlersuche

Der ACPI-Treiber besitzt flexible Möglichkeiten zur Fehlersuche. Sie können sowohl die zu untersuchenden Subsysteme als auch die zu erzeugenden Ausgaben festlegen. Die zu untersuchenden Subsysteme werden als "layer" angegeben und in Komponenten (`ACPI_ALL_COMPONENTS`) und ACPI-Hardware (`ACPI_ALL_DRIVERS`) aufgeteilt. Welche Meldungen ausgegeben werden, wird über "level" gesteuert. Die Level reichen von von `ACPI_LV_ERROR` (es werden nur Fehler ausgegeben) bis zu `ACPI_LV_VERBOSE` (alles wird ausgegeben). Das Level ist eine Bitmaske, sodass verschiedene Stufen auf einmal (durch Leerzeichen getrennt) angegeben werden können. Die erzeugte Ausgabemenge passt vielleicht nicht in den Konsolenpuffer. In diesem Fall sollte die Ausgabe mithilfe einer seriellen Konsole gesichert werden. Die möglichen Werte für "layers" und "level" werden in acpi(4) beschrieben.

Die Ausgaben zur Fehlersuche sind in der Voreinstellung nicht aktiviert. Wenn ACPI im Kernel enthalten ist, fügen Sie `options ACPI_DEBUG` zur Kernelkonfigurationsdatei hinzu. Sie können die Ausgaben zur Fehlersuche global aktivieren, indem Sie in der Datei /etc/make.conf die Zeile `ACPI_DEBUG=1` einfügen. Das Modul acpi.ko können Sie wie folgt neu übersetzen:

```
# cd /sys/modules/acpi/acpi && make clean && make ACPI_DEBUG=1
```

Kopieren Sie anschließend acpi.ko ins Verzeichnis /boot/kernel. In /boot/loader.conf stellen Sie "level" und "layer" ein. Das folgende Beispiel aktiviert die Ausgabe von Fehlern für alle ACPI-Komponenten und alle Hardwaretreiber:

```
debug.acpi.layer="ACPI_ALL_COMPONENTS ACPI_ALL_DRIVERS"
debug.acpi.level="ACPI_LV_ERROR"
```

Wenn ein Problem durch ein bestimmtes Ereignis, beispielsweise den Start nach einem Ruhezustand, hervorgerufen wird, können Sie die Einstellungen für "level" und "layer" auch mit dem Kommando `sysctl` vornehmen. In diesem Fall müssen Sie /boot/loader.conf nicht editieren. Auf der Kommandozeile geben Sie über `sysctl` dieselben Variablennamen wie in /boot/loader.conf an.

Sobald Sie die Fehlerinformationen gesammelt haben, schicken Sie diese an FreeBSD ACPI, sodass die Betreuer des FreeBSD-ACPI-Subsystems diese Informationen zur Analyse und für die Entwicklung einer Lösung verwenden können.



Bevor Sie einen Fehlerbericht an diese Mailingliste einreichen, stellen Sie bitte sicher, dass das BIOS und die Firmware des Controllers aktuell sind.

Wenn Sie einen Fehlerbericht einsenden, fügen Sie bitte die folgenden Informationen ein:

- Beschreiben Sie den Fehler und alle Umstände, unter denen der Fehler auftritt. Geben Sie ebenfalls den Typ und das Modell Ihres Systems an. Wenn Sie einen neuen Fehler entdeckt haben, versuchen Sie möglichst genau zu beschreiben, wann der Fehler das erste Mal aufgetreten ist.
- Die Ausgabe von `dmesg` nach der Eingabe von `boot -v`. Geben Sie auch alle Fehlermeldungen an, die erscheinen, wenn Sie den Fehler provozieren.
- Die Ausgabe von `dmesg` nach der Eingabe von `boot -v` und mit deaktiviertem ACPI, wenn das Problem ohne ACPI nicht auftritt.
- Die Ausgabe von `sysctl hw.acpi`. Dieses Kommando zeigt die vom System unterstützten ACPI-Funktionen an.
- Die URL, unter der die ASL liegt. Schicken Sie bitte *nicht* die ASL an die Mailingliste, da die ASL sehr groß sein kann. Eine Kopie der ASL erstellen Sie mit dem nachstehenden Befehl:

  ```
  # acpidump -td > name-system.asl
  ```

  Setzen Sie für *name* den Namen des Kontos und für *system* den Hersteller und das Modell des Systems ein. Zum Beispiel: njl-FooCo6000.asl.

Obwohl die meisten Entwickler die Mailingliste FreeBSD-CURRENT lesen, sollten Sie Fehlerberichte an die Liste FreeBSD ACPI schicken. Seien Sie bitte geduldig; wir haben alle Arbeit außerhalb des Projekts. Wenn der Fehler nicht offensichtlich ist, bitten wir Sie vielleicht, einen offiziellen Fehlerbericht (PR) einzusenden. Geben Sie im Fehlerbericht bitte dieselben Informationen wie oben an. Mithilfe der PRs verfolgen und lösen wir Probleme. Senden Sie bitte keinen PR ein, ohne vorher den Fehlerbericht an die Liste FreeBSD ACPI zu senden. Es kann sein, dass der Fehler schon von jemand anderem gemeldet wurde.

## 29.5. Referenzen

Weitere Informationen über ACPI finden Sie hier:

- Die FreeBSD ACPI Mailingliste (https://lists.freebsd.org/pipermail/freebsd-acpi/)

- Die ACPI Spezifikation
- acpi(4), acpi_thermal(4), acpidump(8), iasl(8) und acpidb(8)

# Kapitel 30. FreeBSDs Bootvorgang

## 30.1. Übersicht

Das Starten des Computers und das Laden des Betriebssystems wird im Allgemeinen als "Bootstrap-Vorgang", oder als "Booten" bezeichnet. FreeBSDs Bootvorgang ermöglicht große Flexibilität, was das Anpassen dessen anbelangt, was passiert, wenn das System gestartet wird. Es kann zwischen verschiedenen Betriebssystemen, die auf demselben Computer installiert sind oder verschiedenen Versionen desselben Betriebssystems oder installierten Kernels gewählt werden.

Dieses Kapitel zeigt die zur Verfügung stehenden Konfigurationsmöglichkeiten und wie man den Bootvorgang anpasst. Dies schließt alles ein, bis der Kernel gestartet worden ist, der dann alle Geräte gefunden hat und init(8) gestartet hat. Dies passiert, wenn die Farbe des Textes während des Bootvorgangs von weiß zu grau wechselt.

Dieses Kapitel informiert über folgende Punkte:

- Die Komponenten des FreeBSD-Bootvorgangs und deren Interaktion.
- Die Optionen, mit denen der FreeBSD-Bootvorgang gesteuert werden kann.
- Wie ein angepasster Willkommensbildschirm beim Booten konfiguriert wird.
- Wie Geräte mit device.hints(5) konfiguriert werden.
- Wie das System in den Single-User-Modus und in den Mehrbenutzer-Modus gestartet wird und wie ein FreeBSD-System ordnungsgemäß heruntergefahren wird.



Dieses Kapitel erklärt den Bootvorgang von FreeBSD auf Intel x86- und amd64-Plattformen.

## 30.2. FreeBSDs Bootvorgang

Wenn der Computer eingeschaltet wird und das Betriebssystem gestartet werden soll, entsteht ein interessantes Dilemma, denn der Computer weiß per Definition nicht, wie er irgendetwas tut, bis das Betriebssystem gestartet wurde. Das schließt das Starten von Programmen, die sich auf der Festplatte befinden, ein. Wenn der Computer kein Programm von der Festplatte starten kann, sich das Betriebssystem aber genau dort befindet, wie wird es dann gestartet?

Dieses Problem ähnelt einer Geschichte des Barons von Münchhausen. Dort war eine Person in einen Sumpf gefallen und hat sich selbst an den Riemen seiner Stiefel (engl. bootstrap) herausgezogen. In den jungen Jahren des Computerzeitalters wurde mit dem Begriff Bootstrap dann die Technik das Betriebssystem zu laden bezeichnet. Seither wurde es mit "booten" abgekürzt.

Auf x86-Plattformen ist das Basic Input/Output System (BIOS) dafür verantwortlich, das Betriebssystem zu laden. Das BIOS liest den Master Boot Record (MBR) aus, der sich an einer bestimmten Stelle auf der Festplatte befinden muss. Das BIOS kann den MBR selbstständig laden und ausführen und geht davon aus, dass dieser die restlichen Dinge, die für das Laden des Betriebssystems notwendig sind, selbst oder mit Hilfe des BIOS erledigen kann.



FreeBSD ermöglicht das Booten sowohl über den alten MBR-Standard, als auch über die neuere GUID-Partitionstabelle (GPT). GPT-Partitionen finden sich häufig auf Systemen mit dem *Unified Extensible Firmware Interface* (UEFI). FreeBSD kann allerdings mit Hilfe von gptboot(8) auch GPT-Partitionen über das alte BIOS booten. An der Unterstützung für ein direktes Booten über UEFI wird derzeit gearbeitet.

Der Code innerhalb des MBRs wird für gewöhnlich als *Boot-Manager* bezeichnet, insbesondere, wenn eine Interaktion mit dem Anwender stattfindet. Der Boot-Manager verwaltet zusätzlichen Code im ersten *Track* der Platte oder des Dateisystems. Zu den bekanntesten Boot-Managern gehören boot0, der auch als Boot Easy bekannte Standard-Boot-Manager von FreeBSD, sowie Grub, welches in vielen Linux®-Distributionen verwendet wird.

Falls nur ein Betriebssystem installiert ist, sucht der MBR nach dem ersten bootbaren Slice (das dabei als *active* gekennzeichnet ist) auf dem Laufwerk und führt den dort vorhandenen Code aus, um das restliche Betriebssystem zu laden. Wenn mehrere Betriebssysteme installiert sind, kann ein anderer Boot-Manager installiert werden, der eine Liste der verfügbaren Betriebssysteme anzeigt, so dass der Benutzer wählen kann, welches Betriebssystem er booten möchte.

Das restliche FreeBSD-Bootstrap-System ist in drei Phasen unterteilt. Die erste Phase besitzt gerade genug Funktionalität um den Computer in einen bestimmten Status zu verhelfen und die zweite Phase zu starten. Die zweite Phase führt ein wenig mehr Operationen durch und startet schließlich die dritte Phase, die das Laden des Betriebssystems abschließt. Der ganze Prozess wird in drei Phasen durchgeführt, weil der MBR die Größe der Programme, die in Phase eins und zwei ausgeführt werden, limitiert. Das Verketten der durchzuführenden Aufgaben ermöglicht es FreeBSD, ein sehr flexibles Ladeprogramm zu besitzen.

Als nächstes wird der Kernel gestartet, der zunächst nach Geräten sucht und sie für den Gebrauch initialisiert. Nach dem Booten des Kernels übergibt dieser die Kontrolle an den Benutzer Prozess init(8), der erst sicherstellt, dass alle Laufwerke benutzbar sind und die Ressourcen Konfiguration auf Benutzer Ebene startet. Diese wiederum mountet Dateisysteme, macht die Netzwerkkarten für die Kommunikation mit dem Netzwerk bereit und startet alle Prozesse, die konfiguriert wurden, um beim Hochfahren gestartet zu werden.

Dieser Abschnitt beschreibt die einzelnen Phasen und wie sie mit dem FreeBSD-Bootvorgang interagieren.

### 30.2.1. Der Boot-Manager

Der Boot-Manager Code im MBR wird manchmal auch als *stage zero* des Boot-Prozesses bezeichnet. In der Voreinstellung verwendet FreeBSD den boot0 Boot-Manager.

Der vom FreeBSD-Installationsprogramm in der Voreinstelung installierte MBR basiert auf /boot/boot0. Die Größe und Leistungsfähigkeit von boot0 ist auf 446 Bytes beschränkt, weil der restliche Platz für die Partitionstabelle sowie den `0x55AA`-Identifier am Ende des MBRs benötigt wird. Wenn boot0 und mehrere Betriebssysteme installiert sind, wird beim Starten des Computers eine Anzeige ähnlich der folgenden zu sehen sein:

*Beispiel 27. boot0-Screenshot*

```
F1 Win
F2 FreeBSD

Default: F2
```

Diverse Betriebssysteme überschreiben den existierenden MBR, wenn sie nach FreeBSD installiert werden. Falls dies passiert, kann mit folgendem Kommando der momentane MBR durch den FreeBSD-MBR ersetzt werden:

```
# fdisk -B -b /boot/boot0 Gerät
```

Bei *Gerät* handelt es sich um das Gerät, von dem gebootet wird, also beispielsweise ad0 für die erste IDE-Festplatte, ad2 für die erste IDE-Festplatte am zweiten IDE-Controller, da0 für die erste SCSI-Festplatte. Um eine angepasste Konfiguration des MBR zu erstellen, lesen Sie boot0cfg(8).

### 30.2.2. Phase Eins und Phase Zwei

Im Prinzip sind die erste und die zweite Phase Teile desselben Programms, im selben Bereich auf der Festplatte. Aufgrund von Speicherplatz-Beschränkungen wurden sie in zwei Teile aufgeteilt, welche jedoch immer zusammen installiert werden. Beide werden entweder vom FreeBSD-Installationsprogramm oder `bsdlabel` aus der kombinierten /boot/boot kopiert.

Beide Phasen befinden sich außerhalb des Dateisystems im Bootsektor des Boot-Slices, wo boot0 oder ein anderer Boot-Manager ein Programm erwarten, das den weiteren Bootvorgang durchführen kann.

Die erste Phase, boot1, ist ein sehr einfaches Programm, da es nur 512 Bytes groß sein darf. Es besitzt gerade genug Funktionalität, um FreeBSDs *bsdlabel*, das Informationen über den Slice enthält, auszulesen, und um boot2 zu finden und auszuführen.

Die zweite Phase, boot2, ist schon ein wenig umfangreicher und besitzt genügend Funktionalität, um Dateien in FreeBSDs Dateisystem zu finden. Es kann eine einfache Schnittstelle bereitstellen, die es ermöglicht, den zu ladenden Kernel oder Loader auszuwählen. Es lädt den loader, der einen weitaus größeren Funktionsumfang bietet und eine Konfigurationsdatei zur Verfügung stellt. Wenn der Boot-Prozess während der zweiten Phase unterbrochen wird, erscheint der folgende Bildschrim:

*Beispiel 28. boot2-Screenshot*

```
>> FreeBSD/i386 BOOT
Default: 0:ad(0,a)/boot/loader
boot:
```

Um das installierte boot1 und boot2 zu ersetzen, benutzen Sie `bsdlabel`, wobei *diskslice* das Laufwerk und die Slice darstellt, von dem gebootet wird, beispielsweise ad0s1 für die erste Slice auf der ersten IDE-Festplatte:

```
# bsdlabel -B diskslice
```



Wenn man nur den Festplatten-Namen benutzt, beispielsweise ad0, wird `bsdlabel` eine "dangerously dedicated disk" erstellen, ohne Slices. Das ist ein Zustand, den man meistens nicht hervorrufen möchte. Aus diesem Grund sollte man das *diskslice* noch einmal prüfen, bevor `Return` gedrückt wird.

### 30.2.3. Phase Drei

Der loader ist der letzte von drei Schritten im Bootstrap-Prozess. Er kann im Dateisystem normalerweise als /boot/loader gefunden werden.

Der loader soll eine interaktive Konfigurations-Schnittstelle mit eingebauten Befehlssatz sein, ergänzt durch einen umfangreichen Interpreter mit einem komplexeren Befehlssatz.

Der loader sucht während seiner Initialisierung nach Konsolen und Laufwerken, findet heraus, von welchem Laufwerk er gerade bootet, und setzt dementsprechend bestimmte Variablen. Dann wird ein Interpreter gestartet, der Befehle interaktiv oder von einem Skript empfangen kann.

Danach liest der loader /boot/loader.rc, welche ihn standardmäßig anweist /boot/defaults/loader.conf zu lesen, wo sinnvolle Standardeinstellungen für diverse Variablen festgelegt werden und wiederum /boot/loader.conf für lokale Änderungen an diesen Variablen ausgelesen wird. Anschließend arbeitet dann loader.rc entsprechend dieser Variablen und lädt die ausgewählten Module und den gewünschten Kernel.

In der Voreinstellung wartet der loader 10 Sekunden lang auf eine Tastatureingabe und bootet den Kernel, falls keine Taste betätigt wurde. Falls doch eine Taste betätigt wurde wird dem Benutzer eine Eingabeaufforderung angezeigt. Sie nimmt einen Befehlssatz entgegen, der es dem Benutzer erlaubt, Änderungen an Variablen vorzunehmen, Module zu laden, alle Module zu entladen oder schließlich zu booten oder neu zu booten.

*Tabelle 9. Die eingebauten Befehle des Loaders*

| Variable | Beschreibung |
|---|---|
| autoboot Sekunden | Es wird mit dem Booten des Kernels fortgefahren, falls keine Taste in der gegebenen Zeitspanne betätigt wurde. In der gegebenen Zeitspanne, Vorgabe sind 10 Sekunden, wird ein Countdown angezeigt. |
| boot [-Optionen] [Kernelname] | Bewirkt das sofortige Booten des Kernels mit allen gegebenen Optionen, oder dem angegebenen Kernelnamen. Das übergeben eines Kernelnamens ist nur nach einem `unload` anwendbar, andernfalls wird der zuvor verwendete Kernel benutzt. Wenn nicht der vollständige Pfad für *Kernelname* angegeben wird, dann sucht der Loader den Kernel unter /boot/kernel und /boot/modules. |

| Variable | Beschreibung |
|---|---|
| boot-conf | Bewirkt die automatische Konfiguration der Module, abhängig von den entsprechenden Variablen (üblicherweise `kernel`). Dies nur dann sinnvoll, wenn zuvor `unload` benutzt wurde. |
| help [`Thema`] | Zeigt die Hilfe an, die zuvor aus der Datei /boot/loader.help gelesen wird. Falls `index` als Thema angegeben wird, wird die Liste der zur Verfügung stehenden Hilfe-Themen angezeigt. |
| include `Dateiname` ... | Das Einlesen und Interpretieren der angegebenen Datei geschieht Zeile für Zeile und wird im Falle eines Fehlers umgehend unterbrochen. |
| load [`-t Typ`] *Dateiname* | Lädt den Kernel, das Kernel-Modul, oder die Datei des angegebenen Typs. Argumente, die auf *Dateiname* folgen, werden der Datei übergeben. Wenn nicht der vollständige Pfad für *Dateiname* angegeben wird, dann sucht der Loader die Datei unter /boot/kernel und /boot/modules. |
| ls [`-l`] [`Pfad`] | Listet die Dateien im angegebenen Pfad auf, oder das Root-Verzeichnis, falls kein Pfad angegeben wurde. Die Option `-l` bewirkt, dass die Dateigrößen ebenfalls angezeigt werden. |
| lsdev [`-v`] | Listet alle Geräte auf, für die Module geladen werden können. Die Option `-v` bewirkt eine ausführliche Ausgabe. |
| lsmod [`-v`] | Listet alle geladenen Module auf. Die Option `-v` bewirkt eine ausführliche Ausgabe. |
| more `Dateiname` | Zeigt den Dateinhalt der angegebenen Datei an, wobei eine Pause alle `LINES` Zeilen gemacht wird. |
| reboot | Bewirkt einen umgehenden Neustart des Systems. |
| set `Variable`, set `Variable=Wert` | Setzt die angegebenen Umgebungsvariablen. |
| unload | Entlädt sämtliche geladenen Module. |

Hier ein paar praktische Beispiele für die Bedienung des Loaders. Um den gewöhnlichen Kernel im Single-User Modus zu starten:

```
boot -s
```

Um alle gewöhnlichen Kernelmodule zu entladen und dann den alten, oder einen anderen Kernel zu laden:

```
unload
/pfad/zur/kerneldatei
```

Verwenden Sie /boot/GENERIC/kernel, um auf den allgemeinen Kernel zu verweisen, der bei jeder Installation dabei ist. /boot/kernel.old/kernel hingegen bezeichnet den Kernel, der vor dem System-Upgrade installiert war.

Der folgende Befehl lädt die gewöhnlichen Module mit einem anderen Kernel:

```
unload
set kernel="meinkernel"
boot-conf
```

Um ein automatisiertes Kernelkonfigurations-Skript zu laden, geben Sie ein:

```
load -t userconfig_script /boot/kernel.conf
```

### 30.2.4. Die letzte Phase

Sobald der Kernel einmal geladen ist, entweder durch den loader oder durch boot2, welches den Loader umgeht, dann überprüft er vorhandene Boot-Flags und passt sein Verhalten nach Bedarf an. In Interaktion mit dem Kernel während des Bootens sind die gebräuchlichsten Boot-Flags aufgelistet. Informationen zu den anderen Boot-Flags finden Sie in boot(8).

*Tabelle 10. Interaktion mit dem Kernel während des Bootens*

| Option | Beschreibung |
|---|---|
| `-a` | Bewirkt, dass während der Kernel-Initialisierung gefragt wird, welches Gerät als Root-Dateisystem eingehängt werden soll. |
| `-C` | Das Root-Dateisystem wird von CD-ROM gebootet. |
| `-s` | Bootet in den Single-User Modus |
| `-v` | Zeigt mehr Informationen während des Starten des Kernels an. |

Nachdem der Kernel den Bootprozess abgeschlossen hat, übergibt er die Kontrolle an den Benutzer-Prozess init(8). Dieses Programm befindet sich in /sbin/init, oder dem Pfad, der durch die Variable `init_path` im `loader` spezifiziert wird.

Der automatische Reboot-Vorgang stellt sicher, dass alle Dateisysteme des Systems konsistent sind. Falls dies nicht der Fall ist und die Inkonsistenz des UFS-Dateisystems nicht durch `fsck` behebbar ist, schaltet `init` das System in den Single-User-Modus, damit der Systemadministrator sich des Problems annehmen kann. Andernfalls startet das System in den Mehrbenutzermodus.

#### Der Single-User Modus

Der Wechsel in den Single-User Modus kann beim Booten durch die Option `-s`, oder das Setzen der Variable `boot_single` in loader erreicht werden. Zudem kann er auch im Mehrbenutzermodus über den Befehl `shutdown now` erreicht werden. Der Single-User Modus beginnt mit dieser Meldung:

```
Enter full path of shell or RETURN for /bin/sh:
```

Wenn Sie die Eingabetaste drücken, wird das System die Bourne Shell starten. Falls Sie eine andere Shell starten möchten, geben Sie den vollständigen Pfad zur Shell ein.

Der Single-User Modus wird normalerweise zur Reparatur verwendet, beispielsweise wenn das System aufgrund eines inkonsistenten Dateisystems oder einem Fehler in einer Konfigurationsdatei nicht bootet. Der Modus wird auch verwendet, um das Passwort von `root` zurückzusetzen, falls dieses nicht mehr bekannt ist. Dies alles ist möglich, da der Single-User Modus vollen Zugriff auf das lokale System und die Konfigurationsdateien gewährt. Einen Zugang zum Netzwerk bietet dieser Modus allerdings nicht.

Obwohl der Single-User Modus für Reparaturen am System sehr nützlich ist, stellt es ein Sicherheitsrisiko dar, wenn sich das System an einem physisch unsicheren Standort befindet. In der Voreinstellung hat jeder Benutzer, der physischen Zugriff auf ein System erlangen kann, volle Kontrolle über das System, nachdem in den Single-User Modus gebootet wurde.

Falls die System-Konsole (`console`) in /etc/ttys auf `insecure` (dt.: unsicher) gesetzt ist, fordert das System zur Eingabe des `root` Passworts auf, bevor es den Single-User Modus aktiviert. Dadurch gewinnen Sie zwar ein gewisses Maß an Sicherheit, aber Sie können dann nicht mehr das Passwort von `root` zurücksetzen, falls es nicht bekannt ist.

*Beispiel 29. Auf insecure gesetzte Konsole in /etc/ttys*

```
# name  getty                           type    status          comments
#
# If console is marked "insecure", then init will ask for the root password
# when going to single-user mode.
console none                            unknown off insecure
```

Eine Konsole sollte auf `insecure` gesetzt sein, wenn die physikalische Sicherheit der Konsole nicht gegeben ist und sichergestellt werden soll, dass nur Personen, die das Passwort von `root` kennen, den Single-User Modus benutzen können.

**Mehrbenutzermodus**

Stellt init fest, dass das Dateisystem in Ordnung ist, oder der Benutzer den Single-User-Modus mit `exit` beendet, schaltet das System in den Mehrbenutzermodus, in dem dann die Ressourcen Konfiguration des Systems gestartet wird.

Das Ressourcen Konfigurationssystem (engl. resource configuration, rc) liest seine Standardkonfiguration von /etc/defaults/rc.conf und System-spezifische Details von /etc/rc.conf. Dann mountet es die Dateisysteme gemäß /etc/fstab, startet die Netzwerkdienste, diverse System Daemons und führt schließlich die Start-Skripten der lokal installierten Anwendungen aus.

Lesen Sie rc(8) und ebenso die Skripte in /etc/rc.d, um mehr über das Ressourcen Konfigurationssystem zu erfahren.

## 30.3. Willkommensbildschirme während des Bootvorgangs konfigurieren

Wenn ein FreeBSD-System startet, gibt es normalerweise eine Reihe von Meldungen auf der

Konsole aus. Ein Willkommensbildschirm erzeugt einen alternativen Boot-Bildschirm, der alle Bootmeldungen und Meldungen über startende Dienste versteckt. Ein paar Meldungen des Bootloaders, einschließlich das Menü mit den Bootoptionen und dem Warte-Countdown werden dennoch zur Bootzeit angezeigt, auch wenn der Willkommensbildschirm aktiviert ist. Der Willkommensbildschirm kann während des Bootvorgangs mit einem beliebigen Tastendruck ausgeschaltet werden.

Es existieren zwei grundlegende Umgebungen in FreeBSD. Die erste ist die altbekannte, auf virtuellen Konsolen basierte Kommandozeile. Nachdem das System den Bootvorgang abgeschlossen hat, wird ein Anmeldebildschirm auf der Konsole anzeigt. Die zweite Umgebung ist eine konfigurierte, graphische Umgebung. Das X-Window-System enthält weitere Informationen zur Installation und Konfiguration eines graphischen Display-Managers und Login-Managers.

Der Willkommensbildschirm ist standardmäßig so eingestellt, dass er als Bildschirmschoner verwendet wird. Nach einer bestimmten Zeit der Untätigkeit wird der Willkommensbildschirm angezeigt und wechselt durch verschiedene Stufen der Intensität von hell zu einem sehr dunklen Bild und wieder zurück. Das Verhalten des Willkommensbildschirms kann durch hinzufügen einer `saver=`-Zeile in /etc/rc.conf geändert werden. Es gibt mehrere eingebaute Bildschirmschoner, die in splash(4) beschrieben werden. Die `saver=`-Option bezieht sich nur auf virtuelle Konsolen und hat keinen Effekt bei grafischen Display-Managern.

Durch die Installation des Ports oder Pakets sysutils/bsd-splash-changer werden Willkommensbildschirme von einer zufällig ausgewählten Sammlung von Bildern bei jedem Neustart angezeigt. Die Willkommensbildschirm-Funktionalität unterstützt 256-Farben in den Formaten Bitmap (.bmp), ZSoft PCX (.pcx) oder TheDraw (.bin). Die Willkommensbildschirm-Datei .bmp, .pcx oder .bin muss in der Root-Partition, beispielsweise unterhalb von /boot abgelegt werden. Willkommensbildschirm-Dateien dürfen eine Auflösung von 320 mal 200 Pixeln oder weniger besitzen, damit Standard-VGA Geräte damit arbeiten können. Für eine Standard-Auflösung von 256-Farben, 320 mal 200 Pixel oder weniger, fügen Sie folgende Zeilen in /boot/loader.conf ein und ersetzen Sie *splash.bmp* mit dem Namen der Bitmap-Datei:

```
splash_bmp_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bmp"
```

Wenn Sie anstelle der Bitmap-Datei eine PCX-Datei verwenden:

```
splash_pcx_load="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.pcx"
```

Für ASCII-Art im TheDraw-Format schreiben Sie:

```
splash_txt="YES"
bitmap_load="YES"
bitmap_name="/boot/splash.bin"
```

Weitere interessante Optionen für loader.conf sind:

`beastie_disable="YES"`

: Diese Option verhindert die Anzeige des Menüs mit den Bootoptionen, aber der Countdown ist immer noch aktiv. Selbst wenn das Bootmenü deaktiviert ist, kann während des Countdowns eine der korrespondierenden Optionen ausgewählt werden.

`loader_logo="beastie"`

: Dies ersetzt die Standardanzeige des Wortes "FreeBSD". Stattdessen wird auf der rechten Seite des Bootmenüs das bunte Beastie-Logo angezeigt.

Weitere Informationen finden Sie in splash(4), loader.conf(5) und vga(4).

## 30.4. Konfiguration von Geräten

Der Boot-Loader liest während des Systemstarts die Datei device.hints(5), die Variablen, auch "device hints" genannt, zur Konfiguration von Geräten enthält.

Die Variablen können auch mit Kommandos in Phase 3 des Boot-Loaders, wie in Phase Drei beschrieben, bearbeitet werden. Neue Variablen werden mit `set` gesetzt, `unset` löscht schon definierte Variablen und `show` zeigt Variablen an. Variablen aus /boot/device.hints können zu diesem Zeitpunkt überschrieben werden. Die hier durchgeführten Änderungen sind nicht permanent und beim nächsten Systemstart nicht mehr gültig.

Nach dem Systemstart können alle Variablen mit kenv(1) angezeigt werden.

Pro Zeile enthält /boot/device.hints eine Variable. Kommentare werden durch `#` eingeleitet. Die verwendete Syntax lautet:

```
hint.driver.unit.keyword="value"
```

Der Boot-Loader verwendet die nachstehende Syntax:

```
set hint.driver.unit.keyword=value
```

Der Gerätetreiber wird mit `driver`, die Nummer des Geräts mit `unit` angegeben. `keyword` ist eine Option aus der folgenden Liste:

- `at`: Gibt den Bus, auf dem sich das Gerät befindet, an.
- `port`: Die Startadresse des I/O-Bereichs.
- `irq`: Gibt die zu verwendende Unterbrechungsanforderung (IRQ) an.
- `drq`: Die Nummer des DMA Kanals.
- `maddr`: Die physikalische Speicheradresse des Geräts.
- `flags`: Setzt verschiedene gerätespezifische Optionen.
- `disabled`: Deaktiviert das Gerät, wenn der Wert auf `1` gesetzt wird.

Ein Gerätetreiber kann mehr Optionen, als die hier beschriebenen, besitzen oder benötigen. Es wird empfohlen, die Optionen in der Manualpage des Treibers nachzuschlagen. Weitere Informationen finden Sie in device.hints(5), kenv(1), loader.conf(5) und loader(8).

## 30.5. Der Shutdown-Vorgang

Im Falle eines regulären Herunterfahrens durch shutdown(8) führt init(8) /etc/rc.shutdown aus, sendet dann sämtlichen Prozessen ein `TERM` Signal und schließlich ein `KILL` Signal an alle Prozesse, die sich nicht rechtzeitig beendet haben.

FreeBSD-Systeme, die Energieverwaltungsfunktionen unterstützen, können mit `shutdown -p now` ausgeschaltet werden. Zum Neustart des Systems wird `shutdown -r now` benutzt. Das Kommando shutdown(8) kann nur von `root` oder Mitgliedern der Gruppe `operator` benutzt werden. Man kann auch halt(8) und reboot(8) verwenden. Weitere Informationen finden Sie in den Hilfeseiten der drei Kommandos.

Das Ändern der Gruppenmitgliedschaft wird in “Benutzer und grundlegende Account-Verwaltung” beschrieben.



Die Energieverwaltungsfunktionen erfordern, dass die Unterstützung für acpi(4) als Modul geladen, oder statisch in einen angepassten Kernel kompiliert wird.

# Kapitel 31. Sicherheit

## 31.1. Übersicht

Sicherheit, ob nun physisch oder virtuell, ist ein so breit gefächertes Thema, dass sich eine ganze Industrie darum gebildet hat. Es wurden bereits hunderte Verfahren zur Sicherung von Systemen und Netzwerken verfasst, und als Benutzer von FreeBSD ist es unumgänglich zu verstehen, wie Sie sich gegen Angreifer und Eindringlinge schützen können.

In diesem Kapitel werden einige Grundlagen und Techniken diskutiert. Ein FreeBSD-System implementiert Sicherheit in mehreren Schichten, und viele weitere Programme von Drittanbietern können zur Verbesserung der Sicherheit beitragen.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie:

- Grundlegende auf FreeBSD bezogene Sicherheitsaspekte kennen.
- Die verschiedenen Verschlüsselungsmechanismen von FreeBSD kennen.
- Wissen, wie Sie ein Einmalpasswörter zur Authentifizierung verwenden.
- TCP Wrapper für inetd(8) einrichten können.
- Wissen, wie Sie Kerberos unter FreeBSD einrichten.
- Wissen, wie Sie IPsec konfigurieren und ein VPN einrichten.
- Wissen, wie Sie OpenSSH unter FreeBSD konfigurieren und benutzen.
- Wissen, wie Sie ACLs für Dateisysteme benutzen.
- pkg anwenden können, um Softwarepakete aus der Ports-Sammlung auf bekannte Sicherheitslücken hin zu überprüfen.
- Mit FreeBSD-Sicherheitshinweisen umgehen können.
- Eine Vorstellung davon haben, was Prozessüberwachung (Process Accounting) ist und wie Sie diese Funktion unter FreeBSD aktivieren können.
- Wissen, wie Sie Login-Klassen oder die Ressourcen-Datenbank benutzen, um die Ressourcen für Benutzer zu steuern.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Grundlegende Konzepte von FreeBSD und dem Internet verstehen.

Dieses Buch behandelt weitere Sicherheitsthemen. Beispielsweise werden verbindliche Zugriffskontrollen im Verbindliche Zugriffskontrolle und Firewalls im Firewalls besprochen.

## 31.2. Einführung

Sicherheit ist die Verantwortung eines jeden Einzelnen. Ein schwacher Einstiegspunkt in einem System kann einem Eindringling Zugriff auf wichtige Informationen verschaffen, was sich verheerend auf das gesamte Netzwerk auswirken kann. Eines der Grundprinzipien der

Informationssicherheit sind die Vertraulichkeit, Integrität und Verfügbarkeit von Informationssystemen.

Diese Grundprinzipien sind ein fundamentales Konzept der Computer-Sicherheit, da Kunden und Benutzer erwarten, dass ihre Daten geschützt sind. Zum Beispiel erwartet ein Kunde, dass seine Kreditkarteninformationen sicher gespeichert werden (Vertraulichkeit), dass seine Aufträge nicht hinter den Kulissen geändert werden (Integrität) und dass er zu jeder Zeit Zugang zu seinen Informationen hat (Verfügbarkeit).

Um diese Grundprinzipien zu implementieren, wenden Sicherheitsexperten das sogenannte Defense-in-Depth-Konzept an. Die Idee dahinter ist, mehrere Sicherheitsschichten zu addieren, so dass nicht die gesamte Systemsicherheit gefährdet ist, wenn eine einzelne Sicherheitsschicht kompromittiert wird. Beispielsweise ist es nicht ausreichend, ein Netzwerk oder ein System nur mit einer Firewall zu sichern. Der Systemadministrator muss auch Benutzerkonten überwachen, die Integrität von Binärdateien prüfen und sicherstellen, dass keine bösartigen Programme installiert sind. Um eine effektive Sicherheitsstrategie zu implementieren, muss man Bedrohungen verstehen und wissen, wie man sich dagegen verteidigen kann.

Was ist eine Bedrohung, wenn es um Computer-Sicherheit geht? Bedrohungen beschränken sich nicht nur auf entfernte Angreifer, die sich unerlaubten Zugriff auf ein System verschaffen wollen. Zu den Bedrohungen zählen auch Mitarbeiter, bösartige Software, nicht autorisierte Netzwerkgeräte, Naturkatastrophen, Sicherheitslücken und sogar konkurrierende Unternehmen.

Der Zugriff auf Netzwerke und Systeme erfolgt ohne Erlaubnis, manchmal durch Zufall, oder von entfernten Angreifern, und in einigen Fällen durch Industriespionage oder ehemalige Mitarbeiter. Als Anwender müssen Sie vorbereitet sein und auch zugeben, wenn ein Fehler zu einer Sicherheitsverletzung geführt hat. Melden Sie Probleme umgehend dem verantwortlichen Sicherheitspersonal. Als Administrator ist es wichtig, Bedrohungen zu kennen und darauf vorbereitet zu sein, mögliche Schäden zu mildern.

Wenn Sicherheit auf Systeme angewendet wird, empfiehlt es sich mit der Sicherung der Benutzerkonten zu beginnen und dann die Netzwerkschicht zu sichern. Dabei ist zu beachten, dass die Sicherheitsrichtlinien des Systems und des Unternehmens eingehalten werden. Viele Unternehmen haben bereits eine Sicherheitsrichtlinie, welche die Konfiguration von technischen Geräten abdeckt. Die Richtlinie sollte die Konfiguration von Arbeitsplatzrechnern, Desktops, mobilen Geräten, Mobiltelefonen, Produktions- und Entwicklungsservern umfassen. In einigen Fällen ist bereits eine Standardvorgehensweise vorhanden. Fragen Sie im Zweifelsfall das Sicherheitspersonal.

Der übrige Teil dieser Einführung beschreibt, wie einige grundlegende Sicherheitskonfigurationen auf einem FreeBSD-System durchgeführt werden. Der Rest des Kapitels zeigt einige spezifische Werkzeuge, die verwendet werden können, um eine Sicherheitsrichtlinie auf einem FreeBSD-System zu implementieren.

### 31.2.1. Anmeldungen am System verhindern

Ein guter Ausgangspunkt für die Absicherung des Systems ist die Prüfung der Benutzerkonten. Stellen Sie sicher, dass `root` ein starkes Passwort besitzt und dass dieses Passwort nicht weitergegeben wird. Deaktivieren Sie alle Konten, die keinen Zugang zum System benötigen.

Es existieren zwei Methoden, um die Anmeldung über ein Benutzerkonto zu verweigern. Die erste Methode ist, das Konto zu sperren. Dieses Beispiel sperrt das Benutzerkonto `toor`:

```
# pw lock toor
```

Bei der zweiten Methode wird der Anmeldevorgang verhindert, indem die Shell auf /usr/sbin/nologin gesetzt wird. Nur der Superuser kann die Shell für andere Benutzer ändern:

```
# chsh -s /usr/sbin/nologin toor
```

Die Shell /usr/sbin/nologin verhindert, dass dem Benutzer bei der Anmeldung am System eine Shell zugeordnet wird.

### 31.2.2. Gemeinsame Nutzung von Benutzerkonten

In manchen Fällen wird die Systemadministration auf mehrere Benutzer aufgeteilt. FreeBSD bietet zwei Methoden, um solche Situationen zu handhaben. Bei der ersten und nicht empfohlenen Methode wird ein gemeinsames root Passwort der Mitglieder der Gruppe `wheel` verwendet. Hier gibt der Benutzer `su` und das Passwort für `wheel` ein, wenn er die Rechte des Superusers benötigt. Der Benutzer sollte dann nach der Beendigung der administrativen Aufgaben `exit` eingeben. Um einen Benutzer zu dieser Gruppe hinzuzufügen, bearbeiten Sie /etc/group und fügen Sie den Benutzer an das Ende des Eintrags `wheel` hinzu. Die Benutzer müssen durch Komma und ohne Leerzeichen getrennt werden.

Die zweite und empfohlene Methode ein Benutzerkonto zu teilen wird über den Port oder das Paket security/sudo realisiert. Dieses Programm bietet zusätzliche Prüfungen, bessere Benutzerkontrolle und es kann auch konfiguriert werden, einzelnen Benutzern Zugriff auf bestimme, privilegierte Befehle zu gestatten.

Benutzen Sie nach der Installation `visudo`, um /usr/local/etc/sudoers zu bearbeiten. Dieses Beispiel erstellt eine neue Gruppe `webadmin` und fügt das Benutzerkonto `trhodes` dieser Gruppe hinzu. Anschließend wird die Gruppe so konfiguriert, dass es Gruppenmitgliedern gestattet wird apache24 neu zu starten:

```
# pw groupadd webadmin -M trhodes -g 6000
# visudo
%webadmin ALL=(ALL) /usr/sbin/service apache24 *
```

### 31.2.3. Passwort-Hashes

Passwörter sind ein notwendiges Übel. Wenn sie verwendet werden müssen, sollten sie sehr komplex sein und dazu sollte eine leistungsfähige Hash-Funktion gewählt werden, um die Version des Passworts zu verschlüsseln, die in der Passwortdatenbank gespeichert wird. FreeBSD unterstützt die Hash-Funktionen DES, MD5, SHA256, SHA512, sowie Blowfish Hash-Funktionen in seiner `crypt()`-Bibliothek. Das in der Voreinstellung verwendete SHA512 sollte nicht durch eine weniger sichere Hash-Funktion getauscht werden. Es kann jedoch durch den besseren Blowfish-

Algorithmus ersetzt werden.



Blowfish ist nicht Bestandteil von AES und ist nicht kompatibel mit allen Federal Information Processing Standards (FIPS). Die Verwendung wird in einigen Umgebungen vielleicht nicht gestattet.

Um zu bestimmen, welche Hash-Funktion das Passwort eines Benutzers verschlüsselt, kann der Superuser den Hash für den Benutzer in der Passwortdatenbank von FreeBSD nachsehen. Jeder Hash beginnt mit einem Zeichen, mit dem die verwendete Hash-Funktion identifiziert werden kann. Bei DES gibt es allerdings kein führendes Zeichen. MD5 benutzt das Zeichen `$`. SHA256 und SHA512 verwenden das Zeichen `$6$`. Blowfish benutzt das Zeichen `$2a$`. In diesem Beispiel wird das Passwort von `dru` mit dem Hash-Algorithmus SHA512 verschlüsselt, da der Hash mit `$6$` beginnt. Beachten Sie, dass der verschlüsselte Hash und nicht das Passwort selbst, in der Passwortdatenbank gespeichert wird:

```
# grep dru /etc/master.passwd
dru:$6$pzIjSvCAn.PBYQBA$PXpSeWPx3g5kscj3IMiM7tUEUSPmGexxta.8Lt9TGSi2lNQqYGKszsBPuGME0:
1001:1001::0:0:dru:/usr/home/dru:/bin/csh
```

Der Hash-Mechanismus wird in der Login-Klasse des Benutzers festgelegt. In diesem Beispiel wird die voreingestellte Login-Klasse für den Benutzer verwendet. Der Hash-Algorithmus wird mit dieser Zeile in /etc/login.conf gesetzt:

```
        :passwd_format=sha512:\
```

Um den Algorithmus auf Blowfish zu ändern, passen Sie die Zeile wie folgt an:

```
        :passwd_format=blf:\
```

Führen Sie anschließend `cap_mkdb /etc/login.conf` aus, wie in Login-Klassen konfigurieren beschrieben. Beachten Sie, dass vorhandene Passwort-Hashes durch diese Änderung nicht beeinträchtigt werden. Das bedeutet, dass alle Passwörter neu gehasht werden sollten, indem die Benutzer mit `passwd` ihr Passwort ändern.

Für die Anmeldung auf entfernten Rechnern sollte eine Zwei-Faktor-Authentifizierung verwendet werden. Ein Beispiel für eine Zwei-Faktor-Authentifizierung ist "etwas, was Sie besitzen" (bspw. einen Schlüssel) und "etwas, was Sie wissen" (bspw. das Passwort für diesen Schlüssel). Da OpenSSH Teil des FreeBSD-Basissystems ist, sollten alle Anmeldungen über das Netzwerk über eine verschlüsselte Verbindung mit einer schlüsselbasierten Authentifizierung stattfinden. Passwörter sollten hier nicht verwendet werden. Weitere Informationen finden Sie in OpenSSH. Kerberos-Benutzer müssen eventuell zusätzliche Änderungen vornehmen, um OpenSSH in Ihrem Netzwerk zu implementieren. Diese Änderungen sind in Kerberos beschrieben.

### 31.2.4. Durchsetzung einer Passwort-Richtlinie

Die Durchsetzung einer starken Passwort-Richtlinie für lokale Benutzerkonten ist ein wesentlicher Aspekt der Systemsicherheit. In FreeBSD kann die Länge, Stärke und Komplexität des Passworts mit den Pluggable Authentication Modules (PAM) implementiert werden.

In diesem Abschnitt wird gezeigt, wie Sie die minimale und maximale Passwortlänge und die Durchsetzung von gemischten Zeichen mit dem Modul pam_passwdqc.so konfigurieren. Dieses Modul wird aufgerufen, wenn ein Benutzer sein Passwort ändert.

Um dieses Modul zu konfigurieren, müssen Sie als Superuser die Zeile mit `pam_passwdqc.so` in /etc/pam.d/passwd auskommentieren. Anschließend bearbeiten Sie die Zeile, so dass sie den vorliegenden Passwort-Richtlinien entspricht:

```
password        requisite       pam_passwdqc.so min=disabled,disabled,disabled,12,10
similar=deny retry=3 enforce=users
```

Dieses Beispiel legt gleich mehrere Anforderungen für neue Passwörter fest. Die Einstellung `min` kontrolliert die Passwortlänge. Es verfügt über fünf Werte, weil dieses Modul fünf verschiedene Arten von Passwörtern definiert, basierend auf der Komplexität. Die Komplexität wird durch die Art von Zeichen definiert, die in einem Passwort vorhanden sind, wie zum Beispiel Buchstaben, Zahlen und Sonderzeichen. Die verschiedenen Arten von Passwörtern werden in pam_passwdqc(8) beschrieben. In diesem Beispiel sind die ersten drei Arten von Passwörtern deaktiviert, was bedeutet, dass Passwörter, die dieser Komplexitätsstufe entsprechen, nicht akzeptiert werden, unabhängig von der Länge des Passworts. Die `12` legt eine Richtlinie von mindestens zwölf Zeichen fest, wenn das Passwort auch drei Arten von Komplexität aufweist. Die `10` legt eine Richtlinie fest, die auch Passwörter mit mindestens zehn Zeichen zulassen, wenn das Passwort Zeichen mit vier Arten von Komplexität aufweist.

Die Einstellung `similar` verbietet Passwörter, die dem vorherigen Passwort des Benutzers ähnlich sind. Die Einstellung `retry` bietet dem Benutzer drei Möglichkeiten, ein neues Passwort einzugeben.

Sobald diese Datei gespeichert wird, sehen Benutzer bei der Änderung ihres Passworts die folgende Meldung:

```
% passwd
Changing local password for trhodes
Old Password:

You can now choose the new password.
A valid password should be a mix of upper and lower case letters,
digits and other characters.  You can use a 12 character long
password with characters from at least 3 of these 4 classes, or
a 10 character long password containing characters from all the
classes.  Characters that form a common pattern are discarded by
the check.
Alternatively, if noone else can see your terminal now, you can
pick this as your password: "trait-useful&knob".
```

```
Enter new password:
```

Wenn ein Passwort nicht den Richtlinien entspricht, wird es mit einer Warnung abgelehnt und der Benutzer bekommt die Möglichkeit, es erneut zu versuchen, bis die Anzahl an Wiederholungen erreicht ist.

Die meisten Passwort-Richtlinien erzwingen, dass Passwörter nach einer bestimmten Anzahl von Tagen ablaufen. Um dieses Limit in FreeBSD zu konfigurieren, setzen Sie es für die Login-Klasse des Benutzers in /etc/login.conf. Die voreingestellte Login-Klasse enthält dazu ein Beispiel:

```
#       :passwordtime=90d:\
```

Um für diese Login-Klasse das Passwort nach 90 Tagen ablaufen zu lassen, entfernen Sie das Kommentarzeichen (`#`), speichern Sie die Änderungen und führen Sie `cap_mkdb /etc/login.conf` aus.

Um das Passwort für einzelne Benutzer ablaufen zu lassen, geben Sie `pw` ein Ablaufdatum oder die Anzahl von Tagen, zusammen mit dem Benutzer an:

```
# pw usermod -p 30-apr-2015 -n trhodes
```

Wie zu sehen ist, wird das Ablaufdatum in der Form von Tag, Monat und Jahr angegeben. Weitere Informationen finden Sie in pw(8).

### 31.2.5. Erkennen von Rootkits

Ein *Rootkit* ist eine nicht autorisierte Software die versucht, Root-Zugriff auf ein System zu erlangen. Einmal installiert, wird diese bösartige Software normalerweise eine Hintertür für den Angreifer installieren. Realistisch betrachtet sollte ein durch ein Rootkit kompromittiertes System nach der Untersuchung von Grund auf neu installiert werden. Es besteht jedoch die enorme Gefahr, dass sogar das Sicherheitspersonal oder Systemingenieure etwas übersehen, was ein Angreifer dort platziert hat.

Wird ein Rootkit erkannt, ist dies bereits ein Zeichen dafür, dass das System an einem bestimmten Zeitpunkt kompromittiert wurde. Meist neigen diese Art von Anwendungen dazu, sehr gut versteckt zu sein. Dieser Abschnitt zeigt das Werkzeug security/rkhunter, mit dem Rootkits erkannt werden können.

Nach der Installation dieses Ports oder Pakets kann das System mit dem folgenden Kommando überprüft werden. Das Programm generiert eine ganze Menge Informationen und Sie werden diverse Male `ENTER` drücken müssen:

```
# rkhunter -c
```

Nachdem der Prozess abgeschlossen ist, wird eine Statusmeldung auf dem Bildschirm ausgegeben. Die Meldung enthält die Anzahl der überprüften Dateien, verdächtige Dateien, mögliche Rootkits und weitere Informationen. Während der Überprüfung erscheinen allgemeine

Sicherheitswarnungen, zum Beispiel über versteckte Dateien, die Auswahl von OpenSSH-Protokollen und bekannte, anfällige Versionen installierter Anwendungen. Diese können nun direkt, oder nach detaillierter Analyse untersucht werden.

Jeder Administrator sollte wissen, was auf den Systemen läuft, für die er verantwortlich ist. Werkzeuge von Drittanbietern, wie rkhunter oder sysutils/lsof, sowie native Befehle wie `netstat` oder `ps`, können eine große Menge an Informationen über das System anzeigen. Machen Sie sich Notizen darüber, was "normal" ist, und fragen Sie nach, wenn Ihnen etwas suspekt erscheint. Eine Beeinträchtigung zu verhindern ist ideal, aber die Erkennung einer Beeinträchtigung ist ein Muss.

### 31.2.6. Überprüfung von Binärdateien

Die Überprüfung von System- und Binärdateien ist wichtig, da sie Systemadministratoren Informationen über Systemänderungen zur Verfügung stellt. Eine Software, die das System auf Änderungen überwacht wird Intrustion Detection System (IDS) genannt.

FreeBSD bietet native Unterstützung für ein einfaches IDS-System. Obwohl die täglichen Sicherheits-E-Mails den Administrator über Änderungen in Kenntnis setzen, werden diese Informationen lokal gespeichert und es besteht die Möglichkeit, dass ein Angreifer diese Informationen manipulieren kann, um Änderungen am System zu verbergen. Daher ist es empfehlenswert, einen eigenen Satz an Signaturen zu erstellen und diese dann in einem schreibgeschützten Verzeichnis, oder vorzugsweise auf einem USB-Stick oder auf einem entfernten Server zu speichern.

Das im Basissystem enthaltene Werkzeug mtree kann verwendet werden, um eine Spezifikation des Inhalts eines Verzeichnisses zu erzeugen. Hierbei wird ein Startwert (Seed) oder eine numerische Konstante benutzt, um die Spezifikation zu erstellen und um sicherzustellen, dass sich die Spezifikation nicht geändert hat. Dadurch kann festgestellt werden, ob eine Datei oder eine Binärdatei verändert wurde. Da ein Angreifer den Seed nicht kennt, ist es ihm fast unmöglich die Prüfsummen von Dateien zu manipulieren. Das folgende Beispiel generiert einen Satz mit SHA256-Prüfsummen für jede Binärdatei unterhalb von /bin und speichert diese Werte in einer versteckten Datei im Heimatverzeichnis von `root` unter dem Namen /root/.bin_chksum_mtree:

```
# mtree -s 3483151339707503 -c -K cksum,sha256digest -p /bin > /root/.bin_chksum_mtree
# mtree: /bin checksum: 3427012225
```

*3483151339707503* stellt den Seed dar. Diesen Wert sollten Sie sich merken, aber nicht mit anderen Personen teilen.

Die Ausgabe von /root/.bin_chksum_mtree sollte ähnlich der folgenden sein:

```
#          user: root
#       machine: dreadnaught
#          tree: /bin
#          date: Mon Feb  3 10:19:53 2014

# .
/set type=file uid=0 gid=0 mode=0555 nlink=1 flags=none
```

```
.               type=dir mode=0755 nlink=2 size=1024 \
                time=1380277977.000000000
    \133        nlink=2 size=1170 time=1380277977.000000000 \
                cksum=484492447 \

sha256digest=6207490fbdb5ed1904441fbfa941279055c3e24d3a4049aeb45094596400662a
    cat         size=12096 time=1380277975.000000000 cksum=3909216944 \

sha256digest=65ea347b9418760b247ab10244f47a7ca2a569c9836d77f074e7a306900c1e69
    chflags     size=8168 time=1380277975.000000000 cksum=3949425175 \

sha256digest=c99eb6fc1c92cac335c08be004a0a5b4c24a0c0ef3712017b12c89a978b2dac3
    chio        size=18520 time=1380277975.000000000 cksum=2208263309 \

sha256digest=ddf7c8cb92a58750a675328345560d8cc7fe14fb3ccd3690c34954cbe69fc964
    chmod       size=8640 time=1380277975.000000000 cksum=2214429708 \

sha256digest=a435972263bf814ad8df082c0752aa2a7bdd8b74ff01431ccbd52ed1e490bbe7
```

Der Report enthält den Rechnernamen, das Datum und die Uhrzeit der Spezifikation, sowie den Namen des Benutzers, der die Spezifikation erstellt hat. Für jede Binärdatei im Verzeichnis gibt es eine Prüfsumme, Größe, Uhrzeit und einen SHA256-Hashwert.

Um sicherzustellen, dass die binären Signaturen nicht verändert wurden, vergleichen Sie den Inhalt des aktuellen Verzeichnisses mit der zuvor erstellen Spezifikation. Speichern Sie die Ergebnisse in einer Datei. Dieses Kommando benötigt den Seed, der verwendet wurde um die ursprüngliche Spezifikation zu erstellen:

```
# mtree -s 3483151339707503 -p /bin < /root/.bin_chksum_mtree >>
/root/.bin_chksum_output
# mtree: /bin checksum: 3427012225
```

Dies sollte die gleiche Prüfsumme für /bin produzieren, wie die ursprüngliche Spezifikation. Wenn keine Änderungen an den Binärdateien in diesem Verzeichnis aufgetreten sind, wird die Ausgabedatei /root/.bin_chksum_output leer sein. Um eine Änderung zu simulieren, ändern Sie mit `touch` das Datum von /bin/cat und führen Sie die Verifikation erneut aus:

```
# touch /bin/cat
# mtree -s 3483151339707503 -p /bin < /root/.bin_chksum_mtree >>
/root/.bin_chksum_output
# more /root/.bin_chksum_output
cat changed
    modification time expected Fri Sep 27 06:32:55 2013 found Mon Feb  3 10:28:43 2014
```

Es wird empfohlen, Spezifikationen für Verzeichnisse zu erstellen, welche Binärdateien, Konfigurationsdateien und sensible Daten enthalten. In der Regel werden Spezifikationen für /bin, /sbin, /usr/bin, /usr/sbin, /usr/local/bin, /usr/local/sbin, /etc und /usr/local/etc erstellt.

Mit security/aide steht ein fortgeschrittenes IDS-System zur Verfügung, aber in den meisten Fällen bietet `mtree` die Funktionalität, die von Administratoren benötigt wird. Es ist jedoch sehr wichtig den Seed und die Prüfsummen in der Ausgabe vor böswilligen Benutzern verborgen zu halten. Weitere Informationen zu `mtree` finden Sie in mtree(8).

### 31.2.7. System-Tuning für Sicherheit

Unter FreeBSD können viele Systemfunktionen mit `sysctl` konfiguriert werden. Dieser Abschnitt behandelt ein paar Sicherheitsmerkmale mit denen Denial of Service (DoS) verhindert werden sollen. Weitere Informationen über die Benutzung von `sysctl` und wie Werte vorübergehend oder auch permanent geändert werden können, finden Sie in “Einstellungen mit sysctl(8)”.



Jedes Mal wenn eine Einstellung mit `sysctl` geändert wird, vergrößert sich die Wahrscheinlichkeit eines unerwünschten Schadens, was die Verfügbarkeit des Systems beeinflusst. Alle Änderungen sollten überwacht und wenn möglich, vorher auf einem Testsystem ausprobiert werden, bevor sie auf einem Produktivsystem verwendet werden.

In der Voreinstellung startet FreeBSD in der Sicherheitsstufe (Securelevel) `-1`. Dieser Modus wird "unsicherer Modus" genannt, da die unveränderlichen Datei-Flags ausgeschaltet werden können und dadurch von allen Geräten gelesen und geschrieben werden kann. Solange die Einstellung nicht über `sysctl` oder in den Startskripten geändert wird, verbleibt die Sicherheitsstufe auf `-1`. Die Sicherheitsstufe kann während des Systemstarts erhöht werden. Dazu muss in /etc/rc.conf`kern_securelevel_enable` auf `YES` und `kern_securelevel` auf den gewünschten Wert gesetzt werden. Weitere Informationen zu diesen Einstellungen und den verfügbaren Sicherheitsstufen finden Sie in security(7) und init(8).



Das Erhöhen der Sicherheitsstufe kann zu Problemen mit Xorg führen.

Die Einstellungen `net.inet.tcp.blackhole` und `net.inet.udp.blackhole` können benutzt werden, um eingehende SYN-Pakete an geschlossenen Ports zu blockieren, ohne ein RST-Paket als Antwort zu senden. Standardmäßig wird jedoch ein RST-Paket gesendet, um zu zeigen, dass der Port geschlossen ist. Das ändern dieser Voreinstellung bietet einen gewissen Schutz gegen Portscans. Diese Portscans versuchen herauszufinden, welche Anwendungen auf einem System ausgeführt werden. Setzen Sie `net.inet.tcp.blackhole` auf `2` und `net.inet.udp.blackhole` auf `1`. Weitere Informationen zu diesen Einstellungen finden Sie in blackhole(4).

Die Einstellung `net.inet.icmp.drop_redirect` hilft dabei, sogenannte Redirect-Angriffe zu verhindern. Ein Redirect-Angriff ist eine Art von DoS, die massenhaft ICMP-Pakete Typ 5 versendet. Da solche Pakete nicht benötigt werden, setzen Sie `net.inet.icmp.drop_redirect` auf `1` und `net.inet.ip.redirect` auf `0`.

Source Routing zur Erfassung und zum Zugriff auf nicht-routbare Adressen im internen Netzwerk. Dies sollte deaktiviert werden, da nicht-routbare Adressen in der Regel nicht absichtlich geroutet werden. Um diese Funktion zu deaktivieren, setzen Sie `net.inet.ip.sourceroute` und `net.inet.accept_sourceroute` auf `0`.

Wenn ein Netzwerkgerät Nachrichten an alle Rechner in einem Subnetz senden muss, wird eine

ICMP-Echo-Request Nachricht an die Broadcast-Adresse gesendet. Allerdings gibt es keinen guten Grund für externe Rechner, solche Nachrichten zu verschicken. Um alle externen Broadcast-Anfragen abzulehnen, setzen Sie `net.inet.icmp.bmcastecho` auf `0`.

Einige zusätzliche Einstellungen sind in security(7) dokumentiert.

## 31.3. Einmalpasswörter

In der Voreinstellung unterstützt FreeBSD One-time Passwords in Everything (OPIE). OPIE wurde konzipiert um Replay-Angriffe zu verhindern, bei dem ein Angreifer das Passwort eines Benutzers ausspäht und es benutzt, um Zugriff auf ein System zu erlangen. Da ein Passwort unter OPIE nur einmal benutzt wird, ist ein ausgespähtes Passwort für einen Angreifer nur von geringem Nutzen. OPIE verwendet eine sichere Hash-Funktion und ein Challenge/Response-System, um Passwörter zu verwalten. Die FreeBSD-Implementation verwendet in der Voreinstellung die MD5-Hash-Funktion.

OPIE verwendet drei verschiedene Arten von Passwörtern. Das erste ist das normale UNIX®- oder Kerberos-Passwort. Das zweite ist das Einmalpasswort, das von `opiekey` generiert wird. Das dritte Passwort ist das "geheime Passwort", das zum Erstellen der Einmalpasswörter verwendet wird. Das geheime Passwort steht in keiner Beziehung zum UNIX®-Passwort und beide Passwörter sollten unterschiedlich sein.

Es gibt noch zwei weitere Werte, die für OPIE wichtig sind. Der erste ist der "Initialwert" (engl. seed oder key), der aus zwei Buchstaben und fünf Ziffern besteht. Der zweite Wert ist der "Iterationszähler", eine Zahl zwischen 1 und 100. OPIE generiert das Einmalpasswort, indem es den Initialwert und das geheime Passwort aneinander hängt und dann die MD5-Hash-Funktion so oft, wie durch den Iterationszähler gegeben, anwendet. Das Ergebnis wird in sechs englische Wörter umgewandelt, die das Einmalpasswort ergeben. Das Authentifizierungssystem (meistens PAM) merkt sich das zuletzt benutzte Einmalpasswort und der Benutzer ist authentifiziert, wenn die Hash-Funktion des Passworts dem vorigen Passwort entspricht. Da nicht umkehrbare Hash-Funktionen benutzt werden, ist es unmöglich, aus einem bekannten Passwort weitere gültige Einmalpasswörter zu berechnen. Der Iterationszähler wird nach jeder erfolgreichen Anmeldung um eins verringert und stellt so die Synchronisation zwischen Benutzer und Login-Programm sicher. Wenn der Iterationszähler den Wert `1` erreicht, muss OPIE neu initialisiert werden.

Es gibt ein paar Programme, die in diesen Prozess einbezogen werden. Ein Einmalpasswort oder eine Liste von Einmalpasswörtern, die von opiekey(1) durch Angabe eines Iterationszählers, eines Initalwertes und einem geheimen Passwort generiert wird. opiepasswd(1) wird benutzt, um Passwörter, Iterationszähler oder Initialwerte zu ändern. opieinfo(1) hingegen gibt den momentanen Iterationszähler und Initialwert eines Benutzers aus, den es aus /etc/opiekeys ermittelt.

Dieser Abschnitt beschreibt vier verschiedene Arten von Tätigkeiten. Zuerst wird erläutert, wie Einmalpasswörter über eine gesicherte Verbindung konfiguriert werden. Als nächstes wird erklärt, wie `opiepasswd` über eine nicht gesicherte Verbindung eingesetzt wird. Als drittes wird beschrieben, wie man sich über eine nicht gesicherte Verbindung anmeldet. Die vierte Tätigkeit beschreibt, wie man eine Reihe von Schlüsseln generiert, die man sich aufschreiben oder ausdrucken kann, um sich von Orten anzumelden, die über keine gesicherten Verbindungen verfügen.

### 31.3.1. OPIE initialisieren

Um OPIE erstmals zu initialisieren, rufen Sie opiepasswd(1) über eine gesicherte Verbindung auf:

```
% opiepasswd -c
[grimreaper] ~ $ opiepasswd -f -c
Adding unfurl:
Only use this method from the console; NEVER from remote. If you are using
telnet, xterm, or a dial-in, type ^C now or exit with no password.
Then run opiepasswd without the -c parameter.
Using MD5 to compute responses.
Enter new secret pass phrase:
Again new secret pass phrase:

ID unfurl OTP key is 499 to4268
MOS MALL GOAT ARM AVID COED
```

Die Option `-c` startet den Konsolen-Modus, der davon ausgeht, dass der Befehl von einem sicherem Ort ausgeführt wird. Dies kann beispielsweise der eigene Rechner sein, oder über eine mit SSH gesicherte Verbindung zum eigenen Rechner.

Geben Sie das geheime Passwort ein, wenn Sie danach gefragt werden. Damit werden die Einmalpasswörter generiert. Dieses Passwort sollte schwer zu erraten sein und sich ebenfalls vom Passwort des Bentuzerkontos unterscheiden. Es muss zwischen 10 und 127 Zeichen lang sein. Prägen Sie sich dieses Passwort gut ein!

Die Zeile, die mit "ID" beginnt, enthält den Login-Namen (`unfrul`), den voreingestellten Iterationszähler (`499`) und den Initialwert (`to4268`). Das System erinnert sich an diese Parameter und wird sie bei einem Anmeldeversuch anzeigen. Sie brauchen sich diese Dinge also nicht merken. Die letzte Zeile enthält das generierte Einmalpasswort, das aus den Parametern und dem geheimen Passwort ermittelt wurde. Bei der nächsten Anmeldung muss dann diese Einmalpasswort benutzt werden.

### 31.3.2. Initialisierung über eine nicht gesicherte Verbindung

Um Einmalpasswörter über eine nicht gesicherte Verbindung zu initialisieren, oder das geheime Passwort zu ändern, müssen Sie über eine gesicherte Verbindung zu einer Stelle verfügen, an der Sie `opiekey` ausführen können. Dies kann etwa die Eingabeaufforderung auf einer Maschine sein, der Sie vertrauen. Zudem müssen Sie einen Iterationszähler vorgeben (100 ist ein guter Wert) und einen Initialwert wählen, wobei Sie auch einen zufällig generierten benutzen können. Benutzen Sie opiepasswd(1) über die ungesicherte Verbindung zu der Maschine, die Sie einrichten wollen:

```
% opiepasswd

Updating unfurl:
You need the response from an OTP generator.
Old secret pass phrase:
        otp-md5 498 to4268 ext
```

```
        Response: GAME GAG WELT OUT DOWN CHAT
New secret pass phrase:
        otp-md5 499 to4269
        Response: LINE PAP MILK NELL BUOY TROY

ID mark OTP key is 499 gr4269
LINE PAP MILK NELL BUOY TROY
```

Drücken Sie `Return`, um die Vorgabe für den Initialwert zu akzeptieren. Bevor Sie nun das Zugriffspasswort (engl. access password) eingeben, rufen Sie über die gesicherte Verbindung `opikey` mit denselben Parametern auf:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Don not use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Gehen Sie zurück zu der nicht gesicherten Verbindung und geben dort das eben generierte Einmalpasswort ein.

### 31.3.3. Erzeugen eines einzelnen Einmalpasswortes

Nachdem Sie OPIE eingerichtet haben, werden Sie beim nächsten Anmelden wie folgt begrüßt:

```
% telnet example.com
Trying 10.0.0.1...
Connected to example.com
Escape character is '^]'.

FreeBSD/i386 (example.com) (ttypa)

login: <username>
otp-md5 498 gr4269 ext
Password:
```

OPIE besitzt eine nützliche Eigenschaft. Wenn Sie an der Eingabeaufforderung `Return` drücken, wird die echo-Funktion eingeschaltet, das heißt Sie sehen, was Sie tippen. Dies ist besonders nützlich, wenn Sie ein generiertes Passwort von einem Ausdruck abtippen müssen.

Jetzt müssen Sie das Einmalpasswort generieren, um der Anmeldeaufforderung nachzukommen. Dies muss auf einem gesicherten System geschehen, auf dem Sie opiekey(1) ausführen können. Dieses Programm gibt es auch für Windows®, Mac OS® und FreeBSD. Es benötigt den Iterationszähler sowie den Initialwert als Parameter, die Sie mittels "cut-and-paste" direkt von der Login-Aufforderung nehmen können.

Auf dem sicheren System:

```
% opiekey 498 to4268
Using the MD5 algorithm to compute response.
Reminder: Do not use opiekey from telnet or dial-in sessions.
Enter secret pass phrase:
GAME GAG WELT OUT DOWN CHAT
```

Sobald das Einmalpasswort generiert wurde, können Sie die Anmeldeprozedur fortsetzen.

### 31.3.4. Erzeugen von mehreren Einmalpasswörtern

Manchmal haben Sie keinen Zugriff auf eine sichere Maschine oder eine sichere Verbindung. In diesem Fall können Sie vorher mit opiekey(1) einige Einmalpasswörter generieren. Zum Beispiel:

```
% opiekey -n 5 30 zz99999
Using the MD5 algorithm to compute response.
Reminder: Do not use opiekey from telnet or dial-in sessions.
Enter secret pass phrase: <secret password>
26: JOAN BORE FOSS DES NAY QUIT
27: LATE BIAS SLAY FOLK MUCH TRIG
28: SALT TIN ANTI LOON NEAL USE
29: RIO ODIN GO BYE FURY TIC
30: GREW JIVE SAN GIRD BOIL PHI
```

Mit `-n 5` fordern Sie fünf Passwörter der Reihe nach an. Der letzte Iterationszähler wird durch `30` gegeben. Beachten Sie bitte, dass die Passwörter in der *umgekehrten* Reihenfolge, in der sie zu benutzen sind, ausgeben werden. Wirklich paranoide Benutzer können sich jetzt die Passwörter aufschreiben oder ausdrucken. Sie sollten die Passwörter nach Gebrauch durchstreichen.

### 31.3.5. Einschränken der Benutzung von System-Passwörtern

OPIE kann die Verwendung von UNIX®-Passwörtern abhängig von der IP-Adresse einschränken. Die dazu nötigen Einstellungen werden in /etc/opieaccess vorgenommen, die bei der Installation des Systems automatisch erzeugt wird. Weitere Informationen über diese Datei und Sicherheitshinweise zu ihrer Verwendung finden Sie in opieaccess(5).

opieaccess könnte beispielsweise die folgende Zeile enthalten:

```
permit 192.168.0.0 255.255.0.0
```

Diese Zeile erlaubt es Benutzern, die sich von einer der angegebenen IP-Adressen anmelden, ihr UNIX®-Passwort zu verwenden. Beachten Sie bitte, dass eine IP-Adresse leicht gefälscht werden kann.

Findet sich in opieaccess kein passender Eintrag, muss die Anmeldung mit OPIE erfolgen.

## 31.4. TCP Wrapper

TCP Wrapper ist ein rechnerbasiertes Zugriffskontrollsystem, das die Fähigkeiten von "Der inetd Super-Server" erweitert. Beispielsweise können Verbindungen protokolliert, Nachrichten zurückgesandt oder nur interne Verbindungen angenommen werden. Weitere Informationen über TCP Wrapper und dessen Funktionen finden Sie in tcpd(8).

TCP Wrapper sollten nicht als Ersatz für eine ordentlich konfigurierte Firewall angesehen werden. Stattdessen sollten TCP Wrapper in Verbindung mit einer Firewall und anderen Sicherheitsmechanismen eingesetzt werden, um bei der Umsetzung einer Sicherheitsrichtlinie eine weitere Sicherheitsschicht zu bieten.

### 31.4.1. Konfiguration

Um TCP Wrapper unter FreeBSD zu aktivieren, fügen Sie die folgenden Zeilen in /etc/rc.conf ein:

```
inetd_enable="YES"
inetd_flags="-Ww"
```

Anschließend muss /etc/hosts.allow richtig konfiguriert werden.



Im Gegensatz zu anderen Implementierungen der TCP Wrapper wird unter FreeBSD vom Gebrauch der Datei hosts.deny abgeraten. Die Konfiguration sollte sich vollständig in /etc/hosts.allow befinden.

In der einfachsten Konfiguration werden Dienste abhängig von den Optionen in /etc/hosts.allow erlaubt oder gesperrt. Unter FreeBSD wird in der Voreinstellung jeder von inetd gestartete Dienst erlaubt.

Eine Konfigurationszeile ist wie folgt aufgebaut: `Dienst : Adresse : Aktion`. `Dienst` ist der von inetd gestartete Dienst (auch Daemon genannt). Die `Adresse` ist ein gültiger Rechnername, eine IP-Adresse oder eine IPv6-Adresse in Klammern (`[ ]`). Der Wert `allow` im Feld `Aktion` erlaubt Zugriffe, der Wert `deny` verbietet Zugriffe. Die Zeilen in hosts.allow werden für jede Verbindung der Reihe nach abgearbeitet. Trifft eine Zeile auf eine Verbindung zu, wird die entsprechende Aktion ausgeführt und die Abarbeitung ist beendet.

Um beispielsweise einkommende POP3-Verbindungen für den Dienst mail/qpopper zu erlauben, sollte hosts.allow um die nachstehende Zeile erweitert werden:

```
# This line is required for POP3 connections:
qpopper : ALL : allow
```

Jedes Mal, wenn diese Datei bearbeitet wird, muss inetd neu gestartet werden:

```
# service inetd restart
```

### 31.4.2. Erweiterte Konfiguration

TCP Wrapper besitzen weitere Optionen, die bestimmen, wie Verbindungen behandelt werden. In einigen Fällen ist es gut, wenn bestimmten Rechnern oder Diensten eine Nachricht geschickt wird. In anderen Fällen soll vielleicht der Verbindungsaufbau protokolliert oder eine E-Mail an einen Administrator versandt werden. Oder ein Dienst soll nur für das lokale Netz bereitstehen. Dies alles ist mit so genannten Wildcards, Metazeichen und der Ausführung externer Programme möglich.

Stellen Sie sich vor, eine Verbindung soll verhindert werden und gleichzeitig soll dem Rechner, der die Verbindung aufgebaut hat, eine Nachricht geschickt werden. Solch eine Aktion ist mit `twist` möglich. `twist` führt beim Verbindungsaufbau ein Kommando oder ein Skript aus. Ein Beispiel ist in hosts.allow enthalten:

```
# Alle anderen Dienste sind geschützt
ALL : ALL \
        : severity auth.info \
        : twist /bin/echo "You are not welcome to use %d from %h."
```

Für jeden Dienst, der nicht vorher in hosts.allow konfiguriert wurde, wird die Meldung "You are not allowed to use *daemon name* from *hostname*." zurückgegeben. Dies ist nützlich, wenn die Gegenstelle sofort benachrichtigt werden soll, nachdem die Verbindung getrennt wurde. Der Text der Meldung *muss* in Anführungszeichen (") stehen.



Ein so konfigurierter Server ist anfällig für Denial-of-Service-Angriffe. Ein Angreifer kann die gesperrten Dienste mit Verbindungsanfragen überfluten.

Eine weitere Möglichkeit bietet `spawn`. Wie `twist` verbietet `spawn` die Verbindung und führt externe Kommandos aus. Allerdings sendet `spawn` dem Rechner keine Rückmeldung. Sehen Sie sich die nachstehende Konfigurationsdatei an:

```
# Verbindungen von example.com sind gesperrt:
ALL : .example.com \
    : spawn (/bin/echo %a from %h attempted to access %d >> \
      /var/log/connections.log) \
    : deny
```

Damit sind Verbindungen von der Domain `*.example.com` gesperrt. Jeder Verbindungsaufbau wird zudem in /var/log/connections.log protokolliert. Das Protokoll enthält den Rechnernamen, die IP-Adresse und den Dienst, der angesprochen wurde. In diesem Beispiel wurden die Metazeichen `%a` und `%h` verwendet. Eine vollständige Liste der Metazeichen finden Sie in hosts_access(5).

Die Wildcard `ALL` passt auf jeden Dienst, jede Domain oder jede IP-Adresse. Eine andere Wildcard ist `PARANOID`. Sie passt auf jeden Rechner, dessen IP-Adresse möglicherweise gefälscht ist. Dies ist beispielsweise der Fall, wenn der Verbindungsaufbau von einer IP-Adresse erfolgt, die nicht zu dem übermittelten Rechnernamen passt. In diesem Beispiel werden alle Verbindungsanfragen zu Sendmail abgelehnt, wenn die IP-Adresse nicht zum Rechnernamen passt:

```
# Block possibly spoofed requests to sendmail:
sendmail : PARANOID : deny
```



Die Wildcard `PARANOID` wird Verbindungen ablehnen, wenn der Client oder der Server eine fehlerhafte DNS-Konfiguration besitzt.

Weitere Informationen über Wildcards und deren Funktion finden Sie in hosts_access(5).



Wenn Sie neue Einträge zur Konfiguration hinzufügen, sollten Sie sicherstellen, dass nicht benötigte Einträge in hosts.allow auskommentiert werden.

## 31.5. Kerberos

Kerberos ist ein Netzwerk-Authentifizierungsprotokoll, das ursprünglich am Massachusetts Institute of Technology (MIT) entwickelt wurde. Es bietet die Möglichkeit zur sicheren Authentifizierung über ein potentiell unsicheres Netzwerk. Das Kerberos-Protokoll benutzt eine starke Kryptographie, um die Identität von Clients und Servern nachweisen zu können. Dabei werden keine unverschlüsselten Daten über das Netzewrk gesendet. Kerberos kann als eine Art Proxy zur Identitätsprüfung, oder als vertrauenswürdiges Authentifizierungssystem betrachtet werden.

Kerberos hat nur eine Aufgabe: Die sichere Prüfung der Identität eines Benutzers (Authentifizierung) über das Netzwerk. Das System überprüft weder die Berechtigungen der Benutzer (Autorisierung), noch verfolgt es die durchgeführten Aktionen (Audit). Daher sollte Kerberos zusammen mit anderen Sicherheits-Systemen eingesetzt werden, die diese Funktionen bereitstellen.

Die aktuelle Version des Protokolls ist Version 5, die in RFC 4120 beschrieben ist. Es existieren mehrere freie Implementierungen dieses Protokolls für eine Reihe von Betriebssystemen. Das MIT entwickelt auch weiterhin seine Kerberos-Version weiter. Es wird in den vereinigten Staaten als Kryptographie-Produkt eingesetzt und unterlag in der Vergangenheit US-Exportbeschränkungen. In FreeBSD ist MIT-Kerberos als Port oder Paket security/krb5 verfügbar. Die Kerberos-Implementation von Heimdal wurde außerhalb der USA entwickelt und unterliegt daher keinen Export-Beschränkungen. Heimdal-Kerberos ist im Basissystem von FreeBSD enthalten. Mit security/heimdal aus der Ports-Sammlung steht eine weitere Distribution, mit mehr konfigurierbaren Optionen zur Verfügung.

Unter Kerberos werden Benutzer und Dienste als "Prinzipale" bezeichnet, die innerhalb einer administrativen Domäne, dem sogenannten "Realm" enthalten sind. Ein typisches Benutzer-Prinzipal hätte das Format `user@REALM` (Realms sind traditionell in Großbuchstaben).

Die folgenden Anweisungen beschreiben, wie Sie das mit FreeBSD gelieferte Heimdal-Kerberos einrichten.

Die Beschreibung der Kerberos-Installation benutzt folgende Namensräume:

- Die DNS-Domain ("Zone") heißt `example.org`.

- Das Kerberos-Realm heißt `EXAMPLE.ORG`.



Benutzen Sie echte Domain-Namen, wenn Sie Kerberos einrichten. Damit vermeiden Sie DNS-Probleme und stellen die Zusammenarbeit mit anderen Kerberos-Realms sicher.

### 31.5.1. Das Heimdal KDC einrichten

Kerberos authentifiziert Benutzer an einer zentralen Stelle: dem Key Distribution Center (KDC). Das KDC verteilt *Tickets*, mit denen ein Dienst die Identität eines Benutzers feststellen kann. Weil alle Mitglieder eines Kerberos-Realms dem KDC vertrauen, gelten für das KDC erhöhte Sicherheitsanforderungen. Der direkte Zugriff auf das KDC sollte daher eingeschränkt sein.

Obwohl der Kerberos-Server wenig Ressourcen benötigt, sollte das KDC wegen der Sicherheitsanforderungen auf einem separaten Rechner installiert werden.

Installieren Sie zunächst das Paket security/heimdal wie folgt:

```
# pkg install heimdal
```

Als nächstes aktualisieren Sie /etc/rc.conf mittels `sysrc`:

```
# sysrc kdc_enable=yes
# sysrc kadmind_enable=yes
```

Danach wird /etc/krb5.conf wie folgt bearbeitet:

```
[libdefaults]
    default_realm = EXAMPLE.ORG
[realms]
    EXAMPLE.ORG = {
    kdc = kerberos.example.org
    admin_server = kerberos.example.org
    }
[domain_realm]
    .example.org = EXAMPLE.ORG
```

Diese Einstellungen setzen voraus, dass der voll qualifizierte Name des KDCs `kerberos.example.org` ist. Der Rechnername des KDC muss im DNS auflösbar sein.

In großen Netzwerken mit einem ordentlich konfigurierten DNS-Server kann die Datei aus dem obigen Beispiel verkürzt werden:

```
[libdefaults]
      default_realm = EXAMPLE.ORG
[domain_realm]
```

```
    .example.org = EXAMPLE.ORG
```

Die Zonendatei von `example.org` muss dann die folgenden Zeilen enthalten:

```
_kerberos._udp      IN  SRV     01 00 88 kerberos.example.org.
_kerberos._tcp      IN  SRV     01 00 88 kerberos.example.org.
_kpasswd._udp       IN  SRV     01 00 464 kerberos.example.org.
_kerberos-adm._tcp  IN  SRV     01 00 749 kerberos.example.org.
_kerberos           IN  TXT     EXAMPLE.ORG
```



Damit die Clients die Kerberos-Dienste benutzen können, *muss* sie entweder eine vollständig konfigurierte /etc/krb5.conf enthalten, oder eine minimale Konfiguration *und* zusätzlich ein richtig konfigurierter DNS-Server.

Im nächsten Schritt wird die Kerberos-Datenbank eingerichtet. Die Datenbank enthält die Schlüssel aller Prinzipale und ist mit einem Passwort geschützt. Dieses Passwort brauchen Sie sich nicht merken, da ein davon abgeleiteter Schlüssel in /var/heimdal/m-key gespeichert wird. Es wäre durchaus sinnvoll, ein 45-stelliges Zufallspasswort für diesen Zweck zu benutzten. Um den Schlüssel zu erstellen, rufen Sie `kstash` auf und geben Sie ein Passwort ein:

```
# kstash
Master key: xxxxxxxxxxxxxxxxxxxxxx
Verifying password - Master key: xxxxxxxxxxxxxxxxxxxxxx
```

Nachdem der Schlüssel erstellt wurde, sollte die Datenbank initialisiert werden. Das Kerberos-Werkzeug kadmin(8) kann die Datenbank mit `kadmin -l` direkt bearbeiten, ohne dabei den Netzwerkdienst kadmind(8) zu benutzen. An der Eingabeaufforderung von `kadmin` kann mit `init` die Datenbank des Realms initialisiert werden:

```
# kadmin -l
kadmin> init EXAMPLE.ORG
Realm max ticket life [unlimited]:
```

Zuletzt wird in `kadmin` mit `add` das erste Prinzipal erstellt. Benutzen Sie vorerst die voreingestellten Optionen für das Prinzipal. Die Optionen können später mit `modify` verändert werden. An der Eingabeaufforderung von kadmin(8) zeigt `?` die verfügbaren Optionen an.

```
kadmin> add tillman
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
Password: xxxxxxxx
```

```
Verifying password - Password: xxxxxxxx
```

Jetzt können die KDC-Dienste wie folgt gestartet werden:

```
# service kdc start
# service kadmind start
```

Obwohl zu diesem Zeitpunkt noch keine kerberisierten Dienste laufen, kann die Funktion des KDC schon überprüft werden, indem Sie für den eben angelegten Benutzer ein Ticket anfordern:

```
% kinit tillman
tillman@EXAMPLE.ORG's Password:
```

Überprüfen Sie, ob das Ticket erfolgreich ausgestellt wurde:

```
% klist
Credentials cache: FILE: /tmp/krb5cc_1001
        Principal: tillman@EXAMPLE.ORG

  Issued                Expires               Principal
Aug 27 15:37:58 2013  Aug 28 01:37:58 2013  krbtgt/EXAMPLE.ORG@EXAMPLE.ORG
```

Nachdem der Test abgeschlossen ist, kann das temporäre Ticket zurückgezogen werden:

```
% kdestroy
```

### 31.5.2. Kerberos-Dienste auf dem Server einrichten

Bei der Konfiguration eines Servers für die Kerberos-Authentifizierung muss zuerst sichergestellt werden, dass /etc/krb5.conf richtig konfiguriert ist. Die Datei kann entweder vom KDC kopiert, oder auf dem neuen System generiert werden.

Als nächstes muss auf dem Server die /etc/krb5.keytab erzeugt werden. Dies ist der Hauptbestandteil um Dienste zu "kerberisieren" und entspricht der Erzeugung eines geheimen Schlüssels zwischen dem Dienst und dem KDC. Das Geheimnis ist ein kryptographischer Schlüssel, der in einem keytab> abgelegt wird. Diese Datei enthält den Schlüssel des Servers, mit dem sich der Server und das KDC gegenseitig authentifizieren können. Sie muss in einer sicheren Art und Weise an den Server übertragen werden, da ansonsten die Sicherheit des Servers gefährdet ist, wenn z.B. die Schlüssel öffentlich werden. In der Regel wird die keytab auf einem vertrauenswürdigen Rechner mit `kadmin` erzeugt und anschließend sicher auf den Server übertragen, beispielsweise mit scp(1). Wenn die Sicherheitsrichtlinien es erlauben, kann die Datei auch direkt auf dem Server erzeugt werden. Es ist sehr wichtig, dass die keytab auf sichere Weise auf den Server übertragen wird. Wenn der Schlüssel einer anderen Partei bekannt wird, kann sich diese Partei den Benutzern als Server ausgeben! Da der Eintrag für das Host-Prinzipal für die KDC-Datenbank auch mit `kadmin`

erstellt wird, ist es praktisch, `kadmin` direkt auf dem Server zu benutzen.

Natürlich ist auch `kadmin` ein kerberisierter Dienst: ein Kerberos-Ticket ist erforderlich, um sich gegenüber dem Netzwerkdienst zu authentifizieren und um sicherzustellen, dass der Benutzer, der `kadmin` ausführt, tatsächlich vorhanden ist. `kadmin` wird nach dem Passwort fragen, um ein neues Ticket zu generieren. Das Prinzipal, das sich mit dem kadmin-Dienst authentifiziert, muss über die Zugriffskontrollliste /var/heimdal/kadmin.acl dazu berechtigt sein. Weitere Informationen über Zugriffskontrolllisten finden Sie in den Heimdal-Info-Seiten (`info heimdal`) im Abschnitt "Remote administration". Wenn der Zugriff auf `kadmin` von entfernten Rechnern verboten ist, kann sich der Administrator entweder über die lokale Konsole oder über ssh(1) mit dem KDC verbinden, um die lokale Administration mit `kadmin -l` durchzuführen.

Nach der Installation von /etc/krb5.conf, können Sie das Kommando `add --random-key` in `kadmin` ausführen, um das Host-Prinzipal in die Datenbank zu schreiben. Das Kommando `ext` extrahiert den Schlüssel des Prinzipals in eine eigene keytab:

```
# kadmin
kadmin> add --random-key host/myserver.example.org
Max ticket life [unlimited]:
Max renewable life [unlimited]:
Principal expiration time [never]:
Password expiration time [never]:
Attributes []:
kadmin> ext_keytab host/myserver.example.org
kadmin> exit
```

Beachten Sie, dass `ext_keytab` den extrahierten Schlüssel standardmäßig in /etc/krb5.keytab speichert. Das ist gut, wenn das Kommando auf dem kerberisierten Server ausgeführt wird, ansonsten sollte das Argument `--keytab pfad/zur/datei` benutzt werden, wenn die keytab an einen anderen Ort extrahiert wird:

```
# kadmin
kadmin> ext_keytab --keytab=/tmp/example.keytab host/myserver.example.org
kadmin> exit
```

Anschließend kann die erzeugte keytab sicher mit scp(1) auf Server oder auf einen Wechseldatenträger kopiert werden. Geben Sie auf jeden Fall einen anderen Namen für die keytab an, um unnötige Schlüssel in der keytab des Systems zu vermeiden.

Mit Hilfe der Datei krb5.conf kann der Server nun mit dem KDC kommunizieren und seine Identität mithilfe der Datei krb5.keytab nachweisen. Jetzt können die kerberisierten Dienste aktiviert werden. Einer der gebräuchlichsten Dienste ist sshd(8), der Kerberos über GSS-API unterstützt. Fügen Sie folgende Zeile in /etc/ssh/sshd_config ein:

```
GSSAPIAuthentication yes
```

Nach dieser Änderung muss sshd(8) mit `service sshd restart` neu gestartet werden, damit die neue Konfiguration wirksam wird.

### 31.5.3. Kerberos auf dem Client einrichten

Genau wie der Server, benötigt auch der Client eine Konfiguration in /etc/krb5.conf. Kopien Sie die Datei (sicher) vom KDC auf den Client, oder schreiben Sie die Datei bei Bedarf einfach neu.

Testen Sie den Client, indem Sie mit `kinit` Tickets anfordern, mit `klist` Tickets anzeigen und mit `kdestroy` Tickets löschen. Kerberos-Anwendungen sollten auch kerberisierte Server ansprechen können. Wenn das nicht funktioniert, Sie aber Tickets anfordern können, hat wahrscheinlich der kerberisierte Server ein Problem und nicht der Client oder das KDC. Im Falle eines kerberisierten ssh(1) ist GSS-API in der Voreinstellung deaktiviert. Testen Sie daher mit `ssh -o GSSAPIAuthentication=yes hostname`.

Wenn Sie die kerberisierten Anwendungen testen, können Sie einen Paket-Sniffer wie `tcpdump` benutzen, um sicherzustellen, dass keine sensiblen Informationen im Klartext übertragen werden.

Es stehen verschiedene Kerberos-Anwendungen zur Verfügung. Die Anwendungen, die SASL benutzen, können dann auch GSS-API benutzen. Viele Arten von Anwendungen können Kerberos zur Authentifizierung verwenden, vom Jabber-Client bis zum IMAP-Client.

Normalerweise wird ein Kerberos-Prinzipal auf ein lokales Benutzerkonto abgebildet. Manchmal wird aber Zugriff auf ein lokales Benutzerkonto benötigt, zu dem es keinen passenden Kerberos-Prinzipal gibt. Der Prinzipal `tillman@EXAMPLE.ORG` bräuchte beispielsweise Zugriff auf das Konto `webdevelopers`. Ebenso könnten andere Prinzipale auf dieses Konto zugreifen wollen.

Die Dateien .k5login und .k5users im Heimatverzeichnis eines Benutzers können verwendet werden, um dieses Problem zu lösen. Mit der folgenden .k5login im Heimatverzeichnis des Benutzers `webdevelopers` haben beide Prinzipale auch ohne das gemeinsame Passwort Zugriff auf das Konto:

```
tillmann@example.org
jdoe@example.org
```

Weitere Informationen zu .k5users finden Sie in ksu(1).

### 31.5.4. Unterschiede zur MIT-Implementation

Der Hauptunterschied zwischen der MIT- und der Heimdal-Implementation ist das Kommando `kadmin`. Die Befehlssätze des Kommandos (obwohl funktional gleichwertig) und das verwendete Protokoll unterscheiden sich in beiden Varianten. Das KDC lässt sich nur mit dem `kadmin` Kommando der passenden Kerberos-Variante verwalten.

Für dieselbe Funktion können auch die Client-Anwendungen leicht geänderte Kommandozeilenoptionen besitzen. Folgen Sie der Anleitung auf http://web.mit.edu/Kerberos/www/. Achten Sie besonders auf den Suchpfad für Anwendungen. Der MIT-Port wird unter FreeBSD standardmäßig in /usr/local/ installiert. Wenn die Umgebungsvariable

PATH zuerst die Systemverzeichnisse enthält, werden die Systemprogramme anstelle der MIT-Programme ausgeführt.

Wenn Sie MIT-Kerberos verwenden, sollten Sie außerdem folgende Änderungen an /etc/rc.conf vornehmen:

```
kdc_program="/usr/local/sbin/kdc"
kadmind_program="/usr/local/sbin/kadmind"
kdc_flags=""
kdc_enable="YES"
kadmind_enable="YES"
```

### 31.5.5. Tipps und Fehlersuche

Während der Konfiguration und bei der Fehlersuche sollten die folgenden Punkte beachtet werden:

- Wenn Sie Heimdal- oder MIT-Kerberos benutzen, muss in der Umgebungsvariable PATH der Pfad zu den Kerberos-Programmen vor dem Pfad zu den Programmen des Systems stehen.
- Wenn die Clients im Realm ihre Uhrzeit nicht synchronisieren, schlägt vielleicht die Authentifizierung fehl. “Die Uhrzeit mit NTP synchronisieren” beschreibt, wie Sie mithilfe von NTP die Uhrzeiten synchronisieren.
- Wenn Sie den Namen eines Rechners ändern, müssen Sie auch den host/-Prinzipal ändern und die keytab aktualisieren. Dies betrifft auch spezielle Einträge wie den HTTP/-Prinzipal für Apaches www/mod_auth_kerb.
- Alle Rechner in einem Realm müssen vor- und rückwärts aufgelöst werden können. Entweder über DNS, zumindest aber über /etc/hosts. CNAME-Einträge im DNS funktionieren, aber die entsprechenden A- und PTR-Einträge müssen vorhanden und richtig sein. Wenn sich Namen nicht auflösen lassen, ist die Fehlermeldung nicht gerade selbstsprechend: `Kerberos5 refuses authentication because Read req failed: Key table entry not found`.
- Einige Betriebssysteme installieren ksu mit falschen Zugriffsrechten; es fehlt das Set-UID-Bit für root. Das hat zur Folge, dass ksu nicht funktioniert. Dies ist ein Fehler in den Zugriffsrechten und kein Fehler des KDCs.
- Wenn Sie für einen Prinzipal unter MIT-Kerberos Tickets mit einer längeren Gültigkeit als der vorgegebenen zehn Stunden einrichten wollen, müssen Sie zwei Sachen ändern. Benutzen Sie `modify_principal` am Prompt von kadmin(8), um die maximale Gültigkeitsdauer für den Prinzipal selbst und den Prinzipal krbtgt zu erhöhen. Das Prinzipal kann dann mit `kinit -l` ein Ticket mit einer längeren Gültigkeit beantragen.
- Mit einem Packet-Sniffer können Sie feststellen, dass Sie sofort nach dem Aufruf von kinit eine Antwort vom KDC bekommen - noch bevor Sie überhaupt ein Passwort eingegeben haben! Das ist in Ordnung: Das KDC händigt ein Ticket-Granting-Ticket (TGT) auf Anfrage aus, da es durch einen vom Passwort des Benutzers abgeleiteten Schlüssel geschützt ist. Wenn das Passwort eingegeben wird, wird es nicht zum KDC gesendet, sondern zum Entschlüsseln der Antwort des KDCs benutzt, die kinit schon erhalten hat. Wird die Antwort erfolgreich entschlüsselt, erhält der Benutzer einen Sitzungs-Schlüssel für die künftige verschlüsselte Kommunikation mit dem KDC und das TGT. Das TGT wiederum ist mit dem Schlüssel des KDCs verschlüsselt. Diese

Verschlüsselung ist für den Benutzer völlig transparent und erlaubt dem KDC, die Echtheit jedes einzelnen TGT zu prüfen.

- Host-Prinzipale können Tickets mit längerer Gültigkeit besitzen. Wenn der Prinzipal eines Benutzers über ein Ticket verfügt, das eine Woche gültig ist, das Ticket des Host-Prinzipals aber nur neun Stunden gültig ist, funktioniert der Ticket-Cache nicht wie erwartet. Im Cache befindet sich dann ein abgelaufenes Ticket des Host-Prinzipals.
- Wenn Sie mit krb5.dict die Verwendung schlechter Passwörter verhindern wollen, wie in kadmin(8) beschrieben, geht das nur mit Prinzipalen, denen eine Passwort-Policy zugewiesen wurde. Das Format von krb5.dict enthält pro Zeile ein Wort. Sie können daher einen symbolischen Link auf /usr/shared/dict/words erstellen.

### 31.5.6. Beschränkungen von Kerberos

Kerberos muss ganzheitlich verwendet werden. Jeder über das Netzwerk angebotene Dienst muss mit Kerberos zusammenarbeiten oder auf anderen Wegen gegen Angriffe aus dem Netzwerk geschützt sein. Andernfalls können Berechtigungen gestohlen und wiederverwendet werden. Es ist beispielsweise nicht sinnvoll, für Remote-Shells Kerberos zu benutzen, dagegen aber POP3-Zugriff auf einem Mail-Server zu erlauben, da POP3 Passwörter im Klartext versendet.

Das KDC ist verwundbar und muss daher genauso abgesichert werden, wie die auf ihm befindliche Passwort-Datenbank. Auf dem KDC sollten absolut keine anderen Dienste laufen und der Rechner sollte physikalisch gesichert sein. Die Gefahr ist groß, da Kerberos alle Passwörter mit einem Schlüssel, dem Haupt-Schlüssel, verschlüsselt. Der Haupt-Schlüssel wiederum wird in einer Datei auf dem KDC gespeichert.

Ein kompromittierter Haupt-Schlüssel ist nicht ganz so schlimm wie allgemein angenommen. Der Haupt-Schlüssel wird nur zum Verschlüsseln der Passwort-Datenbank und zum Initialisieren des Zufallsgenerators verwendet. Solange der Zugriff auf das KDC abgesichert ist, kann ein Angreifer wenig mit dem Haupt-Schlüssel anfangen.

Wenn das KDC nicht zur Verfügung steht, sind auch die Netzwerkdienste nicht benutzbar, da eine Authentifizierung nicht durchgeführt werden kann. Das KDC ist also ein optimales Ziel für einen Denial-of-Service Angriff. Sie können diesem Angriff entgegenwirken, indem Sie einen KDC-Master und einen oder mehrere Slaves verwenden. Der Rückfall auf ein sekundäres KDC mittels PAM-Authentifizierung muss sorgfältig eingerichtet werden.

Mit Kerberos können sich Benutzer, Rechner und Dienste gegenseitig authentifizieren. Allerdings existiert kein Mechanismus, der das KDC gegenüber Benutzern, Rechnern oder Diensten authentifiziert. Ein verändertes `kinit` könnte beispielsweise alle Benutzernamen und Passwörter abfangen. Die von veränderten Programmen ausgehende Gefahr können Sie lindern, indem Sie die Integrität von Dateien mit Werkzeugen wie security/tripwire prüfen.

### 31.5.7. Weiterführende Dokumentation

- The Kerberos FAQ
- Designing an Authentication System: a Dialogue in Four Scenes
- RFC 4120, The Kerberos Network Authentication Service (V5)

- MIT Kerberos-Seite
- Heimdal Kerberos-Wiki

## 31.6. OpenSSL

OpenSSL ist eine Open Source Implementierung der SSL und TLS-Protokolle. Es bietet eine verschlüsselte Transportschicht oberhalb der normalen Kommunikationsschicht und kann daher zusammen mit vielen Netzdiensten benutzt werden.

Das in FreeBSD integrierte OpenSSL stellt die Protokolle Secure Sockets Layer 3.0 (SSLv3) und Transport Layer Security 1.0/1.1/1.2 (TLSv1/TLSv1.1/TLSv1.2) zur Verfügung. Die OpenSSL-Bibliotheken stellen kryptographische Funktionen bereit. FreeBSD 12.0-RELEASE und neuere Versionen enthalten OpenSSL mit Unterstützung für Transport Layer Security 1.3 (TLSv1.3).

Anwendungsbeispiele für OpenSSL sind die verschlüsselte Authentifizierung von E-Mail-Clients oder Web-Transaktionen wie das Bezahlen mit Kreditkarte. Einige Ports, wie www/apache24 und databases/postgresql11-server, haben eine Option für den Bau mit OpenSSL. Bei Auswahl dieser Option, wird OpenSSL aus dem Basissystem benutzt. Wenn Sie für den Bau der Anwendung stattdessen OpenSSL aus dem Port security/openssl benutzten wollen, fügen Sie folgende Zeile in /etc/make.conf ein:

```
DEFAULT_VERSIONS+= ssl=openssl
```

OpenSSL wird auch eingesetzt, um Zertifikate für Anwendungen bereitzustellen. Die Zertifikate stellen die Identität einer Firma oder eines Einzelnen sicher. Wenn ein Zertifikat nicht von einer Zertifizierungsstelle (Certificate Authority, CA) gegengezeichnet wurde, erhalten Sie normalerweise eine Warnung. Eine Zertifizierungsstelle ist eine Firma wie VeriSign, die Zertifikate von Personen oder Firmen gegenzeichnet und damit die Korrektheit der Zertifikate bestätigt. Diese Prozedur kostet Geld, ist aber keine Voraussetzung für den Einsatz von Zertifikaten, beruhigt aber sicherheitsbewusste Benutzer.

Dieser Abschnitt beschreibt, wie Sie auf einem FreeBSD-System Zertifikate erstellen und benutzen. “Konfiguration eines LDAP-Servers” beschreibt, wie Sie eine CA erstellen um die eigenen Zertifikate zu signieren.

Weitere Informationen über SSL finden Sie im kostenlosen OpenSSL Cookbook.

### 31.6.1. Zertifikate erzeugen

Um ein Zertifikat zu erzeugen, das von einer externen CA signiert werden soll, geben Sie folgenden Befehl und die angeforderten Informationen ein. Diese Informationen werden in das Zertifikat geschrieben. Für `Common Name` geben Sie den vollqualifizierten Namen des Systems ein, auf dem das Zertifikat später installiert wird. Wenn der Name nicht übereinstimmt, wird die Anwendung, die das Zertifikat überprüft, dem Benuzter eine Warnung anzeigen. Die Überprüfung würde fehlschlagen und das Zertifikat damit unbrauchbar machen.

```
# openssl req -new -nodes -out req.pem -keyout cert.key -sha256 -newkey rsa:2048
```

```
Generating a 2048 bit RSA private key
..................+++
.............................................................+++
writing new private key to 'cert.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (eg, YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:Another Name
```

Bei der Erzeugung des Zertifikates können noch weitere Optionen, wie die Gültigkeitsdauer und alternative Verschlüsselungsalgorithmen, angegeben werden. openssl(1) beschreibt die zur Verfügung stehenden Optionen.

Das folgende Kommando erstellt zwei Dateien im aktuellen Verzeichnis: Die Anforderung für ein neues Zertifikat wird in req.pem gespeichert. Diese Datei können Sie an eine CA senden, wo die Angaben geprüft werden. Nach erfolgreicher Prüfung wird das Zertifikat signiert und an Sie zurückgesandt. cert.key, enthält den privaten Schlüssel für das Zertifikat und darf auch keine Fall in fremde Hände geraten, da ein Angreifer sonst in der Lage ist, anderen Personen oder Rechnern vorzugaukeln, dass es sich bei ihm um Sie handelt.

Wenn Sie keine Signatur einer Zertifizierungsstelle benötigen, können Sie ein selbst signiertes Zertifikat erstellen. Erzeugen Sie dazu zuerst einen RSA-Schlüssel:

```
# openssl genrsa -rand -genkey -out cert.key 2048
0 semi-random bytes loaded
Generating RSA private key, 2048 bit long modulus
.........................................+++
.....................................................................................
.........................+++
e is 65537 (0x10001)
```

Benutzen Sie diesen Schlüssel, um ein selbst signiertes Zertifikat zu erzeugen. Folgen Sie wieder den Anweisungen am Prompt:

```
# openssl req -new -x509 -days 365 -key cert.key -out cert.crt -sha256
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:PA
Locality Name (eg, city) []:Pittsburgh
Organization Name (eg, company) [Internet Widgits Pty Ltd]:My Company
Organizational Unit Name (eg, section) []:Systems Administrator
Common Name (e.g. server FQDN or YOUR name) []:localhost.example.org
Email Address []:trhodes@FreeBSD.org
```

Dieses Kommando erstellt zwei neue Dateien im aktuellen Verzeichnis: Der Schlüssel der Zertifizierungsstelle cert.key und das Zertifikat selbst, cert.crt. Sie sollten in einem Verzeichnis, vorzugsweise unterhalb von /etc/ssl/ abgelegt werden, das nur von `root` lesbar ist. Die Zugriffsrechte der Dateien können mit `chmod` auf `0700` gesetzt werden.

### 31.6.2. Zertifikate benutzen

Mit einem Zertifikat können beispielsweise die Verbindungen zu Sendmail verschlüsselt werden, um eine Klartext-Authentifizierung zu verhindern.



Einige E-Mail-Programme geben Warnungen aus, wenn ein Zertifikat nicht lokal installiert ist. Weitere Informationen zur Installation von Zertifikaten finden Sie in der Dokumentation der entsprechenden Software.

Unter FreeBSD 10.0-RELEASE und neueren Versionen ist es möglich, ein selbst signiertes Zertifikat für Sendmail automatisch erzeugen zu lassen. Um diese Funktionalität zu aktivieren, fügen Sie die folgenden Zeilen in /etc/rc.conf ein:

```
sendmail_enable="YES"
sendmail_cert_enable="YES"
sendmail_cert_cn="localhost.example.org"
```

Dadurch wird automatisch ein selbst signiertes Zertifikat (/etc/mail/certs/host.cert), der Schlüssel für die CA (/etc/mail/certs/host.key und das Zertifikat der CA (/etc/mail/certs/cacert.pem erzeugt. Das Zertifikat wird den in `sendmail_cert_cn` festgelegten `Common Name` verwenden. Nachdem Sie die Änderungen gespeichert haben, starten Sie Sendmail neu:

```
# service sendmail restart
```

Wenn alles gut ging, erscheinen keine Fehlermeldungen in /var/log/maillog. Für einen einfachen

Test, bauen Sie mit Hilfe von `telnet` eine Verbindung zum Mailserver auf:

```
# telnet example.com 25
Trying 192.0.34.166...
Connected to example.com.
Escape character is '^]'.
220 example.com ESMTP Sendmail 8.14.7/8.14.7; Fri, 18 Apr 2014 11:50:32 -0400 (EDT)
ehlo example.com
250-example.com Hello example.com [192.0.34.166], pleased to meet you
250-ENHANCEDSTATUSCODES
250-PIPELINING
250-8BITMIME
250-SIZE
250-DSN
250-ETRN
250-AUTH LOGIN PLAIN
250-STARTTLS
250-DELIVERBY
250 HELP
quit
221 2.0.0 example.com closing connection
Connection closed by foreign host.
```

Wenn die Zeile `STARTTLS` erscheint, hat alles funktioniert.

## 31.7. VPN mit IPsec

Internet Protocol Security (IPsec) ist ein Satz von Protokollen, die auf dem Internet-Protokoll (IP) aufbauen. Durch Authentifizierung und Verschlüsselung jedes einzelnen IP-Pakets, können mehrere Systeme geschützt miteinander kommunizieren. FreeBSDs IPSsec Netzwerk-Stack basiert auf der http://www.kame.net Implementierung und unterstützt sowohl IPv4 als auch IPv6.

IPsec besteht aus den folgenden Protokollen:

- *Encapsulated Security Payload (ESP)*: dieses Protokoll verschlüsselt IP-Pakete mit einem symmetrischen Verfahren wie Blowfish oder 3DES. Damit werden die Pakete vor Manipulationen Dritter geschützt.
- *Authentication Header (AH)*: dieses Protokoll enthält eine kryptographische Prüfsumme, die sicher stellt, dass ein IP-Paket nicht verändert wurde. Der Authentication-Header folgt nach dem normalen IP-Header und erlaubt dem Empfänger eines IP-Paketes, dessen Integrität zu prüfen.
- *IP Payload Compression Protocol (IPComp)*: dieses Protokoll versucht durch Komprimierung der IP-Nutzdaten die Menge der gesendeten Daten zu reduzieren und somit die Kommunikationsleistung zu verbessern.

Diese Protokolle können, je nach Situation, zusammen oder einzeln verwendet werden.

IPsec unterstützt zwei Modi: Der *Transport-Modus* verschlüsselt die Daten zwischen zwei Systemen.

Der *Tunnel-Modus* verbindet zwei Subnetze miteinander. Durch einen Tunnel können dann verschlüsselte Daten übertragen werden. Ein Tunnel wird auch als Virtual-Private-Network (VPN) bezeichnet. Detaillierte Informationen über das IPsec-Subsystem von FreeBSD finden Sie in ipsec(4).

Seit FreeBSD 11 ist IPsec in der Voreinstellung aktiviert. Um die Unterstützung für IPsec in älteren Versionen zu aktivieren, fügen Sie folgenden Optionen in die Kernelkonfigurationsdatei ein und erstellen Sie einen neuen Kernel, wie in Konfiguration des FreeBSD-Kernels beschrieben.

```
options   IPSEC        IP security
device    crypto
```

Wenn Sie zur Fehlersuche im IPsec-Subsystem Unterstützung wünschen, sollten Sie die folgende Option ebenfalls aktivieren:

```
options   IPSEC_DEBUG  debug for IP security
```

Der Rest dieses Kapitels beschreibt die Einrichtung eines IPsec-VPN zwischen einem Heimnetzwerk und einem Firmennetzwerk. Für das folgende Beispiel gilt:

- Beide Netzwerke sind über ein FreeBSD-Gateway mit dem Internet verbunden.
- Der Gateway jedes Netzwerks besitzt mindestens eine externe IP-Adresse. In diesem Beispiel ist die externe IP-Adresse des Firmennetzwerks (LAN) `172.16.5.4` und das Heimnetzwerk (LAN) hat die externe IP-Adresse `192.168.1.12`.
- Die intern verwendeten IP-Adressen können private oder öffentliche Adressen sein. Sie dürfen sich jedoch nicht überschneiden. Zum Beispiel sollten nicht beide Netze `192.168.1.x` benutzen. In diesem Beispiel ist die interne IP-Adresse des Firmennetzwerks (LAN) `10.246.38.1` und das Heimnetzwerk (LAN) hat die interne IP-Adresse `10.0.0.5`.

### 31.7.1. Konfiguration eines VPN unter FreeBSD

Als erstes muss security/ipsec-tools aus der Ports-Sammlung installiert werden. Diese Software enthält einige Anwendungen, die bei der Konfiguration von IPsec hilfreich sind.

Als nächstes müssen zwei gif(4)-Pseudogeräte angelegt werden, um die Pakete zu tunneln und dafür zu sorgen, dass beide Netzwerke richtig miteinander kommunizieren können. Geben Sie als `root` die folgenden Befehle ein, wobei Sie *intern* und *extern* durch die realen internen und externen IP-Adressen der Gateways ersetzen müssen:

```
# ifconfig gif0 create
# ifconfig gif0 intern1 intern2
# ifconfig gif0 tunnel extern1 extern2
```

Überprüfen Sie mit `ifconfig` die Konfiguration auf beiden Gateways. Hier folgt die Ausgabe von Gateway 1:

```
gif0: flags=8051 mtu 1280
tunnel inet 172.16.5.4 --> 192.168.1.12
inet6 fe80::2e0:81ff:fe02:5881%gif0 prefixlen 64 scopeid 0x6
inet 10.246.38.1 --> 10.0.0.5 netmask 0xffffff00
```

Hier folgt die Ausgabe von Gateway 2:

```
gif0: flags=8051 mtu 1280
tunnel inet 192.168.1.12 --> 172.16.5.4
inet 10.0.0.5 --> 10.246.38.1 netmask 0xffffff00
inet6 fe80::250:bfff:fe3a:c1f%gif0 prefixlen 64 scopeid 0x4
```

Wenn Sie fertig sind, sollten beide internen Adressen über ping(8) erreichbar sein:

```
priv-net# ping 10.0.0.5
PING 10.0.0.5 (10.0.0.5): 56 data bytes
64 bytes from 10.0.0.5: icmp_seq=0 ttl=64 time=42.786 ms
64 bytes from 10.0.0.5: icmp_seq=1 ttl=64 time=19.255 ms
64 bytes from 10.0.0.5: icmp_seq=2 ttl=64 time=20.440 ms
64 bytes from 10.0.0.5: icmp_seq=3 ttl=64 time=21.036 ms
--- 10.0.0.5 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max/stddev = 19.255/25.879/42.786/9.782 ms

corp-net# ping 10.246.38.1
PING 10.246.38.1 (10.246.38.1): 56 data bytes
64 bytes from 10.246.38.1: icmp_seq=0 ttl=64 time=28.106 ms
64 bytes from 10.246.38.1: icmp_seq=1 ttl=64 time=42.917 ms
64 bytes from 10.246.38.1: icmp_seq=2 ttl=64 time=127.525 ms
64 bytes from 10.246.38.1: icmp_seq=3 ttl=64 time=119.896 ms
64 bytes from 10.246.38.1: icmp_seq=4 ttl=64 time=154.524 ms
--- 10.246.38.1 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 28.106/94.594/154.524/49.814 ms
```

Wie erwartet, können nun beiden Seiten ICMP-Pakete von ihren privaten Adressen senden und empfangen. Als nächstes müssen beide Gateways so konfiguriert werden, dass sie die Pakete des anderen Netzwerkes richtig routen. Dazu werden folgende Befehle verwendet:

```
corp-net# route add 10.0.0.0 10.0.0.5 255.255.255.0
corp-net# route add net 10.0.0.0: gateway 10.0.0.5
priv-net# route add 10.246.38.0 10.246.38.1 255.255.255.0
priv-net# route add host 10.246.38.0: gateway 10.246.38.1
```

Ab jetzt sollten die Rechner von den Gateways sowie von den Rechnern hinter den Gateways erreichbar sein. Dies können Sie wieder mit ping(8) überprüfen:

```
corp-net# ping 10.0.0.8
PING 10.0.0.8 (10.0.0.8): 56 data bytes
64 bytes from 10.0.0.8: icmp_seq=0 ttl=63 time=92.391 ms
64 bytes from 10.0.0.8: icmp_seq=1 ttl=63 time=21.870 ms
64 bytes from 10.0.0.8: icmp_seq=2 ttl=63 time=198.022 ms
64 bytes from 10.0.0.8: icmp_seq=3 ttl=63 time=22.241 ms
64 bytes from 10.0.0.8: icmp_seq=4 ttl=63 time=174.705 ms
--- 10.0.0.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.870/101.846/198.022/74.001 ms

priv-net# ping 10.246.38.107
PING 10.246.38.1 (10.246.38.107): 56 data bytes
64 bytes from 10.246.38.107: icmp_seq=0 ttl=64 time=53.491 ms
64 bytes from 10.246.38.107: icmp_seq=1 ttl=64 time=23.395 ms
64 bytes from 10.246.38.107: icmp_seq=2 ttl=64 time=23.865 ms
64 bytes from 10.246.38.107: icmp_seq=3 ttl=64 time=21.145 ms
64 bytes from 10.246.38.107: icmp_seq=4 ttl=64 time=36.708 ms
--- 10.246.38.107 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 21.145/31.721/53.491/12.179 ms
```

Das Konfigurieren der Tunnel ist der einfache Teil. Die Konfiguration einer sicheren Verbindung geht viel mehr in die Tiefe. Die folgende Konfiguration benutzt pre-shared (PSK) RSA-Schlüssel. Abgesehen von den IP-Adressen, sind beide /usr/local/etc/racoon/racoon.conf identisch und sehen ähnlich aus:

```
path    pre_shared_key  "/usr/local/etc/racoon/psk.txt"; #location of pre-shared key
file
log     debug;  #log verbosity setting: set to 'notify' when testing and debugging is
complete

padding # options are not to be changed
{
        maximum_length  20;
        randomize       off;
        strict_check    off;
        exclusive_tail  off;
}

timer   # timing options. change as needed
{
        counter         5;
        interval        20 sec;
        persend         1;
#       natt_keepalive  15 sec;
        phase1          30 sec;
        phase2          15 sec;
}
```

```
listen  # address [port] that racoon will listen on
{
        isakmp          172.16.5.4 [500];
        isakmp_natt     172.16.5.4 [4500];
}

remote  192.168.1.12 [500]
{
        exchange_mode   main,aggressive;
        doi             ipsec_doi;
        situation       identity_only;
        my_identifier   address 172.16.5.4;
        peers_identifier        address 192.168.1.12;
        lifetime        time 8 hour;
        passive         off;
        proposal_check  obey;
#       nat_traversal   off;
        generate_policy off;

                        proposal {
                                encryption_algorithm    blowfish;
                                hash_algorithm          md5;
                                authentication_method   pre_shared_key;
                                lifetime time           30 sec;
                                dh_group                1;
                        }
}

sainfo  (address 10.246.38.0/24 any address 10.0.0.0/24 any)    # address
$network/$netmask $type address $network/$netmask $type ( $type being any or esp)
{                                                               # $network must be the
two internal networks you are joining.
        pfs_group       1;
        lifetime        time    36000 sec;
        encryption_algorithm    blowfish,3des;
        authentication_algorithm        hmac_md5,hmac_sha1;
        compression_algorithm   deflate;
}
```

Eine Beschreibung der verfügbaren Optionen finden Sie in der Manualpage von racoon.conf.

Die Security Policy Database (SPD) muss noch konfiguriert werden, so dass FreeBSD und racoon in der Lage sind den Netzwerkverkehr zwischen den Hosts zu ver- und entschlüsseln.

Dies wird durch ein Shellskript ähnlich wie das folgende, das auf dem Firmennetzwerk-Gateway liegt, ausgeführt. Diese Datei wird während der Systeminitialisierung ausgeführt und sollte unter /usr/local/etc/racoon/setkey.conf gespeichert werden.

```
flush;
```

```
spdflush;

# To the home network
spdadd 10.246.38.0/24 10.0.0.0/24 any -P out ipsec esp/tunnel/172.16.5.4-
192.168.1.12/use;
spdadd 10.0.0.0/24 10.246.38.0/24 any -P in ipsec esp/tunnel/192.168.1.12-
172.16.5.4/use;
```

Nachdem die Datei gespeichert wurde, kann racoon durch das folgende Kommando auf beiden Gateways gestartet werden:

```
# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf -l
/var/log/racoon.log
```

Die Ausgabe sollte so ähnlich aussehen:

```
corp-net# /usr/local/sbin/racoon -F -f /usr/local/etc/racoon/racoon.conf
Foreground mode.
2006-01-30 01:35:47: INFO: begin Identity Protection mode.
2006-01-30 01:35:48: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:35:55: INFO: received Vendor ID: KAME/racoon
2006-01-30 01:36:04: INFO: ISAKMP-SA established 172.16.5.4[500]-192.168.1.12[500]
spi:623b9b3bd2492452:7deab82d54ff704a
2006-01-30 01:36:05: INFO: initiate new phase 2 negotiation:
172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]-
>172.16.5.4[0] spi=28496098(0x1b2d0e2)
2006-01-30 01:36:09: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]-
>192.168.1.12[0] spi=47784998(0x2d92426)
2006-01-30 01:36:13: INFO: respond new phase 2 negotiation:
172.16.5.4[0]192.168.1.12[0]
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 192.168.1.12[0]-
>172.16.5.4[0] spi=124397467(0x76a279b)
2006-01-30 01:36:18: INFO: IPsec-SA established: ESP/Tunnel 172.16.5.4[0]-
>192.168.1.12[0] spi=175852902(0xa7b4d66)
```

Um sicherzustellen, dass der Tunnel richtig funktioniert, wechseln Sie auf eine andere Konsole und benutzen Sie tcpdump(1) mit dem folgenden Befehl, um sich den Netzwerkverkehr anzusehen. Tauschen Sie `em0` durch die richtige Netzwerkkarte aus:

```
# tcpdump -i em0 host 172.16.5.4 and dst 192.168.1.12
```

Die Ausgabe der Konsole sollte dem hier ähneln. Wenn nicht, gibt es ein Problem und ein Debuggen der ausgegebenen Daten ist notwendig.

```
01:47:32.021683 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
```

```
ESP(spi=0x02acbf9f,seq=0xa)
01:47:33.022442 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
ESP(spi=0x02acbf9f,seq=0xb)
01:47:34.024218 IP corporatenetwork.com > 192.168.1.12.privatenetwork.com:
ESP(spi=0x02acbf9f,seq=0xc)
```

An diesem Punkt sollten beide Netzwerke verfügbar sein und den Anschein haben, dass sie zum selben Netzwerk gehören. Meistens sind beide Netzwerke durch eine Firewall geschützt. Um den Netzwerkverkehr zwischen den beiden Netzwerken zu erlauben, ist es notwendig Regeln zu erstellen. Für die ipfw(8) Firewall fügen Sie folgende Zeilen in die Firewall-Konfigurationsdatei ein:

```
ipfw add 00201 allow log esp from any to any
ipfw add 00202 allow log ah from any to any
ipfw add 00203 allow log ipencap from any to any
ipfw add 00204 allow log udp from any 500 to any
```



Die Regelnummern müssen eventuell, je nach Hostkonfiguration, angepasst werden.

Für Benutzer der pf(4)- oder ipf(8)-Firewall sollte folgendes funktionieren:

```
pass in quick proto esp from any to any
pass in quick proto ah from any to any
pass in quick proto ipencap from any to any
pass in quick proto udp from any port = 500 to any port = 500
pass in quick on gif0 from any to any
pass out quick proto esp from any to any
pass out quick proto ah from any to any
pass out quick proto ipencap from any to any
pass out quick proto udp from any port = 500 to any port = 500
pass out quick on gif0 from any to any
```

Zum Ende, um dem Computer den Start vom VPN während der Systeminitialisierung zu erlauben, fügen Sie folgende Zeilen in ihre /etc/rc.conf: ein

```
ipsec_enable="YES"
ipsec_program="/usr/local/sbin/setkey"
ipsec_file="/usr/local/etc/racoon/setkey.conf" # allows setting up spd policies on
boot
racoon_enable="yes"
```

## 31.8. OpenSSH

OpenSSH stellt Werkzeuge bereit, um sicher auf entfernte Maschinen zuzugreifen. Zusätzlich können TCP/IP-Verbindungen sicher durch SSH getunnelt oder weitergeleitet werden. OpenSSH

verschlüsselt alle Verbindungen. Dadurch wird beispielsweise verhindert, dass die Verbindung abgehört oder übernommen (Hijacking) werden kann. Weitere Informationen zu OpenSSH finden Sie auf http://www.openssh.com/.

Dieser Abschnitt enthält einen Überblick über die integrierten Client-Werkzeuge, mit denen Sie sicher auf entfernte Systeme zugreifen können, oder mit denen Sie sicher Dateien austauschen können. Der Abschnitt beschreibt auch die Konfiguration eines SSH-Servers auf einem FreeBSD-System. Weitere Informationen finden Sie in den hier erwähnten Manualpages.

### 31.8.1. Die SSH Client-Werkzeuge benutzen

Benutzen Sie ssh zusammen mit einem Benutzernamen und einer IP-Adresse oder dem Hostnamen, um sich an einem SSH-Server anzumelden. Ist dies das erste Mal, dass eine Verbindung mit dem angegebenen Server hergestellt wird, wird der Benutzer aufgefordert, zuerst den Fingerabdruck des Servers zu prüfen:

```
# ssh user@example.com
The authenticity of host 'example.com (10.0.0.1)' can't be established.
ECDSA key fingerprint is 25:cc:73:b5:b3:96:75:3d:56:19:49:d2:5c:1f:91:3b.
Are you sure you want to continue connecting (yes/no)? yes
Permanently added 'example.com' (ECDSA) to the list of known hosts.
Password for user@example.com: user_password
```

SSH speichert einen Fingerabdruck des Serverschlüssels um die Echtheit des Servers zu überprüfen, wenn der Client eine Verbindung herstellt. Wenn der Benutzer den Fingerabdruck mit yes bestätigt, wird eine Kopie des Schlüssels in .ssh/known_hosts im Heimatverzeichnis des Benutzers gespeichert. Zukünftige Verbindungen zu dem Server werden gegen den gespeicherten Fingerabdruck des Schlüssels geprüft und der Client gibt eine Warnung aus, wenn sich der empfangene Fingerabdruck von dem gespeicherten unterscheidet. Wenn dies passiert, sollte zunächst geprüft werden, ob sich der Schlüssel geändert hat, bevor die Verbindung hergestellt wird.

In der Voreinstellung akzeptieren aktuelle Versionen von OpenSSH nur SSHv2 Verbindungen. Wenn möglich, wird der Client versuchen Version 2 zu verwenden, ist dies nicht möglich, fällt er auf Version 1 zurück. Der Client kann gezwungen werden, nur eine der beiden Versionen zu verwenden, indem die Option -1 oder -2 übergeben wird. Weitere Optionen sind in ssh(1) beschrieben.

Mit scp(1) lassen sich Dateien in einer sicheren Weise auf entfernte Maschinen übertragen. Dieses Beispiel kopiert die Datei COPYRIGHT von einem entfernten System in eine Datei mit dem gleichen Namen auf das lokale System:

```
#  scp user@example.com:/COPYRIGHT COPYRIGHT
Password for user@example.com: *******
COPYRIGHT            100% |*****************************|  4735
00:00
#
```

Da der Fingerabdruck für diesen Rechner bereits bestätigt wurde, wird er automatisch überprüft, bevor der Benutzer zur Eingabe des Passworts aufgefordert wird.

Die Argumente, die `scp` übergeben werden, gleichen denen von `cp` in der Beziehung, dass die ersten Argumente die zu kopierenden Dateien sind und das letzte Argument den Bestimmungsort angibt. Da die Dateien über das Netzwerk kopiert werden, können ein oder mehrere Argumente die Form `user@host:<path_to_remote_file>` besitzen. Beachten Sie, das `scp` die Option `-r` verwendet um Dateien rekursiv zu kopieren, während `cp -R` benutzt.

Mit `sftp` können Dateien über eine interaktive Sitzung kopiert werden. sftp(1) beschreibt die verfügbaren Befehle, die während einer `sftp`-Sitzung zur Verfügung stehen.

#### Schlüsselbasierte Authentifizierung

Ein Client kann bei der Verbindung auch Schlüssel anstelle von Passwörtern verwenden. Benutzen Sie `ssh-keygen` um RSA-Schlüssel erzeugen. Geben Sie das entsprechende Protokoll an, wenn Sie einen öffentlichen und einen privaten Schlüssel erzeugen. Folgen Sie anschließend den Anweisungen des Programms. Es wird empfohlen, die Schlüssel mit einer einprägsamen, aber schwer zu erratenen Passphrase zu schützen.

```
% ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/user/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):  ①
Enter same passphrase again:                 ②
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:54Xm9Uvtv6H4NOo6yjP/YCfODryvUU7yWHzMqeXwhq8 user@host.example.com
The key's randomart image is:
+---[RSA 2048]----+
|                 |
|                 |
|                 |
|        . o..    |
|       .S*+*o    |
|      . O=Oo . . |
|       = Oo= oo..|
|      .oB.* +.oo.|
|       =OE**.o..=|
+----[SHA256]-----+
```

① Geben Sie hier die Passphrase ein. Diese darf auch Leer- und Sonderzeichen enthalten.

② Geben Sie die Passphrase zur Überprüfung erneut ein.

Der private Schlüssel wird in ~/.ssh/id_rsa und der öffentliche Schlüssel in ~/.ssh/id_rsa.pub gespeichert. Der *öffentliche* Schlüssel muss zuerst auf den entfernten Rechner nach ~/.ssh/authorized_keys kopiert werden, damit die schlüsselbasierte Authentifizierung funktioniert.



Viele Benutzer denken, dass die Verwendung von Schlüsseln generell sicher ist. Sie verwenden dann einen Schlüssel ohne eine Passphrase. Dies ist jedoch sehr *gefährlich*. Ein Administrator kann überprüfen, ob ein Schlüsselpaar mit einer Passphrase geschützt ist. Wenn die Datei mit dem privaten Schlüssel den Text `ENCRYPTED` enthält, dann hat der Benutzer eine Passphrase verwendet. Um die Benutzer zusätzlich zu schützen, kann ein `from`-Feld in der Datei des öffentlichen Schlüssels hinzugefügt werden. Zum Beispiel würde das Hinzufügen von `from="192.168.10.5"` vor dem `ssh-rsa`-Präfix dafür sorgen, dass sich ein bestimmter Benutzer nur noch von dieser IP-Adresse anmelden darf.

Die Optionen und Dateinamen sind abhängig von der eingesetzten Version von OpenSSH. Die für das System gültigen Optionen finden Sie in ssh-keygen(1).

Wenn bei der Erzeugung des Schlüssels eine Passphrase angegeben wurde, wird der Benutzer bei jeder Anmeldung am Server zur Eingabe der Passphrase aufgefordert. Mit ssh-agent(1) und ssh-add(1) ist es möglich, SSH-Schlüssel in den Speicher zu laden, damit die Passphrase nicht jedes Mal eingegeben werden muss.

`ssh-agent` übernimmt die Authentifizierung mit den geladenen privaten Schlüsseln. `ssh-agent` kann dazu verwendet werden, ein anderes Programm zu starten, beispielsweise eine Shell oder einen Window-Manager.

Um `ssh-agent` in einer Shell zu verwenden, muss es mit einer Shell als Argument aufgerufen werden. Die zu verwaltende Identität muss mit `ssh-add` sowie der Passphrase für den privaten Schlüssel übergeben werden. Danach kann sich der Benutzer mit `ssh` auf jedem Rechner anmelden, der einen entsprechenden öffentlichen Schlüssel besitzt. Dazu ein Beispiel:

```
% ssh-agent csh
% ssh-add
Enter passphrase for /usr/home/user/.ssh/id_rsa:  ①
Identity added: /usr/home/user/.ssh/id_rsa (/home/user/.ssh/id_rsa)
%
```

① Geben Sie hier die Passphrase für den Schlüssel ein.

Um `ssh-agent` unter Xorg zu verwenden, muss ein Eintrag für das Programm in ~/.xinitrc aufgenommen werden. Dadurch können alle unter Xorg gestarteten Programme die Dienste von `ssh-agent` nutzen. ~/.xinitrc könnte etwa so aussehen:

```
exec ssh-agent startxfce4
```

Dadurch wird bei jedem Start von Xorg zuerst `ssh-agent` aufgerufen, das wiederum XFCE startet. Nachdem diese Änderung durchgeführt wurde, muss Xorg neu gestartet werden. Danach können Sie mit `ssh-add` die SSH-Schlüssel laden.

### SSH-Tunnel

Mit OpenSSH ist es möglich, einen Tunnel zu erstellen, in dem ein anderes Protokoll verschlüsselt übertragen wird.

Im folgenden Kommando erzeugt `ssh` einen Tunnel für `telnet`:

```
% ssh -2 -N -f -L 5023:localhost:23 user@foo.example.com
%
```

Dieses Beispiel verwendet die folgenden Optionen:

`-2`
: Zwingt `ssh` dazu, die Version 2 des Protokolls zu verwenden, um sich mit dem Server zu verbinden.

`-N`
: Zeigt an, dass ein Tunnel erstellt werden soll. Ohne diese Option würde `ssh` eine normale Sitzung öffnen.

`-f`
: Zwingt `ssh` im Hintergrund zu laufen.

`-L`
: Ein lokaler Tunnel wird in der Form *localport:remotehost:remoteport* angegeben. Die Verbindung wird dabei von dem lokalen Port *localport* auf einen entfernten Rechner weitergeleitet.

`user@foo.example.com`
: Gibt den Anmeldenamen auf dem entfernten SSH-Server an.

Ein SSH-Tunnel erzeugt einen Socket auf `localhost` und dem angegebenen lokalen Port. Jede Verbindung, die auf dem angegebenen Socket aufgemacht wird, wird dann auf den spezifizierten entfernten Rechner und Port weitergeleitet. Im Beispiel wird der lokale Port `5023` an die entfernte Maschine auf Port `23` weitergeleitet. Da der Port 23 für `telnet` reserviert ist, erzeugt das eine sichere telnet(1)-Verbindung durch einen SSH-Tunnel.

Wie in den folgenden Beispielen zu sehen ist, kann diese Vorgehensweise genutzt werden, um jedes unsichere TCP-Protokoll, wie SMTP, POP3 und FTP, weiterzuleiten.

*Beispiel 30. Einen sicheren Tunnel für SMTP erstellen*

```
% ssh -2 -N -f -L 5025:localhost:25 user@mailserver.example.com
user@mailserver.example.com's password: *****
% telnet localhost 5025
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
```

```
220 mailserver.example.com ESMTP
```

Zusammen mit `ssh-keygen` und zusätzlichen Benutzer-Accounts können leicht benutzbare SSH-Tunnel aufgebaut werden. Anstelle von Passwörtern können Schlüssel benutzt werden und jeder Tunnel kann unter einem eigenen Benutzer laufen.

*Beispiel 31. Sicherer Zugriff auf einen POP3-Server*

In diesem Beispiel gibt es einen SSH-Server, der Verbindungen von außen akzeptiert. Im selben Netzwerk befindet sich zudem noch ein Mail-Server, der POP3 spricht. Um E-Mails auf sichere Weise abzurufen, bauen Sie eine SSH-Verbindung zu dem SSH-Server im Netzwerk auf und tunneln von dort zum Mail-Server weiter.

```
% ssh -2 -N -f -L 2110:mail.example.com:110 user@ssh-server.example.com
user@ssh-server.example.com's password: ******
```

Wenn Sie den Tunnel eingerichtet haben, konfigurieren Sie den Mail-Client so, dass er POP3 Anfragen zu `localhost` auf Port 2110 sendet. Diese Verbindung wird dann über den gesicherten Tunnel zu `mail.example.com` weitergeleitet.

*Beispiel 32. Umgehen einer Firewall*

Einige Firewalls filtern sowohl eingehende als auch ausgehende Verbindungen. Zum Beispiel könnte eine Firewall den Zugriff auf entfernte Rechner auf die Ports 22 und 80 beschränken, um lediglich SSH und Web-Inhalte zu erlauben. Dies würde den Zugriff auf Dienste verhindern, die nicht die Ports 22 oder 80 benutzen.

Die Lösung hier ist es, eine SSH-Verbindung zu einer Maschine außerhalb der Firewall aufzumachen und durch diese zum gewünschten Dienst zu tunneln:

```
% ssh -2 -N -f -L 8888:music.example.com:8000 user@unfirewalled-system.example.org
user@unfirewalled-system.example.org's password: *******
```

In diesem Beispiel benutzt ein Ogg Vorbis Client `localhost` und Port 8888. Die Verbindung wird dann zu `music.example.com` Port 8000 weitergeleitet. Die Firewall wurde somit erfolgreich umgangen.

### 31.8.2. Den SSH-Server aktivieren

Neben den integrierten SSH Client-Werkzeugen, die zur Verfügung stehen, kann ein FreeBSD-System auch als SSH-Server konfiguriert werden, um Verbindungen von anderen SSH-Clients zu akzeptieren.

Benutzen Sie den Kommando service(8), um zu prüfen ob der sshd ausgeführt wird:

```
# service sshd status
```

Wenn der Dienst nicht ausgeführt wird, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
sshd_enable="YES"
```

Diese Zeile startet `sshd`, den OpenSSH-Daemon, beim nächsten Systemstart. Geben Sie folgendes ein, um den Dienst jetzt zu starten:

```
# service sshd start
```

Wenn `sshd` erstmalig gestartet wird, werden die Host-Schlüssel des Systems erzeugt und der Fingerabdruck wird auf der Konsole angezeigt. Stellen Sie den Fingerabdruck den Benutzern zur Verfügung, sodass sie ihn überprüfen können, wenn sie das erste Mal eine Verbindung mit dem Server herstellen.

sshd(8) enthält die verfügbaren Optionen für den Start von `sshd` und weitere Informationen zur Authentifizierung, den Anmeldeprozess und die verschiedenen Konfigurationsdateien.

Ab jetzt sollte sshd für alle Benutzer mit einem Benutzernamen und Kennwort zur Verfügung stehen.

### 31.8.3. SSH Server Sicherheit

Obwohl sshd das am weitesten verbreitete Remote-Administrations-Werkzeug ist, sind Brute-Force- und Drive-by-Angriffe auf öffentliche Netzwerke weit verbreitet. Daher stehen mehrere Optionen zur Verfügung, um diese Art von Angriffen zu verhindern. Diese Optionen werden in diesem Abschnitt beschrieben.

Es ist in der Regel ein gute Idee, festzulegen, welche Benutzer sich von welchem Rechner aus anmelden können. Dies lässt sich beispielsweise über die Option `AllowUsers` festlegen. Soll sich etwa nur `root` vom Rechner mit der IP-Adresse `192.168.1.32` aus einwählen dürfen, würden Sie folgenden Eintrag in /etc/ssh/sshd_config aufnehmen:

```
AllowUsers root@192.168.1.32
```

Damit sich `admin` von jedem Rechner aus anmelden kann, geben Sie nur den Benutzernamen an:

```
AllowUsers admin
```

Sie können auch mehrere Benutzer in einer Zeile aufführen:

```
AllowUsers root@192.168.1.32 admin
```

Nachdem Sie /etc/ssh/sshd_config angepasst haben, muss `sshd` seine Konfigurationsdateien neu einlesen. Dazu geben Sie Folgendes ein:

```
# /etc/rc.d/sshd reload
```



Wenn die Option `AllowUsers` verwendet wird, ist es wichtig, jeden Benutzer aufzulisten, der sich an diesem Rechner anmelden muss. Benutzer, die nicht in dieser Liste aufgeführt sind, dürfen sich nicht anmelden. Die Optionen für die Konfigurationsdatei von OpenSSH unterscheiden zwischen Groß- und Kleinschreibung. Wenn Sie eine Option falsch schreiben, so wird sie ingnoriert. Testen Sie immer die Änderungen, um sicherzustellen, dass sie wie erwartet funktionieren. Weitere Informationen zu den verfügbaren Optionen finden Sie in sshd_config(5).

Darüber hinaus können Benutzer gezwungen werden, eine Zwei-Faktor-Authentifizierung mit einem öffentlichen und einem privaten Schlüssel zu benutzen. Bei Bedarf kann der Benutzer ein Schlüsselpaar mit ssh-keygen(1) erzeugen und dem Administrator den öffentlichen Schlüssel zukommen lassen. Der Schlüssel wird, wie weiter oben beschrieben, in authorized_keys platziert. Um den Benutzer zu zwingen, ausschließlich Schlüssel zu benutzen, kann die folgende Option konfiguriert werden:

```
AuthenticationMethods publickey
```



Verwechseln Sie nicht /etc/ssh/sshd_config mit /etc/ssh/ssh_config (beachten Sie das zusätzliche `d` im ersten Dateinamen). Die erste Datei konfiguriert den Server und die zweite Datei konfiguriert den Client. ssh_config(5) enthält eine Auflistung der verfügbaren Client-Einstellungen.

## 31.9. Zugriffskontrolllisten für Dateisysteme (ACL)

*Zugriffskontrolllisten* (Access Control Lists, ACL) erweitern die normalen Zugriffsrechte von UNIX® Systemen auf eine kompatible (POSIX®.1e) Weise und bieten feiner granulierte Sicherheitsmechanismen.

Der GENERIC-Kernel von FreeBSD bietet ACL-Unterstützung für UFS-Dateisysteme. Benutzer, die es vorziehen einen eigenen Kernel zu übersetzen, müssen die folgende Option in die Kernelkonfigurationsdatei aufnehmen:

```
options UFS_ACL
```

Das System gibt eine Warnung aus, wenn ein Dateisystem mit ACLs eingehangen werden soll und die Unterstützung für ACLs nicht im Kernel aktiviert ist. ACLs bauen auf den erweiterten Attributen auf, die von UFS2 standardmäßig unterstützt werden.

Dieses Kapitel beschreibt, wie ACL-Unterstützung aktiviert wird. Zudem werden einige Anwendungsbeispiele vorgestellt.

### 31.9.1. ACL-Unterstützung aktivieren

Die Option `acl` in /etc/fstab aktiviert Zugriffskontrollisten für ein Dateisystem. Die bevorzugte Möglichkeit ist die Verwendung von Zugriffskontrollisten mit tunefs(8) (Option `-a`), im Superblock des Dateisystems festzuschreiben. Diese Möglichkeit hat mehrere Vorteile:

- Nochmaliges Einhängen eines Dateisystems (Option `-u` von mount(8)) verändert den Status der Zugriffskontrollisten nicht. Die Verwendung von Zugriffskontrollisten kann nur durch Abhängen und erneutes Einhängen eines Dateisystems verändert werden. Das heißt auch, dass Zugriffskontrollisten nicht nachträglich auf dem Root-Dateisystem aktiviert werden können.
- Die Zugriffskontrollisten auf den Dateisystemen sind, unabhängig von den Optionen in /etc/fstab oder Namensänderungen der Geräte, immer aktiv. Dies verhindert auch, dass Zugriffskontrollisten aus Versehen auf Dateisystemen ohne Zugriffskontrollisten aktiviert werden.



Es kann sein, dass sich der Status von Zugriffskontrollisten später durch nochmaliges Einhängen des Dateisystems (Option `-u` von mount(8)) ändern lässt. Die momentane Variante ist aber sicherer, da der Status der Zugriffskontrollisten nicht versehentlich geändert werden kann. Allgemein sollten Zugriffskontrollisten auf einem Dateisystem, auf dem sie einmal verwendet wurden, nicht deaktiviert werden, da danach die Zugriffsrechte falsch sein können. Werden Zugriffskontrollisten auf einem solchen Dateisystem wieder aktiviert, werden die Zugriffsrechte von Dateien, die sich zwischenzeitlich geändert haben, überschrieben, was zu erneuten Problemen führt.

Die Zugriffsrechte einer Datei werden durch ein `+` (Plus) gekennzeichnet, wenn die Datei durch Zugriffskontrollisten geschützt ist:

```
drwx------  2 robert  robert  512 Dec 27 11:54 private
drwxrwx---+ 2 robert  robert  512 Dec 23 10:57 directory1
drwxrwx---+ 2 robert  robert  512 Dec 22 10:20 directory2
drwxrwx---+ 2 robert  robert  512 Dec 27 11:57 directory3
drwxr-xr-x  2 robert  robert  512 Nov 10 11:54 public_html
```

In diesem Beispiel sind die Verzeichnisse directory1, directory2 und directory3 durch Zugriffskontrollisten geschützt, wohingegen das Verzeichnis public_html nicht geschützt ist.

### 31.9.2. Zugriffskontrollisten benutzen

`getfacl` zeigt Zugriffskontrollisten an. Das folgende Kommando zeigt die ACLs auf der Datei test:

```
% getfacl test
    #file:test
    #owner:1001
```

```
    #group:1001
    user::rw-
    group::r--
    other::r--
```

`setfacl` ändert oder entfernt ACLs auf Dateien. Um alle ACLs einer Datei zu entfernen, können Sie die Option `-k` benutzen. Es ist jedoch empfehlenswert die Option `-b` zu verwenden, da sie die erforderlichen Felder, die für ACLs benötigt werden, beibehält.

```
# setfacl -k test
```

Benutzen Sie `-m` um die Einträge der ACL zu verändern:

```
% setfacl -m u:trhodes:rwx,g:web:r--,o::--- test
```

In diesem Beispiel gab es keine vordefinierten Einträge, da sie durch den vorhergehenden Befehl entfernt wurden. Mit diesem Kommando werden die eben entfernten Zugriffskontrolllisten wiederhergestellt. Der Befehl gibt die Fehlermeldung `Invalid argument` aus, wenn Sie nicht existierende Benutzer oder Gruppen als Parameter angeben.

Weitere Informationen zu den Optionen dieser Kommandos finden Sie in getfacl(1) und setfacl(1).

## 31.10. Sicherheitsprobleme in Software von Drittanbietern überwachen

In den letzten Jahren wurden zahlreiche Verbesserungen in der Einschätzung und dem Umgang mit Sicherheitsproblemen erzielt. Die Gefahr von Einbrüchen in ein System wird aber immer größer, da Softwarepakete von Dritten auf nahezu jedem Betriebssystem installiert und konfiguriert werden.

Die Einschätzung der Verletzlichkeit eines Systems ist ein Schlüsselfaktor für dessen Sicherheit. FreeBSD veröffentlicht zwar Sicherheitshinweise (security advisories) für das Basissystem, das Projekt ist allerdings nicht dazu in der Lage, dies auch für die zahlreichen Softwarepakete von Dritten zu tun. Dennoch gibt es einen Weg, auch diese Programmpakete zu überwachen. Das FreeBSD Dienstprogramm pkg enthält Optionen für genau diesen Anwendungsfall.

pkg fragt dazu eine Datenbank auf bekannte Sicherheitsprobleme ab. Diese Datenbank wird vom FreeBSD Security Team sowie den Ports-Entwicklern aktualisiert und gewartet.

Anweisungen zur Installation von pkg finden Sie im Benutzen von pkg zur Verwaltung von Binärpaketen.

Die Installation enthält Konfigurationsdateien für periodic(8), welche die Datenbank von pkg verwaltet und aktualisiert. Diese Funktionalität wird aktiviert, wenn in periodic.conf(5) die Variable `daily_status_security_pkgaudit_enable` auf `YES` gesetzt wird. Stellen Sie auf jeden Fall sicher, dass diese (an das E-Mail-Konto von `root` gesendeten) Sicherheitsberichte auch gelesen

werden.

Nach der Installation kann ein Administrator mit dem folgenden Kommando die Datenbank aktualisieren und sich die Sicherheitslücken in installierten Paketen anzeigen lassen:

```
# pkg audit -F
```

pkg zeigt dann die Schwachstellen in installierten Pakete an:

```
Affected package: cups-base-1.1.22.0_1
Type of problem: cups-base -- HPGL buffer overflow vulnerability.
Reference: <https://www.FreeBSD.org/ports/portaudit/40a3bca2-6809-11d9-a9e7-
0001020eed82.html>

1 problem(s) in your installed packages found.

You are advised to update or deinstall the affected package(s) immediately.
```

Wenn Sie die angegebene URL über einen Internetbrowser aufrufen, erhalten Sie weitere Informationen über die bestehende Sicherheitslücke, wie die betroffenen Versionen, die Version des FreeBSD-Ports sowie Hinweise auf weitere Seiten, die ebenfalls Sicherheitshinweise zu diesem Problem bieten.

pkg ist ein mächtiges Werkzeug und insbesondere in Zusammenarbeit mit ports-mgmt/portmaster äußerst hilfreich.

## 31.11. FreeBSD Sicherheitshinweise

Wie viele andere Hersteller von hochwertigen Betriebssystemen, hat auch das FreeBSD-Projekt ein Sicherheitsteam, das für die Bestimmung des End-of-Life (EoL) Datum verantwortlich ist. Das Sicherheitsteam stellt zudem sicher, dass Sicherheitsupdates für unterstützte Versionen, welche noch nicht ihr EoL erreicht haben, zur Verfügung gestellt werden. Weitere Informationen über das FreeBSD Sicherheitsteam und den unterstützten Versionen finden Sie auf der Webseite FreeBSD Security.

Zu den Aufgaben des Sicherheitsteams zählt es, auf gemeldete Sicherheitslücken im FreeBSD-Betriebssystem zu reagieren. Sobald eine Sicherheitslücke bestätigt wird, überprüft das Sicherheitsteam die notwendigen Schritte, um die Schwachstelle zu beheben und den Quellcode mit der Korrektur zu aktualisieren. Anschließend veröffentlicht es die Details in einem Sicherheitshinweis (Security Advisory). Die Sicherheitshinweise werden auf der FreeBSD Webseite und auf den Mailinglisten FreeBSD security notifications, FreeBSD security und FreeBSD announcements veröffentlicht.

Dieser Abschnitt beschreibt das Format eines FreeBSD Sicherheitshinweises.

### 31.11.1. Format eines Sicherheitshinweis

Hier ist ein Beispiel für einen FreeBSD Sicherheitshinweis:

```
=============================================================================
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

=============================================================================
FreeBSD-SA-14:04.bind                                        Security Advisory
                                                           The FreeBSD Project

Topic:          BIND remote denial of service vulnerability

Category:       contrib
Module:         bind
Announced:      2014-01-14
Credits:        ISC
Affects:        FreeBSD 8.x and FreeBSD 9.x
Corrected:      2014-01-14 19:38:37 UTC (stable/9, 9.2-STABLE)
                2014-01-14 19:42:28 UTC (releng/9.2, 9.2-RELEASE-p3)
                2014-01-14 19:42:28 UTC (releng/9.1, 9.1-RELEASE-p10)
                2014-01-14 19:38:37 UTC (stable/8, 8.4-STABLE)
                2014-01-14 19:42:28 UTC (releng/8.4, 8.4-RELEASE-p7)
                2014-01-14 19:42:28 UTC (releng/8.3, 8.3-RELEASE-p14)
CVE Name:       CVE-2014-0591

For general information regarding FreeBSD Security Advisories,
including descriptions of the fields above, security branches, and the
following sections, please visit <URL:http://security.FreeBSD.org/>.

I.   Background

BIND 9 is an implementation of the Domain Name System (DNS) protocols.
The named(8) daemon is an Internet Domain Name Server.

II.  Problem Description

Because of a defect in handling queries for NSEC3-signed zones, BIND can
crash with an "INSIST" failure in name.c when processing queries possessing
certain properties.  This issue only affects authoritative nameservers with
at least one NSEC3-signed zone.  Recursive-only servers are not at risk.

III. Impact

An attacker who can send a specially crafted query could cause named(8)
to crash, resulting in a denial of service.

IV.  Workaround
```

```
No workaround is available, but systems not running authoritative DNS service
with at least one NSEC3-signed zone using named(8) are not vulnerable.

V.   Solution

Perform one of the following:

1) Upgrade your vulnerable system to a supported FreeBSD stable or
release / security branch (releng) dated after the correction date.

2) To update your vulnerable system via a source code patch:

The following patches have been verified to apply to the applicable
FreeBSD release branches.

a) Download the relevant patch from the location below, and verify the
detached PGP signature using your PGP utility.

[FreeBSD 8.3, 8.4, 9.1, 9.2-RELEASE and 8.4-STABLE]
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-release.patch
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-release.patch.asc
# gpg --verify bind-release.patch.asc

[FreeBSD 9.2-STABLE]
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-stable-9.patch
# fetch http://security.FreeBSD.org/patches/SA-14:04/bind-stable-9.patch.asc
# gpg --verify bind-stable-9.patch.asc

b) Execute the following commands as root:

# cd /usr/src
# patch < /path/to/patch

Recompile the operating system using buildworld and installworld as
described in <URL:https://www.FreeBSD.org/handbook/makeworld.html>.

Restart the applicable daemons, or reboot the system.

3) To update your vulnerable system via a binary patch:

Systems running a RELEASE version of FreeBSD on the i386 or amd64
platforms can be updated via the freebsd-update(8) utility:

# freebsd-update fetch
# freebsd-update install

VI.  Correction details

The following list contains the correction revision numbers for each
affected branch.
```

```
Branch/path                                                         Revision
- ------------------------------------------------------------------------
stable/8/                                                           r260646
releng/8.3/                                                         r260647
releng/8.4/                                                         r260647
stable/9/                                                           r260646
releng/9.1/                                                         r260647
releng/9.2/                                                         r260647
- ------------------------------------------------------------------------

To see which files were modified by a particular revision, run the
following command, replacing NNNNNN with the revision number, on a
machine with Subversion installed:

# svn diff -cNNNNNN --summarize svn://svn.freebsd.org/base

Or visit the following URL, replacing NNNNNN with the revision number:

<URL:https://svnweb.freebsd.org/base?view=revision&revision=NNNNNN>

VII. References

<URL:https://kb.isc.org/article/AA-01078>

<URL:http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-0591>

The latest revision of this advisory is available at
<URL:http://security.FreeBSD.org/advisories/FreeBSD-SA-14:04.bind.asc>
-----BEGIN PGP SIGNATURE-----

iQIcBAEBCgAGBQJS1ZTYAAoJEO1n7NZdz2rnOvQP/2/68/s9Cu35PmqNtSZVVxVG
ZSQP5EGWx/lramNf9566iKxOrLRMq/h3XWcC4goVd+gZFrvITJSVOWSa7ntDQ7TO
XcinfRZ/iyiJbs/Rg2wLHc/t5oVSyeouyccqODYFbOwOlk35JjOTMUG1YcX+Zasg
ax8RV+7Zt1QSBkMlOz/myBLXUjlTZ3Xg2FXVsfFQW5/g2CjuHpRSFx1bVNX6ysoG
9DT58EQcYxIS8WfkHRbbXKh9I1nSfZ7/Hky/kTafRdRMrjAgbqFgHkYTYsBZeav5
fYWKGQRJulYfeZQ90yMTvlpF42DjCC3uJYamJnwDIu8OhS1WRBI8fQfr9DRzmRua
OK3BK9hUiScDZOJB6OqeVzUTfe7MAA4/UwrDtTYQ+PqAenv1PK8DZqwXyxA9ThHb
zKO3OwuKOVHJnKvpOcr+eNwo7jbnHlis0oBksj/mrq2P9m2ueF9gzCiq5Ri5Syag
Wssb1HUoMGwqU0roS8+pRpNC8YgsWpsttvUWSZ8u6Vj/FLeHpiV3mYXPVMaKRhVm
067BA2uj4Th1JKtGleox+Em0R7OFbCc/9aWC67wiqI6KRyit9pYiF3npph+7D5Eq
7zPsUdDd+qc+UTiLp3liCRp5w6484wWdhZO6wRtmUgxGjNkxFoNnX8CitzF8AaqO
UWWemqWuz3lAZuORQ9KX
=OQzQ
-----END PGP SIGNATURE-----
```

Jeder Sicherheitshinweis verwendet das folgende Format:

- Jeder Sicherheitshinweis wird mit dem PGP-Schlüssel des Sicherheitsbeauftragten unterzeichnet. Der öffentliche Schlüssel des Sicherheitsbeauftragten kann in OpenPGP-Schlüssel überprüft werden.

- Der Name des Sicherheitshinweises beginnt immer mit `FreeBSD-SA-` (für FreeBSD Security Advisory), gefolgt vom Jahr im zweistelligen Format (`14:`), gefolgt von der Anzahl von Sicherheitshinweisen für dieses Jahr (`04.`), gefolgt vom Namen der Anwendung oder des betroffenen Subsystems (`bind`). Der hier gezeigte Sicherheitshinweis ist der vierte Hinweis für das Jahr 2014 und betrifft die Anwendung BIND.
- Das Feld `Topic` enthält eine Beschreibung der Schwachstelle.
- Das Feld `Category` beschreibt den betroffenen Systemteil. Mögliche Werte für dieses Feld sind `core`, `contrib` oder `ports`. Die Kategorie `core` gilt für Komponenten des FreeBSD-Betriebssystems, die Kategorie `contrib` beschreibt zum Basissystem gehörende Software Dritter, beispielsweise BIND. Die Kategorie `ports` beschreibt Software, die Teil der Ports-Sammlung ist.
- Das Feld `Module` beschreibt die betroffene Komponente. Im diesem Beispiel ist das `bind`-Modul betroffen, dass heißt dieses Problem betrifft eine Anwendung aus dem Betriebssystem.
- Das Feld `Announced` gibt den Zeitpunkt der Bekanntgabe des Sicherheitshinweises an. Das bedeutet, dass das Sicherheitsteam das Problem bestätigt hat und das eine entsprechende Korrektur bereits im FreeBSD Quellcode-Repository zur Verfügung steht .
- Das Feld `Credits` gibt die Person oder Organisation an, die das Sicherheitsproblem bemerkt und gemeldet hat.
- Das Feld `Affects` listet die FreeBSD-Releases auf, die von dem Problem betroffen sind.
- Das Feld `Corrected` zeigt an, wann das Problem in welchem Release behoben wurde. Der Teil in Klammern zeigt an, in welchem Zweig die Aktualisierung eingeflossen ist und die entsprechende Versionsnummer und Patch-Level des Release. Der Patch-Level besteht aus dem Buchstaben `p`, gefolgt von einer Nummer. Dies erlaubt es dem Benutzer festzustellen, welche Korrekturen bereits auf dem System eingespielt wurden.
- Reserviert für Informationen, über die auf cve.mitre.org nach Sicherheitslücken gesucht werden kann.
- Im Feld `Background` wird das betroffene Modul beschrieben.
- Im Feld `Problem Description` wird das Sicherheitsproblem beschrieben. Hier wird fehlerhafter Code beschrieben oder geschildert, wie ein Werkzeug ausgenutzt werden könnte.
- Das Feld `Impact` beschreibt die Auswirkungen des Sicherheitsproblems auf ein System.
- Im Feld `Workaround` wird eine Umgehung des Sicherheitsproblems beschrieben. Die Umgehung ist für Administratoren gedacht, die das System aus Zeitnot, Netzwerk-technischen oder anderen Gründen nicht aktualisieren können.
- Das Feld `Solution` enthält eine getestete Schritt-für-Schritt Anleitung, die das Sicherheitsproblem behebt.
- Das Feld `Correction Details` enthält die Subversion-Tags der betroffenen Dateien zusammen mit zugehörigen Revisionsnummern, in denen das Problem behoben wurde.
- Im Feld `References` finden sich Verweise auf weitere Informationsquellen.

## 31.12. Prozess-Überwachung

Prozess-Überwachung (Process accounting) ist ein Sicherheitsverfahren, bei dem ein Administrator verfolgt, welche Systemressourcen verwendet werden und wie sich diese auf die einzelnen

Anwender verteilen. Dadurch kann das System überwacht werden und es ist sogar möglich, zu kontrollieren, welche Befehle ein Anwender eingibt.

Die Überwachung von Prozessen hat sowohl Vor- als auch Nachteile. Positiv ist, dass man einen Einbruchsversuch bis an den Anfang zurückverfolgen kann. Von Nachteil ist allerdings, dass durch diesen Prozess Unmengen an Protokolldateien erzeugt werden, die auch dementsprechenden Plattenplatz benötigen. Dieser Abschnitt beschreibt die Grundlagen der Prozess-Überwachung.



Wenn Sie eine differenzierte Prozess-Überwachung benötigen, lesen Sie Security Event Auditing.

### 31.12.1. Die Prozess-Überwachung aktivieren und konfigurieren

Bevor Sie die Prozess-Überwachung verwenden können, müssen Sie diese über die folgenden Befehle aktivieren:

```
# sysrc accounting_enable=yes
# service accounting start
```

Die Informationen werden unterhalb von /var/account gespeichert. Das Verzeichnis wird beim ersten Start des Dienstes automatisch erstellt. Die Dateien enthalten sensible Informationen, einschließlich aller Befehle, die von allen Benutzern ausgeführt wurden. Der Schreibzugriff auf diese Dateien ist auf `root` beschränkt, der Lesezugriff auf `root` und Mitgliedern der Gruppe `wheel`. Um zu verhindern, dass die Mitglieder der Gruppe `wheel` die Dateien lesen können, ändern Sie den Modus des Verzeichnisses /var/account so, dass der Zugriff nur durch `root` möglich ist.

Einmal aktiviert, wird sofort mit der Überwachung von CPU-Statistiken, Befehlen und anderen Vorgängen begonnen. Protokolldateien werden in einem nur von Maschinen lesbaren Format gespeichert und können mit `sa` aufgerufen werden. Ohne Optionen gibt `sa` Informationen wie die Anzahl der Aufrufe pro Anwender, die abgelaufene Zeit in Minuten, die gesamte CPU- und Anwenderzeit in Minuten und die durchschnittliche Anzahl der Ein- und Ausgabeoperationen aus. sa(8) enthält eine Liste der Optionen, welche die Ausgabe steuern.

Benutzen Sie `lastcomm`, um die von den Benutzern ausgeführten Befehle anzuzeigen. Dieses Beispiel zeigt die Nutzung von `ls` durch `trhodes` auf dem Terminal `ttyp1`:

```
# lastcomm ls trhodes ttyp1
```

Zahlreiche weitere nützliche Optionen finden Sie lastcomm(1), acct(5) sowie sa(8).

## 31.13. Einschränkung von Ressourcen

FreeBSD bietet dem Systemadministrator mehrere Möglichkeiten die System-Ressourcen, die ein einzelner Benutzer verwenden kann, einzuschränken. Festplatten-Kontingente schränken den Plattenplatz, der einem Benutzer zur Verfügung steht, ein. Kontingente werden im Disk Quotas diskutiert.

Einschränkungen auf andere Ressourcen, wie CPU und Speicher, können über eine Konfigurationsdatei oder über die Kommandozeile konfiguriert werden. Traditionell werden Login-Klassen in /etc/login.conf definiert. Obwohl diese Methode immer noch untersützt wird, muss nach jeder Änderung an dieser Datei die Ressourcen-Datenbank neu gebaut werden. Zudem müssen Sie die notwendigen Änderungen in /etc/master.passwd vornehmen und die Passwort-Datenbnak neu bauen. Dieser Prozess kann, abhängig davon, wie viele Benutzer bearbeitet werden müssen, sehr zeitaufwändig sein.

Mit `rctl` Ressourcen für Benutzer sehr detailliert gesteuert werden. Dieser Befehl unterstützt nicht nur die Kontrolle der Ressourcen für Benutzer, sondern auch die Beschränkung auf Prozesse und Jails.

In diesem Abschnitt werden beide Methoden vorgestellt. Angefangen wird mit der traditionellen Methode.

### 31.13.1. Login-Klassen konfigurieren

Bei der traditionellen Methode werden Login-Klassen und Ressourcenbeschränkungen in /etc/login.conf definiert. Jeder Benutzer kann einer Login-Klasse zugewiesen werden (standardmäßig `default`) und jede Login-Klasse ist mit einem Satz von Login-Fähigkeiten verbunden. Eine Login-Fähigkeit ist ein `Name=Wert` Paar, in dem *Name* die Fähigkeit bezeichnet und *Wert* ein beliebiger Text ist, der in Abhänigkeit von *Name* entsprechend verarbeitet wird.



Immer wenn /etc/login.conf verändert wurde, muss die /etc/login.conf.db mit dem folgenden Kommando aktualisiert werden:

```
# cap_mkdb /etc/login.conf
```

Ressourcenbeschränkungen unterscheiden sich von normalen Login-Fähigkeiten zweifach. Erstens gibt es für jede Beschränkung ein aktuelles und ein maximales Limit. Das aktuelle Limit kann vom Benutzer oder einer Anwendung beliebig bis zum maximalen Limit verändert werden. Letzteres kann der Benutzer nur heruntersetzen. Zweitens gelten die meisten Ressourcenbeschränkungen für jeden vom Benutzer gestarteten Prozess.

Ressourcenbeschränkungen für Login-Klassen listet die gebräuchlichen Ressourcenbeschränkungen auf. Alle verfügbaren Ressourcenbeschränkungen und Fähigkeiten sind im Detail in login.conf(5) beschrieben.

*Tabelle 11. Ressourcenbeschränkungen für Login-Klassen*

| Ressourcenbeschränkung | Beschreibung |
|---|---|
| coredumpsize | Das Limit der Größe einer core-Datei, die von einem Programm generiert wird, unterliegt aus offensichtlichen Gründen anderen Limits der Festplattenbenutzung, zum Beispiel `filesize` oder Festplattenkontingenten. Es wird oft als weniger harte Methode zur Kontrolle des Festplattenplatz-Verbrauchs verwendet. Da Benutzer die core-Dateien selbst nicht erstellen und sie oft nicht löschen, kann diese Option davor schützen, dass kein Festplattenspeicher mehr zur Verfügung steht, sollte ein großes Programm abstürzen. |
| cputime | Die maximale Rechenzeit, die ein Prozess eines Benutzers verbrauchen darf. Überschreitet ein Prozess diesen Wert, wird er vom Kernel beendet. Beachten Sie, dass die Rechen*zeit* limitiert wird, nicht die prozentuale Prozessorenbenutzung, wie es in einigen Feldern von `top` und `ps` dargestellt wird. |
| filesize | Hiermit lässt sich die maximale Größe einer Datei bestimmen, die der Benutzer besitzen darf. Im Gegensatz zu Festplattenkontingenten ist diese Beschränkung nur für jede einzelne Datei gültig und nicht für den Platz, den alle Dateien eines Benutzers verwenden. |
| maxproc | Das ist die maximale Anzahl von Prozessen, die ein Benutzer starten darf, und beinhaltet sowohl Vordergrund- als auch Hintergrundprozesse. Dieser Wert nicht höher sein als das System-Limit, das in `kern.maxproc` angegeben ist. Vergessen Sie nicht, dass ein zu kleiner Wert den Benutzer in seiner Produktivität einschränken könnte, wenn beispielsweise ein großes Programm übersetzt wird oder viele Prozesse gestartet sind. |
| memorylocked | Dieses Limit gibt an, wie viel virtueller Speicher von einem Prozess maximal im Arbeitsspeicher festgesetzt werden kann (siehe auch mlock(2)). Ein paar systemkritische Programme, wie amd(8), verhindern damit einen Systemzusammenbruch, der auftreten könnte, wenn sie aus dem Speicher genommen werden. |
| memoryuse | Bezeichnet den maximalen Speicher, den ein Prozess benutzen darf und beinhaltet sowohl Arbeitsspeicher-, als auch Swap-Benutzung. Es ist kein allübergreifendes Limit für den Speicherverbrauch, aber ein guter Anfang. |
| openfiles | Mit diesem Limit lässt sich die maximale Anzahl der von einem Prozess des Benutzers geöffneten Dateien festlegen. In FreeBSD werden Dateien auch verwendet, um Sockets und >IPC>-Kanäle darzustellen. Setzen Sie es deshalb nicht zu niedrig. Das System-Limit ist in `kern.maxfiles` definiert. |
| sbsize | Dieses Limit beschränkt den Netzwerk-Speicher, den ein Benutzer verbrauchen darf. Es kann generell dazu benutzt werden Netzwerk-Verbindungen zu beschränken. |

| Ressourcenbeschränkung | Beschreibung |
|---|---|
| stacksize | Das ist die maximale Größe, auf die der Stack eines Prozesses heranwachsen darf. Das allein ist natürlich nicht genug, um den Speicher zu beschränken, den ein Programm verwenden darf. Es sollte deshalb in Verbindung mit anderen Limits verwendet werden. |

Beim Setzen von Ressourcenbeschränkungen sind noch andere Dinge zu beachten:

- Von /etc/rc beim Hochfahren des Systems gestartete Prozesse werden der `daemon` Login-Klasse zugewiesen.
- Obwohl die voreingestellte /etc/login.conf sinnvolle Limits enthält, sind sie evtl. nicht für jedes System geeignet. Ein zu hohes Limit kann das System für Missbrauch anfällig machen, und ein zu niedriges Limit kann der Produktivität schaden.
- Xorg beansprucht selbst eine Menge Ressourcen und verleitet die Benutzer dazu, mehrere Programme gleichzeitig laufen zu lassen.
- Bedenken Sie, dass viele Limits für einzelne Prozesse gelten und nicht für den Benutzer selbst. Setzt man zum Beispiel `openfiles` auf `50`, kann jeder Prozess des Benutzers bis zu `50` Dateien öffnen. Dadurch ist die maximale Anzahl von Dateien, die von einem Benutzer geöffnet werden können, `openfiles` mal `maxproc`. Das gilt auch für den Speicherverbrauch.

Weitere Informationen über Ressourcenbeschränkungen, Login-Klassen und -Fähigkeiten finden Sie in cap.mkdb(1), getrlimit(2) und login.conf(5).

### 31.13.2. Einschränkung von Ressourcen aktivieren und konfigurieren

Die Variable `kern.racct.enable` muss auf einen Wert ungleich Null eingestellt sein. Angepasste Kernel benötigen eine spezielle Konfiguration:

```
options           RACCT
options         RCTL
```

Sobald das System mit dem neuen Kernel gestartet wird, kann `rctl` benutzt werden, um die Regeln für das System festzulegen.

Die Syntax der Regeln wird durch *subject*, *subject-id*, *resource* und *action* gesteuert, wie in diesem Beispiel zu sehen ist:

```
user:trhodes:maxproc:deny=10/user
```

Diese Regel zeigt den grundlegenden Aufbau, hier mit dem Subjekt `user` und der Subjekt-ID `trhodes`. `maxproc` definiert die Anzahl der Prozesse. Die "Aktion" `deny` verhindert, dass neue Prozesse erstellt werden. Im vorherigen Beispiel wurde für den Benutzer `trhodes` eine Beschränkung von `10` Prozessen konfiguriert. Zu den weiteren Aktionen zählen beispielsweise die Protokollierung auf der Konsole, Benachrichtigungen an devd(8) oder das Senden eines `SIGTERM` an einen Prozess.

Beim hinzufügen von Regeln müssen einige Dinge beachtet werden. Das obige Beispiel würde den Benutzer sogar daran hindern, einfachste Dinge zu tun, nachdem er sich anmeldet und eine `screen` Sitzung gestartet hat. Sobald die Begrenzung für eine Ressource erreicht ist, wird folgende Fehlermeldung ausgegeben:

```
# man test
/usr/bin/man: Cannot fork: Resource temporarily unavailable
eval: Cannot fork: Resource temporarily unavailable
```

rctl(8) kann auch benutzt werden, um einer Jail eine Speichergrenze zuzuweisen. Eine solche Regel könnte wie folgt festgelegt werden:

```
# rctl -a jail:httpd:memoryuse:deny=2G/jail
```

Damit die Regeln auch nach einem Neustart erhalten bleiben, müssen sie in /etc/rctl.conf hinzugefügt werden. Dazu schreiben Sie einfach die Regel, ohne das vorhergehende Kommando. Zum Beispiel:

```
# Block jail from using more than 2G memory:
jail:httpd:memoryuse:deny=2G/jail
```

Mit `rctl` können auch Regeln entfernt werden:

```
# rctl -r user:trhodes:maxproc:deny=10/user
```

rctl(8) zeigt auch eine Möglichkeit, alle Regeln zu entfernen. Falls es erforderlich ist alle Regeln für einen einzelnen Benutzer zu entfernen, kann dieser Befehl verwendet werden:

```
# rctl -r user:trhodes
```

Es gibt noch viele weitere Ressourcen, die verwendet werden können, um zusätzliche `subjects` zu kontrollieren. Weitere Informationen zu diesem Thema finden Sie in rctl(8).

## 31.14. Gemeinsame Administration mit Sudo

Systemadministratoren benötigen häufig die Möglichkeit, Benutzern erweiterte Berechtigungen zu gewähren, damit diese privilegierte Aufgaben ausführen können. Die Idee, dass Teammitglieder einen Zugang zu einem FreeBSD-System zur Verfügung gestellt bekommen, um ihre spezifischen Aufgaben erledigen zu können, stellt den Administrator vor eine große Herausforderung. Diese Teammitglieder benötigen in der Regel nur einen eingeschränkten Zugang. Für manche Aufgaben werden jedoch die Rechte des Superusers benötigt. Zum Glück gibt es keinen Grund, diesen Mitgliedern einen solchen Zugang zu geben, da es Werkzeuge für genau diesen Anwendungsfall gibt.

Bislang wurde in diesem Kapitel immer versucht, den Zugriff für autorisierte Benutzer zu gewähren und den Zugriff für nicht autorisierte Benutzer zu verhindern. Ein weiteres Problem entsteht, sobald autorisierte Benutzer Zugriff auf die Ressourcen des Systems haben. In vielen Fällen benötigen einige Benutzer Zugriff auf Startskripte von Anwendungen. In anderen Fällen muss eine Gruppe von Administratoren das System verwalten. Traditionell wird der Zugriff über Benutzer, Gruppen, Dateiberechtigungen und manchmal sogar su(1) verwaltet. Und da immer mehr Anwendungen einen Zugriff brauchen und immer mehr Benutzer Zugriff auf die Systemressourcen benötigen, ist ein besserer Lösungsansatz erforderlich. Die am häufigsten verwendete Anwendung in solchen Fällen ist derzeit Sudo.

Sudo erlaubt dem Administrator eine rigide Konfiguration des Zugriffs auf bestimmte Kommandos und stellt einige erweiterte Protokollfunktionen zur Verfügung. Dieses Werkzeug kann als Port oder Paket security/sudo installiert werden. Das Paket wird wie folgt installiert:

```
# pkg install sudo
```

Nach der Installation können Sie `visudo` benutzen, um die Konfiguration in einem Texteditor zu öffnen. Es wird ausdrücklich `visudo` empfohlen, da dieses Programm die Syntax auf Fehler überprüft, bevor die Konfigurationsdatei gespeichert wird.

Die Konfigurationsdatei besteht aus mehreren kleinen Abschnitten, die eine umfangreiche Konfiguration ermöglichen. Im folgenden Beispiel soll der Webentwickler (`user1`) den Dienst *webservice* starten und stoppen dürfen. Um ihm dieses Recht zu gewähren, fügen Sie folgende Zeile an das Ende von /usr/local/etc/sudoers ein:

```
user1   ALL=(ALL)       /usr/sbin/service webservice *
```

Der Benutzer kann jetzt *webservice* über dieses Kommando starten:

```
% sudo /usr/sbin/service webservice start
```

Diese Konfiguration gestattet den Zugriff auf den webservice für einen einzelnen Benutzer. Jedoch ist in den meisten Organisationen ein ganzes Team für die Verwaltung eines solchen Dienstes verantwortlich. Mit einer weiteren Zeile ist es möglich, einer ganzen Gruppe diesen Zugriff zu geben. Die folgenden Schritte erstellen eine Gruppe mit den entsprechenden Benutzern. Der Gruppe wird es dann ermöglicht, diesen Dienst zu verwalten:

```
# pw groupadd -g 6001 -n webteam
```

Nun werden die Benutzer mit Hilfe von pw(8) in die Gruppe *webteam* hinzugefügt:

```
# pw groupmod -m user1 -n webteam
```

Zuletzt wird folgende Zeile in /usr/local/etc/sudoers hinzugefügt, damit jedes Mitglied von *webteam*

den Dienst *webservice* verwalten kann:

```
%webteam   ALL=(ALL)       /usr/sbin/service webservice *
```

Im Gegensatz zu su(1), benötigt Sudo nur das Passwort des Benutzers.

Benutzer, die mit Hilfe von Sudo Programme ausführen, müssen lediglich ihr eigenes Passwort eingeben. Dies ist sicherer und bietet eine bessere Kontrolle als su(1), wo der Benutzer das `root` -Passwort eingibt und damit alle Rechte von `root` erlangt.



Viele Organisationen haben bereits auf eine Zwei-Faktor-Authentifizierung umgestellt. In diesen Fällen hat der Benutzer möglicherweise gar kein Passwort, welches er eingeben könnte. Sudo bietet für solche Fälle die Variable `NOPASSWD`. Wenn die Variable in die obige Konfiguration hinzugefügt wird, dürfen die Mitglieder der Gruppe *webteam* den Dienst verwalten, ohne ein Passwort eingeben zu müssen:

```
%webteam   ALL=(ALL)       NOPASSWD: /usr/sbin/service webservice *
```

### 31.14.1. Protokollierung

Ein Vorteil von Sudo ist, dass Sitzungen protokolliert werden können. Mit den integrierten Protokollmechanismen und dem Befehl sudoreplay können alle über Sudo ausgelösten Befehle protokolliert und zu einem späteren Zeitpunkt überprüft werden. Um diese Funktion zu aktivieren, fügen Sie einen Eintrag für das Verzeichnis der Protokolle hinzu. Dieses Beispiel verwendet eine Benutzervariable. Weitere Informationen finden Sie in der Manualpage von sudoreplay.

```
Defaults iolog_dir=/var/log/sudo-io/%{user}
```



Dieses Verzeichnis wird automatisch nach der Konfiguration erstellt. Um auf der sicheren Seite zu sein, ist es am besten, das System die Verzeichnisse mit Standardberechtigungen erstellen zu lassen. Dieser Eintrag wird auch ein Protokoll für Administratoren erstellen, wenn diese den Befehl sudoreplay benutzen. Um dieses Verhalten zu ändern, kommentieren Sie die entsprechenden Zeilen in sudoers aus.

Nachdem dieser Eintrag in die Datei sudoers hinzugefügt wurde, kann die Konfiguration der Benutzer für die Protokollierung aktualisiert werden. In dem gezeigten Beispiel würde der aktualisierte Eintrag für das *webteam* zusätzlich folgende Änderung benötigen:

```
%webteam ALL=(ALL) NOPASSWD: LOG_INPUT: LOG_OUTPUT: /usr/sbin/service webservice *
```

Von nun an wird jede Änderung am *webservice* protokolliert, wenn sie von einem Mitglied der Gruppe *webteam* initiiert wurde. Eine Liste der Sitzungen kann wie folgt angezeigt werden:

```
# sudoreplay -l
```

Wenn Sie eine bestimmte Sitzung wiedergeben möchten, suchen Sie in der Ausgabe nach dem Eintrag `TSID=` und übergeben Sie den Wert ohne weitere Optionen an sudoreplay. Zum Beispiel:

```
# sudoreplay user1/00/00/02
```



Obwohl die Sitzungen protokolliert werden, kann ein böswilliger Administrator wahllos die Sitzungsprotokolle löschen. Daher ist es eine gute Idee, eine tägliche Kontrolle mit einem Intrusion Detection System (IDS) oder einer ähnlichen Software durchzuführen, so dass andere Administratoren auf manuelle Änderungen aufmerksam gemacht werden.

sudoreplay ist extrem erweiterbar. Lesen Sie die Dokumentation für weitere Informationen.

# Kapitel 32. Jails

## 32.1. Übersicht

Da die Systemadministration eine schwierige Aufgabe ist, wurden viele Werkzeuge entwickelt, die Administratoren bei der Installation, Konfiguration und Wartung ihrer Systeme unterstützen sollen. Eines dieser Werkzeuge, die verwendet werden können um die Sicherheit eines FreeBSD-Systems zu erhöhen, sind *Jails*. Jails sind seit FreeBSD 4.X verfügbar und werden ständig in ihrer Nützlichkeit, Leistung, Zuverlässigkeit und Sicherheit verbessert. Jails können als eine Art von Betriebssystem-Virtualisierung angesehen werden.

Jails setzen auf dem chroot(2)-Konzept auf, das dazu verwendet wird das root-Verzeichnis einer Reihe von Prozessen zu ändern, um so eine separate, sichere Umgebung zu schaffen. Prozesse, die in einer chroot-Umgebung erstellt wurden, können nicht auf Dateien oder Ressourcen zugreifen, die sich außerhalb dieser Umgebung befinden. Dadurch ist es einem kompromittierten Dienst nicht möglich, das gesamte System zu kompromittieren. Im Laufe der Zeit wurden viele Wege gefunden, um aus einer chroot-Umgebung auszubrechen, so dass es für die Sicherung von Diensten nicht die ideale Lösung ist.

Jails verbessern das traditionelle chroot-Konzept auf unterschiedlichste Art und Weise. In einer traditionellen chroot-Umgebung sind Prozesse auf den Bereich des Dateisystems beschränkt, auf den sie zugreifen können. Der Rest der Systemressourcen (wie zum Beispiel eine Reihe von Systembenutzern, die laufenden Prozesse oder das Netzwerk-Subsystem) teilen sich die chroot-Prozesse mit dem Host-System. Jails erweitern dieses Modell nicht nur auf die Virtualisierung des Zugriffs auf das Dateisystem, sondern auch auf eine Reihe von Benutzern und das Netzwerk-Subsystem. Zudem stehen weitere Möglichkeiten zur Verfügung, den Zugriff auf eine Jail-Umgebung zu kontrollieren.

Eine Jail zeichnet sich durch folgende Merkmale aus:

- Ein Unterverzeichnisbaum: dies ist der Ausgangspunkt der Jail. Einem Prozess, der innerhalb der Jail läuft, ist es nicht mehr möglich, aus diesem Unterverzeichnis auszubrechen.
- Ein Hostname: dieser Name wird für die Jail verwendet.
- Eine IP Adresse: diese Adresse wird der Jail zugewiesen. Die IP-Adresse einer Jails ist üblicherweise ein Adress-Alias auf eine existierende Netzwerkschnittstelle.
- Ein Kommando: der Pfad einer ausführbaren Datei, die innerhalb der Jail ausgeführt werden soll. Dieser Pfad wird relativ zum root-Verzeichnis der Jail-Umgebung angegeben.

Jails haben einen eigenen Satz von Benutzern und ihren eigenen `root`-Konto. Die Rechte dieser Benutzer sind nur auf die Jail-Umgebung beschränkt. Der Benutzer `root` der Jail-Umgebung ist nicht dazu berechtigt, kritische Operationen am System außerhalb der angebundenen Jail-Umgebung durchzuführen.

Dieses Kapitel bietet einen Überblick über die Terminologie und die Kommandos zur Verwaltung von FreeBSD Jails. Jails sind ein sehr mächtiges Werkzeug für Administratoren und fortgeschrittene Anwender.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- Wissen, was eine Jail ist und welche Verwendungszwecke es dafür unter FreeBSD gibt.
- Wissen, wie man eine Jail erstellt, startet und anhält.
- Die Grundlagen der Jail-Administration (sowohl innerhalb als auch außerhalb des Jails) kennen.



Jails sind ein mächtiges Werkzeug, aber sie sind kein Sicherheits-"Allheilmittel". Es ist wichtig zu beachten, dass es für einen Prozess in der Jail nicht möglich ist, von selbst auszubrechen. Es gibt jedoch Möglichkeiten, in denen ein unprivilegierter Benutzer außerhalb der Jail, mit einem privilegierten Benutzer innerhalb der Jail kooperiert, und somit erhöhte Rechte in der Host-Umgebung erlangt.

Den meisten dieser Angriffe kann vorgebeugt werden, indem sichergestellt wird, dass das Rootverzeichnis der Jail für unprivilegierte Benutzer der Host-Umgebung nicht zugänglich ist.

## 32.2. Jails - Definitionen

Um die für den Einsatz von Jails benötigten os;-Funktionen, deren Interna sowie die Art und Weise, mit der diese mit anderen Teilen des Betriebssystems interagieren, zu erläutern, werden in diesem Kapitel folgende Definitionen verwendet:

**chroot(8) (-Befehl)**

Ein Werkzeug, das den FreeBSD-Systemaufruf chroot(2) verwendet, um das Wurzelverzeichnis eines Prozesses und all seiner Nachkömmlinge zu ändern.

**chroot(2) (-Umgebung)**

Die Umgebung eines Prozesses, der in einem "chroot" läuft. Diese beinhaltet Ressourcen, wie zum Beispiel sichtbare Abschnitte des Dateisystems, verfügbare Benutzer- und Gruppenkennungen, Netzwerkschnittstellen und weitere IPC-Mechanismen und so weiter.

**jail(8) (-Befehl)**

Das Systemadministrationswerkzeug, welches es erlaubt, Prozesse innerhalb der Jail-Umgebung zu starten.

**Host (-Benutzer, -Prozess, -System)**

Das verwaltende System einer Jail-Umgebung. Das Host-System hat Zugriff auf alle verfügbaren Hardwareressourcen und kann sowohl innerhalb als auch außerhalb der Jail-Umgebung Prozesse steuern. Einer der wichtigsten Unterschiede des Host-System einer Jails ist, dass die Einschränkungen, welche für die Superuser-Prozesse innerhalb eines Jails gelten, nicht für die Prozesse des Host-Systems gelten.

**Gast (-Benutzer, -Prozess, -System)**

Ein Prozess, ein Benutzer oder eine andere Instanz, deren Zugriff durch eine FreeBSD-Jail eingeschränkt ist.

## 32.3. Einrichtung und Verwaltung von Jails

Einige Administratoren unterscheiden zwei verschiedene Jail-Arten: "Komplette" Jails, die ein echtes FreeBSD darstellen und Jails für einen bestimmten "Dienst", die nur einer bestimmten Anwendung oder einem Dienst (der möglicherweise mit besonderen Privilegien laufen soll) gewidmet sind. Dies ist aber nur eine konzeptuelle Unterscheidung, die Einrichtung einer Jail bleibt davon gänzlich unberührt. Bei der Erstellung einer kompletten Jail gibt es zwei Optionen für die Quelle des Userlands: vorkompilierte Binärpakete (im Lieferumfang der Installationsmedien enthalten) oder die Kompilierung aus dem Quelltext.

### 32.3.1. Eine Jail installieren

#### Eine Jail aus dem Internet installieren

Der Werkzeug bsdinstall(8) kann verwendet werden, um die für eine Jail benötigten Binärdateien zu holen und zu installieren. Dies geht durch die Auswahl eines Spiegelservers, welche Distributionen in das Zielverzeichnis installiert werden sollen, sowie die grundlegende Konfiguration einer Jail:

```
# bsdinstall jail /pfad/zur/jail
```

Nachdem der Befehl ausgeführt wurde, wird der Host für den Betrieb der Jail konfiguriert.

#### Eine Jail aus einer ISO-Datei installieren

Um das Basissystem von Installationsmedien zu installieren, erstellen Sie zunächst das Rootverzeichnis für die Jail. Dazu setzen Sie `DESTDIR` auf das entsprechende Verzeichnis.

Starten Sie eine Shell und legen Sie `DESTDIR` fest:

```
# sh
# export DESTDIR=/hier/ist/die/jail
```

Hängen Sie das Installationsmedium wie in mdconfig(8) beschrieben ein, wenn Sie von einem ISO-Abbild installieren:

```
# mount -t cd9660 /dev/`mdconfig -f cdimage.iso` /mnt
# cd /mnt/usr/freebsd-dist/
```

Extrahieren Sie die Binärdateien aus den Archiven des Installationsmediums in das entsprechende Verzeichnis. Es wird mindestens das "base"-Set benötigt, aber Sie können auch eine komplette Installation durchführen, wenn Sie dies bevorzugen.

Um lediglich das Basissystem zu installieren, führen Sie dieses Kommando aus:

```
# tar -xf base.txz -C $DESTDIR
```

Führen Sie folgendes Kommando aus, um alles außer den Kernel zu installieren:

```
# for set in base ports; do tar -xf $set.txz -C $DESTDIR ; done
```

#### Eine Jail aus den Quellen bauen und installieren

Die Manualpage jail(8) beschreibt die Erstellung einer Jail wie folgt:

```
# setenv D /hier/ist/die/jail
# mkdir -p $D      ①
# cd /usr/src
# make buildworld  ②
# make installworld DESTDIR=$D  ③
# make distribution DESTDIR=$D  ④
# mount -t devfs devfs $D/dev   ⑤
```

① Das Festlegen des Installationsorts für das Jail eignet sich am besten als Startpunkt. Hier wird sich die Jail innerhalb des Host-Dateisystems befinden. Eine gute Möglichkeit wäre etwa /usr/jail/name_der_jail, wobei `name_der_jail` den Hostname darstellt, über den die Jail identifiziert werden soll. /usr/ stellt normalerweise ausreichend Platz für eine Jail zur Verfügung. Bedenken Sie, dass eine "komplette" Jail ein Replikat einer jeden Datei der Standardinstallation des FreeBSD-Basissystems enthält.

② Wenn Sie bereits ihre Systemanwendungen mittels `make world` oder `make buildworld` neu erstellt haben, können Sie diesen Schritt überspringen und die Systemanwendungen in die neue Jail installieren.

③ Dieser Befehl wird den Verzeichnisbaum mit allen notwendigen Binärdateien, Bibliotheken, Manualpages usw. erstellen.

④ Der `distribution`-Befehl lässt make alle benötigten Konfigurationsdateien installieren, es werden also alle installierbaren Dateien aus /usr/src/etc/ in das Verzeichnis /etc der Jail installiert (also nach $D/etc/).

⑤ Das Einhängen des devfs(8)-Dateisystems innerhalb der Jail ist nicht unbedingt notwendig. Allerdings benötigt fast jede Anwendung Zugriff auf wenigstens ein Gerät. Es ist daher sehr wichtig, den Zugriff auf Devices aus der Jail heraus zu kontrollieren, da unsaubere Einstellungen es einem Angreifer erlauben könnten, in das System einzudringen. Die Kontrolle über devfs(8) erfolgt durch die in den Manualpages devfs(8) und devfs.conf(5) beschriebenen Regeln.

### 32.3.2. Den Host konfigurieren

Ist die Jail erst einmal erstellt, kann sie durch jail(8) gestartet werden. jail(8) benötigt zwingend mindestens vier Argumente, die in Übersicht des Handbuchs beschrieben sind. Weitere Argumente sind möglich, um beispielsweise die Jail mit den Berechtigungen eines bestimmten Benutzers laufen zu lassen. Das Argument `command` hängt vom Typ der Jail ab; für ein *virtuelles System* ist

/etc/rc eine gute Wahl, da dies dem Startvorgang eines echten FreeBSD-Systems entspricht. Bei einer *Service*-Jail hängt dieses von der Art des Dienstes ab, der in der Jail laufen soll.

Jails werden häufig mit dem Betriebssystem gestartet, da der rc-Mechanismus von FreeBSD dafür eine einfach zu realisierende Möglichkeit bietet.

- Konfigurieren Sie die Jail in /etc/jail.conf:

```
www {
    host.hostname = www.example.org;          # Hostname
    ip4.addr = 192.168.0.10;                  # IP address of the jail
    path = "/usr/jail/www";                   # Path to the jail
    devfs_ruleset = "www_ruleset";            # devfs ruleset
    mount.devfs;                              # Mount devfs inside the jail
    exec.start = "/bin/sh /etc/rc";           # Start command
    exec.stop = "/bin/sh /etc/rc.shutdown";   # Stop command
}
```

  Um die Jails mit dem Betriebssystem zu starten, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
jail_enable="YES"   # Set to NO to disable starting of any jails
```

  Beim Start einer in jail.conf(5) konfigurierten Jail wird das /etc/rc-Skript der Jail (das "annimmt", dass es sich in einem kompletten System befindet) aufgerufen. Für Service-Jails sollten die Startskripte der Jail durch das Setzen der Option `exec.start` entsprechend angepasst werden.



Eine vollständige Liste der Optionen findet sich in der Manualpage jail.conf(5).

service(8) kann zum manuellen Starten und Stoppen der Jail genutzt werden, wenn ein Eintrag in jail.conf angelegt wurde:

```
# service jail start www
# service jail stop www
```

Jails können mit jexec(8) heruntergefahren werden. Führen Sie zunächst jls(8) aus, um die `JID` der Jail ausfindig zu machen. Anschließend können Sie jexec(8) benutzen, um das Shutdown-Skript in der Jail auszuführen.

```
# jls
   JID  IP Address      Hostname                      Path
     3  192.168.0.10    www                           /usr/jail/www
# jexec 3 /etc/rc.shutdown
```

Weitere Informationen zu diesem Thema finden Sie in der Manualpage jail(8).

## 32.4. Feinabstimmung und Administration

Es gibt verschiedene Optionen, die für jede Jail gesetzt werden können und verschiedene Wege, ein FreeBSD-Host-System mit Jails zu kombinieren. Dieser Abschnitt zeigt Ihnen:

- Einige zur Verfügung stehende Optionen zur Abstimmung des Verhaltens und der Sicherheitseinstellungen, die mit einer Jail-Installation ausgeführt werden können.
- Einige der Anwendungsprogramme für das Jail-Management, die über die FreeBSD Ports-Sammlung verfügbar sind und genutzt werden können, um Jail-basierte Lösungen allumfassend umzusetzen.

### 32.4.1. Systemwerkzeuge zur Feinabstimmung von Jails in FreeBSD

Die Feinabstimmung einer Jail-Konfiguration erfolgt zum Großteil durch das Setzen von sysctl(8) -Variablen. Es gibt einen speziellen sysctl-Zweig, der als Basis für die Organisation aller relevanten Optionen dient: Die `security.jail.*`-Hierarchie der FreeBSD-Kerneloptionen. Die folgende Liste enthält alle jail-bezogenen sysctls (inklusiver ihrer Voreinstellungen). Die Namen sollten selbsterklärend sein, für weitergehende Informationen lesen Sie bitte die Manualpages jail(8) und sysctl(8).

- `security.jail.set_hostname_allowed: 1`
- `security.jail.socket_unixiproute_only: 1`
- `security.jail.sysvipc_allowed: 0`
- `security.jail.enforce_statfs: 2`
- `security.jail.allow_raw_sockets: 0`
- `security.jail.chflags_allowed: 0`
- `security.jail.jailed: 0`

Diese Variablen können vom Administrator des *Host-Systems* genutzt werden, um Beschränkungen hinzuzufügen oder aufzuheben, die dem Benutzer `root` als Vorgabe auferlegt sind. Beachten Sie, dass es einige Beschränkungen gibt, die nicht verändert werden können. Der Benutzer `root` darf innerhalb der jail(8) keine Dateisysteme mounten und unmounten. Ebenso ist es ihm untersagt, das devfs(8)-Regelwerk zu laden oder zu entladen. Er darf weder Firewallregeln setzen, noch administrative Aufgaben erledigen, die Modifikationen am Kernel selbst erfordern (wie beispielsweise das Setzen des `Securelevels` für den Kernel).

Das FreeBSD-Basissystem enthält einen Basissatz an Werkzeugen, um Informationen über aktive Jails zu erlangen und einer Jail administrative Befehle zuzuordnen. Die Befehle jls(8) und jexec(8) sind Teil des FreeBSD-Basissystems und können für folgende Aufgaben verwendet werden:

- Das Anzeigen einer Liste der aktiven Jails und ihrer zugehörigen Jail Identifier (JID), ihrer IP-Adresse, ihres Hostnames und ihres Pfades.
- Das Herstellen einer Verbindung mit einer laufenden Jail, das Starten eines Befehls aus dem Gastsystem heraus oder das Ausführen einer administrativen Aufgabe innerhalb der Jail selbst. Dies ist insbesondere dann nützlich, wenn der Benutzer `root` die Jail sauber herunterfahren möchte. jexec(8) kann auch zum Starten einer Shell innerhalb der Jail genutzt werden, um

administrative Aufgaben durchzuführen:

```
# jexec 1 tcsh
```

### 32.4.2. High-Level-Werkzeuge zur Jail-Administration in der FreeBSD Ports-Sammlung

Unter den zahlreichen Werkzeugen für die Administration von Jails ist sysutils/ezjail am vollständigsten und brauchbarsten. Dabei handelt es sich um eine Sammlung von Skripten, die das jail(8)-Management vereinfachen. Weitere Informationen zu diesem Werkzeug finden Sie im Abschnitt über ezjail.

### 32.4.3. Jails auf dem aktuellen Stand halten

Jails sollten immer vom Host-System auf dem neuesten Stand gehalten werden, da eine Aktualisierung aus einer Jail heraus wahrscheinlich fehlschlägt, da in der Voreinstellung von FreeBSD die Verwendung von chflags(1) in einem Jail nicht erlaubt ist und somit der Austausch einiger Dateien verhindert wird. Es ist zwar möglich, dieses Verhalten zu ändern, aber es wird empfohlen, freebsd-update(8) zu benutzen, um die Jails zu aktualisieren. Verwenden Sie `-b` mit dem Pfad der Jail, die Sie aktualisieren möchten.

Um die Jail auf das neueste Patch-Release der bereits installierten FreeBSD-Version zu aktualisieren, führen Sie auf dem Host die folgenden Befehle aus:

```
# freebsd-update -b /hier/ist/die/jail fetch
# freebsd-update -b /hier/ist/die/jail install
```

Um die Jail auf eine neue Haupt- oder Unterversion zu aktualisieren, wird zunächst eine Aktualisierung des Host-Systems durchgeführt, wie in "Aktualisierungen an Haupt- und Unterversionen" beschrieben. Nachdem der Host aktualisiert und neu gestartet wurde, kann die Jail aktualisiert werden. Führen Sie folgende Befehle auf dem Host aus, um von 12.0-RELEASE auf 12.1-RELEASE zu aktualisieren:

```
# freebsd-update -b /hier/ist/die/jail --currently-running 12.0-RELEASE -r 12.1-
RELEASE upgrade
# freebsd-update -b /hier/ist/die/jail install
# service jail restart myjail
# freebsd-update -b /hier/ist/die/jail install
```

Wenn es sich um eine Aktualisierung einer Hauptversion handelte, installieren Sie alle installierten Pakete neu und starten Sie die Jail erneut. Dies ist notwendig, da sich die ABI-Version bei einer Aktualisierung zwischen Hauptversionen von FreeBSD ändert. Führen Sie folgende Befehle auf dem Host-System aus:

```
# pkg -j mymail upgrade -f
```

```
# service jail restart myjail
```

## 32.5. Mehrere Jails aktualisieren

Die Verwaltung von mehreren Jails kann problematisch sein, da jede Jail bei jedem Upgrade komplett neu gebaut werden muss. Dieser Prozess kann sehr zeitaufwändig sein, wenn eine große Anzahl von Jails erstellt oder manuell aktualisiert werden müssen.

Dieser Abschnitt beschreibt eine Methode zur Lösung dieses Problems, indem so viel wie möglich zwischen Jails, auf sichere Art und Weise, durch den Einsatz von mount_nullfs(8)-Mounts geteilt wird. Dadurch werden Aktualisierungen erleichtert und das Verteilen von verschiedenen Diensten, wie HTTP, DNS und SMTP, auf verschiedene Jails wird attraktiver. Außerdem bietet dieses Verfahren einen einfachen Weg, Jails zu erstellen, zu entfernen und zu aktualisieren.



Es existieren auch einfachere Lösungen, wie zum Beispiel ezjail, das einfachere Methoden zur Administration von Jails verwendet und daher nicht so anspruchsvoll ist, wie der hier beschriebene Aufbau. ezjail wird in Verwaltung von Jails mit ezjail ausführlich behandelt.

Die Ziele des in diesem Abschnitt beschriebenen Aufbaus sind:

- Das Erstellen einer einfachen und gut verständlichen Jail Struktur, die es nicht erfordert für jede Jail ein vollständiges installworld laufen lassen zu müssen.
- Es einfach zu machen, neue Jails zu erstellen oder alte zu entfernen.
- Es einfach zu machen, bestehende Jails zu aktualisieren.
- Es einfach zu machen, einen angepassten FreeBSD-Zweig zu nutzen.
- Paranoid bezüglich Sicherheit zu sein und Angriffsmöglichkeiten weitgehend zu reduzieren.
- Soviel Platz und Inodes wie möglich einzusparen.

Dieses Design ist darauf angewiesen, dass eine read-only-Hauptvorlage in jede Jail hinein gemountet wird und dass jede Jail über wenigstens ein beschreibbares Gerät verfügt. Das Gerät kann hierbei eine separate physikalische Platte oder ein vnode unterstütztes Speichergerät sein. Im folgenden Beispiel wird ein read/write nullfs-Mount genutzt.

Das Layout des Dateisystems ist wie folgt:

- Die Jails befinden sich unterhalb der /home Partition.
- Jede Jail wird unterhalb des /home/j-Verzeichnisses gemountet.
- /home/j/mroot ist die Vorlage für jede Jail und die nur lesbare Partition für alle Jails.
- Unterhalb von /home/j wird für jede Jail ein leeres Verzeichnis angelegt.
- Jede Jail bekommt ein /s-Verzeichnis, das zum read/write-Teilbereich des Systems verlinkt wird.
- Jede Jail bekommt ihr eigenes read/write-System, das auf /home/j/skel basiert.
- Der read/write-Teilbereich jeder Jail wird in /home/js erstellt.

### 32.5.1. Erstellen der Vorlage

Dieser Abschnitt beschreibt die Schritte, die zum Erstellen der Hauptvorlage notwendig sind.

Es wird empfohlen, zunächst das FreeBSD Host-System nach den Anweisungen in "FreeBSD aus den Quellen aktualisieren" auf den aktuellen -RELEASE-Zweig zu aktualisieren. Darüber hinaus verwendet diese Vorlage sysutils/cpdup, sowie portsnap zum herunterladen der FreeBSD Ports-Sammlung.

1. Zuerst erstellen wir eine Verzeichnisstruktur für das read-only-Dateisystem, das die FreeBSD-Binärdateien für die Jails enthalten wird. Anschließend wechseln wir in den FreeBSD-Quellcodebaum und installieren das read-only-Dateisystem in die (Vorlage-)Jail.

```
# mkdir /home/j /home/j/mroot
# cd /usr/src
# make installworld DESTDIR=/home/j/mroot
```

2. Als nächstes bereiten wir die Ports-Sammlung für die Jails vor und kopieren den FreeBSD Quellcodebaum in die Jail, da dieser für mergemaster benötigt wird:

```
# cd /home/j/mroot
# mkdir usr/ports
# portsnap -p /home/j/mroot/usr/ports fetch extract
# cpdup /usr/src /home/j/mroot/usr/src
```

3. Danach wird die Struktur für den read/write-Bereich des Systems erstellt:

```
# mkdir /home/j/skel /home/j/skel/home /home/j/skel/usr-X11R6
/home/j/skel/distfiles
# mv etc /home/j/skel
# mv usr/local /home/j/skel/usr-local
# mv tmp /home/j/skel
# mv var /home/j/skel
# mv root /home/j/skel
```

4. Nutzen Sie mergemaster, um fehlende Konfigurationsdateien zu installieren. Anschließend werden die von mergemaster erstellten Extra-Verzeichnisse entfernt:

```
# mergemaster -t /home/j/skel/var/tmp/temproot -D /home/j/skel -i
# cd /home/j/skel
# rm -R bin boot lib libexec mnt proc rescue sbin sys usr dev
```

5. Nun wird das read/write-Dateisystem mit dem read-only-Dateisystem verlinkt. Vergewissern Sie sich, dass die symbolischen Links an den korrekten s/ Positionen erstellt werden, weil echte Verzeichnisse oder an falschen Positionen erstellte Verzeichnisse die Installation fehlschlagen lassen.

```
# cd /home/j/mroot
# mkdir s
# ln -s s/etc etc
# ln -s s/home home
# ln -s s/root root
# ln -s s/usr-local usr/local
# ln -s s/usr-X11R6 usr/X11R6
# ln -s s/distfiles usr/ports/distfiles
# ln -s s/tmp tmp
# ln -s s/var var
```

6. Zuletzt erstellen Sie eine allgemeine /home/j/skel/etc/make.conf mit folgendem Inhalt:

```
WRKDIRPREFIX?=  /s/portbuild
```

   Dies erlaubt es, die FreeBSD-Ports innerhalb jeder Jail zu kompilieren. Das Ports-Verzeichnis ist Teil des read-only System. Der angepasste Pfad des `WRKDIRPREFIX` macht es möglich, innerhalb des read/write-Bereichs der Jail Ports zu bauen.

### 32.5.2. Jails erstellen

Die Jailvorlage kann nun verwendet werden, um die Jails einzurichten und in /etc/rc.conf zu konfigurieren. In diesem Beispiel werden drei Jails erstellt: `NS`, `MAIL` und `WWW`.

1. Fügen Sie die folgenden Zeilen in /etc/fstab ein, damit die read-only-Vorlage und der read/write-Bereich für alle Jails verfügbar sind:

```
/home/j/mroot   /home/j/ns     nullfs  ro  0   0
/home/j/mroot   /home/j/mail   nullfs  ro  0   0
/home/j/mroot   /home/j/www    nullfs  ro  0   0
/home/js/ns     /home/j/ns/s   nullfs  rw  0   0
/home/js/mail   /home/j/mail/s nullfs  rw  0   0
/home/js/www    /home/j/www/s  nullfs  rw  0   0
```

   Um zu verhindern, dass fsck die nullfs-Mounts während des Bootens überprüft oder dass dump die Mounts sichert, müssen die letzten beiden Spalten auf `0` gesetzt werden.

2. Konfigurieren Sie die Jails in /etc/rc.conf:

```
jail_enable="YES"
jail_set_hostname_allow="NO"
jail_list="ns mail www"
jail_ns_hostname="ns.example.org"
jail_ns_ip="192.168.3.17"
jail_ns_rootdir="/usr/home/j/ns"
jail_ns_devfs_enable="YES"
```

```
jail_mail_hostname="mail.example.org"
jail_mail_ip="192.168.3.18"
jail_mail_rootdir="/usr/home/j/mail"
jail_mail_devfs_enable="YES"
jail_www_hostname="www.example.org"
jail_www_ip="62.123.43.14"
jail_www_rootdir="/usr/home/j/www"
jail_www_devfs_enable="YES"
```

Die Variable `jailnamerootdir` zeigt nach /usr/home statt nach /home, da der physikalische Pfad von /home unter FreeBSD /usr/home lautet. Die Variable `jailnamerootdir` darf im Pfad aber *keinen symbolischen Link* enthalten, weil die Jail ansonsten nicht gestartet werden kann.

3. Erstellen Sie die notwendigen Mountpunkte für die nur lesbaren Bereiche jeder Jail:

```
# mkdir /home/j/ns /home/j/mail /home/j/www
```

4. Installieren Sie mit sysutils/cpdup die read/write-Vorlage in jede Jail:

```
# mkdir /home/js
# cpdup /home/j/skel /home/js/ns
# cpdup /home/j/skel /home/js/mail
# cpdup /home/j/skel /home/js/www
```

5. An dieser Stelle werden die Jails erstellt und für den Betrieb vorbereitet. Mounten Sie zuerst die notwendigen Dateisysteme für jede Jail. Danach starten Sie die Jails:

```
# mount -a
# service jail start
```

Die Jails sollten nun laufen. Um zu prüfen, ob sie korrekt gestartet wurden, verwenden Sie `jls`. Die Ausgabe sollte ähnlich der folgenden sein:

```
# jls
   JID  IP Address      Hostname                      Path
     3  192.168.3.17    ns.example.org                /home/j/ns
     2  192.168.3.18    mail.example.org              /home/j/mail
     1  62.123.43.14    www.example.org               /home/j/www
```

An diesem Punkt sollte es möglich sein, sich an jeder Jail anzumelden, Benutzer anzulegen und Dienste zu konfigurieren. Die Spalte `JID` gibt die Jail-Identifikationsnummer jeder laufenden Jail an. Nutzen Sie den folgenden Befehl, um administrative Aufgaben in der Jail mit der `JID``3` durchzuführen:

```
# jexec 3 tcsh
```

### 32.5.3. Jails aktualisieren

Das Design dieses Aufbaus bietet einen einfachen Weg, bestehende Jails zu aktualisieren, während die Ausfallzeiten minimiert werden. Außerdem bietet es die Möglichkeit, zu älteren Versionen zurückzukehren, falls irgendwelche Probleme auftreten.

1. Im ersten Schritt wird das Host-System aktualisiert. Anschließend wird eine temporäre neue read-only Vorlage /home/j/mroot2 erstellt.

   ```
   # mkdir /home/j/mroot2
   # cd /usr/src
   # make installworld DESTDIR=/home/j/mroot2
   # cd /home/j/mroot2
   # cpdup /usr/src usr/src
   # mkdir s
   ```

   `installworld` erzeugt einige unnötige Verzeichnisse, die nun entfernt werden sollten:

   ```
   # chflags -R 0 var
   # rm -R etc var root usr/local tmp
   ```

2. Erzeugen Sie neue symbolische Links für das Hauptdateisystem:

   ```
   # ln -s s/etc etc
   # ln -s s/root root
   # ln -s s/home home
   # ln -s ../s/usr-local usr/local
   # ln -s ../s/usr-X11R6 usr/X11R6
   # ln -s s/tmp tmp
   # ln -s s/var var
   ```

3. Nun können die Jails gestoppt werden:

   ```
   # service jail stop
   ```

4. Hängen Sie die originalen Dateisysteme aus, da die read/write-Systeme an das read-only System (/s) angeschlossen sind:

   ```
   # umount /home/j/ns/s
   # umount /home/j/ns
   # umount /home/j/mail/s
   # umount /home/j/mail
   ```

```
# umount /home/j/www/s
# umount /home/j/www
```

5. Verschieben Sie das alte read-only-Dateisystem und ersetzen Sie es durch das neue Dateisystem. Das alte Dateisystem kann so als Backup dienen, falls etwas schief geht. Die Namensgebung entspricht hier derjenigen bei der Erstellung eines neuen read-only-Dateisystems. Verschieben Sie die originale FreeBSD Ports-Sammlung in das neue Dateisystem, um Platz und Inodes zu sparen:

```
# cd /home/j
# mv mroot mroot.20060601
# mv mroot2 mroot
# mv mroot.20060601/usr/ports mroot/usr
```

6. Nun ist die neue read-only-Vorlage fertig. Sie müssen daher nur noch die Dateisysteme erneut mounten und die Jails starten:

```
# mount -a
# service jail start
```

Nutzen Sie `jls` um zu prüfen, ob die Jails korrekt gestartet wurden. Führen Sie innerhalb jeder Jail `mergemaster` aus, damit die Konfigurationsdateien aktualisiert werden.

# 32.6. Verwaltung von Jails mit ezjail

Das Erstellen und Verwalten von mehreren Jails kann schnell zeitaufwändig und fehleranfällig werden. Dirk Engling's ezjail automatisiert und vereinfacht viele dieser Aufgaben. Als Vorlage wird ein *Basejail* erzeugt. Zusätzliche Jails nutzen mount_nullfs(8) um viele Verzeichnisse aus der Basejail zu teilen, ohne dabei zusätzlichen Speicherplatz zu belegen. Jedes weitere Jail benötigt daher nur wenige Megabyte an Speicherplatz, bevor die Anwendungen installiert werden.

Weitere Vorteile und Merkmale werden im Detail auf der Webseite von ezjail beschrieben: https://erdgeist.org/arts/software/ezjail/.

## 32.6.1. ezjail installieren

Für die Installation von ezjail wird zunächst eine Loopback-Schnittstelle für die Jails benötigt. Anschließend kann ezjail installiert und der dazugehörige Dienst aktiviert werden.

1. Damit der Verkehr auf der Loopback-Schnittstelle des Jails vom Host-System separiert ist, wird eine zweite Loopback-Schnittstelle in /etc/rc.conf erstellt:

```
cloned_interfaces="lo1"
```

   Die zusätzliche Schnittstelle `lo1` wird erstellt, wenn das System neu gestartet wird. Die

Schnittstelle kann auch ohne Neustart manuell erstellt werden:

```
# service netif cloneup
Created clone interfaces: lo1.
```

Jails können die Aliase dieser sekundären Schnittstelle verwenden, ohne dabei das Host-System zu stören.

Der Zugang zur Loopback-Adresse `127.0.0.1` wird an die erste IP-Adresse umgeleitet, die dem Jail zugewiesen ist. Damit die Loopback-Schnittstelle des Jails der neuen `lo1`-Schnittstelle zugeordnet werden kann, muss beim Erstellen der Jail diese Schnittstelle als erstes in der Liste der IP-Adressen angegeben werden.

Teilen Sie jedem Jail eine Loopback-Adresse aus dem Netzblock `127.0.0.0`/8` zu.

2. Installieren Sie sysutils/ezjail:

```
# cd /usr/ports/sysutils/ezjail
# make install clean
```

3. Aktivieren Sie ezjail, indem Sie folgende Zeile in /etc/rc.conf hinzufügen:

```
ezjail_enable="YES"
```

4. Der Dienst wird automatisch gestartet, wenn das System bootet. Er kann auch direkt für die aktuelle Sitzung gestartet werden:

```
# service ezjail start
```

### 32.6.2. Einrichtung

Nach erfolgreicher Installation von ezjail kann die Verzeichnisstruktur für die Basejail erstellt und befüllt werden. Dieser Schritt wird einmalig auf dem Host-System ausgeführt.

In diesen beiden Beispielen wird `-p` verwendet, um die Ports-Sammlung mit portsnap(8) in die Basejail herunterzuladen. Diese Kopie kann dann von allen Jails gemeinsam genutzt werden. Eine separate Kopie der Ports-Sammlung für die Jails ermöglicht die Isolierung der Ports vom Host-System. Die FAQ von ezjail erklärt dies im Detail: https://erdgeist.org/arts/software/ezjail/#FAQ.

1. Die Jail mit FreeBSD-RELEASE installieren

   Benutzen Sie `install`, wenn das FreeBSD-RELEASE für die Jail der Version auf dem Host-System entspricht. Wenn beispielsweise auf dem Host-System FreeBSD 10-STABLE installiert ist, wird in der Jail das neueste RELEASE von FreeBSD-10 installiert:

```
# ezjail-admin install -p
```

2. Die Jail mit `installworld` installieren

   Mit `ezjail-admin update` kann die Basejail mit den Binärdateien aus dem Host-System befüllt werden. Diese Dateien wurden auf dem Host-System mittels `buildworld` erzeugt.

   In diesem Beispiel wird FreeBSD 10-STABLE aus den Quellen gebaut. Die Verzeichnisse für die Jail wurden bereits erstellt. Anschließend wird `installworld` ausgeführt, das /usr/obj aus dem Host-System in die Basejail installiert.

   ```
   # ezjail-admin update -i -p
   ```

   In der Voreinstellung wird /usr/src des Host-Systems verwendet. Ein anderes Quellverzeichnis kann durch die Angabe von `-s`, oder durch Setzen der Variable `ezjail_sourcetree` in /usr/local/etc/ezjail.conf definiert werden.



Die Ports-Sammlung der Basejail wird mit den anderen Jails geteilt, jedoch werden die heruntergeladenen Distfiles im jeweiligen Jail gespeichert. In der Voreinstellung werden diese Dateien in /var/ports/distfiles der Jail gespeichert. Wenn die Ports gebaut werden, wird /var/ports im Jail als Arbeitsverzeichnis genutzt.



Zum herunterladen der Pakete, für die Installation in der Basejail, wird in der Voreinstellung das FTP-Protokoll verwendet. Firewalls und Proxies können jedoch bei der FTP-Übertragung Probleme verursachen. Das HTTP-Protokoll arbeitet anderes und vermeidet diese Probleme. Sie können eine URL für einen bestimmten Spiegel in /usr/local/etc/ezjail.conf eintragen:

```
ezjail_ftphost=http://ftp.FreeBSD.org
```

Im “FTP-Server” finden Sie eine Liste mit Spiegeln.

### 32.6.3. Eine neue Jail erstellen und starten

Neue Jails werden mit `ezjail-admin create` erstellt. In diesen Beispielen wird die `lo1` Loopback-Schnittstelle, wie oben beschrieben, verwendet.

*Procedure: Eine neue Jail erstellen und starten*

1. Geben Sie bei der Erstellung der Jail einen Namen und die verwendeten Loopback- und Netzwerk-Schnittstellen mit den IP-Adressen an. In diesem Beispiel trägt die Jail den Namen `dnsjail`.

   ```
   # ezjail-admin create dnsjail 'lo1|127.0.1.1,em0|192.168.1.50'
   ```



Die meisten Netzwerkdienste laufen problemlos in einer Jail. Ein paar wenige Netzwerkdienste, vor allem ping(8) verwenden Netzwerk-Sockets. Aus Sicherheitsgründen werden Netzwerk-Sockets innerhalb der Jails deaktiviert, so dass Dienste, die diese Sockets benötigten, nicht funktionieren werden. Gelegentlich benötigt ein Jail jedoch den Zugriff auf Raw-Sockets. Beispielsweise verwenden Netzwerk-Monitoring-Anwendungen ping(8), um die Verfügbarkeit von anderen Rechnern zu überprüfen. Sollten diese Sockets tatsächlich benötigt werden, können sie durch einen Eintrag in der Konfigurationsdatei von ezjail, /usr/local/etc/jailname, für einzelne Jails aktiviert werden. Bearbeiten Sie den Eintrag `parameters`:

```
export jail_jailname_parameters="allow.raw_sockets=1"
```

Aktivieren Sie keine Netzwerk-Sockets, solange die Dienste im Jail sie nicht tatsächlich benötigen.

2. Starten Sie die Jail:

```
# ezjail-admin start dnsjail
```

3. Starten Sie eine Konsole in der Jail:

```
# ezjail-admin console dnsjail
```

Die Jail ist jetzt in Betrieb und die zusätzliche Konfiguration kann nun abgeschlossen werden. Typische Einstellungen an dieser Stelle sind:

1. Das `root`-Passwort setzen

   Verbinden Sie sich mit der Jail und setzen Sie das Passwort für den Benutzer `root`:

```
# ezjail-admin console dnsjail
# passwd
Changing local password for root
New Password:
Retype New Password:
```

2. Konfiguration der Zeitzone

   Die Zeitzone kann innerhalb der Jail mit tzsetup(8) gesetzt werden. Um störende Fehlermeldungen zu vermeiden, kann der Eintrag adjkerntz(8) in /etc/crontab auskommentiert werden. Dieser Job versucht die Uhr des Rechners zu aktualisieren, was jedoch in einem Jail fehlschlägt, da die Jail nicht auf diese Hardware zugreifen darf.

3. DNS-Server

Tragen Sie die Zeilen für die Nameserver der Domäne in /etc/resolv.conf ein, damit die Namensauflösung in der Jail funktioniert.

4. /etc/hosts anpassen

   Ändern Sie die Adresse und fügen Sie den Namen der Jail zu den `localhost`-Einträgen in /etc/hosts hinzu.

5. /etc/rc.conf konfigurieren

   Tragen Sie Konfigurationseinstellungen in /etc/rc.conf ein. Der Rechnername und die IP-Adresse werden nicht eingestellt, da diese Werte bereits durch die Jail-Konfiguration zur Verfügung gestellt werden.

Nach der Konfiguration der Jail können die Anwendungen, für die die Jail erstellt wurde, installiert werden.



Einige Ports müssen mit speziellen Optionen gebaut werden, damit sie in der Jail verwendet werden können. Zum Beispiel haben die Netzwerk-Monitoring-Pakete net-mgmt/nagios-plugins und net-mgmt/monitoring-plugins eine Option `JAIL`, die aktiviert werden muss, damit diese Werkzeuge innerhalb einer Jail funktionieren.

### 32.6.4. Jails aktualisieren

#### Das Betriebssystem aktualisieren

Da das Basissystem der Basejail von den anderen Jails gemeinsam genutzt wird, werden bei einem Update der Basejail automatisch alle anderen Jails aktualisiert. Die Aktualisierung kann entweder über den Quellcode oder über binäre Updates erfolgen.

Um das Basissystem auf dem Host-System zu bauen und in der Basejail zu installieren, geben Sie folgendes ein:

```
# ezjail-admin update -b
```

Wenn das Basissystem bereits auf dem Host-System gebaut wurde, kann es in der Basejail installiert werden:

```
# ezjail-admin update -i
```

Binär-Updates verwenden freebsd-update(8). Das Update unterliegt dabei den gleichen Einschränkungen, als wenn freebsd-update(8) direkt ausgeführt würde. Vor allem stehen mit dieser Methode nur -RELEASE Versionen von FreeBSD zur Verfügung.

Aktualisieren Sie die Basejail auf die neueste FreeBSD-Version des Host-Systems. Zum Beispiel von RELEASE-p1 auf RELEASE-p2.

```
# ezjail-admin update -u
```

Damit das Basejail aktualisiert werden kann, muss zunächst das Host-System, wie in "Aktualisierungen an Haupt- und Unterversionen" beschrieben, aktualisiert werden. Sobald das Host-System aktualisiert und neu gestartet wurde, kann die Basejail aktualisiert werden. Da freebsd-update(8) keine Möglichkeit besitzt, die derzeit installierte Version der Basejail zu bestimmen, muss die ursprüngliche Version beim Aufruf mit angegeben werden. Benutzen Sie file(1) um die ursprüngliche Version der Basejail zu bestimmen:

```
# file /usr/jails/basejail/bin/sh
/usr/jails/basejail/bin/sh: ELF 64-bit LSB executable, x86-64, version 1 (FreeBSD),
dynamically linked (uses shared libs), for FreeBSD 9.3, stripped
```

Nutzen Sie diese Information, um die Aktualisierung von `9.3-RELEASE` auf die aktuelle Version des Host-Systems durchzuführen:

```
# ezjail-admin update -U -s 9.3-RELEASE
```

Nachdem die Basejail aktualisiert ist, muss in jeder Jail mergemaster(8) ausgeführt werden, um die Konfigurationsdateien zu aktualisieren.

Wie mergemaster(8) verwendet wird, hängt stark vom Zweck und Vertrauenswürdigkeit der Jail ab. Wenn die Dienste oder Benutzer nicht vertrauenswürdig sind, dann sollte mergemaster(8) nur innerhalb der Jail ausgeführt werden:

*Beispiel 33. mergemaster(8) in einer nicht vertrauenswürdigen Jail ausführen*

Entfernen Sie die Verknüpfung von /usr/src des Jails zur Basejail und erstellen Sie ein neues /usr/src als Mountpunkt für die Jail. Hängen Sie /usr/src vom Host-System schreibgeschützt in den Mountpunkt für die Jail ein:

```
# rm /usr/jails/jailname/usr/src
# mkdir /usr/jails/jailname/usr/src
# mount -t nullfs -o ro /usr/src /usr/jails/jailname/usr/src
```

Öffnen Sie eine Konsole in der Jail:

```
# ezjail-admin console jailname
```

Innerhalb der Jail führen Sie dann mergemaster(8) aus. Danach verlassen Sie die Konsole:

```
# cd /usr/src
# mergemaster -U
```

```
# exit
```

Abschließend können Sie /usr/src aus der Jail aushängen:

```
# umount /usr/jails/jailname/usr/src
```

*Beispiel 34. mergemaster(8) in einer vertrauenswürdigen Jail ausführen*

Wenn den Benutzern und den Diensten in der Jail vertraut wird, kann mergemaster(8) auf dem Host-System ausgeführt werden:

```
# mergemaster -U -D /usr/jails/jailname
```



Nach einem größeren Versionsupdate empfiehlt sysutils/ezjail, sicherzustellen, dass `pkg` die richtige Version hat. Geben Sie dazu den folgenden Befehl ein, um auf die entsprechende Version zu aktualisieren:

```
# pkg-static upgrade -f pkg
```

**Ports aktualisieren**

Die Ports-Sammlung der Basejail wird von den anderen Jails gemeinsam genutzt. Somit genügt es, die Ports-Sammlung in der Basejail zu aktualisieren.

Die Ports-Sammlung der Basejail wird mit portsnap(8) aktualisiert:

```
# ezjail-admin update -P
```

## 32.6.5. Jails verwalten

**Jails starten und stoppen**

ezjail startet automatisch alle Jails, wenn das System hochfährt. Jails können auch manuell mit `stop` und `start` gestoppt und neu gestartet werden:

```
# ezjail-admin stop sambajail
Stopping jails: sambajail
```

In der Voreinstellung werden die Jails automatisch gestartet, wenn das Host-System hochfährt. Der automatische Start kann mit `config` deaktiviert werden:

```
# ezjail-admin config -r norun seldomjail
```

Diese Einstellung wird nach einem Neustart des Host-Systems aktiviert. Eine Jail, die bereits läuft, wird hiermit nicht gestoppt.

Der automatische Start kann auch aktiviert werden:

```
# ezjail-admin config -r run oftenjail
```

#### Jails archivieren und wiederherstellen

Benutzen Sie `archive` um ein .tar.gz-Archiv einer Jail zu erstellen. Der Dateiname wird aus dem Namen der Jail und dem aktuellen Datum zusammengesetzt. Archivdateien werden in /usr/jails/ezjail_archives abgelegt. Ein alternatives Verzeichnis für die Ablage kann in der Variable `ezjail_archivedir` der Konfigurationsdatei definiert werden.

Die Archivdatei kann an anderer Stelle als Sicherung gespeichert werden, oder eine andere Jail kann daraus mit `restore` wiederhergestellt werden. Eine neue Jail kann auch aus dem Archiv erstellt werden, was eine bequeme Möglichkeit bietet, bestehende Jails zu klonen.

Die Jail `wwwserver` stoppen und archivieren:

```
# ezjail-admin stop wwwserver
Stopping jails: wwwserver.
# ezjail-admin archive wwwserver
# ls /usr/jails/ezjail-archives/
wwwserver-201407271153.13.tar.gz
```

Erstellen Sie aus dem eben erzeugten Archiv eine neue Jail namens `wwwserver-clone`. Verwenden Sie die Schnittstelle em1 und weisen Sie eine neue IP-Adresse zu, um einen Konflikt mit dem Original zu vermeiden:

```
# ezjail-admin create -a /usr/jails/ezjail_archives/wwwserver-201407271153.13.tar.gz
wwwserver-clone 'lo1|127.0.3.1,em1|192.168.1.51'
```

### 32.6.6. Vollständiges Beispiel: BIND in einer Jail

Einen BINDDNS-Server innerhalb einer Jail zu betreiben erhöht die Sicherheit, da der Dienst isoliert wird. Dieses Beispiel erstellt einen einfachen caching-only Nameserver.

- Die Jail bekommt den Namen `dns1`.
- Die Jail erhält die IP-Adresse `192.168.1.240` auf der Schnittstelle `re0` des Host-Systems.
- Die Upstream-DNS-Server des ISPs lauten `10.0.0.62` und `10.0.0.61`.
- Die Basejail wurde bereits erstellt und die Ports-Sammlung installiert, wie in Einrichtung

beschrieben.

*Beispiel 35. BIND in einer Jail laufen lassen*

Erstellen Sie eine geklonte Loopback-Schnittstelle durch einen Eintrag in /etc/rc.conf:

```
cloned_interfaces="lo1"
```

Erzeugen Sie jetzt die Loopback-Schnittstelle:

```
# service netif cloneup
Created clone interface: lo1
```

Erstellen Sie die Jail:

```
# ezjail-admin create dns1 'lo1|127.0.2.1,re0|192.168.1.240'
```

Starten Sie die Jail, verbinden Sie sich mit der Konsole und führen Sie die grundlegende Konfiguration durch:

```
# ezjail-admin start dns1
# ezjail-admin console dns1
# passwd
Changing local password for root
New Password:
Retype New Password:
# tzsetup
# sed -i .bak -e '/adjkerntz/ s/^/#/' /etc/crontab
# sed -i .bak -e 's/127.0.0.1/127.0.2.1/g; s/localhost.my.domain/dns1.my.domain
dns1/' /etc/hosts
```

Setzen Sie vorübergehend die Upstream-DNS-Server in /etc/resolv.conf, damit die Ports-Sammlung heruntergeladen werden kann:

```
nameserver 10.0.0.62
nameserver 10.0.0.62
```

Immer noch in der Konsole der Jail, installieren Sie dns/bind99.

```
# make -C /usr/ports/dns/bind99 install clean
```

Konfigurieren Sie den Nameserver in /usr/local/etc/namedb/named.conf.

Erstellen Sie eine Zugriffskontrollliste (ACL) der Adressen und Netzwerke, die DNS-Anfragen

an diesen Nameserver senden dürfen. Diese Sektion wird vor der Sektion `options` hinzugefügt, die sich bereits in der Datei befindet:

```
...
// or cause huge amounts of useless Internet traffic.

acl "trusted" {
    192.168.1.0/24;
    localhost;
    localnets;
};

options {
...
```

Verwenden Sie die IP-Adresse der Jail in der Direktive `listen-on`, um DNS-Anfragen von anderen Rechnern aus dem Netzwerk zu akzeptieren:

```
    listen-on   { 192.168.1.240; };
```

Entfernen Sie die Kommentarzeichen `/ und/`. Tragen Sie die IP-Adressen der Upstream-DNS-Server ein. Unmittelbar nach der Sektion `forwarders` fügen Sie Verweise auf die bereits definierten ACLs ein:

```
    forwarders {
        10.0.0.62;
        10.0.0.61;
    };

    allow-query       { any; };
    allow-recursion   { trusted; };
    allow-query-cache { trusted; };
```

Aktivieren Sie den Dienst in /etc/rc.conf:

```
named_enable="YES"
```

Starten und testen Sie den Nameserver:

```
# service named start
wrote key file "/usr/local/etc/namedb/rndc.key"
Starting named.
# /usr/local/bin/dig @192.168.1.240 freebsd.org
```

Beinhaltet die Antwort

```
;; Got answer;
```

dann funktioniert der Nameserver. Eine längere Verzögerung, gefolgt von der Antwort

```
;; connection timed out; no servers could be reached
```

weist auf ein Problem hin. Überprüfen Sie die Konfigurationseinstellungen und stellen Sie sicher, dass alle lokalen Firewalls den DNS-Zugriff auf die Upstream-DNS-Server erlauben.

Wie auch jeder andere lokale Rechner, kann der DNS-Server Anfragen für Namensauflösung an sich selbst stellen. Tragen Sie die Adresse des DNS-Servers in die /etc/resolv.conf der Client-Rechner:

```
nameserver 192.168.1.240
```

Ein lokaler DHCP-Server kann die Adresse eines lokalen DNS-Servers automatisch für alle DHCP-Clients zur Verfügung stellen.

# Kapitel 33. Verbindliche Zugriffskontrolle

## 33.1. Übersicht

In FreeBSD 5.X wurden neue Sicherheits-Erweiterungen verfügbar, die aus dem TrustedBSD-Projekt übernommen wurden und auf dem Entwurf POSIX®.1e basieren. Die beiden bedeutendsten neuen Sicherheits-Mechanismen sind Berechtigungslisten (Access Control Lists, ACL) und die verbindliche Zugriffskontrolle (Mandatory Access Control, MAC). Durch die MAC können Module geladen werden, die neue Sicherheitsrichtlinien bereitstellen. Mit Hilfe einiger Module kann beispielsweise ein eng umgrenzter Bereich des Betriebssystems gesichert werden, indem die Sicherheitsfunktionen spezieller Dienste unterstützt bzw. verstärkt werden. Andere Module wiederum betreffen in ihrer Funktion das gesamte System - alle vorhandenen Subjekte und Objekte. Das "Verbindliche" in der Namensgebung erwächst aus dem Fakt, dass die Kontrolle allein Administratoren und dem System obliegt und nicht dem Ermessen der Nutzer, wie es mit Hilfe der benutzerbestimmbaren Zugriffskontrolle (Discrectionary Access Control / DAC), dem Zugriffstandard für Dateien, gar der System V IPC in FreeBSD, normalerweise umgesetzt wird.

Dieses Kapitel wird sich auf die Grundstruktur der Verbindlichen Zugriffskontrolle und eine Auswahl der Module, die verschiedenste Sicherheitsfunktionen zur Verfügung stellen, konzentrieren.

Beim Durcharbeiten dieses Kapitels erfahren Sie:

- Welche MAC Module für Sicherheitsrichtlinien derzeit in FreeBSD eingebettet sind und wie die entsprechenden Mechanismen funktionieren.
- Was die einzelnen MAC Module an Funktionen realisieren und auch, was der Unterschied zwischen einer Richtlinie, die *mit* Labels arbeitet, und einer, die *ohne* Labels arbeitet, ist.
- Wie Sie die MAC in ein System einbetten und effizient einrichten.
- Wie die verschiedenen Richtlinienmodule einer MAC konfiguriert werden.
- Wie mit einer MAC und den gezeigten Beispielen eine sicherere Umgebung erstellt werden kann.
- Wie die Konfiguration einer MAC auf korrekte Einrichtung getestet wird.

Vor dem Lesen dieses Kapitels sollten Sie bereits:

- Grundzüge von UNIX® und FreeBSD verstanden haben. (Grundlagen des FreeBSD Betriebssystems).
- Mit den Grundzügen der Kernelkonfiguration und -kompilierung vertraut sein (Konfiguration des FreeBSD-Kernels).
- Einige Vorkenntnisse über Sicherheitskonzepte im Allgemeinen und deren Umsetzung in FreeBSD im Besonderen mitbringen (Sicherheit)).



Der unsachgemäße Gebrauch der in diesem Kapitel enthaltenen Informationen kann den Verlust des Systemzugriffs, Ärger mit Nutzern oder die Unfähigkeit, grundlegende Funktionen des X-Windows-Systems zu nutzen, verursachen.

> Wichtiger noch ist, dass man sich nicht allein auf die MAC verlassen sollte, um ein System zu sichern. Die MAC verbessert und ergänzt lediglich die schon existierenden Sicherheits-Richtlinien - ohne eine gründliche und fundierte Sicherheitspraxis und regelmäßige Sicherheitsprüfungen wird Ihr System nie vollständig sicher sein.
>
> Außerdem sollte angemerkt werden, dass die Beispiele in diesem Kapitel auch genau dasselbe sein sollen, nämlich Beispiele. Es wird nicht empfohlen, diese bestimmten Beispiele auf einem Arbeitssystem umzusetzen. Das Einarbeiten der verschiedenen Sicherheitsmodule erfordert eine Menge Denkarbeit und viele Tests. Jemand, der nicht versteht, wie diese Module funktionieren, kann sich schnell darin wiederfinden, dass er (oder sie) das ganze System durchforsten und viele Dateien und Verzeichnisse neu konfigurieren muß.

### 33.1.1. Was in diesem Kapitel nicht behandelt wird

Dieses Kapitel behandelt einen großen Teil sicherheitsrelevanter Themen, bezogen auf die Verbindliche Zugriffskontrolle (MAC). Die gegenwärtige Entwicklung neuer MAC Module ist nicht abgedeckt. Einige weitere Module, die im MAC Framework enthalten sind, haben besondere Charakteristika, die zum Testen und Entwickeln neuer Module gedacht sind. Dies sind unter anderem mac_test(4), mac_stub(4) und mac_none(4). Für weitere Informationen zu diesen Modulen und den entsprechend angebotenen Funktionen lesen Sie bitte die Manpages.

## 33.2. Schlüsselbegriffe

Bevor Sie weiterlesen, müssen noch einige Schlüsselbegriffe geklärt werden. Dadurch soll jegliche auftretende Verwirrung von vornherein beseitigt und die plötzliche Einführung neuer Begriffe und Informationen vermieden werden.

- *Verbund*: Ein Verbund ist ist ein Satz von Programmen und Daten, die speziell und zusammen abgeschottet wurden, um Nutzern Zugriff auf diese ausgewiesenen Systembereiche zu gewähren. Man kann sagen, ein solcher Verbund ist eine Gruppierung, ähnlich einer Arbeitsgruppe, einer Abteilung, einem Projekt oder einem Thema. Durch die Nutzung von Verbünden (*compartments*) kann man Sicherheitsrichtlinien erstellen, die alles notwendige Wissen und alle Werkzeuge zusammenfassen.
- *Hochwassermarkierung*: Eine solche Richtlinie erlaubt die Erhöhung der Sicherheitsstufe in Abhängigkeit der Klassifikation der gesuchten bzw. bereitzustellenden Information. Normalerweise wird nach Abschluss des Prozesses die ursprüngliche Sicherheitsstufe wieder hergestellt. Derzeit enthält die MAC Grundstruktur keine Möglichkeit, eine solche Richtlinie umzusetzen, der Vollständigkeit halber ist die Definition hier jedoch aufgeführt.
- *Integrität*: Das Schlüsselkonzept zur Klassifizierung der Vertraulichkeit von Daten nennt man Integrität. Je weiter die Integrität erhöht wird, umso mehr kann man den entsprechenden Daten vertrauen.
- *Label*: Ein Label ist ein Sicherheitsmerkmal, welches mit Dateien, Verzeichnissen oder anderen Elementen im System verbunden wird. Man sollte es wie einen Vertraulichkeitsstempel auffassen, der Dateien angehört wie beispielsweise die Zugriffszeit, das Erstellungsdatum oder

auch der Name; sobald Dateien derart gekennzeichnet werden, bezeichnen diese Label die sicherheitsrelevanten Eigenschaften. Zugriff ist nur noch dann möglich, wenn das zugreifende Subjekt eine korrespondierende Kennzeichnung trägt. Die Bedeutung und Verarbeitung der Label-Werte ist von der Einrichtung der Richtlinie abhängig: Während einige Richtlinien das Label zum Kennzeichnen der Vertraulichkeit oder Geheimhaltungsstufe eines Objekts nutzen, können andere Richtlinien an derselben Stelle Zugriffsregeln festschreiben.

- *Level*: Eine erhöhte oder verminderte Einstellung eines Sicherheitsmerkmals. Wenn das Level erhöht wird, wird auch die ensprechende Sicherheitsstufe angehoben.
- *Niedrigwassermarkierung*: Eine solche Richtlinie erlaubt das Herabstufen des Sicherheitslevels, um weniger sensible Daten verfügbar zu machen. In die meisten Fällen wird das ursprüngliche Sicherheitslevel des Nutzers wiederhergestellt, sobald der Vorgang abgeschlossen ist. Das einzige Modul in FreeBSD, welches von dieser Richtlinie Gebrauch macht, ist mac_lomac(4).
- *Multilabel*: Die Eigenschaft `multilabel` ist eine Dateisystemoption, die entweder im Einzelbenutzermodus mit Hilfe des Werkzeugs tunefs(8), während des Bootvorgangs in der Datei fstab(5) oder aber beim Erstellen einen neues Dateisystems aktiviert werden kann. Diese Option erlaubt einem Administrator, verschiedenen Objekten unterschiedliche Labels zuzuordnen - kann jedoch nur zusammen mit Modulen angewendet werden, die auch tatsächlich mit Labels arbeiten.
- *Objekt*: Ein Objekt oder auch Systemobjekt ist theoretisch eine Einheit, durch welche Information fließt, und zwar unter der Lenkung eines *Subjektes*. Praktisch schliesst diese Definition Verzeichnisse, Dateien, Felder, Bildschirme, Tastaturen, Speicher, Bandlaufwerke, Drucker und jegliche anderen Datenspeicher- oder -verarbeitungsgeräte ein. Im Prinzip ist ein Objekt ein Datenkontainer oder eine Systemressource - Zugriff auf ein *Objekt* bedeutet, auf Daten zuzugreifen.
- *Richtlinie*: Eine Sammlung von Regeln, die definiert, wie Zielvorgaben umgesetzt werden, nennt man Richtlinie. Eine *Richtlinie* dokumentiert normalerweise, wie mit bestimmten Elementen umgegangen wird. Dieses Kapitel faßt den Begriff in diesem Kontext als *Sicherheitsrichtlinie* auf; als eine Sammlung von Regeln, die den Fluß von Daten und Informationen kontrolliert und die gleichzeitig definiert, wer auf diese Daten und Informationen zugreifen darf.
- *Anfälligkeit*: Dieser Begriff wird normalerweise verwendet, wenn man über MLS (Multi Level Security) spricht. Das Anfälligkeits-Level beschreibt, wie wichtig oder geheim die Daten sein sollen. Um so höher das Anfälligkeits-Level, um so wichtiger die Geheimhaltung bzw. Vertraulichkeit der Daten.
- *Einzel-Label*: Von einem Einzel-Label spricht man, wenn für ein ganzes Dateisystem lediglich ein einziges Label verwendet wird, um Zugriffskontrolle über den gesamten Datenfluss zu erzwingen. Sobald diese Option verwendet wird - und das ist zu jeder Zeit, wenn die Option `multilabel` nicht explizit gesetzt wurde - sind alle Dateien und Verzeichnisse mit dem gleichen Label gekennzeichnet.
- *Subjekt*: Ein Subjekt ist jedwede Einheit, die Information in Fluss zwischen Objekten bringt: Zum Beispiel ein Nutzer, ein Nutzerprozessor, ein Systemprozeß usw. In FreeBSD handelt es sich meistens um einen Thread, der als Prozeß im Namen eines Nutzers arbeitet.

## 33.3. Erläuterung

Mit all diesen neuen Begriffen im Kopf können wir nun überlegen, wie die Möglichkeiten der verbindlichen Zugriffskontrolle (MAC) die Sicherheit eines Betriebssystems als Ganzes erweitern. Die verschiedenen Module, die durch die MAC bereitgestellt werden, können verwendet werden, um das Netzwerk oder Dateisysteme zu schützen, Nutzern den Zugang zu bestimmten Ports oder Sockets zu verbieten und vieles mehr. Die vielleicht beste Weise, die Module zu verwenden, ist, sie miteinander zu kombinieren, indem mehrere Sicherheitsrichtlinienmodule gleichzeitig eine mehrschichtige Sicherheitsumgebung schaffen. Das ist etwas anderes als singuläre Richtlinien wie zum Beispiel die Firewall, die typischerweise Elemente eines Systems stabilisiert, das nur für einen speziellen Zweck verwendet wird. Der Verwaltungsmehraufwand ist jedoch von Nachteil, zum Beispiel durch die Verwendung von mehreren Labels oder dem eigenhändigen Erlauben von Netzwerkzugriffen für jeden einzelnen Nutzer.

Solche Nachteile sind allerdings gering im Vergleich zum bleibenden Effekt der erstellten Struktur. Die Möglichkeit zum Beispiel, für konkrete Anwendungen genau die passenden Richtlinien auszuwählen und einzurichten, senkt gleichzeitig die Arbeitskosten. Wenn man unnötige Richtlinien aussortiert, kann man die Gesamtleistung des Systems genauso steigern wie auch eine höhere Anpassungsfähigkeit gewährleisten. Eine gute Umsetzung der MAC beinhaltet eine Prüfung der gesamten Sicherheitsanforderungen und einen wirksamen Einsatz der verschiedenen Module.

Ein System, auf dem eine MAC verwendet wird, muß zumindest garantieren, dass einem Nutzer nicht gestattet wird, Sicherheitsmerkmale nach eigenem Ermessen zu verändern; dass Arbeitswerkzeuge, Programme und Skripte, innerhalb der Beschränkungen arbeiten können, welche die Zugriffsregeln der ausgewählten Module dem System auferlegen; und dass die volle Kontrolle über die Regeln der MAC beim Administrator ist und bleibt.

Es ist die einsame Pflicht des zuständigen Administrators, die richtigen Module sorgfältig auszuwählen. Einige Umgebungen könnten eine Beschränkung der Zugriffe über die Netzwerkschnittstellen benötigen - hier wären die Module mac_portacl(4), mac_ifoff(4) und sogar mac_biba(4) ein guter Anfang. In anderen Fällen muß man sehr strenge Vertraulichkeit von Dateisystemobjekten gewährleisten - dafür könnte man mac_bsdextended(4) oder mac_mls(4) einsetzen.

Die Entscheidung, welche Richtlinien angewandt werden, kann auch anhand der Netzwerk-Konfiguration getroffen werden. Nur bestimmten Benutzern soll erlaubt werden, via ssh(1) auf das Netzwerk oder Internet zuzugreifen - mac_portacl(4) wäre eine gute Wahl. Aber für was entscheidet man sich im Falle eines Dateisystems? Soll der Zugriff auf bestimmte Verzeichnisse von spezifischen Nutzern oder Nutzergruppen separiert werden? Oder wollen wir den Zugriff durch Nutzer oder Programme auf spezielle Dateien einschränken, indem wir gewisse Objekte als geheim einstufen?

Der Zugriff auf Objekte kann einigen vertraulichen Nutzern gestattet werden, anderen wiederum verwehrt. Als Beispiel sei hierzu ein großes Entwicklerteam angeführt, das in kleine Gruppen von Mitarbeitern aufgeteilt wurde. Die Entwickler von Projekt A dürfen nicht auf Objekte zugreifen, die von den Entwicklern von Projekt B geschrieben wurden. Sie müssen aber trotzdem auf Objekte zugreifen können, die von einem dritten Entwicklerteam geschaffen wurden - alles in allem eine verzwickte Situation. Wenn man die verschiedenen Module der MAC richtig verwendet, können Anwender in solche Gruppen getrennt und ihnen der Zugriff zu den gewünschten Systemobjekten

gestattet werden - ohne Angst haben zu müssen, dass Informationen in die falschen Hände geraten.

So hat jedes Modul, das eine Sicherheitsrichtlinie verfügbar macht, einen eigenen Weg, die Sicherheit des Systems zu verstärken. Die Auswahl der Module sollte auf einem gut durchdachten Sicherheitskonzept gründen. In vielen Fällen muß das gesamte Konzept eines Systems überarbeitet und neu eingepflegt werden. Ein guter Überblick über die Möglichkeiten der verschiedenen von der MAC angebotenen Module hilft einem Administrator, die besten Richtlinien für seine spezielle Situation auszuwählen.

Im FreeBSD-Standardkernel ist die Option zur Verwendung der MAC nicht enthalten. Daher muß die Zeile

```
options     MAC
```

der Kernelkonfiguration hinzugefügt und der Kernel neu übersetzt und installiert werden.



Verschiedenen Anleitungen für die MAC empfehlen, die einzelnen Module direkt in den Kernel einzuarbeiten. Dabei ist es jedoch möglich, das System aus dem Netzwerk auszusperren oder gar schlimmeres. Die Arbeit mit der MAC ist ähnlich der Arbeit mit einer Firewall - man muß, wenn man sich nicht selbst aus dem System aussperren will, genau aufpassen. Man sollte sich eine Möglichkeit zurechtlegen, wie man eine Implementation einer MAC rückgängig machen kann - genauso wie eine Ferninstallation über das Netzwerk nur mit äußerster Vorsicht vorgenommen werden sollte. Es wird daher empfohlen, die Module nicht in den Kernel einzubinden, sondern sie beim Systemstart via /boot/loader.conf zu laden.

## 33.4. MAC Labels verstehen

MAC Label sind Sicherheitsmerkmale, die, wenn sie zum Einsatz kommen, allen Subjekten und Objekten im System zugeordnet werden.

Wenn ein Administrator ein solches Merkmal bzw. Attribut setzen will, muß er/sie verstehen können, was da genau passiert. Die Attribute, die im speziellen Fall zu vergeben sind, hängen vom geladenen Modul und den darin jeweils implementierten Richtlinien ab. Jedes dieser Richtlinienmodule setzt die Arbeit mit seinen entsprechenden Attributen in individueller Weise um. Falls der Nutzer nicht versteht, was er da konfiguriert, oder auch, was seine Konfiguration für Begleiterscheinungen mit sich bringt, ergibt sich meist als Resultat ein unerwartetes, ja sogar unerwünschtes Verhalten des gesamten Systems.

Ein Label, einem Objekt verliehen, wird verwendet, um anhand einer Richtlinie eine sicherheitsrelevante Entscheidung über Zugriffsrechte zu fällen. In einigen Richtlinien enthält bereits das Label selbst alle dafür nötigen Informationen. Andere Richtlinien verwenden diese Informationen, um zunächst ein komplexes Regelwerk abzuarbeiten.

Wenn man zum Beispiel einer Datei das Attribut `biba/low` zuordnet, wird dieses durch das Biba Sicherheitsrichtlinienmodul, und zwar mit dem Wert "low", verarbeitet.

Einige der Richtlinienmodule, die die Möglichkeit zum Vergeben von Labels unter FreeBSD

unterstützen, bieten drei vordefinierte Labels an. Dieses nennen sich "high", "low" und "equal". Obwohl die verschiedenen Module die Zugriffskontrolle auf verschiedene Weisen regeln, kann man sich sicher sein, das das "low"-Label der untersten, unsichersten Einstellung entspricht, das "equal"-Label die Verwendung des Moduls für das jeweilige Objekt oder Subjekt deaktiviert - und das "high"-Label die höchstmögliche Einstellung erzwingt. Im Speziellen gilt diese Aussage für die Richtlinien(-module) MLS und Biba.

In den meisten Umgebungen, sogenannten Single Label Environments, wird Objekten nur ein einzelnes Label zugewiesen. Dadurch wird nur ein Regelsatz für die Zugriffskontrolle auf das gesamte System verwendet - und das ist meistens auch tatsächlich ausreichend. Es gibt wenige Fälle, in denen mehrere Labels auf Dateisystemobjekte oder -subjekte verwendet werden. In einem solchen Fall muß das Dateisystem mit der tunefs(8)-Option `multilabel` angepaßt werden, da `single label` die Standardeinstellung ist.

Bei der Verwendung von Biba oder MLS kann man numerische Labels vergeben, die genau das Level angeben, an welcher Stelle in der Hierarchie das Subjekt oder Objekt einzuordnen ist. Dieses numerische Level wird verwendet, um Informationen in verschiedene Gruppen aufzuteilen oder zu sortieren - damit zum Beispiel nur Subjekte, die zu einer gewissen Vertraulichkeitsstufe gehören, Zugang zu einer Gruppe von Objekten erhalten.

In den meisten Fällen wird ein Administrator nur ein einzelnes Label für das gesamte Dateisystem verwenden.

*Moment mal, dass ist doch dasselbe wie DAC! Ich dachte, MAC würde die Kontrolle strengstens an den Administrator binden!* Diese Aussage hält immer noch stand - `root` ist derjenige, der die Kontrolle ausübt und die Richtlinie konfiguriert, so dass Nutzer in die entsprechenden, angemessenen Kategorien / Zugriffsklassen eingeordnet werden. Nunja, einige Module schränken `root` selbst ein. Die Kontrolle über Objekte wird dann einer Gruppe zugewiesen, jedoch hat `root` die Möglichkeit, die Einstellungen jederzeit zu widerrufen oder zu ändern. Dies ist das Hierarchie/Freigabe-Modell, das durch Richtlinien wie MLS oder Biba bereitgestellt wird.

### 33.4.1. Konfigurieren der Labels

Gewissermaßen alle Aspekte der Labelkonfiguration werden durch Werkzeuge das Basissystems umgesetzt. Die entsprechenden Kommandos bieten eine einfache Schnittstelle zum Konfigurieren, Manipulieren und auch Verifizieren der gekennzeichneten Objekte.

Mit den beiden Kommandos setfmac(8) und setpmac(8) kann man eigentlich schon alles machen. Das Kommando `setfmac` wird verwendet, um ein MAC-Label auf einem Systemobjekt zu setzen, `setpmac` hingegen zum Setzen von Labels auf Systemsubjekte. Als Beispiel soll hier dienen:

```
# setfmac biba/high test
```

Wenn bei der Ausführung dieses Kommandos keine Fehler aufgetreten sind, gelangt man zur Eingabeaufforderung zurück. Nur wenn ein Fehler auftritt, verhalten sich diese Kommandos nicht still, ganz wie auch die Kommandos chmod(1) und chown(8). In einigen Fällen wird dieser Fehler `Permission denied` lauten und gewöhnlich dann auftreten, wenn ein Label an einem Objekt angebracht oder verändert werden soll, das bereits (Zugriffs-)Beschränkungen unterliegt. Der

Systemadministrator kann so eine Situation mit Hilfe der folgenden Kommandos überwinden:

```
# setfmac biba/high test
Permission denied
# setpmac biba/low setfmac biba/high test
# getfmac test
test: biba/high
```

Wie wir hier sehen, kann `setpmac` verwendet werden, um die vorhandene Einstellungen zu umgehen, indem dem gestarteten Prozeß ein anderes, valides Label zugeordnet wird. Das Werkzeug `getpmac` wird normalerweise auf gerade laufende Prozesse angewendet. Ähnlich sendmail: Als Argument wird statt eines Kommandos eine eine Prozeß-ID übergeben, es verbirgt sich doch dieselbe Logik dahinter. Wenn ein Nutzer versucht, eine Datei zu verändern, auf die er keinen Zugriff hat, entsprechend der Regeln eines geladenen Richtlinienmoduls, wird der Fehler `Operation not permitted` durch die Funktion `mac_set_link` angezeigt.

### Übliche Typen von Labeln

Wenn man die Module mac_biba(4), mac_mls(4) und mac_lomac(4) verwendet, hat man die Möglichkeit, einfache Label zu vergeben. Diese nennen sich `high`, `low` und `equal`. Es folgt eine kurze Beschreibung, was diese Labels bedeuten:

- Das Label `low` ist definitionsgemäß das niedrigeste Label, das einem Objekt oder Subjekt verliehen werden kann. Wird es gesetzt, kann die entsprechende Entität nicht mehr auf Entitäten zugreifen, die das Label `high` tragen.
- Das Label `equal` wird Entitäten verliehen, die von der Richtlinie ausgenommen sein sollen.
- Das Label `high` verleiht einer Entität die höchstmögliche Einstellung.

Unter Beachtung jedes einzelnen Richtlinienmoduls moduliert und beschränkt jede dieser Einstellungen den Informationsfluß unterschiedlich. Genaue Erklärungen zu den Charakteristika der einfachen Labels in den verschiedenen Modulen finden sich im entsprechenden Unterabschnitt dieses Kapitels oder in den Manpages.

#### Fortgeschrittene Label-Konfiguration

Numerische klassifizierte Labels werden verwendet in der Form `Klasse:Verbund+Verbund`. Demnach ist das Label

```
biba/10:2+3+6(5:2+3-15:2+3+4+5+6)
```

folgendermaßen zu lesen:

"Biba Policy Label"/"effektive Klasse 10" :"Verbund 2,3 und 6": ("Low-Klasse 5:..."- "High-Klasse 15:...")

In diesem Beispiel ist die erstgenannte Klasse als "effektive Klasse" zu bezeichnen. Ihr werden die "effektiven Verbünde" zugeordnet. Die zweite Klasse ist die "Low"-Klasse und die letzte die "high"-

Klasse. Die allermeisten Konfigurationen kommen ohne die Verwendungen von solchen Klassen aus, nichtsdestotrotz kann man sie für erweiterte Konfigurationen verwenden.

Sobald sie auf *Systemsubjekte* angewendet werden, haben diese eine gegenwärtige Klasse/Verbund-Konfiguration und diese muß im definierten Rahmen gegebenenfalls angepaßt (erhöht oder gesenkt) werden. Im Gegensatz dazu haben *Systemobjekte* alle eingestellten (effektive, High- und Low-Klasse) gleichzeitig. Dies ist notwendig, damit auf Sie von den *Systemsubjekten* in den verschiedenen Klassen gleichzeitig zugegriffen werden kann.

Die Klasse und und die Verbünde in einem Subjekt-Objekt-Paar werden zum Erstellen einer sogenannten Dominanz-Relation verwendet, in welcher entweder das Subjekt das Objekt, das Objekt das Subjekt, keines das andere dominiert oder sich beide gegenseitig dominieren. Der Fall, dass sich beide dominieren, tritt dann ein, wenn die beiden Labels gleich sind. Wegen der Natur des Informationsflusses in Biba kann man einem Nutzer Rechte für einen Reihe von Abteilungen zuordnen, die zum Beispiel mit entsprechenden Projekten korrespondieren. Genauso können aber auch Objekten mehrere Abteilungen zugeordnet sein. Die Nutzer müssen eventuell ihre gegenwärtigen Rechte mithilfe von `su` or `setpmac` anpassen um auf Objekte in einer Abteilung zuzugreifen, zu der sie laut ihrer effektiven Klasse nicht berechtigt sind.

**Nutzer- und Label-Einstellungen**

Nutzer selbst brauchen Labels damit ihre Dateien und Prozesse korrekt mit der Sicherheitsrichtlinie zusammenarbeitet, die für das System definiert wurde. Diese werden in der Datei login.conf durch die Verwendung von Login- Klassen zugeordnet. Jedes Richtlinienmodul, das Label verwendet, arbeitet mit diesen Login-Klassen.

Beispielhaft wird der folgende Eintrag, der für jede Richtlinie eine Einstellung enthält, gezeigt:

```
default:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin:\
:manpath=/usr/shared/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
```

```
:label=partition/13,mls/5,biba/10(5-15),lomac/10[2]:
```

Die Label-Option in der letzten Zeile legt fest, welches Standard-Label für einen Nutzer erzwungen wird. Nutzern darf niemals gestattet werden, diese Werte selbst zu verändern, demnach haben Nutzer in dieser Beziehung auch keine Wahlfreiheit. In einer richtigen Konfiguration jedoch wird kein Administrator alle Richtlinienmodule aktivieren wollen. Es wird an dieser Stelle ausdrücklich empfohlen, dieses Kapitel zu Ende zu lesen, bevor irgendein Teil dieser Konfiguration ausprobiert wird.



Nutzer können ihr eigenes Label nach dem Loginvorgang durchaus ändern. Jedoch kann diese Änderung nur unter den Auflagen der gerade gültigen Richtlinie geschehen. Im Beispiel oben wird für die Biba-Richtlinie eine minimale Prozeßintegrität von 5, eine maximale von 15 angegeben, aber die Voreinstellung des tatsächlichen Labels ist 10. Der Nutzerprozeß läuft also mit einer Integrität von 10 bis das Label verändert wird, zum Beispiel durch eine Anwendung des Kommandos `setpmac`, welches jedoch auf den Bereich eingeschränkt wird, der zum Zeitpunkt des Logins angegeben wurde, in diesem Fall von 5 bis 15.

Nach einer Änderung der Datei login.conf muß in jedem Fall die Befähigungsdatenbank mit dem Kommando `cap_mkdb` neu erstellt werden - und das gilt für alle im weiteren Verlauf gezeigten Beispiele und Diskussionspunkte.

Es ist nützlich anzumerken, dass viele Einsatzorte eine große Anzahl von Nutzern haben, die wiederum viele verschiedenen Nutzerklassen angehören sollen. Hier ist eine Menge Planungsarbeit notwendig, da die Verwaltung sehr unübersichtlich und schwierig ist.

#### Netzwerkschnittstellen und die zugehörigen Label

Labels können auch, wenn man sie an Netzwerkschittstellen vergibt, helfen, den Datenfluß durch das Netzwerk zu kontrollieren. Das funktioniert in allen Fällen genau so wie mit Objekten. Nutzer, die in der Biba-Richtlinie das Label `high` tragen, dürfen nicht auf Schnittstellen zugreifen, die `low` markiert sind usw.

Die Option `maclabel` wird via `ifconfig` übergeben. Zum Beispiel

```
# ifconfig bge0 maclabel biba/equal
```

belegt die Schnittstelle mit dem MAC Label `biba/equal`. Wenn eine komplexe Einstellung wie `biba/high(low-high)` verwendet wird, muß das gesamte Label in Anführungszeichen geschrieben werden, da sonst eine Fehlermeldung zurückgegeben wird.

Jedes Richtlinienmodul, das die Vergabe von Labels unterstützt, stellt einen Parameter bereit, mit dem das MAC Label für Netzwerkschnittstellen deaktiviert werden kann. Das Label der Netzwerkschnittstelle auf `equal` zu setzen, führt zum selben Ergebnis. Beachten Sie die Ausgabe von `sysctl`, die Manpages der verschiedenen Richtlinien oder eben die Informationen, die im weiteren Verlauf dieses Kapitels angeboten werden, um mehr zu diesen Parametern zu erfahren.

### 33.4.2. Single- oder Multilabel?

Als Standardeinstellung verwendet das System die Option `single label`. Was bedeutet das für den Administrator? Es gibt einige Unterschiede zwischen `single label` und `multilabel`. In ihrer ureigenen Weise bieten beide Vor- und Nachteile bezogen auf die Flexibilität bei der Modellierung der Systemsicherheit.

Die Option `single label` gibt jedem Subjekt oder Objekt genau ein einziges Label, zum Beispiel `biba/high`. Mit dieser Option hat man einen geringeren Verwaltungsaufwand, aber die Flexibilität beim Einsatzes von Richtlinien ist ebenso gering. Viele Administratoren wählen daher auch die Option `multilabel` im Sicherheitsmodell, wenn die Umstände es erfordern.

Die Option `multilabel` gestattet, jedem einzelnen Subjekt oder Objekt seine eigenen unabhängigen Label zu zuzuordnen. Die Optionen `multilabel` und `singlelabel` betreffen jedoch nur die Richtlinien, die Labels als Leistungsmerkmal verwenden, einschließlich der Richtlinien Biba, Lomac, MLS und SEBSD.

Wenn Richtlinien benutzt werden sollen, die ohne Labels auskommen, wird die Option `multilabel` nicht benötigt. Dies betrifft die Richtlinien `seeotheruids`, `portacl` und `partition`.

Man sollte sich dessen bewußt sein, dass die Verwendung der Option `multilabel` auf einer Partition und die Erstellung eines Sicherheitsmodells auf der Basis der FreeBSD `multilevel` Funktionalität einen hohen Verwaltungsaufwand bedeutet, da alles im Dateisystem ein Label bekommt. Jedes Verzeichnis, jede Datei und genauso jede Schnittstelle.

Das folgende Kommando aktiviert `multilabel` für ein Dateisystem. Dies funktioniert nur im Einzelbenutzermodus:

```
# tunefs -l enable /
```

In einer Swap-Partition wird dies nicht benötigt.



Falls Sie Probleme beim Setzen der Option `multilabel` auf der Root-Partition bemerken, lesen Sie bitte Fehler im MAC beheben dieses Kapitels.

## 33.5. Planung eines Sicherheitsmodells

Wann immer eine neue Technologie eingepflegt werden soll, ist es wichtig, vorher einen Plan zu erstellen. In den verschiedenen Etappen der Planung sollte der Administrator nie das "Große Ganze" aus den Augen verlieren und mindestens die folgenden Punkte beachten:

- Die Anforderungen
- Die Ziele

Wenn Sie MAC verwenden möchten, sind das im Besonderen folgende Punkte:

- Wie werden Informationen und Ressourcen auf den Zielsystemen klassifiziert?

- Welche Arten von Informationen bzw. Ressourcen sollen im Zugang beschränkt sein und welche Art Einschränkung soll verwendet werden?
- Welche(s) MAC Modul(e) wählt man, um sein Ziel zu erreichen?

Es ist immer möglich, die Einstellungen des Systems und der Systemressourcen im Nachhinein zu "optimieren". Es ist aber wirklich lästig, das gesamte Dateisystem zu durchsuchen, um Dateien oder Benutzerkonten zu reparieren. Eine gute Planung hilft dem Administrator, sich einer sorgenfreien und effizienten Umsetzung eines Sicherheitsmodells zu versichern. Testlauf des Sicherheitsmodells *vor* dem Einsatz in seiner richtigen Arbeitsumgebung ist auf jeden Fall empfehlenswert. Die Idee, ein System mit einer MAC einfach loslaufen zu lassen, ist wie direkt auf einen Fehlschlag hinzuarbeiten.

Jede Umgebung hat ihre eigenen Anforderungen. Ein tiefgreifendes und vollständiges Sicherheitsprofil zu erstellen spart weitere Änderungen, nachdem das System in Betrieb genommen wurde. Also werden die folgenden Abschnitte die verschiedenen Module vorstellen, die den Administratoren zur Verfügung gestellt werden, die Nutzung und Konfiguration der einzelnen Module beschreiben; und in einigen Fällen Einblicke gewähren, für welche Situationen welche Module besonders geeignet sind. Zum Beispiel ein Webserver kann von der Verwendung der mac_biba(4) oder der mac_bsdextended(4) Richtlinie profitieren. In anderen Fällen, an einem Rechner mit nur wenigen lokalen Benutzern, ist die mac_partition(4) die Richtlinie der Wahl.

## 33.6. Modulkonfiguration

Jedes Modul, das in der MAC enthalten ist, kann entweder direkt in den Kernel eingefügt werden oder als Kernelmodul in der Laufzeit des Systems geladen werden. Empfohlen wird, den Modulnamen in der Datei /boot/loader.conf anzufügen, so dass das Modul am Anfang des Bootvorgangs eingebunden wird.

Die folgenden Abschnitte werden verschiedene MAC Module und ihre jeweiligen Vor- und Nachteile vorstellen. Außerdem wird erklärt, wie sie in bestimmte Umgebungen eingearbeitet werden können. Einige Module unterstützen die Verwendung von `Labels`, das heißt Zugriffskontrolle durch hinzufügen einer Kennzeichnung in der Art von "dieses ist erlaubt, jenes aber nicht". Eine Label-Konfigurationdatei kontrolliert unter anderem, wie auf Dateien zugegriffen oder wie über das Netzwerk kommuniziert werden darf. Im vorangehenden Abschnitt wurde bereits erläutert, wie die Option `multilabel` auf Dateisysteme angewendet wird, um eine Zugriffskontrolle auf einzelne Dateien oder ganze Dateisysteme zu konfigurieren.

Eine `single label` Konfiguration erzwingt ein einzelnes Label für das gesamte System. Daher wird die `tunefs`-Option `multilabel` genannt.

## 33.7. Das MAC Modul seeotheruids

Modulename: mac_seeotheruids.ko

Parameter in der Kernelkonfiguration: `options MAC_SEEOTHERUIDS`

Bootparameter: `mac_seeotheruids_load="YES"`

Das Modul mac_seeotheruids(4) erweitert die `sysctl`-Variablen `security.bsd.see_other_uids` und `security.bsd.see_other_gids`. Diese Optionen benötigen keine im Vorhinein zu setzenden Labels und können leicht durchschaubar mit den anderen MAC-Modulen zusammenarbeiten.

Nachdem das Modul geladen wurde, können die folgenden `sysctl` Variablen verwendet werden.

- `security.mac.seeotheruids.enabled` dient zur Aktivierung des Moduls, zunächst mit den Standardeinstellungen. Diese verhindern, dass Nutzer Prozesse und Sockets sehen können, die ihnen nicht selbst gehöen.
- `security.mac.seeotheruids.specificgid_enabled` kann eine spezifizierte Nutzergruppe von dieser Richtlinie ausnehmen. Die entsprechende Gruppe muß an den Parameter `security.mac.seeotheruids.specificgid=XXX` übergeben werden, wobei *XXX* die ID der Gruppe ist, die von der Richtlinie ausgenommen werden soll.
- `security.mac.seeotheruids.primarygroup_enabled` kann verwendet werden, um eine spezifische, *primäre* Nutzergruppe von der Richtlinie auszuschliessen. Dieser Parameter und `security.mac.seeotheruids.specificgid_enabled` schließen einander aus.

# 33.8. Das MAC Modul bsdextended

Modulname: mac_bsdextended.ko

Parameter in der Kernelkonfiguration: `options MAC_BSDEXTENDED`

Bootparameter: `mac_bsdextended_load="YES"`

Das Modul mac_bsdextended(4) erstellt eine Firewall für das Dateisystem und ist eine Erweiterung des sonst üblichen Rechtemodells. Es erlaubt einem Administrator einen Regelsatz zum Schutz von Dateien, Werkzeugen und Verzeichnissen in der Dateisystemhierarchie zu erstellen, der einer Firewall ähnelt. Sobald auf ein Objekt im Dateisystem zugegriffen werden soll, wird eine Liste von Regel abgearbeitet, bis eine passende Regel gefunden wird oder die Liste zu Ende ist. Das Verhalten kann durch die Änderung des sysctl(8) Parameters `security.mac.bsdextended.firstmatch_enabled` eingestellt werden. Ähnlich wie bei den anderen Firewallmodulen in FreeBSD wird eine Datei erstellt, welche die Zugriffsregeln enthält. Diese wird beim Systemstart durch eine Variable in rc.conf(5) eingebunden.

Der Regelsatz kann mit dem Programm ugidfw(8) eingepflegt werden, welches eine Syntax bereitstellt, die der von ipfw(8) gleicht. Weitere Werkzeuge können auch selbst erstellt werden, indem die Funktionen der Bibliothek libugidfw(3) verwendet werden.

Bei der Arbeit mit diesem Modul ist äußerste Vorsicht geboten - falscher Gebrauch kann den Zugriff auf Teile des Dateisystems komplett unterbinden.

## 33.8.1. Beispiele

Nachdem das Modul mac_bsdextended(4) erfolgreich geladen wurde, zeigt das folgende Kommando die gegenwärtig aktiven Regeln an:

```
# ugidfw list 0 slots, 0 rules
```

Wie erwartet, sind keine Regeln definiert. Das bedeutet, das auf alle Teile des Dateisystems zugegriffen werden kann. Um eine Regel zu definieren, die jeden Zugriff durch Nutzer blockiert und nur die Rechte von `root` unangetastet läßt, muß lediglich dieses Kommando ausgeführt werden:

```
# ugidfw add subject not uid root new object not uid root mode n
```

Das ist allerdings keine gute Idee, da nun allen Nutzern der Zugriff auf selbst die einfachsten Programme wie `ls` untersagt wird. Angemessener wäre etwas wie:

```
# ugidfw set 2 subject uid user1 object uid user2 mode n
# ugidfw set 3 subject uid user1 object gid user2 mode n
```

Diese Befehle bewirken, dass `user1` keinen Zugriff mehr auf Dateien und Programme hat, die `user2` gehören. Dies schließt das Auslesen von Verzeichniseinträgen ein.

Anstelle `uid user1` könnte auch `not uid user2` als Parameter übergeben werden. Dies würde diesselben Einschränkungen für alle Nutzer bewirken anstatt nur einen einzigen.



`root` ist von diesen Einstellungen nicht betroffen.

Dies sollte als Überblick ausreichen, um zu verstehen, wie das Modul mac_bsdextended(4) helfen kann, das Dateisystem abzuschotten. Weitere Informationen bieten die Manpages mac_bsdextended(4) und ugidfw(8).

## 33.9. Das MAC Modul ifoff

Modulname: mac_ifoff.ko

Parameter für die Kernelkonfiguration: `options MAC_IFOFF`

Bootparameter: `mac_ifoff_load="YES"`

Das Modul mac_ifoff(4) ist einzig dazu da, Netzwerkschnittstellen im laufenden Betrieb zu deaktivieren oder zu verhindern, das Netzwerkschnittstellen während der Bootphase gestartet werden. Dieses Modul benötigt für seinen Betrieb weder Labels, die auf dem System eingerichtet werden müssen, noch hat es Abhängigkeiten zu anderen MAC Modulen.

Der größte Teil der Kontrolle geschieht über die im folgenden aufgelisteten `sysctl`-Parameter:

- `security.mac.ifoff.lo_enabled` schaltet den gesamten Netzwerkverkehr auf der Loopback-Schnittstelle lo(4) an bzw. aus.
- `security.mac.ifoff.bpfrecv_enabled` macht das Gleiche für den Berkeley Paket Filter bpf(4).

- `security.mac.ifoff.other_enabled` schaltet den Verkehr für alle anderen Netzwerkschnittstellen.

Die wahrscheinlich häufigste Nutzung von mac_ifoff(4) ist die Überwachung des Netzwerks in einer Umgebung, in der kein Netzwerkverkehr während des Bootvorgangs erlaubt werden soll. Eine andere mögliche Anwendung wäre ein Script, das mit Hilfe von security/aide automatisch alle Schnittstellen blockiert, sobald Dateien in geschützten Verzeichnissen angelegt oder verändert werden.

## 33.10. Das MAC Modul portacl

Modulname: mac_portacl.ko

Parameter für die Kernelkonfiguration: `options MAC_PORTACL`

Bootparameter: `mac_portacl_load="YES"`

Mit Hilfe des Moduls mac_portacl(4) können die Anbindungen an die lokalen TCP und UDP Ports durch eine Vielzahl von `sysctl` Variablen beschränkt werden. Genauer gesagt ermöglicht mac_portacl(4) Nutzern ohne `root`-Rechten den Zugriff auf zu bestimmende privilegierte Ports, also denen innerhalb der ersten 1024.

Sobald das Modul geladen wurde, ist die Richtlinie für alle Sockets verfügbar. Die folgenden Variablen können für die Konfiguration verwendet werden:

- `security.mac.portacl.enabled` schaltet die Anwendung der Richtlinie ein oder aus.
- `security.mac.portacl.port_high` gibt den höchsten Port an, der von der Richtlinie mac_portacl(4) betroffen sein soll.
- `security.mac.portacl.suser_exempt` nimmt, wenn es einen Wert ungleich Null zugewiesen bekommt, `root` von der Richtlinie aus.
- `security.mac.portacl.rules` enthält als Wert die eigentliche `mac_portacl` Richtlinie.

Die eigentliche Konfiguration der `mac_portacl` Richtlinie wird der `sysctl`-Variablen `security.mac.portacl.rules` als Zeichenkette der Form `rule[,rule,…]` übergeben. Jede einzelne Regel hat die Form `idtype:id:protocol:port`. Der Parameter idtype ist entweder `uid` oder `gid` und wird verwendet, um den Parameter id als Nutzer-ID oder Gruppen-ID zu kennzeichnen. Der Parameter protocol gibt an, ob die Regel ür TCP oder UDP gelten soll (indem man den Wert auf `tcp` oder `udp` setzt). Und der letzte Parameter, port, enthält die Nummer des Ports, auf den der angegebene Nutzer bzw. die angegebene Gruppe Zugriff erhalten soll.



Da der Regelsatz direkt vom Kernel ausgewertet wird, können nur Zahlenwerte übergeben werden. Das heißt, Namen von Nutzern, Gruppen oder Dienstnamen aus der Datei /etc/services funktionieren nicht.

Auf UNIX®-artigen Betriebssystemen sind die Ports kleiner 1024 privilegierten Prozessen vorbehalten, müssen also mit als/von `root` gestartet werden und weiterhin laufen. Damit mac_portacl(4) die Vergabe von Ports kleiner als 1024 an nicht privilegierte Prozesse übernehmen kann, muß die UNIX® Standardeinstellung deaktiviert werden. Dazu ändert man die sysctl(8) Variablen `net.inet.ip.portrange.reservedlow` und `net.inet.ip.portrange.reservedhigh` auf den Wert

"0".

Weiterführende Informationen entnehmen Sie bitte den unten aufgeführten Beispielen oder der Man-Page mac_portacl(4)!

### 33.10.1. Beispiele

Die folgenden Beispiele sollten ein wenig Licht in die obige Diskussion bringen:

```
# sysctl security.mac.portacl.port_high=1023
# sysctl net.inet.ip.portrange.reservedlow=0 net.inet.ip.portrange.reservedhigh=0
```

Zunächst bestimmen wir, dass mac_portacl(4) für alle privilegierten Ports gelten soll und deaktivieren die normale UNIX®-Beschränkung.

```
# sysctl security.mac.portacl.suser_exempt=1
```

Da `root` von dieser Richtlinie nicht beeinträchtigt werden soll, setzen wir hier `security.mac.portacl.suser_exempt` auf einen Wert ungleich Null. Das Modul mac_portacl(4) ist nun so eingerichtet, wie es UNIX®-artige Betriebssysteme normal ebenfalls tun.

```
# sysctl security.mac.portacl.rules=uid:80:tcp:80
```

Nun erlauben wir dem Nutzer mit der UID 80, normalerweise dem Nutzer `www`, den Port 80 zu verwenden. Dadurch kann der Nutzer `www` einen Webserver betreiben, ohne dafür mit `root` -Privilegien ausgestattet zu sein.

```
# sysctl security.mac.portacl.rules=uid:1001:tcp:110,uid:1001:tcp:995
```

Hier wird dem Nutzer mit der UID 1001 erlaubt, die TCP Ports 110 ("pop3") und 995 ("pop3s") zu verwenden. Dadurch kann dieser Nutzer einen Server starten, der Verbindungen an diesen beiden Ports annehmen kann.

## 33.11. Das MAC Modul partition

Modulname: mac_partition.ko

Parameter für die Kernelkonfiguration: `options MAC_PARTITION`

Bootparameter `mac_partition_load="YES"`

Die Richtlinie mac_partition(4) setzt Prozesse in spezielle "Partitionen", entsprechend dem zugewiesenen MAC Label. Man kann sich das vorstellen wie eine spezielle Art jail(8), auch wenn das noch kein wirklich guter Vergleich ist.

Es wird empfohlen, dieses Modul durch einen Eintrag in loader.conf(5) zu aktivieren, so dass die Richtlinie während des Bootvorganges eingebunden wird.

Der Großteil der Konfiguration geschieht mit dem Kommando setpmac(8), wie gleich erklärt wird. Außerdem gibt es folgenden `sysctl` Parameter für diese Richtlinie.

- `security.mac.partition.enabled` erzwingt die Verwendung von MAC Prozeß-Partitionen.

Sobald diese Richtlinie aktiv ist, sehen Nutzer nur noch ihre eigenen Prozesse, und alle anderen Prozesse, die ebenfalls derselben Prozeß-Partition zugeordnet sind. Sie können jedoch nicht auf Prozesse oder Werkzeuge außerhalb des Anwendungsbereich dieser Partition zugreifen. Das bedeutet unter anderem, das ein Nutzer, der einer Klasse `insecure` zugeordnet ist, nicht auf das Kommando `top` zugreifen kann - wie auch auf viele anderen Befehle, die einen eigenen Prozeß erzeugen.

Um einen Befehl einer Prozeß-Partition zuzuordnen, muß dieser durch das Kommando `setpmac` mit einem Label versehen werden:

```
# setpmac partition/13 top
```

Diese Zeile fügt das Kommando `top` dem Labelsatz für Nutzer der Klasse `insecure` hinzu, sofern die Partition 13 mit der Klasse `insecure` übereinstimmt. Beachten Sie, dass alle Prozesse, die von Nutzern dieser Klasse erzeugt werden, das Label `partition/13` erhalten, und dieses auch nicht durch den Nutzer geändert werden kann.

### 33.11.1. Beispiele

Der folgende Befehl listet die vergebenen Label für Prozeß-Partitionen und die laufenden Prozesse auf.

```
# ps Zax
```

Das nächste Kommando liefert das Label der Prozeß-Partition eines anderen Nutzers `trhodes` und dessen gegenwärtig laufenden Prozesse zurück.

```
# ps -ZU trhodes
```



Jeder Nutzer kann die Prozesse in der Prozeß-Partition von `root` betrachten, solange nicht die Richtlinie mac_seeotheruids(4) geladen wurde.

Eine ausgefeilte Umsetzung dieser Richtlinie deaktiviert alle Dienste in /etc/rc.conf und startet diese dann später durch ein Skript, das jedem Dienst das passende Label zuordnet.



Die folgenden Richtlinien verwenden Zahlenwerte anstatt der drei Standardlabels. Diese Optionen, und ihre Grenzen, werden in den zugehörigen Manpages genauer erklärt.

# 33.12. Das MAC Modul Multi-Level Security

Modulname: mac_mls.ko

Parameter für die Kernelkonfiguration: `options MAC_MLS`

Bootparameter: `mac_mls_load="YES"`

Die Richtlinie mac_mls(4) kontrolliert die Zugriffe zwischen Subjekten und Objekten, indem sie den Informationsfluß strengen Regeln unterwirft.

In MLS Umgebungen wird jedem Subjekt oder Objekt ein "Freigabe"-Level zugeordnet, und diese werden wiederum zu einzelnen Verbünden zusammengefaßt. Da diese Freigabe- oder Anfälligkeits-Level Zahlen größer 6000 erreichen können, ist es für jeden Systemadministrator eine undankbare Aufgabe, jede Entität von Grund auf zu konfigurieren. Zum Glück gibt es 3 "instant" Labels, die in der Richtlinie zur Anwendung bereit stehen.

Diese drei Labels heißen `mls/low`, `mls/equal` und `mls/high`. Da sie in den Manpages mac_mls(4) ausführlich beschrieben werden, gibt es hier nur einen kurzen Abriß:

- Das Label `mls/low` ist eine niedrige Einstellung, die von allen anderen dominiert werden darf. Alles, was mit `mls/low` versehen wird, hat ein niedriges Freigabe-Level und darf auf keine Informationen zugreifen, denen ein höheres Freigabe-Level zugeordnet wurde. Einem Objekt mit diesem Label kann außerdem keine Information durch ein Objekt höherer Freigabe übergeben werden, es kann also auch nicht durch solche Objekte editiert oder überschrieben werden.
- Das Label `mls/equal` wird an Objekte vergeben, die von dieser Richtlinie ausgenommen werden sollen.
- Das Label `mls/high` verkörpert das höchstmögliche Freigabe-Level. Objekte, denen dieses Label zugeordnet wird, dominieren alle anderen Objekte des Systems. Trotzdem können sie Objekten mit einem niedrigeren Freigabe-Level keine Informationen zuspielen.

MLS bietet:

- Eine hierarchische Sicherheitsschicht und Zuordnung nichthierarchischer Kategorien;
- Feste Regeln: kein "Read-Up", kein "Write-Down" (ein Subjekt kann nur Objekte gleicher oder *niedrigerer* Stufe lesen, und es kann nur Objekte gleicher oder *höherer* Stufe schreiben);
- Geheimhaltung (indem unangemessene Offenlegung von Daten verhindert wird);
- Eine Basis zum Entwerfen von Systemen, die Daten verschiedener Vertraulichkeitsebenen gleichzeitig handhaben sollen (ohne das geheime und vertrauliche Informationen untereinander ausgetauscht werden können).

Nachfolgend werden die `sysctl`-Variablen vorgestellt, die für die Einrichtung spezieller Dienste und Schnittstellen vorhanden sind.

- `security.mac.mls.enabled` schaltet die Richtlinie MLS ein (oder aus).
- `security.mac.mls.ptys_equal` sorgt dafür, dass während der Initialisierung alle pty(4)-Geräte als

`mls/equal` gekennzeichnet werden.

- `security.mac.mls.revocation_enabled` sorgt dafür, dass die Zugriffsrechte von Objekten wieder zurückgesetzt werden, nachdem deren Label vorübergehend auf ein niedrigeres Freigabe-Level geändert wurde.
- `security.mac.mls.max_compartments` gibt die maximale Anzahl von Verbünden an. Im Prinzip ist es die höchste Nummer eines Verbundes auf dem System.

Um die Labels der MLS Richtlinie zu bearbeiten verwendet man setfmac(8). Um ein Objekt zu kennzeichnen, benutzen Sie folgendes Kommando:

```
# setfmac mls/5 test
```

Um das MLS-Label der Datei test auszulesen, verwenden Sie dieses Kommando:

```
# getfmac test
```

Dies ist eine Zusammenstellung der Merkmale von test. Ein anderer Ansatz ist, für diese Richtlinie eine Konfigurationsdatei in /etc abzulegen, die alle Informationen enthält und mit der dann das Kommando `setfmac` gefüttert wird. Diese Vorgehensweise wird erklärt, nachdem alle Richtlinien vorgestellt wurden.

### 33.12.1. Verbindlicher Vertraulichkeit in der Planungsphase

Mit dem Richtlinienmodul `Multi-Level Security` bereitet sich ein Administrator darauf vor, den Fluß vertraulicher Informationen zu kontrollieren. Beim Starten der Richtlinie ist immer `mls/low` voreingestellt - alles kann auf alles zugreifen. Der Administrator ändert dies während der eigentlichen Konfiguration, indem er die Vertraulichkeit bestimmter Objekte erhöht.

Jenseits der drei Grundeinstellungen des Labels kann der Administrator einzelne Nutzer oder Nutzergruppen nach Bedarf zusammenschließen und den Informationsaustausch zwischen diesen gestatten oder unterbinden. Es ist sicher eine Vereinfachung, die Freigabe-Level mit Begriffen wie `vertraulich`, `geheim` oder `streng geheim` zu bezeichnen. Einige Administratoren erstellen einfach verschiedene Gruppen auf der Ebene von gegenwärtigen Projekten. Ungeachtet der Herangehensweise bei der Klassifizierung muß ein gut durchdachter Plan existieren, bevor eine derart einengende Richtlinie umgesetzt wird.

Exemplarisch für die Anwendung dieses Moduls bzw. dieser Richtlinie seien angeführt:

- Ein E-Commerce Webserver
- Ein Dateiserver, der vertrauliche Informationen einer Firma oder eines Konzerns speichert
- Umgebungen in Finanzeinrichtungen

Der unsinnigste Einsatzort für diese Richtlinie wäre ein Arbeitsplatzrechner mit nur zwei oder drei Benutzern.

# 33.13. Das MAC Modul Biba

Modulname: mac_biba.ko

Parameter für die Kernelkonfiguration: `options MAC_BIBA`

Bootparameter: `mac_biba_load="YES"`

Das Modul mac_biba(4) lädt die MAC Biba Richtlinie. Diese ähnelt stark der MLS Richtlinie, nur das die Regeln für den Informationsfluß ein wenig vertauscht sind. Es wird in diesem Fall der absteigende Fluß sicherheitskritischer Information geregelt, während die MLS Richtlinie den aufsteigenden Fluß regelt. In gewissen Sinne treffen dieses und das vorangegangene Unterkapitel also auf beide Richtlinien zu.

In einer Biba-Umgebung wird jedem Subjekt und jedem Objekt ein "Integritäts"-Label zugeordnet. Diese Labels sind in hierarchischen Klassen und nicht-hierarchischen Komponenten geordnet. Je höher die Klasse, um so höher die Integrität.

Die unterstützten Labels heißen `biba/low`, `biba/equal` und `biba/high`. Sie werden im Folgenden erklärt:

- `biba/low` ist die niedrigste Stufe der Integrität, die einem Objekt verliehen werden kann. Wenn sie einem Objekt oder Subjekt zugeordnet wird, kann dieses auf Objekte oder Subjekte, die biba/high markiert wurden, zwar lesend zugreifen, nicht jedoch schreibend.
- Das Label `biba/equal` ist, wie der aufmerksame Leser sicherlich schon ahnt, für die Ausnahmen dieser Richtlinie gedacht und sollte nur diesen Ausnahmen entsprechenden Objekten verliehen werden.
- `biba/high` markierte Subjekte und Objekte können Objekte niedrigerer Stufe schreiben , nicht jedoch lesen. Es wird empfohlen, dass dieses Label an Objekte vergeben wird, die sich auf Integrität des gesamten Systems auswirken.

Biba stellt bereit:

- Hierarchische Integritätsstufen mit einem Satz nichthierarchischer Integritätskategorien;
- Festgeschriebene Regeln: kein "Write-Up", kein "Read-Down" (der Gegensatz zu MLS - ein Subjekt erhält schreibenden Zugriff auf Objekte gleicher oder geringerer Stufe, aber nicht bei höherer, und lesenden Zugriff bei gleicher Stufe oder höerer, aber nicht bei niedrigerer);
- Integrität (es wird die Echtheit der Daten gewährleistet, indem unangemessene Veränderungen verhindert werden);
- Eine Abstufung der Gewährleistung (im Gegensatz zu MLS, bei der eine Abstufung der Vertraulichkeit vorgenommen wird).

Folgende `sysctl` Parameter werden zur Nutzung der Biba-Richtlinie angeboten:

- `security.mac.biba.enabled` zum Aktivieren/Deaktivieren der Richtlinie auf dem Zielsystem.
- `security.mac.biba.ptys_equal` wird verwendet, um die Biba-Richtlinie auf der pty(4) -Schnittstelle zu deaktivieren.

- `security.mac.biba.revocation_enabled` erzwingt das Zurücksetzen des Labels, falls dieses zeitweise geändert wurde um ein Subjekt zu dominieren.

Um Einstellungen der Biba Richtlinie für Systemobjekte zu verändern werden die Befehle `setfmac` und `getfmac` verwendet:

```
# setfmac biba/low test
# getfmac test
test: biba/low
```

### 33.13.1. Verbindliche Integrität in der Planungsphase

Integrität garantiert, im Unterschied zu Sensitivität, dass Informationen nur durch vertraute Parteien verändert werden können. Dies schließt Informationen ein, die zwischen Subjekten ausgetauscht werden, zwischen Objekt, oder auch zwischen den beiden. Durch Integrität wird gesichert, das Nutzer nur Informationen verändern, oder gar nur lesen können, die sie explizit benötigen.

Das Modul mac_biba(4) eröffnet einem Administrator die Möglichkeit zu bestimmen, welche Dateien oder Programme ein Nutzer oder eine Nutzergruppe sehen bzw. aufrufen darf. Gleichzeitig kann er zusichern, dass dieselben Programme und Dateien frei von Bedrohungen sind und das System die Echtheit gewährleistet - für diesen Nutzer oder die Nutzergruppe.

Während der anfänglichen Phase der Planung muß der Administrator vorbereitet sein, Nutzer in Klassen, Stufen und Bereiche einzuteilen. Der Zugriff auf Dateien und insbesondere auch Programme wird verhindert sowohl vor als auch nachdem sie gestartet wurden. Das System selbst erhält als Voreinstellung das Label `biba/high` sobald das Modul aktiviert wird - und es liegt allein am Administrator, die verschiedenen Klassen und Stufen für die einzelnen Nutzer zu konfigurieren. Anstatt mit Freigaben zu arbeiten, wie weiter oben gezeigt wurde, könnte man auch Überbegriffe für Projekte oder Systemkomponenten entwerfen. Zum Beispiel, ausschließlich Entwicklern den Vollzugriff auf Quellcode, Compiler und Entwicklungswerkzeuge gewähren, während man andere Nutzer in Kategorien wie Tester, Designer oder einfach nur "allgemeiner Nutzer" zusammenfaßt, die für diese Bereiche lediglich lesenden Zugriff erhalten sollen.

Mit seinem ursprünglichen Sicherheits-Standpunkt ist ein Subjekt niedrigerer Integrität unfähig, ein Subjekt höherer Integrität zu verändern. Ein Subjekt höherer Integrität kann ein Subjekt niedrigerer Integrität weder beobachten noch lesen. Wenn man ein Label für die niedrigstmögliche Klasse erstellt, kann man diese allen Subjekten verwehren. Einige weitsichtig eingerichtete Umgebungen, die diese Richtlinie verwenden, sind eingeschränkte Webserver, Entwicklungs- oder Test-Rechner oder Quellcode-Sammlungen. Wenig sinnvoll ist diese Richtlinie auf einer Arbeitsstation, oder auf Rechnern die als Router oder Firewall verwendet werden.

## 33.14. Das MAC Modul LOMAC

Modulname: mac_lomac.ko

Parameter für die Kernelkonfiguration: `options MAC_LOMAC`

Bootparameter: `mac_lomac_load="YES"`

Anders als die Biba Richtlinie erlaubt die mac_lomac(4) Richtlinie den Zugriff auf Objekte niedrigerer Integrität nur, nachdem das Integritätslevel gesenkt wurde. Dadurch wird eine Störung derIntegritätsregeln verhindert.

Die MAC Version der "Low-Watermark" Richtlinie, die nicht mit der älteren -Implementierung verwechselt werden darf, arbeitet fast genauso wie Biba. Anders ist, dass hier "schwebende" Label verwendet werden, die ein Herunterstufen von Subjekten durch Hilfsverbünde ermöglichen. Dieser zweite Verbund wird in der Form `[auxgrade]` angegeben und sollte in etwa aussehen wie `lomac/10[2]`, wobei die Ziffer zwei (2) hier den Hilfsverbund abbildet.

Die MAC Richtlinie `LOMAC` beruht auf einer durchgängigen Etikettierung aller Systemobjekte mit Integritätslabeln, die Subjekten das Lesen von Objekten niedriger Integrität gestatten und dann das Label des Subjektes herunterstufen - um zukünftige Schreibvorgänge auf Objekte hoher Integrität zu unterbinden. Dies ist die Funktion der Option `[auxgrade]`, die eben vorgestellt wurde. Durch sie erhält diese Richtlinie eine bessere Kompatibilität und die Initialisierung ist weniger aufwändig als bei der Richtlinie Biba.

### 33.14.1. Beispiele

Wie schon bei den Richtlinien Biba und MLS werden die Befehle `setfmac` und `setpmac` verwendet, um die Labels an den Systemobjekten zu setzen:

```
# setfmac /usr/home/trhodes lomac/high[low]
# getfmac /usr/home/trhodes lomac/high[low]
```

Beachten Sie, dass hier der Hilfswert auf `low` gesetzt wurde - dieses Leistungsmerkmal ist nur in der MAC `LOMAC` Richtlinie enthalten.

## 33.15. Beispiel 1: Nagios in einer MAC Jail

Die folgende Demonstration setzt eine sichere Umgebung mithilfe verschiedener MAC Module und sorgfältig konfigurierter Richtlinien um. Es handelt sich jedoch nur um einen Test und sollte nicht als Antwort auf jedes Problem in Fragen Sicherheit gesehen werden. Eine Richtlinie nur umzusetzen und dann einfach laufen zu lassen, funktioniert nie und kann eine echte Arbeitsumgebung in eine Katastrophe stürzen.

Bevor es losgeht, muß jedes Dateisystem mit der Option `multilabel`, wie weiter oben beschrieben, markiert werden. Dies nicht zu tun, führt zu Fehlern. Außerdem müssen die Ports net-mngt/nagios-plugins, net-mngt/nagios und www/apache22 installiert und konfiguriert sein, so dass sie ordentlich laufen.

### 33.15.1. Erstellen einer Nutzerklasse `insecure`

Beginnen wir die Prozedur mit dem Hinzufügen einer Nutzerklasse in der Datei /etc/login.conf:

```
insecure:\
:welcome=/etc/motd:\
:setenv=MAIL=/var/mail/$,BLOCKSIZE=K:\
:path=~/bin:/sbin:/bin:/usr/sbin:/usr/bin:/usr/local/sbin:/usr/local/bin--
:manpath=/usr/shared/man /usr/local/man:\
:nologin=/usr/sbin/nologin:\
:cputime=1h30m:\
:datasize=8M:\
:vmemoryuse=100M:\
:stacksize=2M:\
:memorylocked=4M:\
:memoryuse=8M:\
:filesize=8M:\
:coredumpsize=8M:\
:openfiles=24:\
:maxproc=32:\
:priority=0:\
:requirehome:\
:passwordtime=91d:\
:umask=022:\
:ignoretime@:\
:label=biba/10(10-10):
```

Zusätzlich fügen wir beim Standardnutzer folgende Zeile hinzu:

```
:label=biba/high:
```

Anschließend muß die Datenbank neu erstellt werden:

```
# cap_mkdb /etc/login.conf
```

### 33.15.2. Boot-Konfiguration

Starten Sie den Rechner noch nicht neu. Fügen Sie zunächst noch die folgenden Zeilen in die Datei /boot/loader.conf ein, damit die benötigten Module während des Systemstarts geladen werden:

```
mac_biba_load="YES"
mac_seeotheruids_load="YES"
```

### 33.15.3. Nutzer einrichten

Ordnen Sie den Superuser `root` der Klasse `default` zu:

```
# pw usermod root -L default
```

Alle Nutzerkonten, die weder root noch Systemkonten sind, brauchen nun eine Loginklasse, da sie sonst keinen Zugriff auf sonst übliche Befehle erhalten, wie bspw. vi(1). Das folgende sh Skript wird diese Aufgabe erledigen:

```
# for x in `awk -F: '($3 >= 1001) && ($3 != 65534) { print $1 }' \
      /etc/passwd`; do pw usermod $x -L default; done;
```

Verschieben Sie die Nutzer nagios und www in die insecure Klasse:

```
# pw usermod nagios -L insecure
```

```
# pw usermod www -L insecure
```

### 33.15.4. Die Kontextdatei erstellen

Nun muß eine Kontextdatei erstellt werden. Die folgende Beispieldatei soll dazu in /etc/policy.contexts gespeichert werden:

```
# This is the default BIBA policy for this system.

# System:
/var/run                        biba/equal
/var/run/*                      biba/equal

/dev                            biba/equal
/dev/*                          biba/equal

/var                biba/equal
/var/spool                      biba/equal
/var/spool/*                    biba/equal

/var/log                        biba/equal
/var/log/*                      biba/equal

/tmp                biba/equal
/tmp/*                  biba/equal
/var/tmp                biba/equal
/var/tmp/*              biba/equal

/var/spool/mqueue               biba/equal
/var/spool/clientmqueue         biba/equal

# For Nagios:
/usr/local/etc/nagios
/usr/local/etc/nagios/*         biba/10
```

```
/var/spool/nagios               biba/10
/var/spool/nagios/*             biba/10

# For apache
/usr/local/etc/apache           biba/10
/usr/local/etc/apache/*         biba/10
```

Die Richtlinie erzwingt Sicherheit, indem der Informationsfluß Einschränkungen unterworfen wird. In der vorliegenden Konfiguration kann kein Nutzer, weder `root` noch andere, auf Nagios zugreifen. Konfigurationsdateien und die Prozesse, die Teil von Nagios sind, werden durch unsere MAC vollständig abgegrenzt.

Die Kontextdatei kann nun vom System eingelesen werden, indem folgender Befehl ausgeführt wird:

```
# setfmac -ef /etc/policy.contexts /
# setfmac -ef /etc/policy.contexts /
```



Das obenstehende Dateisystem-Layout kann, je nach Umgebung, sehr unterschiedlich aussehen. Außerdem muß es auf jedem einzelnen Dateisystem ausgeführt werden.

In die Datei /etc/mac.conf müssen nun noch diese Änderungen eingetragen werden:

```
default_labels file ?biba
default_labels ifnet ?biba
default_labels process ?biba
default_labels socket ?biba
```

### 33.15.5. Netzwerke einbinden

Tragen Sie die folgende Zeile in die Datei /boot/loader.conf ein:

```
security.mac.biba.trust_all_interfaces=1
```

Und das Folgende gehört in Datei rc.conf zu den Optionen für die Netzwerkkarte. Falls die Netzwerkverbindung(-en) via DHCP konfiguriert werden, muß man dies nach jedem Systemstart eigenhändig nachtragen:

```
maclabel biba/equal
```

### 33.15.6. Testen der Konfiguration

Versichern Sie sich, dass der Webserver und Nagios nicht automatisch geladen werden und starten

Sie den Rechner neu. Prüfen Sie nun, ob `root` wirklich keinen Zugriff auf die Dateien im Konfigurationsverzeichnis von Nagios hat. Wenn `root` den Befehl ls(1) auf /var/spool/nagios ausführen kann, ist irgendwas schief gelaufen. Es sollte ein `permission denied` Fehler ausgegeben werden.

Wenn alles gut aussieht, können Nagios, Apache und Sendmail gestartet werden - allerdings auf eine Weise, die unserer Richtlinie gerecht wird. Zum Beispiel durch die folgenden Kommandos:

```
# cd /etc/mail && make stop && \
setpmac biba/equal make start && setpmac biba/10\(10-10\) apachectl start && \
setpmac biba/10\(10-10\) /usr/local/etc/rc.d/nagios.sh forcestart
```

Versichern Sie sich lieber doppelt, dass alles ordentlich läuft. Wenn nicht, prüfen Sie die Logs und Fehlermeldungen. Verwenden Sie das sysctl(8) Werkzeug um die Sicherheitsrichtlinie sysctl(8) zu deaktivieren und versuchen Sie dann alles noch einmal zu starten.



Der Superuser kann den Vollzug der Richtlinie schalten und die Konfiguration ohne Furcht verändern. Folgender Befehl stuft eine neu gestartete Shell herunter:

```
# setpmac biba/10 csh
```

Um dies zu vermeiden, werden die Nutzer durch login.conf(5) eingeschränkt. Wenn setpmac(8) einen Befehl außerhalb der definierten Schranken ausführen soll, wird ein Fehler zurückgeliefert. In so einem Fall muß `root` auf `biba/high(high-high)` gesetzt werden.

## 33.16. Beispiel 2: User Lock Down

Grundlage dieses Beispiels ist ein relativ kleines System zur Datenspeicherung mit weniger als 50 Benutzern. Diese haben die Möglichkeit, sich einzuloggen und dürfen nicht nur Daten speichern, sondern auch auf andere Ressourcen zugreifen.

Die Richtlinien mac_bsdextended(4) und mac_seeotheruids(4) können gleichzeitig eingesetzt werden. Zusammen kann man mit ihnen nicht nur den Zugriff auf Systemobjekte einschränken, sondern auch Nutzerprozesse verstecken.

Beginnen Sie, indem Sie die folgende Zeile in die Datei /boot/loader.conf eintragen:

```
mac_seeotheruids_load="YES"
```

Die Richtlinie mac_bsdextended(4) wird durch den anschließenden Eintrag in /etc/rc.conf hinzugefügt:

```
ugidfw_enable="YES"
```

Die Standardregeln, welche in /etc/rc.bsdextended gespeichert sind, werden zum Systemstart geladen. Sie müssen aber noch angepaßt werden. Da dieser Computer nur Nutzern dienen soll und weitere Dienste gestartet werden, kann alles bis auf die beiden letzten Zeilen auskommentiert werden. Das sorgt dafür dass jeder Nutzer seine eigenen Systemobjekte erhält.

Nun fügen wir alle benötigten Nutzer auf der Maschine hinzu und starten neu. Zum Testen der Einstellungen loggen Sie sich parallel zwei mal mit unterschiedlichen Nutzernamen ein und starten Sie das Kommando `ps aux`. Dort sehen Sie, dass Sie die Prozesse des anderen Nutzers nicht sehen können. Versuchen Sie, ls(1) auf das Heimatverzeichnis eines anderen Nutzers auszuführen. Auch dieser Versuch wird fehlschlagen.

Solange nicht die speziellen `sysctl`-Variablen geändert wurden, hat der Superuser noch vollen Zugriff. Sobald auch diese Einstellungen angepaßt wurden, führen Sie ruhig auch den obigen Test als `root` aus.



Wenn ein neuer Benutzer hinzugefügt wird, ist für diesen zunächst keine mac_bsdextended(4) Regel im Regelsatz vorhanden. Schnelle Abhilfe schafft hier, einfach das Kernelmodul mit kldunload(8) zu entladen und mit kldload(8) erneut einzubinden.

# 33.17. Fehler im MAC beheben

Während der Entwicklung des Frameworks haben einige Nutzer auf Probleme hingewiesen. Einige davon werden hier aufgeführt:

## 33.17.1. Die Option `multilabel` greift nicht auf der /-Partition

Es scheint, dass etwa jedem fünfzigsten Nutzer dieses Problem widerfährt. Und in der Tat - auch wir kennen es aus der Entwicklung. Genauere Untersuchungen dieses "Bugs" machten uns glauben, dass es sich entweder um einen Fehler in oder eine fehlerhafte Interpretation der Dokumentation handelt. Warum auch immer dieser Fehler auftritt - er kann mit folgender Prozedur behoben werden:

1. Öffnen Sie die Datei /etc/fstab und setzen Sie die Rootpartition auf `ro` wie "read-only".
2. Starten Sie in den Einzelnutzermodus.
3. Rufen Sie `tunefs -l enable` für / auf.
4. Starten Sie in den Mehrbenutzermodus.
5. Führen Sie `mount -urw/` aus und ändern Sie anschließend in der Datei /etc/fstab die Option `ro` zurück in `rw`. Starten Sie das System noch einmal neu.
6. Achten Sie besonders auf die Ausgabe von `mount` um sich zu versichern, dass die `multilabel` korrekt für das root-Dateisystem gesetzt wurde.

## 33.17.2. Mit der aktivierten MAC kann ich keinen X11 Server starten

Dies kann durch die Richtlinie `partition` oder einer fehlerhaften Verwendung einer Richtlinie, die mit Labels arbeitet, auftreten. Zum debuggen versuchen Sie folgendes:

1. Schauen Sie sich die Fehlermeldungen genau an. Wenn der Nutzer einer `insecure` Klasse angehört, ist wahrscheinlich die Richtlinie `partition` die Ursache. Versuchen Sie, die Nutzerklasse auf `default` zu stellen und danach die Datenbank mit `cap_mkdb` zu erneuern. Wenn das Problem dadurch nicht gelöst wird, gehen Sie weiter zu Schritt 2.
2. Gehen Sie die Label-Richtlinien Schritt für Schritt nocheinmal durch. Achten Sie darauf, dass für den Nutzer, bei dem das Problem auftritt, für X11 und das Verzeichnis /dev alle Einstellungen korrekt sind.
3. Falls all dies nicht helfen sollte, senden Sie die Fehlermeldung und eine Beschreibung ihrer Arbeitsumgebung an die (englisch-sprachige) TrustedBSD Diskussionsliste auf der TrustedBSD Webseite oder an die FreeBSD general questions Mailingliste.

### 33.17.3. Error: cannot stat .login_conf

Wenn ich versuche, von `root` zu einem anderen Nutzer des Systems zu wechseln, erhalte ich die Fehlermeldung `_secure_path: unable to state .login_conf`.

Diese Meldung wird gewöhnlich ausgegeben, wenn der Nutzer ein höhere Label-Einstellung hat als der, dessen Identität man annehmen möchte. Ausführlich: Wenn ein Nutzer `joe` als `biba/low` gelabelt wurde, kann `root`, der `biba/high` als Voreinstellung trägt, das Heimatverzeichnis von `joe` nicht einsehen. Das passiert unabhänig davon, ob `root` vorher mit `su` die Identität von `joe` angenommen hat oder nicht, da das Label sich nicht ändert. Hier haben wir also einen Fall, in dem das Gewährleistungsmodell von Biba verhindert, das der Superuser Objekte einer niedrigeren Integrität betrachten kann.

### 33.17.4. Der Nutzer `root` ist kaputt!

Im normalen oder sogar im Einzelbenutzermodus wird `root` nicht anerkannt. Das Kommando `whoami` liefert 0 (null) und `su` liefert `who are you?` zurück. Was geht da vor?

Das kann passieren, wenn eine Label-Richtlinie ausgeschaltet wird - entweder durch sysctl(8) oder wenn das Richtlinienmodul entladen wurde. Wenn eine Richtlinie deaktiviert oder auch nur vorübergehen deaktiviert wird, muß die Befähigungsdatenbank neu konfiguriert werden, d.h. die `label` Option muß entfernt werden. Überprüfen Sie, ob alle `label` Einträge aus der Datei /etc/login.conf entfernt wurden und bauen Sie die Datenbank mit `cap_mkdb` neu.

Dieser Fehler kann auch auftreten, wenn eine Richtlinie den Zugriff auf die Datei master.passwd einschränkt. Normalerweise passiert das nur, wenn ein Administrator ein Label an diese Datei vergibt, das mit der allgemeingültigen Richtlinie, die das System verwendet, in Konflikt steht. In solchen Fällen werden die Nutzerinformationen vom System ausgelesen und jeder weitere Zugriff wird blockiert, sobald das neue Label greift. Wenn man die Richtlinie via sysctl(8) ausschaltet, sollte es erstmal wieder gehen.

# Kapitel 34. Security Event Auditing

## 34.1. Einleitung

FreeBSD bietet Unterstützung für Sicherheits-Auditing. Ereignis-Auditing bietet zuverlässige, feingranulierte und konfigurierbare Aufzeichnung einer Vielzahl von sicherheitsrelevanten Systemereignissen einschließlich Benutzereingaben, Konfigurationsänderungen sowie Datei- und Netzwerkzugriffen. Diese Log-Datensätze können unschätzbar wertvoll sein für direkte Systemüberwachung, Einbruchserkennung und Post-Mortem-Analyse. FreeBSD implementiert Sun™s öffentlich zugängliches Basic Security Module (BSM) Application Programming Interface (API) und Dateiformat, und kann mit den Audit-Implementierungen von Sun™ Solaris™ und Apple® Mac OS® X zusammenarbeiten.

Dieses Kapitel konzentriert sich auf die Installation und Konfiguration des Ereignis-Auditings. Es erklärt Audit-Richtlinien und stellt ein Beispiel einer Audit-Konfiguration vor.

Nach dem Lesen dieses Kapitels werden Sie Folgendes wissen:

- Was Ereignis-Auditing ist und wie es funktioniert.
- Wie man Ereignis-Auditing in FreeBSD für Benutzer und Prozesse konfiguriert.
- Wie man den Audit-Pfad mittels Audit-Reduktion und Revisionswerkzeugen überprüft.

Vor dem Lesen dieses Kapitels sollten Sie:

- Sowohl UNIX® als auch FreeBSD-Basismechanismen beherrschen (Grundlagen des FreeBSD Betriebssystems).
- Mit den grundlegenden Mechanismen der Kernel-Konfiguration und -Kompilierung vertraut sein (Konfiguration des FreeBSD-Kernels).
- Mit den Maßnahmen zur Sicherung von FreeBSD vertraut sein (Sicherheit).

Die Audit-Funktionalität in FreeBSD hat einige bekannte Einschränkungen. Nicht alle sicherheitsrelevanten System-Ereignisse sind auditierbar, und einige Anmelde-Mechanismen, wie beispielsweise Xorg-basierte Bildschirm-Manager und Dienste von Drittanbietern, konfigurieren das Auditing für Benutzeranmeldungen nicht korrekt.



Das Sicherheits-Auditing ist in der Lage, sehr detaillierte Log-Dateien von Systemaktivitäten zu erzeugen. Auf einem ausgelasteten System kann die Pfad-Datei sehr groß werden, wenn sie für hohe Auflösung konfiguriert ist, und im Extremfall pro Woche um mehrere Gigabyte anwachsen. Administratoren sollten daher den benötigten Plattenplatz in Verbindung mit umfangreichen Audit-Konfigurationen berücksichtigen. So kann es wünschenswert sein, ein eigenes Dateisystem für /var/audit einzusetzen, damit andere Dateisysteme nicht betroffen sind, wenn das Dateisystem des Audit voll läuft.

## 34.2. Schlüsselbegriffe

Die folgenden Begriffe stehen im Zusammenhang mit Ereignis-Auditing:

- *event*: ein auditierbares Ereignis ist jedes Ereignis, das mit dem Audit-Subsystem aufgezeichnet werden kann. Beispiele für sicherheitsrelevante Systemereignisse sind etwa das Anlegen von Dateien, das Erstellen einer Netzwerkverbindung oder eine Benutzeranmeldung. Ereignisse sind entweder "attributierbar", können also zu einen authentifizierten Benutzer zurückverfolgt werden, oder sind "nicht-attributierbar". Nicht-attributierbare Ereignisse erfolgen daher vor der Authentifizierung im Anmeldeprozess (beispielsweise die Eingabe eines falschen Passworts).
- *class*: benannte Zusammenstellungen von zusammengehörenden Ereignissen, die in Auswahl-Ausdrücken benutzt werden. Häufig genutzte Klassen von Ereignissen schließen "file creation" (fc, Anlegen von Dateien), "exec" (ex, Ausführung) und "login_logout" (lo, Anmeldung-Abmeldung) ein.
- *record*: ein Audit-Logeintrag, der ein Sicherheitsereignis enthält. Jeder Datensatz enthält einen Ereignistyp, Informationen über den Gegenstand (Benutzer), welcher die Aktion durchführt, Datums- und Zeitinformationen, Informationen über jedes Objekt oder Argument sowie den Zustand hinsichtlich Erfolg oder Scheitern der Operation.
- *trail*: eine Log-Datei bestehend aus einer Reihe von Audit-Datensätzen, die Sicherheitsereignisse beschreiben. Pfade sind in grober zeitlicher Reihenfolge bezüglich des Zeitpunktes, an welchem ein Ereignis beendet wurde. Nur autorisierte Prozesse dürfen Datensätze zum Audit-Pfad hinzufügen.
- *selection expression*: eine Zeichenkette, welche eine Liste von Präfixen und Audit-Ereignisklassennamen enthält, um Ereignisse abzugleichen.
- *preselection*: der Prozess, durch den das System erkennt, welche Ereignisse von Interesse für den Administrator sind, um die Erzeugung von Datensätze zu verhindern, welche nicht von Belang sind. Die Konfiguration der Vorauswahl benutzt eine Reihe von Auswahl-Ausdrücken, um zu erkennen, welche Klassen von Ereignissen für welche Benutzer aufgezeichnet werden sollen sowie globale Einstellungen, welche sowohl auf autorisierte als auch unautorisierte Prozesse angewendet werden.
- *reduction*: Die Reduzierung ist der Prozess, durch den Datensätze von bestehenden Audit-Pfaden ausgewählt werden für Speicherung, Ausdruck oder Analyse. Ebenso der Prozess, durch den unerwünschte Datensätze aus dem Audit-Pfad entfernt werden. Mittels Reduzierung können Administratoren Richtlinien für die Speicherung von Audit-Daten vorgeben. Zum Beispiel können ausführliche Audit-Pfade für einen Monat gespeichert werden, um danach den Pfad für archivarische Zwecke auf die Anmeldeinformationen zu reduzieren.

## 34.3. Audit Konfiguration

Userspace-Unterstützung für Ereignis-Auditing ist Bestandteil des FreeBSD-Betriebssystems. Kernel-Unterstützung ist in der Voreinstellung im GENERIC-Kernel enthalten und auditd(8) kann durch Hinzufügen der folgenden Zeile in /etc/rc.conf aktiviert werden:

```
auditd_enable="YES"
```

Starten Sie anschließend den Audit-Daemon:

```
# service auditd start
```

Benutzer, die es bevorzugen einen angepassten Kernel zu kompilieren, müssen folgende Zeile in die Kernelkonfigurationsdatei aufnehmen:

```
options   AUDIT
```

### 34.3.1. Ereignis-Auswahlausdrücke

Auswahlausdrücke werden an einigen Stellen der Audit-Konfiguration benützt, um zu bestimmen, welche Ereignisse auditiert werden sollen. Die Ausdrücke enthalten eine Liste der Ereignisklassen, welche verglichen werden sollen. Auswahlausdrücke werden von links nach rechts ausgewertet und zwei Ausdrücke werden durch Aneinanderhängen miteinander kombiniert.

Audit-Ereignisklassen fasst die Audit-Ereignisklassen zusammen:

*Tabelle 12. Audit-Ereignisklassen*

| Name der Klasse | Beschreibung | Aktion |
|---|---|---|
| all | all | Vergleicht alle Ereigsnisklassen. |
| aa | authentication and authorization | |
| ad | administrative | Administrative Aktionen, ausgeführt auf dem System als Ganzes. |
| ap | application | Aktionen definiert für Applikationen. |
| cl | file close | Audit-Aufrufe für den Systemaufruf `close`. |
| ex | exec | Ausführung des Audit-Programms. Auditierung von Befehlszeilen-Argumenten und Umgebungsvariablen wird gesteuert durch audit_control(5) mittels der `argv` und `envv` -Parameter gemäß der `Richtlinien`-Einstellungen. |
| fa | file attribute access | Auditierung des Zugriffs auf Objektattribute wie stat(1) und pathconf(2). |
| fc | file create | Audit-Ereignisse, bei denen eine Datei als Ergebnis angelegt wird. |
| fd | file delete | Audit-Ereignisse, bei denen Dateilöschungen vorkommen. |
| fm | file attribute modify | Audit-Ereignisse, bei denen Dateiattribute geändert werden, wie chown(8), chflags(1) und flock(2). |

| Name der Klasse | Beschreibung | Aktion |
|---|---|---|
| fr | file read | Audit-Ereignisse, bei denen Daten gelesen oder Dateien zum lesen geöffnet werden. |
| fw | file write | Audit-Ereignisse, bei denen Daten geschrieben oder Dateien geschrieben oder verändert werden. |
| io | ioctl | Nutzung des Systemaufrufes `ioctl` durch Audit. |
| ip | ipc | Auditierung verschiedener Formen von Inter-Prozess-Kommunikation einschließlich POSIX-Pipes und System V IPC-Operationen. |
| lo | login_logout | Audit-Ereignisse von login(1) und logout(1). |
| na | non attributable | Auditierung nicht-attributierbarer Ereignisse. |
| no | invalid class | Kein Abgleich von Audit-Ereignissen. |
| nt | network | Audit-Ereignisse in Zusammenhang mit Netzwerkaktivitäten wie connect(2) und accept(2) |
| ot | other | Auditierung verschiedener Ereignisse. |
| pc | process | Auditierung von Prozess-Operationen wie exec(3) und exit(3). |

Diese Ereignisklassen können angepasst werden durch Modifizierung der Konfigurationsdateien audit_class und audit_event.

Jede Audit-Klasse kann mit einem Präfix kombiniert werden, welches anzeigt, ob erfolgreiche/gescheiterte Operationen abgebildet werden, und ob der Eintrag den Abgleich hinzufügt oder entfernt für die Klasse und den Typ. Präfixe für Audit-Ereignisklassen fasst die verfügbaren Präfixe zusammen.

*Tabelle 13. Präfixe für Audit-Ereignisklassen*

| Präfix | Aktion |
|---|---|
| + | Auditiert erfolgreiche Ereignisse in dieser Klasse. |
| - | Auditiert fehlgeschlagene Ereignisse in dieser Klasse. |
| ^ | Auditiert weder erfolgreiche noch fehlgeschlagene Ereignisse. |
| ^+ | Auditiert keine erfolgreichen Ereignisse in dieser Klasse. |
| ^- | Auditiert keine fehlgeschlagenen Ereignisse in dieser Klasse. |

Wenn kein Präfix vorhanden ist, werden sowohl erfolgreiche als auch fehlgeschlagene Ereignisse auditiert.

Das folgende Beispiel einer Auswahl-Zeichenkette wählt erfolgreiche und gescheiterte Anmelde/Abmelde-Ereignisse aus, aber nur erfolgreich beendete Ausführungs-Ereignisse:

```
lo,+ex
```

### 34.3.2. Konfigurationsdateien

Die folgenden Konfigurationsdateien für Sicherheits-Auditing befinden sich in /etc/security.

- audit_class: enthält die Definitionen der Audit-Klassen.
- audit_control: steuert die Eigenschaften des Audit-Subsystems, wie Standard-Audit-Klassen, Mindestfestplattenspeicher auf dem Audit-Log-Volume und die maximale Größe des Audit-Trails.
- audit_event: Namen und Beschreibungen der Audit-Ereignisse, und eine Liste von Klassen mit den dazugehörigen Ereignissen.
- audit_user: benutzerspezifische Audit-Anforderungen, kombinierbar mit den globalen Standardeinstellungen bei der Anmeldung.
- audit_warn: ein anpassbares Skript, das von auditd(8) verwendet wird, um in bestimmten Situationen Warnmeldungen zu generieren, z.B. wenn der Platz für Audit-Protokolle knapp wird, oder wenn die Datei des Audit-Trails rotiert wurde.



Konfigurationsdateien von Audit sollten sorgfältig bearbeitet und gepflegt werden, da Fehler in der Konfiguration zu einer fehlerhaften Protokollierung der Ereignisse führen können.

In den meisten Fällen wird der Administrator nur audit_control und audit_user anpassen müssen. Die erste Datei steuert systemweite Audit-Eigenschaften, sowie Richtlinien. Die zweite Datei kann für die Feinabstimmung bei der Auditierung von Benutzern verwendet werden.

#### Die audit_control-Datei

Die audit_control-Datei legt eine Anzahl Vorgabewerte fest:

```
dir:/var/audit
dist:off
flags:lo,aa
minfree:5
naflags:lo,aa
policy:cnt,argv
filesz:2M
expire-after:10M
```

Die Option `dir` wird genutzt, um eines oder mehrere Verzeichnisse festzulegen, in welchen Audit-Protokolle gespeichert werden. Gibt es mehrere Verzeichniseinträge, werden diese in der angegebenen Reihenfolge genutzt, bis sie jeweils gefüllt sind. Es ist üblich, Audit so zu konfigurieren, dass die Audit-Logs auf einem dedizierten Dateisystem abgelegt werden, um Wechselwirkungen zwischen dem Audit-Subsystem und anderen Subsystemen zu verhindern, falls das Dateisystem voll läuft.

Ist die Option `dist` auf `on` oder `yes` gesetzt, wird ein Link der Dateien des Audit-Trails in /var/audit/dist erstellt.

Das `flags`-Feld legt die systemweite Standard-Vorauswahl-Maske für attributierbare (direkt einem Benutzer zuordenbare) Ereignisse fest. Im obigen Beispiel werden alle gescheiterten und erfolgreichen Anmelde- und Abmelde-Ereignisse für alle Benutzer aufgezeichnet.

Die Option `minfree` definiert den minimalen Prozentsatz an freiem Plattenplatz für das Dateisystem, in welchem der Audit-Pfad abgespeichert wird. Wenn diese Schwelle überschritten ist, wird ein Warnhinweis erzeugt.

Die `naflags`-Option bestimmt diejenigen Audit-Klassen, für die nicht-attributierbare Ereignisse aufgezeichnet werden sollen, wie beispielsweise Anmeldeprozesse, Authentifizierung und Autorisierung.

Die Option `policy` legt eine durch Kommata getrennte Liste von policy-Flags fest, welche verschiedene Aspekte des Audit-Verhaltens steuern. Der Flag `cnt` zeigt an, dass das System trotz eines Audit-Fehlers weiterlaufen soll (dieses Flag wird dringend empfohlen). Ein anderes, häufig genutztes Flag ist `argv`, welches dazu führt, dass Befehlszeilen-Argumente für den Systemaufruf execve(2) als Teil der Befehlsausführung aufgezeichnet werden.

Die `filesz`-Option spezifiziert die maximale Größe der Audit-Datei, bevor sie automatisch beendet und rotiert wird. Der Wert `0` setzt die automatische Log-Rotation außer Kraft. Falls die angeforderte Dateigröße unterhalb des Minimums von 512K ist, dann wird die Angabe verworfen und ein Log-Hinweis wird erzeugt.

Die Option `expire-after` legt fest, wann die Audit-Dateien verfallen und entfernt werden.

**Die Datei audit_user**

Die audit_user-Datei erlaubt es dem Administrator, weitere Audit-Erfordernisse für bestimmte Benutzer festzulegen. Jede Zeile konfiguriert das Auditing für einen Benutzer über zwei Felder: `alwaysaudit` gibt eine Ansammlung von Ereignissen vor, welche immer für diesen Benutzer aufgezeichnet werden. `neveraudit` legt Ereignisse fest, die niemals für diesen Benutzer auditiert werden sollen.

Das folgende Beispiel einer audit_user-Datei zeichnet Anmelde/Abmelde-Ereignisse, erfolgreiche Befehlsausführungen für den Benutzer `root`, Anlegen von Dateien und erfolgreiche Befehlsausführungen für den Benutzer `www` auf. Falls die voreingestellte audit_control benutzt wird, dann ist der Eintrag `lo` für `root` überflüssig und Anmelde/Abmelde-Ereignisse werden für `www` ebenfalls aufgezeichnet.

```
root:lo,+ex:no
www:fc,+ex:no
```

# 34.4. Audit-Trails

Weil Audit-Trails werden im binären BSM-Format gespeichert werden, gibt es verschiedene Werkzeuge, um derartige Dateien zu ändern oder sie in Textdateien zu konvertieren. Der Befehl `praudit` wandelt alle Pfad-Dateien in ein einfaches Textformat um. Der Befehl `auditreduce` kann genutzt werden, um die Pfad-Dateien für Analyse, Ausdruck, Archivierung oder andere Zwecke zu

reduzieren. Eine Reihe von Auswahl-Parametern werden von auditreduce(1) unterstützt, einschließlich Ereignistyp, Ereignisklasse, Benutzer, Datum und Uhrzeit des Ereignisses und den Dateipfad oder das Objekt, mit dem gearbeitet wurde.

Der folgende Befehl schreibt den gesamten Inhalt einer angegebenen Audit-Protokolldatei in eine Textdatei:

```
# praudit /var/audit/AUDITFILE
```

*AUDITFILE* ist hier die zu schreibende Protokolldatei.

Audit-Pfade bestehen aus einer Reihe von Datensätzen, die wiederum aus Kürzeln (token) gebildet werden, die von praudit(1) fortlaufend zeilenweise ausgegeben werden. Jedes Kürzel ist von einem bestimmten Typ, z.B. enthält `header` einen audit-Datensatz-Header oder `path` enthält einen Dateipfad von einer Suche. Hier ein Beispiel eines `execve`-Ereignisses:

```
header,133,10,execve(2),0,Mon Sep 25 15:58:03 2006, + 384 msec
exec arg,finger,doug
path,/usr/bin/finger
attribute,555,root,wheel,90,24918,104944
subject,robert,root,wheel,root,wheel,38439,38032,42086,128.232.9.100
return,success,0
trailer,133
```

Dieser Audit stellt einen erfolgreichen `execve`-Aufruf dar, in welchem der Befehl `finger doug` ausgeführt wurde. `exec arg` enthält die Befehlszeile, welche die Shell an den Kernel weiterleitet. Das Kürzel `path` enthält den Pfad zur ausführbaren Datei (wie vom Kernel wahrgenommen). Das Kürzel `attribute` beschreibt die Binärdatei und enthält den Datei-Modus, der genutzt werden kann, um zu bestimmen, ob setuid auf die Applikation angewendet wurde. Das Kürzel `subject` speichert die Audit-Benutzer-ID, effektive Benutzer-ID und Gruppen-ID, wirkliche Benutzer-ID und Gruppen-ID, Prozess-ID, Session- ID, Port-ID und Anmelde-Adresse. Beachten Sie, dass Audit-Benutzer-ID und wirkliche Benutzer-ID abweichen, da der Benutzer `robert` zum Benutzer `root` wurde, bevor er diesen Befehl ausführte, aber er wird auditiert mit dem ursprünglich authentifizierten Benutzer. Das Kürzel `return` zeigt die erfolgreiche Ausführung an und `trailer` schließt den Datensatz ab.

Die Ausgabe im XML-Format wird ebenfalls unterstützt und kann über die Option `-x` ausgewählt werden.

Da Audit-Protokolldateien sehr groß sein können, kann mit Hilfe von `auditreduce` auch nur eine Teilmenge der Datensätze ausgewählt werden. Dieses Beispiel selektiert alle Datensätze des Benutzers `trhodes` aus der Datei AUDITFILE:

```
# auditreduce -u trhodes /var/audit/AUDITFILE | praudit
```

Mitglieder der Gruppe `audit` sind berechtigt, Audit-Pfade in /var/audit zu lesen. In der Voreinstellung ist diese Gruppe leer, daher kann nur der Benutzer `root` die Audit-Pfade lesen.

Benutzer können der Gruppe `audit` hinzugefügt werden, um Rechte für Audit-Reviews zu gewähren. Da die Fähigkeit, Inhalte von Audit-Protokolldateien zu verfolgen, tiefgreifende Einblicke in das Verhalten von Benutzern und Prozessen erlaubt, wird empfohlen, dass die Gewährung von Rechten für Audit-Reviews mit Bedacht erfolgt.

### 34.4.1. Aktive Überwachung mittels Audit-Pipes

Audit-Pipes sind nachgebildete (geklonte) Pseudo-Geräte, welche es Applikationen erlauben, die laufenden Audit-Datensätze anzuzapfen. Dies ist vorrangig für Autoren von Intrusion Detection Software und Systemüberwachungsprogrammen von Bedeutung. Allerdings ist das Audit-Pipe-Gerät ein angenehmer Weg für den Administrator, aktive Überwachung zu gestatten, ohne Gefahr von Problemen durch Besitzerrechte der Audit-Pfad-Datei oder Unterbrechung des Stroms von Ereignissen durch Log-Rotation. Um den laufenden Audit-Ereignisstrom zu verfolgen, geben Sie folgendes ein:

```
# praudit /dev/auditpipe
```

In der Voreinstellung kann nur der Benutzer `root` auf die Audit-Pipe-Geräte-Knotenpunkte zugreifen. Um sie allen Mitgliedern der Gruppe `audit` zugänglich zu machen, fügen Sie eine `devfs` -Regel in /etc/devfs.rules hinzu:

```
add path 'auditpipe*' mode 0440 group audit
```

Lesen Sie devfs.rules(5) für weitere Informationen, wie das devfs-Dateisystem konfiguriert wird.



Es ist sehr leicht, Rückmeldungszyklen von Audit-Ereignissen hervorzurufen, in welcher das Betrachten des Resultates eines Audit-Ereignisses in die Erzeugung von mehr Audit-Ereignissen mündet. Wenn zum Beispiel der gesamte Netzwerk-I/O auditiert wird, während `praudit` in einer SSH-Sitzung gestartet wurde, dann wird ein kontinuierlicher, mächtiger Strom von Audit-Ereignissen erzeugt, da jedes ausgegebene Ereignis wiederum neue Ereignisse erzeugt. Daher ist anzuraten, `praudit` an einem Audit-Pipe-Gerät nur von Sitzungen anzuwenden (ohne feingranuliertes I/O-Auditing), um dies zu vermeiden.

### 34.4.2. Rotation und Komprimierung von Audit-Pfad-Dateien

Audit-Pfade werden vom Kernel geschrieben und vom Audit-Daemon auditd(8) verwaltet. Administratoren sollten nicht versuchen, newsyslog.conf(5) oder andere Werkzeuge zu benutzen, um Audit-Protokolldateien direkt zu rotieren. Stattdessen sollte `audit` benutzt werden, um die Auditierung zu beenden, das Audit-System neu zu konfigurieren und eine Log-Rotation durchzuführen. Der folgende Befehl veranlasst den Audit-Daemon, eine neue Protokolldatei anzulegen und dem Kernel zu signalisieren, die neue Datei zu nutzen. Die alte Datei wird beendet und umbenannt. Ab diesem Zeitpunkt kann sie vom Administrator bearbeitet werden:

```
# audit -n
```

Falls der auditd(8)-Daemon gegenwärtig nicht läuft, wird dieser Befehl scheitern und eine Fehlermeldung wird ausgegeben.

Durch das Hinzufügen der folgenden Zeile in /etc/crontab wird die Log-Rotation alle zwölf Stunden durchgeführt:

```
0     */12      *       *       *       root    /usr/sbin/audit -n
```

Die Änderung wird wirksam, sobald /etc/crontab gespeichert wird.

Die automatische Rotation der Audit-Pfad-Datei in Abhängigkeit von der Dateigröße ist möglich durch die Angabe der Option `filesz` in audit_control. Dieser Vorgang ist in Die audit_control-Datei beschrieben.

Da Audit-Pfad-Dateien sehr groß werden können, ist es oft wünschenswert, Pfade zu komprimieren oder anderweitig zu archivieren, sobald sie vom Audit-Daemon geschlossen wurden. Das Skript audit_warn kann genutzt werden, um angepasste Aktionen für eine Vielzahl von audit-bezogenen Ereignissen auszuführen, einschließlich der sauberen Beendigung von Audit-Pfaden, wenn diese geschlossen werden. Zum Beispiel kann man die folgenden Zeilen in /etc/security/audit_warn aufnehmen, um Audit-Pfade beim Beenden zu komprimieren:

```
#
# Compress audit trail files on close.
#
if [ "$1" = closefile ]; then
        gzip -9 $2
fi
```

Andere Archivierungsaktivitäten können das Kopieren zu einem zentralen Server, die Löschung der alten Pfad-Dateien oder die Reduzierung des alten Audit-Pfades durch Entfernung nicht benötigter Datensätze einschließen. Dieses Skript wird nur dann ausgeführt, wenn die Audit-Pfad-Dateien sauber beendet wurden, daher wird es nicht auf Pfaden laufen, welche durch ein unsauberes Herunterfahren des Systems nicht beendet wurden.

# Kapitel 35. Speichermedien

## 35.1. Übersicht

Dieses Kapitel behandelt die Benutzung von Laufwerken unter FreeBSD. Hierzu zählen SCSI- und IDE-Geräte, CD- und DVD-Medien, speicherbasierte Laufwerke und USB-Geräte.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie Folgendes wissen:

- Wie Sie zusätzliche Laufwerke zu einem FreeBSD-System hinzufügen.
- Wie Sie unter FreeBSD die Partition einer Festplatte vergrößern.
- Wie Sie FreeBSD zur Verwendung von USB-Speichermedien konfigurieren.
- Wie Sie CD- und DVD-Medien unter FreeBSD benutzen.
- Wie Sie die unter FreeBSD erhältlichen Backup-Programme benutzen.
- Wie Sie RAM-Disks einrichten.
- Was Dateisystem-Schnappschüsse sind und wie sie effizient eingesetzt werden.
- Wie Sie mit Quotas die Benutzung von Laufwerken einschränken.
- Wie Sie Festplatten und Swap verschlüsseln, um Daten vor Angreifern zu schützen.
- Wie Sie ein hochverfügbares Speichernetzwerk konfigurieren.

Bevor Sie dieses Kapitel lesen,

- sollten Sie wissen, wie Sie einen neuen FreeBSD-Kernel konfigurieren und installieren.

## 35.2. Hinzufügen von Laufwerken

Dieser Abschnitt beschreibt, wie Sie ein neues SATA-Laufwerk zu einer Maschine hinzufügen, die momentan nur ein Laufwerk hat. Dazu schalten Sie zuerst den Rechner aus und installieren das Laufwerk entsprechend der Anleitungen Ihres Rechners, Ihres Controllers und des Laufwerkherstellers. Starten Sie das System neu und melden Sie sich als Benutzer `root` an.

Kontrollieren Sie /var/run/dmesg.boot, um sicherzustellen, dass das neue Laufwerk gefunden wurde. In diesem Beispiel erscheint das neu hinzugefügte SATA-Laufwerk als ada1.

In diesem Beispiel wird eine einzige große Partition auf der Festplatte erstellt. Verwendet wird das GPT-Partitionsschema, welches gegenüber dem älteren und weniger vielseitigen MBR-Schema bevorzug wird.



Wenn die hinzugefügte Festplatte nicht leer ist, können alte Partitionsinformationen mit `gpart delete` entfernt werden. Details finden Sie in gpart(8).

Zuerst wird das Partitionsschema erstellt und dann eine einzelne Partition angefügt. Zur Verbesserung der Leistung auf neueren Festplatten mit größeren Blockgrößen, wird die Partition

an einer Megabyte-Grenze ausgerichtet:

```
# gpart create -s GPT ada1
# gpart add -t freebsd-ufs -a 1M ada1
```

Je nach Anwendung kann es wünschenswert sein, mehrere kleinere Partitionen zu haben. In gpart(8) finden Sie Optionen zum Erstellen von kleineren Partitionen.

Informationen über die Partitionen der Festplatte werden mit `gpart show` angezeigt:

```
% gpart show ada1
=>        34  1465146988  ada1  GPT  (699G)
          34        2014        - free -  (1.0M)
        2048  1465143296     1  freebsd-ufs  (699G)
  1465145344        1678        - free -  (839K)
```

Ein Dateisystem wird in der neuen Partition erstellt:

```
# newfs -U /dev/ada1p1
```

Ein leeres Verzeichnis wird als Mountpunkt erstellt, also ein Speicherort für die Montage der neuen Festplatte im originalen Dateisystem:

```
# mkdir /newdisk
```

Abschließend wird ein Eintrag in /etc/fstab hinzugefügt, damit die neue Festplatte automatisch beim Start eingehängt wird:

```
/dev/ada1p1 /newdisk      ufs   rw      2     2
```

Die neue Festplatte kann manuell montiert werden, ohne das System neu zu starten:

```
# mount /newdisk
```

## 35.3. Partitionen vergrößern

Die Kapazität einer Festplatte kann sich ohne Änderungen an bereits vorhandenen Daten erhöhen. Dies geschieht üblicherweise mit virtuellen Maschinen, wenn sich herausstellt, dass die virtuelle Festplatte zu klein ist und vergrößert werden soll. Zuweilen wird auch ein Abbild einer Platte auf einen USB-Stick geschrieben, ohne dabei die volle Kapazität zu nutzen. Dieser Abschnitt beschreibt, wie man Platten vergrößert, bzw. *erweitert*, um die Vorteile der erhöhten Kapazität zu nutzen.

Überprüfen Sie /var/run/dmesg.boot, um den Gerätenamen der Festplatte zu bestimmen, die

vergrößert werden soll. In diesem Beispiel gibt es nur eine SATA-Festplatte im System, so dass die Platte als ada0 angezeigt wird.

Um die aktuelle Konfiguration der Partitionen auf der Festplatte anzuzeigen:

```
# gpart show ada0
=>      34  83886013  ada0  GPT  (48G) [CORRUPT]
        34       128     1  freebsd-boot  (64k)
       162  79691648     2  freebsd-ufs  (38G)
  79691810   4194236     3  freebsd-swap  (2G)
  83886046         1        - free -  (512B)
```



Wenn die Festplatte mit dem GPT-Partitionsschema formatiert wurde kann es vorkommen, dass sie als "corrupted" angezeigt wird, weil sich die Sicherung der GPT-Partitionstabellen nicht mehr am Ende des Laufwerks befinden. Reparieren Sie in so einem Fall die Partitionstabelle mit `gpart`:

```
# gpart recover ada0
ada0 recovered
```

Nun steht der zusätzliche Speicherplatz zur Verfügung und kann verwendet werden, um eine neue Partition anzulegen oder eine bestehende Partition zu erweitern:

```
# gpart show ada0
=>       34  102399933  ada0  GPT  (48G)
         34        128     1  freebsd-boot  (64k)
        162   79691648     2  freebsd-ufs  (38G)
   79691810    4194236     3  freebsd-swap  (2G)
   83886046   18513921        - free -  (8.8G)
```

Partitionen können nur auf zusammenhängenden, freien Speicherplatz vergrößert werden. In diesem Beispiel wird die letzte Partition der Platte als Swap-Speicher genutzt, aber die zweite Partition ist die, dessen Größe verändert werden soll. Weil der Swap-Speicher nur temporäre Daten enthält, kann er gefahrlos ausgehangen, gelöscht und nachdem die zweite Partition vergrößert wurde, als dritte Partition neu erstellt werden.

Deaktivieren Sie Swap-Speicher Partition:

```
# swapoff /dev/ada0p3
```

Löschen Sie die dritte Partition, angegeben mit dem Schalter `-i`, der Festplatte *ada0*:

```
# gpart delete -i 3 ada0
ada0p3 deleted
# gpart show ada0
```

```
=>       34  102399933  ada0  GPT  (48G)
         34        128     1  freebsd-boot  (64k)
        162   79691648     2  freebsd-ufs  (38G)
   79691810   22708157        - free -  (10G)
```



Es besteht die Gefahr von Datenverlust, wenn die Partitionstabelle eines eingehangenen Dateisystems verändert wird. Es empfiehlt sich daher, die folgenden Schritte auf einem ausgehangenen Dateisystem durchzuführen, während die Umsetzung über eine Live-CD-ROM oder von einem USB-Gerät erfolgt. Wenn es jedoch absolut notwendig ist, kann ein eingehangenes Dateisystem auch vergrößert werden, nachdem die Sicherheitsfunktionen von GEOM deaktiviert wurden:

```
# sysctl kern.geom.debugflags=16
```

Vergrößern Sie die Partition und lassen Sie Platz, um die Swap-Partition in der gewünschten Größe neu erstellen zu können. Die zu ändernde Partition wird mit `-i` und die neue gewünschte Größe mit `-s` angegeben. Optional wird die Ausrichtung der Partition mit `-a` festgelegt. Dieser Schritt ändert nur die Größe der Partition. Das Dateisystem innerhalb der Partition wird in einem separaten Schritt erweitert.

```
# gpart resize -i 2 -s 47G -a 4k ada0
ada0p2 resized
# gpart show ada0
=>       34  102399933  ada0  GPT  (48G)
         34        128     1  freebsd-boot  (64k)
        162   98566144     2  freebsd-ufs  (47G)
   98566306    3833661        - free -  (1.8G)
```

Erstellen Sie die Swap-Partition neu und aktivieren Sie sie:

```
# gpart add -t freebsd-swap -a 4k ada0
ada0p3 added
# gpart show ada0
=>       34  102399933  ada0  GPT  (48G)
         34        128     1  freebsd-boot  (64k)
        162   98566144     2  freebsd-ufs  (47G)
   98566306    3833661     3  freebsd-swap  (1.8G)
# swapon /dev/ada0p3
```

Erweitern Sie das UFS-Dateisystem, um die Kapazität der vergrößerten Partition zu nutzen:

```
# growfs /dev/ada0p2
Device is mounted read-write; resizing will result in temporary write suspension for
/.
```

```
It's strongly recommended to make a backup before growing the file system.
OK to grow file system on /dev/ada0p2, mounted on /, from 38GB to 47GB? [Yes/No] Yes
super-block backups (for fsck -b #) at:
 80781312, 82063552, 83345792, 84628032, 85910272, 87192512, 88474752,
 89756992, 91039232, 92321472, 93603712, 94885952, 96168192, 97450432
```

Wenn das Dateisystem ZFS ist, wird die Größenänderung mit dem Unterkommando `online` und `-e` ausgelöst:

```
# zfs online -e zroot /dev/ada0p2
```

Sowohl die Partition als auch das Dateisystem wurden jetzt vergrößert, um den neu zur Verfügung stehenden Speicherplatz zu nutzen.

## 35.4. USB Speichermedien

Der Universal Serial Bus (USB) wird von vielen externen Speichern benutzt: Festplatten, USB-Thumbdrives sowie von CD- und DVD-Brennern. FreeBSD bietet Unterstützung für Geräte mit USB 1.x, 2.0 und 3.0.



Die Unterstützung für USB 3.0 ist mit einiger Hardware, einschließlich Haswell (Lynx Point) Chipsätzen, nicht kompatibel. Wenn FreeBSD beim Booten mit dem Fehler `failed with error 19` abbricht, müssen Sie xHCI/USB3 im BIOS deaktivieren.

Unterstützung für USB-Massenspeicher ist im GENERIC-Kernel enthalten. Für einen angepassten Kernel müssen die nachstehenden Zeilen in der Kernelkonfigurationsdatei enthalten sein:

```
device scbus    # SCSI bus (required for ATA/SCSI)
device da   # Direct Access (disks)
device pass # Passthrough device (direct ATA/SCSI access)
device uhci # provides USB 1.x support
device ohci # provides USB 1.x support
device ehci # provides USB 2.0 support
device xhci # provides USB 3.0 support
device usb  # USB Bus (required)
device umass    # Disks/Mass storage - Requires scbus and da
device cd   # needed for CD and DVD burners
```

FreeBSD benutzt den umass(4)-Treiber, der das SCSI-Subsystem verwendet um auf USB-Geräte zuzugreifen. Da alle USB-Geräte vom System als SCSI-Geräte erkannt werden, dürfen Sie *nicht* `device atapicam` in die Kernelkonfigurationsdatei aufnehmen, wenn es sich bei dem Gerät um einen CD- oder DVD-Brenner handelt.

Der übrige Abschnitt beschreibt, wie Sie überprüfen können ob ein USB-Gerät von FreeBSD erkannt wird und wie Sie das Gerät so konfigurieren, dass es verwendet werden kann.

### 35.4.1. Konfiguration von Geräten

Um die USB-Konfiguration zu testen, schließen Sie das USB-Gerät an. Verwenden Sie `dmesg` um zu überprüfen, ob das Gerät in den Systemmeldungen erscheint. Dies sollte in etwa so aussehen:

```
umass0: <STECH Simple Drive, class 0/0, rev 2.00/1.04, addr 3> on usbus0
umass0:  SCSI over Bulk-Only; quirks = 0x0100
umass0:4:0:-1: Attached to scbus4
da0 at umass-sim0 bus 0 scbus4 target 0 lun 0
da0: <STECH Simple Drive 1.04> Fixed Direct Access SCSI-4 device
da0: Serial Number WD-WXE508CAN263
da0: 40.000MB/s transfers
da0: 152627MB (312581808 512 byte sectors: 255H 63S/T 19457C)
da0: quirks=0x2<NO_6_BYTE>
```

Fabrikat, Gerätedatei (da0), Geschwindigkeit und Kapazität werden je nach Gerät unterschiedlich sein.

Da ein USB-Gerät als SCSI-Gerät erkannt wird, kann `camcontrol` benutzt werden, um die mit dem System verbundenen USB-Massenspeicher anzuzeigen:

```
# camcontrol devlist
<STECH Simple Drive 1.04>          at scbus4 target 0 lun 0 (pass3,da0)
```

Alternativ kann `usbconfig` benutzt werden, um die Geräte aufzulisten. Weitere Informationen zu diesem Kommando finden Sie in usbconfig(8).

```
# usbconfig
ugen0.3: <Simple Drive STECH> at usbus0, cfg=0 md=HOST spd=HIGH (480Mbps) pwr=ON (2mA)
```

Wenn das Gerät noch nicht formatiert ist, finden Sie in Hinzufügen von Laufwerken Informationen, wie Sie USB-Laufwerke formatieren und Partitionen einrichten. Wenn das Laufwerk bereits ein Dateisystem enthält, kann es von `root` nach den Anweisungen in “Anhängen und Abhängen von Dateisystemen” eingehängt werden.



Aus Sicherheitsgründen sollten Sie Benutzern, denen Sie nicht vertrauen, das Einhängen (z.B. durch die unten beschriebene Aktivierung von `vfs.usermount`) beliebiger Medien verbieten. Die meisten Dateisysteme wurden nicht entwickelt, um sich vor böswilligen Geräten zu schützen.

Um auch normalen Anwendern das Einhängen des Laufwerks zu gestatten, könnten Sie beispielsweise mit pw(8) alle potentiellen Benutzer dieser Gerätedateien in die Gruppe `operator` aufnehmen. Außerdem muss sichergestellt werden, dass `operator` Schreib- und Lesezugriff auf diese Gerätedateien haben. Hierfür werden die folgenden Zeilen in /etc/devfs.rules hinzugefügt:

```
[localrules=5]
```

```
add path 'da*' mode 0660 group operator
```



Verfügt das System über interne SCSI-Laufwerke, so verändern Sie die zweite Zeile wie folgt:

```
add path 'da[3-9]*' mode 0660 group operator
```

Dies wird die ersten drei SCSI-Laufwerke (da0 bis da2) davon ausschließen, in die Gruppe `operator` aufgenommen zu werden. Ersetzen Sie `3` durch die Anzahl der SCSI-Laufwerke. Weitere Informationen zu dieser Datei finden Sie in devfs.rules(5).

Aktivieren Sie nun die Regeln in /etc/rc.conf:

```
devfs_system_ruleset="localrules"
```

Als nächstes müssen Sie das System anweisen, auch normalen Benutzern das mounten von Dateisystemen zu erlauben, indem Sie die folgende Zeile in /etc/sysctl.conf hinzufügen:

```
vfs.usermount=1
```

Da diese Einstellung erst nach einem Neustart wirksam wird, können Sie diese Variable mit `sysctl` auch direkt setzen:

```
# sysctl vfs.usermount=1
vfs.usermount: 0 -> 1
```

Zuletzt müssen Sie noch ein Verzeichnis anlegen, in das das USB-Laufwerk eingehängt werden soll. Dieses Verzeichnis muss dem Benutzer gehören, der das USB-Laufwerk in den Verzeichnisbaum einhängen will. Dazu legen Sie als `root` ein Unterverzeichnis /mnt/username an, wobei Sie *username* durch den Login des jeweiligen Benutzers sowie *usergroup* durch die primäre Gruppe des Benutzers ersetzen:

```
# mkdir /mnt/username
# chown username:usergroup /mnt/username
```

Wenn Sie nun beispielsweise einen USB-Stick anschließen, wird automatisch die Gerätedatei /dev/da0s1 erzeugt. Ist das Gerät mit einem FAT-Dateisystem formatiert, kann es der Benutzer mit dem folgenden Befehl in den Verzeichnisbaum einhängen:

```
% mount -t msdosfs -o -m=644,-M=755 /dev/da0s1 /mnt/username
```

Bevor das Gerät entfernt werden kann, *muss* es abgehängt werden:

```
# umount /mnt/username
```

Nach Entfernen des Geräts stehen in den Systemmeldungen Einträge, ähnlich der folgenden:

```
umass0: at uhub3, port 2, addr 3 (disconnected)
da0 at umass-sim0 bus 0 scbus4 target 0 lun 0
da0: <STECH Simple Drive 1.04> s/n WD-WXE508CAN263          detached
(da0:umass-sim0:0:0:0): Periph destroyed
```

### 35.4.2. Automatisches Einhängen von Wechselmedien

Damit USB-Geräte automatisch eingehängt werden, muss der Kommentar für folgende Zeile in /etc/auto_master entfernt werden:

```
/media      -media      -nosuid
```

Anschließend fügen Sie folgende Zeilen in /etc/devd.conf hinzu:

```
notify 100 {
    match "system" "GEOM";
    match "subsystem" "DEV";
    action "/usr/sbin/automount -c";
};
```

Falls autofs(5) und devd(8) bereits ausgeführt werden, müssen Sie die Konfiguration neu einlesen:

```
# service automount restart
# service devd restart
```

autofs(5) wird beim Booten automatisch gestartet, wenn Sie folgende Zeile in /etc/rc.conf hinzufügen:

```
autofs_enable="YES"
```

Damit autofs(5) funktioniert, muss devd(8) aktiviert sein, was aber in der Voreinstellung der Fall ist.

Starten Sie jetzt die Dienste:

```
# service automount start
# service automountd start
# service autounmountd start
```

```
# service devd start
```

Jedes Dateisystem, das automatisch eingehängt werden kann, erscheint als ein Verzeichnis unterhalb von media. Das Verzeichnis wird nach dem Dateisystemlabel benannt, bzw. nach dem Gerätenamen, falls kein Label existiert.

Das Dateisystem wird transparent beim ersten Zugriff in den Verzeichnisbaum eingehängt und auch nach gewisser Zeit der Inaktivität wieder ausgehängt. Laufwerke können auch manuell ausgehängt werden:

```
# automount -fu
```

Diese Methode wird in der Regel bei Speicherkarten und USB-Sticks verwendet. Sie funktioniert aber mit allen Blockgeräten, einschließlich optischen Laufwerken und iSCSI-LUNs.

# 35.5. Erstellen und Verwenden von CDs

CDs besitzen einige Eigenschaften, die sie von konventionellen Laufwerken unterscheiden. Sie wurden so entworfen, dass sie ununterbrochen, ohne Verzögerungen durch Kopfbewegungen zwischen den Spuren, gelesen werden können. CDs besitzen Spuren, aber damit ist der Teil Daten gemeint, der ununterbrochen gelesen wird, und nicht eine physikalische Eigenschaft der CD. Das ISO 9660-Dateisystem wurde entworfen, um mit diesen Unterschieden umzugehen.

Die FreeBSD Ports-Sammlung bietet einige Werkzeuge zum Brennen und Kopieren von Audio- und Daten-CDs. Dieses Kapitel beschreibt die Verwendung von mehreren Kommandozeilen-Werkzeugen. Wenn Sie eine graphische Oberfläche zum Brennen von CDs benutzen, können Sie sysutils/xcdroast oder sysutils/k3b installieren.

## 35.5.1. Unterstützte Geräte

Der GENERIC-Kernel enthält Unterstützung für SCSI, USB und ATAPICD Lesegeräte und Brenner. Wird ein angepasster Kernel erstellt, unterscheiden sich die Optionen für die Kernelkonfigurationsdatei je nach Art des Geräts.

Für einen SCSI-Brenner müssen folgende Optionen vorhanden sein:

```
device scbus      # SCSI bus (required for ATA/SCSI)
device da     # Direct Access (disks)
device pass   # Passthrough device (direct ATA/SCSI access)
device cd     # needed for CD and DVD burners
```

Für einen USB-Brenner müssen folgende Optionen vorhanden sein:

```
device scbus      # SCSI bus (required for ATA/SCSI)
device da     # Direct Access (disks)
device pass   # Passthrough device (direct ATA/SCSI access)
```

```
device cd>     # needed for CD and DVD burners
device uhci   # provides USB 1.x support
device ohci   # provides USB 1.x support
device ehci   # provides USB 2.0 support
device xhci   # provides USB 3.0 support
device usb    # USB Bus (required)
device umass      # Disks/Mass storage - Requires scbus and da
```

Für einen ATAPI-Brenner müssen folgende Optionen vorhanden sein:

```
device ata    # Legacy ATA/SATA controllers
device scbus      # SCSI bus (required for ATA/SCSI)
device pass   # Passthrough device (direct ATA/SCSI access)
device cd     # needed for CD and DVD burners
```



Unter FreeBSD Versionen kleiner 10.x wird auch diese Option in der Kernelkonfigurationsdatei benötigt, falls der Brenner ein ATAPI-Gerät ist:

```
device atapicam
```

Alternativ kann folgende Zeile in /boot/loader.conf hinzugefügt werden, um den Treiber beim Booten automatisch zu laden:

```
atapicam_load="YES"
```

Hierzu ist ein Neustart des Systems erforderlich, da dieser Treiber nur beim Booten geladen werden kann.

Mit `dmesg` können Sie prüfen, ob das Gerät von FreeBSD erkannt wurde. Unter FreeBSD Versionen kleiner 10.x lautet der Gerätename acd0 anstelle von cd0.

```
% dmesg | grep cd
cd0 at ahcich1 bus 0 scbus1 target 0 lun 0
cd0: <HL-DT-ST DVDRAM GU70N LT20> Removable CD-ROM SCSI-0 device
cd0: Serial Number M3OD3S34152
cd0: 150.000MB/s transfers (SATA 1.x, UDMA6, ATAPI 12bytes, PIO 8192bytes)
cd0: Attempt to query device size failed: NOT READY, Medium not present - tray closed
```

### 35.5.2. Eine CD brennen

Unter FreeBSD kann `cdrecord` zum Brennen von CDs benutzt werden. Dieses Programm wird aus dem Port oder Paket sysutils/cdrtools installiert.

Obwohl `cdrecord` viele Optionen besitzt, ist die grundlegende Benutzung sehr einfach. Geben Sie den Namen der zu brennenden ISO-Datei an. Wenn das System über mehrere Brenner verfügt,

müssen Sie auch den Namen des Gerätes angeben:

```
# cdrecord dev=device imagefile.iso
```

Benutzen Sie `-scanbus` um den Gerätenamen des Brenners zu bestimmen. Die Ausgabe könnte wie folgt aussehen:

```
# cdrecord -scanbus
ProDVD-ProBD-Clone 3.00 (amd64-unknown-freebsd10.0) Copyright (C) 1995-2010 Jörg
Schilling
Using libscg version 'schily-0.9'
scsibus0:
        0,0,0     0) 'SEAGATE ' 'ST39236LW       ' '0004' Disk
        0,1,0     1) 'SEAGATE ' 'ST39173W        ' '5958' Disk
        0,2,0     2) *
        0,3,0     3) 'iomega  ' 'jaz 1GB         ' 'J.86' Removable Disk
        0,4,0     4) 'NEC     ' 'CD-ROM DRIVE:466' '1.26' Removable CD-ROM
        0,5,0     5) *
        0,6,0     6) *
        0,7,0     7) *
scsibus1:
        1,0,0   100) *
        1,1,0   101) *
        1,2,0   102) *
        1,3,0   103) *
        1,4,0   104) *
        1,5,0   105) 'YAMAHA  ' 'CRW4260        ' '1.0q' Removable CD-ROM
        1,6,0   106) 'ARTEC   ' 'AM12S          ' '1.06' Scanner
        1,7,0   107) *
```

Benutzen Sie die drei durch Kommas separierten Zahlen, die für den CD-Brenner angegeben sind, als Argument für `dev`. Im Beispiel ist das Yamaha-Gerät `1,5,0`, so dass die passende Eingabe `dev=1,5,0` ist. Einfachere Wege das Argument anzugeben, sowie Informationen über Audiospuren und das Einstellen der Geschwindigkeit, sind in der Manualpage von `cdrecord` beschrieben.

Alternativ können Sie den folgenden Befehl ausführen, um die Geräteadresse des Brenners zu ermitteln:

```
# camcontrol devlist
<MATSHITA CDRW/DVD UJDA740 1.00>   at scbus1 target 0 lun 0 (cd0,pass0)
```

Verwenden Sie die numerischen Werte für `scbus`, `target` und `lun`. Für dieses Beispiel wäre `1,0,0` als Gerätename zu verwenden.

### 35.5.3. Daten auf ISO-Dateisystem schreiben

Die Datendateien müssen vorbereitet sein, bevor sie auf eine CD gebrannt werden. In FreeBSD wird

`mkisofs` vom Paket oder Port sysutils/cdrtools installiert. Dieses Programm kann aus einem UNIX® Verzeichnisbaum ein ISO 9660-Dateisystem erzeugen. Im einfachsten Fall müssen Sie lediglich den Namen der zu erzeugenden ISO-Datei und den Pfad zu den Dateien angeben, die auf dem ISO 9660-Dateisystem platziert werden:

```
# mkisofs -o imagefile.iso /path/to/tree
```

Bei diesem Kommando werden die Dateinamen auf Namen abgebildet, die den Restriktionen des ISO 9660-Dateisystem entsprechen. Dateien, die diesem Standard nicht entsprechen bleiben unberücksichtigt.

Es gibt einige Optionen, um die Beschränkungen dieses Standards zu überwinden. Die unter UNIX® Systemen üblichen Rock-Ridge-Erweiterungen werden durch `-R` aktiviert und `-J` aktiviert die von Microsoft® Systemen benutzten Joliet-Erweiterungen.

Für CDs, die nur auf FreeBSD-Systemen verwendet werden sollen, kann `-U` genutzt werden, um alle Beschränkungen für Dateinamen aufzuheben. Zusammen mit `-R` wird ein Abbild des Dateisystems, identisch zu angegebenen FreeBSD-Dateibaum erstellt, selbst wenn dies den ISO 9660 Standard verletzt.

Die letzte übliche Option ist `-b`. Sie wird benutzt, um den Ort eines Bootimages einer "El Torito" bootbaren CD anzugeben. Das Argument zu dieser Option ist der Pfad zu einem Bootimage ausgehend von der Wurzel des Baumes, der auf die CD geschrieben werden soll. In der Voreinstellung erzeugt `mkisofs` ein ISO-Image im "Diskettenemulations"-Modus. Dabei muss das Image genau 1200, 1440 oder 2880 KB groß sein. Einige Bootloader, darunter der auf den FreeBSD Installationsmedien verwendete, kennen keinen Emulationsmodus. Daher sollte in diesen Fällen `-no-emul-boot` verwendet werden. Wenn /tmp/myboot ein bootbares FreeBSD-System enthält, dessen Bootimage sich in /tmp/myboot/boot/cdboot befindet, dann würde folgendes Kommando /tmp/bootable.iso erstellen:

```
# mkisofs -R -no-emul-boot -b boot/cdboot -o /tmp/bootable.iso /tmp/myboot
```

Das resultierende ISO-Abbild kann als speicherbasiertes Laufwerk eingehängt werden:

```
# mdconfig -a -t vnode -f /tmp/bootable.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

Jetzt können Sie überprüfen, dass /mnt und /tmp/myboot identisch sind.

Sie können das Verhalten von `mkisofs` mit einer Vielzahl von Optionen beeinflussen. Details dazu entnehmen Sie bitte mkisofs(8).



Es ist möglich eine Daten-CD in eine Datei zu kopieren, die einem Image entspricht, das mit `mkisofs` erstellt wurde. Verwenden Sie dazu `dd` mit dem Gerätenamen als Eingabedatei und den Namen der ISO als Ausgabedatei:

```
# dd if=/dev/cd0 of=file.iso bs=2048
```

Das resultierende Abbild kann auf eine CD gebrannt werden, wie in Eine CD brennen beschrieben.

### 35.5.4. Einhängen von Daten-CDs

Sobald ein Abbild auf eine CD gebrannt wurde, kann es durch Angabe des Dateisystemtyp, des CD-Laufwerks und des Mountpunktes eingehangen werden:

```
# mount -t cd9660 /dev/cd0 /mnt
```

Da `mount` davon ausgeht, dass ein Dateisystem vom Typ `ufs` ist, würde die Fehlermeldung `Incorrect super block` erscheinen, wenn Sie beim Einhängen einer Daten-CD auf die Angabe `-t cd9660` verzichten.

Auf diese Weise können Daten-CDs von jedem Hersteller verwendet werden. Es kann allerdings zu Problemen mit CDs kommen, die verschiedene ISO 9660-Erweiterungen benutzen. So speichern Joliet-CDs alle Dateinamen unter Verwendung von zwei Byte langen Unicode-Zeichen. Tauchen statt bestimmter Zeichen nur Fragezeichen auf, so muss über die Option `-C` der benötigte Zeichensatz angegeben werden. Weitere Informationen zu diesem Problem finden Sie in mount_cd9660(8).



Damit der Kernel diese Zeichenkonvertierung (festgelegt durch die Option `-C`) erkennt, müssen Sie das Kernelmodul cd9660_iconv.ko laden. Dazu fügen Sie folgende Zeile in loader.conf ein:

```
cd9660_iconv_load="YES"
```

Danach müssen Sie allerdings Ihr System neu starten. Alternativ können Sie das Kernelmodul auch direkt über `kldload` laden.

Manchmal werden Sie die Meldung `Device not configured` erhalten, wenn Sie versuchen, eine Daten-CD einzuhängen. Für gewöhnlich liegt das daran, dass das Laufwerk keine CD erkannt hat, oder dass das Laufwerk auf dem Bus nicht erkannt wird. Es kann einige Sekunden dauern, bevor das Laufwerk die CD erkennt. Seien Sie also geduldig.

Manchmal wird ein SCSI-CD nicht erkannt, weil es keine Zeit hatte, auf das Zurücksetzen des Busses zu antworten. Um dieses Problem zu lösen, fügen Sie die folgende Zeile in die Kernelkonfiguration ein und erstellen Sie einen angepassten Kernel nach den Anweisungen in “Einen angepassten Kernel bauen und installieren”:

```
options SCSI_DELAY=15000
```

Die Zeile bewirkt, dass nach dem Zurücksetzen des SCSI-Busses beim Booten 15 Sekunden gewartet

wird, um dem CD-Laufwerk genügend Zeit zu geben, darauf zu antworten.



Es ist möglich eine Datei auch direkt auf eine CD zu brennen, ohne vorher auf ihr ein ISO 9660-Dateisystem einzurichten. Man sagt auch, Daten werden roh auf die CD gebrannt. Einige Leute nutzen dies, um Datensicherungen durchzuführen.

Eine auf diese Weise gefertigte Daten-CD kann nicht in das Dateisystem eingehangen werden. Um auf die Daten einer solchen CD zuzugreifen, müssen die Daten vom rohen Gerät gelesen werden. Beispielsweise würde dieser Befehl eine komprimierte tar-Datei auf dem zweiten CD-Laufwerk in das aktuelle Verzeichnis extrahieren:

```
# tar xzvf /dev/cd1
```

Um eine Daten-CD in das System einzuhängen, müssen die Daten mit `mkisofs` geschrieben werden.

### 35.5.5. Kopieren von Audio-CDs

Um eine Kopie einer Audio-CD zu erstellen, kopieren Sie die Stücke der CD in einzelne Dateien und brennen diese Dateien dann auf eine leere CD.

Procedure: Eine Audio-CD kopieren beschreibt, wie eine Audio-CD kopiert und gebrannt wird. Wenn die Version älter als FreeBSD 10.0 ist und ein ATAPI-Gerät verwendet wird, muss zunächst das Modul `atapicam` nach den Anweisungen in Unterstützte Geräte geladen werden.

*Procedure: Eine Audio-CD kopieren*

1. Der Port oder das Paket sysutils/cdrtools installiert `cdda2wav`. Mit diesem Kommando können Audiodaten in das aktuelle Verzeichnis extrahiert werden, wobei jede Datei in eine separate WAV-Datei geschrieben wird:

   ```
   % cdda2wav -vall -B -Owav
   ```

   Wenn das System nur über ein CD-Laufwerk verfügt, muss der Gerätename nicht angegeben werden. Lesen Sie die Manualpage von `cdda2wav` für Anweisungen, wie ein Gerät spezifiziert wird und weitere verfügbare Optionen für dieses Kommando.

2. Die erzeugten .wav Dateien schreiben Sie mit `cdrecord` auf eine leere CD:

   ```
   % cdrecord -v dev=2,0 -dao -useinfo *.wav
   ```

   Das Argument von `dev` gibt das verwendete Gerät an, das wie in Eine CD brennen ermittelt werden kann.

# 35.6. DVDs benutzen

Nach der CD ist die DVD die nächste Generation optischer Speichermedien. Auf einer DVD können mehr Daten als auf einer CD gespeichert werden. DVDs werden als Standardmedium für Videos verwendet.

Für beschreibbare DVDs existieren fünf Medienformate:

- DVD-R: Dies war das erste verfügbare Format. Das Format wurde vom DVD-Forum festgelegt. Die Medien sind nur einmal beschreibbar.
- DVD-RW: Dies ist die wiederbeschreibbare Version des DVD-R Standards. Eine DVD-RW kann ungefähr 1000 Mal beschrieben werden.
- DVD-RAM: Dies ist ein wiederbeschreibbares Format, das wie ein Wechsellaufwerk betrachtet werden kann. Allerdings sind die Medien nicht kompatibel zu den meisten DVD-ROM-Laufwerken und DVD-Video-Spielern, da das DVD-RAM-Format nur von wenigen Brennern unterstützt wird. Informationen zur Nutzung von DVD-RAM finden Sie in DVD-RAM.
- DVD+RW: Ist ein wiederbeschreibbares Format, das von der DVD+RW Alliance festgelegt wurde. Eine DVD+RW kann ungefähr 1000 Mal beschrieben werden.
- DVD+R: Dieses Format ist die nur einmal beschreibbare Variante des DVD+RW Formats.

Auf einer einfach beschichteten DVD können 4.700.000.000 Bytes gespeichert werden. Das sind 4,38 GB oder 4485 MB (1 Kilobyte sind 1024 Bytes).



Die physischen Medien sind unabhängig von der Anwendung. Ein DVD-Video ist eine spezielle Anordnung von Dateien, die auf irgendein Medium, beispielsweise DVD-R, DVD+R oder DVD-RW geschrieben werden kann. Bevor Sie ein Medium auswählen, müssen Sie sicherstellen, dass der Brenner und der DVD-Spieler mit dem Medium umgehen können.

## 35.6.1. Konfiguration

Benutzen Sie growisofs(1), um DVDs zu beschreiben. Das Kommando ist Bestandteil von sysutils/dvd+rw-tools, und kann mit allen DVD-Medien umgehen.

Diese Werkzeuge verwenden das SCSI-Subsystem, um auf die Geräte zuzugreifen. Daher muss ATAPI/CAM-Unterstützung geladen, oder statisch in den Kernel kompiliert werden. Sollte der Brenner jedoch die USB-Schnittstelle nutzen, wird diese Unterstützung nicht benötigt. Weitere Informationen zur Konfiguration von USB-Geräten finden Sie in USB Speichermedien.

Für ATAPI-Geräte müssen ebenfalls DMA-Zugriffe aktiviert werden. Dazu wird die folgende Zeile in /boot/loader.conf eingefügt:

```
hw.ata.atapi_dma="1"
```

Bevor Sie dvd+rw-tools benutzen, lesen Sie bitte die Hardware-Informationen auf der Seite Hardware Compatibility Notes.



Für eine grafische Oberfläche sollten Sie sich sysutils/k3b ansehen, das eine benutzerfreundliche Schnittstelle zu growisofs(1) und vielen anderen Werkzeugen bietet.

### 35.6.2. Daten-DVDs brennen

growisofs(1) erstellt mit dem Programm mkisofs das Dateisystem und brennt anschließend die DVD. Vor dem Brennen braucht daher kein Abbild der Daten erstellt zu werden.

Wenn Sie von den Daten im Verzeichnis /path/to/data eine DVD+R oder eine DVD-R brennen wollen, benutzen Sie das nachstehende Kommando:

```
# growisofs -dvd-compat -Z /dev/cd0 -J -R /path/to/data
```

In diesem Beispiel wird `-J -R` an mkisofs(8) durchgereicht und dient zum Erstellen des Dateisystems (hier: ein ISO-9660-Dateisystem mit Joliet- und Rock-Ridge-Erweiterungen). Weiteres entnehmen Sie bitte der Hilfeseite mkisofs(8).

Die Option `-Z` wird für die erste Aufnahme einer Single- oder Multisession benötigt. Ersetzen Sie */dev/cd0* mit dem Gerätenamen des DVD-Gerätes. Die Nutzung von `-dvd-compat` schließt das Medium, weitere Daten können danach nicht mehr angehängt werden. Dies sollte auch eine bessere Kompatibilität mit anderen DVD-ROM-Laufwerken bieten.

Um ein vorher erstelltes Abbild der Daten zu brennen, beispielsweise *imagefile.iso*, verwenden Sie:

```
# growisofs -dvd-compat -Z /dev/cd0=imagefile.iso
```

Die Schreibgeschwindigkeit hängt von den verwendeten Medium sowie dem verwendeten Gerät ab und sollte automatisch gesetzt werden. Um die Schreibgeschwindigkeit vorzugeben, verwenden Sie `-speed=`. Beispiele finden Sie in growisofs(1).



Um größere Dateien als 4.38GB zu unterstützen, ist es notwendig ein UDF/ISO-9660 Hybrid-Dateisystem zu erstellen. Dieses Dateisystem muss mit zusätzlichen Parametern `-udf -iso-level 3` bei mkisofs(8) und allen relevanten Programmen, wie beispielsweise growisofs(1)) erzeugt werden. Dies ist nur notwendig, wenn Sie ein ISO-Image erstellen oder direkt auf eine DVD schreiben wollen. DVDs, die in dieser Weise hergestellt worden sind, müssen als UDF-Dateisystem mit mount_udf(8) eingehangen werden. Sie sind nur auf Betriebssystemen, die UDF unterstützen brauchbar, ansonsten sieht es so aus, als ob sie kaputte Dateien enthalten würden.

Um diese Art von ISO-Datei zu erstellen:

```
% mkisofs -R -J -udf -iso-level 3 -o imagefile.iso /path/to/data
```

Um Daten direkt auf eine DVD zu brennen, geben Sie den folgenden Befehl ein:

```
# growisofs -dvd-compat -udf -iso-level 3 -Z /dev/cd0 -J -R
/path/to/data
```

Wenn ein ISO-Abbild bereits große Dateien enthält, sind keine weiteren Optionen für growisofs(1) notwendig, um das Abbild auf die DVD zu brennen.

Achten Sie darauf, eine aktuelle Version von sysutils/cdrtools zu verwenden, welche mkisofs(8) enthält, da ältere Versionen keinen Support für große Dateien enthalten. Falls die neueste Version nicht funktioniert, installieren Sie sysutils/cdrtools-devel und lesen Sie mkisofs(8).

### 35.6.3. DVD-Videos brennen

Ein DVD-Video ist eine spezielle Anordnung von Dateien, die auf den ISO-9660 und den micro-UDF (M-UDF) Spezifikationen beruht. Da DVD-Video auf eine bestimmte Datei-Hierarchie angewiesen ist, müssen DVDs mit speziellen Programmen wie multimedia/dvdauthor erstellt werden.

Ist bereits ein Abbild des Dateisystems eines DVD-Videos vorhanden, kann es auf die gleiche Weise wie jedes andere Abbild gebrannt werden. Wenn `dvdauthor` verwendet wurde, um die DVD zu erstellen und die Resultate in /path/to/video liegen, kann das folgende Kommando verwendet werden, um ein DVD-Video zu brennen:

```
# growisofs -Z /dev/cd0 -dvd-video /path/to/video
```

`-dvd-video` wird an mkisofs(8) weitergereicht, um die Datei-Hierarchie für ein DVD-Video zu erstellen. Weiterhin bewirkt diese Option, dass growisofs(1) mit `-dvd-compat` aufgerufen wird.

### 35.6.4. DVD+RW-Medien benutzen

Im Gegensatz zu CD-RW-Medien müssen DVD+RW-Medien erst formatiert werden, bevor sie benutzt werden können. Es wird *empfohlen* growisofs(1) einzusetzen, da das Programm Medien automatisch formatiert, wenn es erforderlich ist. Es ist jedoch möglich, auch `dvd+rw-format` zu nutzen, um die DVD+RW zu formatieren:

```
# dvd+rw-format /dev/cd0
```

Dieser Vorgang muss nur einmal durchgeführt werden. Denken Sie daran, dass nur neue DVD+RWs formatiert werden müssen. Anschließend können DVD+RWs, wie gewohnt gebrannt werden.

Wenn Sie auf einer DVD+RW ein neues Dateisystem erstellen wollen, brauchen Sie die DVD+RW vorher nicht zu löschen. Überschreiben Sie einfach das vorige Dateisystem indem Sie eine neue Session anlegen:

```
# growisofs -Z /dev/cd0 -J -R /path/to/newdata
```

Das DVD+RW-Format erlaubt es, Daten an eine vorherige Aufnahme anzuhängen. Dazu wird eine neue Session mit der schon bestehenden zusammengeführt. Es wird keine Multi-Session geschrieben, sondern growisofs(1) *vergrößert* das ISO-9660-Dateisystem auf dem Medium.

Das folgende Kommando fügt weitere Daten zu einer vorher erstellten DVD+RW hinzu:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Wenn Sie eine DVD+RW erweitern, verwenden Sie dieselben mkisofs(8)-Optionen wie beim Erstellen der DVD+RW.



Verwenden Sie `-dvd-compat`, um bessere Kompatibilität mit DVD-ROM-Laufwerken zu gewährleisten. Zu einem DVD+RW-Medium können Sie mit dieser Option auch weiterhin Daten hinzufügen.

Um das Medium zu löschen, verwenden Sie:

```
# growisofs -Z /dev/cd0=/dev/zero
```

### 35.6.5. DVD-RW-Medien benutzen

Eine DVD-RW kann mit zwei Methoden beschrieben werden: *Sequential-Recording* oder *Restricted-Overwrite*. Voreingestellt ist Sequential-Recording.

Eine neue DVD-RW kann direkt beschrieben werden; sie muss nicht vorher formatiert werden. Allerdings muss eine DVD-RW, die mit Sequential-Recording aufgenommen wurde, zuerst gelöscht werden, bevor eine neue Session aufgenommen werden kann.

Der folgende Befehl löscht eine DVD-RW im Sequential-Recording-Modus:

```
# dvd+rw-format -blank=full /dev/cd0
```



Das vollständige Löschen mit `-blank=full` dauert mit einem 1x Medium ungefähr eine Stunde. Wenn die DVD-RW im Disk-At-Once-Modus (DAO) aufgenommen wurde, kann sie mit `-blank` schneller gelöscht werden. Um eine DVD-RW im DAO-Modus zu brennen, benutzen Sie das folgende Kommando:

```
# growisofs -use-the-force-luke=dao -Z /dev/cd0=imagefile.iso
```

Die Option `-use-the-force-luke=dao` sollte nicht erforderlich sein, da growisofs(1) den DAO-Modus automatisch erkennt.

Der Restricted-Overwrite-Modus sollte mit jeder DVD-RW verwendet werden, da er flexibler als der voreingestellte Sequential-Recording-Modus ist.

Um Daten auf eine DVD-RW im Sequential-Recording-Modus zu schreiben, benutzen Sie dasselbe Kommando wie für die anderen DVD-Formate:

```
# growisofs -Z /dev/cd0 -J -R /path/to/data
```

Um weitere Daten zu einer Aufnahme hinzuzufügen, benutzen Sie `-M` mit growisofs(1). Werden die Daten im Sequential-Recording-Modus hinzugefügt, wird eine neue Session erstellt. Das Ergebnis ist ein Multi-Session-Medium.

Eine DVD-RW im Restricted-Overwrite-Modus muss nicht gelöscht werden, um eine neue Session aufzunehmen. Das Medium kann einfach mit `-Z` überschrieben werden. Mit `-M` kann das ISO-9660-Dateisystem, wie mit einer DVD+RW, vergrößert werden. Die DVD enthält danach eine Session.

Benutzen sie das nachstehende Kommando, um den Restricted-Overwrite-Modus einzustellen:

```
# dvd+rw-format /dev/cd0
```

Das folgende Kommando stellt den Modus wieder auf Sequential-Recording zurück:

```
# dvd+rw-format -blank=full /dev/cd0
```

### 35.6.6. Multi-Session

Nur wenige DVD-ROM-Laufwerke unterstützen Multi-Session-DVDs und lesen meist nur die erste Session. Mehrere Sessions werden von DVD+R, DVD-R und DVD-RW im Sequential-Recording-Modus unterstützt. Im Modus Restricted-Overwrite gibt nur eine Session.

Wenn das Medium noch nicht geschlossen ist, erstellt das nachstehende Kommando eine neue Session auf einer DVD+R, DVD-R oder DVD-RW im Sequential-Recording-Modus:

```
# growisofs -M /dev/cd0 -J -R /path/to/nextdata
```

Wird dieses Kommando mit DVD+RW- oder DVD-RW-Medien im Restricted-Overwrite-Modus benutzt, werden die neuen Daten mit den Daten der bestehenden Session zusammengeführt. Das Medium enthält danach eine Session. Nutzen Sie diese Methode, um neue Daten zu einer bestehenden Session hinzuzufügen.



Für den Anfang und das Ende einer Session wird auf dem Medium zusätzlicher Platz verbraucht. Um den Speicherplatz auf dem Medium optimal auszunutzen, sollten Sie daher Sessions mit vielen Daten hinzufügen. Auf ein DVD+R-Medium passen maximal 154 Sessions, 2000 Sessions auf ein DVD-R-Medium und 127 Sessions auf eine DVD+R Double Layer.

### 35.6.7. Weiterführendes

`dvd+rw-mediainfo /dev/cd0` zeigt Informationen über eine im Laufwerk liegende DVD an.

Weiteres zu dvd+rw-tools finden Sie in growisofs(1), auf der dvd+rw-tools Web-Seite und in den Archiven der cdwrite-Mailingliste.



Wenn Sie einen Problembericht zur Nutzung der dvd+rw-tools erstellen, fügen Sie immer die Ausgabe von `dvd+rw-mediainfo` hinzu.

### 35.6.8. DVD-RAM

DVD-RAM-fähige Brenner nutzten die SCSI- oder ATAPI-Schnittstelle. Für ATAPI-Geräte muss der DMA-Modus aktiviert werden, indem die folgende Zeile in /boot/loader.conf hinzugefügt wird:

```
hw.ata.atapi_dma="1"
```

Eine DVD-RAM kann mit einer Wechselplatte verglichen werden. Wie diese, muss auch eine DVD-RAM vor dem ersten Einsatz formatiert werden. In diesem Beispiel wird das gesamte Medium mit dem Standard-UFS2-Dateisystem formatiert:

```
# dd if=/dev/zero of=/dev/acd0 bs=2k count=1
# bsdlabel -Bw acd0
# newfs /dev/acd0
```

Denken Sie dabei daran, dass Sie gegebenenfalls die Gerätedatei (hier acd0) an Ihre Konfiguration anpassen müssen.

Nachdem die DVD-RAM formatiert ist, kann sie wie eine normale Festplatte gemountet werden:

```
# mount /dev/acd0 /mnt
```

Danach kann schreibend und lesend auf das DVD-RAM Medium zugegriffen werden.

## 35.7. Disketten benutzen

Dieser Abschnitt beschreibt die Formatierung von 3,5 Zoll Disketten in FreeBSD.

*Procedure: Disketten formatieren*

Bevor eine Diskette benutzt werden kann, muss sie (low-level) formatiert werden, was normalerweise der Hersteller schon gemacht hat. Sie können die Diskette allerdings noch einmal formatieren, um das Medium zu überprüfen. Benutzen Sie fdformat(1), um Disketten unter FreeBSD zu formatieren. Achten Sie dabei auf Fehlermeldungen, die schlechte Speichermedien anzeigen.

1. Um eine Diskette zu formatieren, legen Sie eine 3,5 Zoll Diskette in das erste Diskettenlaufwerk ein und führen das folgende Kommando aus:

   ```
   # /usr/sbin/fdformat -f 1440 /dev/fd0
   ```

2. Nach dem Formatieren muss auf der Diskette ein Disklabel erstellt werden, um die Größe und Geometrie der Diskette zu erkennen. Eine Liste der unterstützten Geometrien finden Sie in /etc/disktab.

   Erstellen Sie nun das Label mit bsdlabel(8):

   ```
   # /sbin/bsdlabel -B -w /dev/fd0 fd1440
   ```

3. Auf der Diskette kann nun ein Dateisystem erstellt werden (high-level Formatierung). Das Dateisystem der Diskette kann entweder UFS oder FAT sein, wobei FAT für Disketten in der Regel die bessere Wahl ist.

   Um die Diskette mit FAT zu formatieren, geben Sie folgendes Kommando ein:

   ```
   # /sbin/newfs_msdos /dev/fd0
   ```

Die Diskette kann nun benutzt werden. Um die Diskette zu verwenden, kann sie mit mount_msdosfs(8) eingehängt werden. Man kann auch emulators/mtools aus der Ports-Sammlung installieren, um mit der Diskette zu arbeiten.

## 35.8. Datensicherung

Die Planung und Umsetzung einer Backup-Strategie ist unerlässlich, um Daten in bestimmten Situationen wiederherstellen zu können, zum Beispiel bei Plattendefekten, versehentlichem Löschen von Dateien, willkürlicher Korrumpierung von Dateien oder der vollständigen Zerstörung des Systems und der Backups, die am gleichen Ort aufbewahrt werden.

Die Art und der Zeitplan des Backups kann variieren, abhängig von der Wichtigkeit der Daten, der benötigten Granularität zur Wiederherstellung von Dateien und der Dauer einer akzeptablen Ausfallzeit. Zu den möglichen Backup-Strategien gehören unter anderem:

- Die Archivierung des kompletten Systems auf externen Datenträgern. Dieser Ansatz schützt zwar vor allen oben aufgeführten Problemen, ist aber zeitaufwändig und unbequem bei der Wiederherstellung, insbesondere für nicht privilegierte Benutzer.
- Dateisystem-Snapshots sind nützlich bei der Wiederherstellung von gelöschten Dateien, bzw. früheren Versionen von Dateien.
- Kopien ganzer Dateisysteme oder Festplatten, die mit einem anderen System im Netzwerk mittels net/rsync synchronisiert werden.
- Hardware oder Software RAID, was im Falle von Plattendefekten die Ausfallzeit minimiert oder

vermeidet.

Üblicherweise wird eine Mischung aus verschiedenen Strategien verwendet. Es kann zum Beispiel ein Sicherungsplan erstellt und automatisiert werden, um eine wöchentliche, vollständige Systemsicherung, ergänzt mit stündlichen ZFS-Snapshots, zu erstellen. Darüber hinaus könnte man eine manuelle Sicherung einzelner Verzeichnisse oder Dateien machen, bevor diese bearbeitet oder gelöscht werden.

Dieser Abschnitt beschreibt einige Programme, die zur Erstellung und Verwaltung von Sicherungen unter FreeBSD verwendet werden können.

### 35.8.1. Sicherung von Dateisystemen

Die traditionellen UNIX®-Programme zum Sichern und Wiederherstellen von Dateisystemen sind dump(8) und restore(8). Diese Programme arbeiten auf der Block-Ebene der Festplatte, also unterhalb des Abstraktionslevels von Dateien, Links und Verzeichnissen, die die Grundlage des Dateisystemkonzepts bilden. Im Gegensatz zu anderen Backup-Programmen sichert `dump` ein ganzes Dateisystem und nicht nur einen Teil des Dateisystems, oder einen Verzeichnisbaum, der mehr als ein Dateisystem umfasst. Anstatt Dateien oder Verzeichnisse zu schreiben, schreibt `dump` die Blöcke, aus denen die Dateien und Verzeichnisse bestehen.



Wird `dump` benutzt, um das Root-Verzeichnis zu sichern, werden /home, /usr und viele andere Verzeichnisse nicht gesichert, da dies normalerweise Mountpunkte für andere Dateisysteme oder symbolische Links zu diesen Dateisystemen sind.

Wenn `restore` zum Extrahieren von Daten verwendet wird, werden temporäre Dateien standardmäßig in /tmp abgelegt. Wenn Sie von einer Platte mit einem kleinen /tmp-Verzeichnis zurücksichern, setzen Sie die Umgebungsvariable `TMPDIR` auf ein Verzeichnis mit mehr freiem Speicherplatz, damit die Wiederherstellung gelingt.

Beachten Sie bei der Verwendung von `dump`, dass es einige Eigenarten aus den frühen Tagen der Version 6 von AT&T UNIX® (ca. 1975) beibehalten hat. Die Standardparameter gehen davon aus, dass auf einem 9-Spur-Band gesichert wird, und nicht auf ein anderes Medium oder auf Sicherungsbänder mit hoher Dichte. Diese Standardwerte müssen auf der Kommandozeile überschrieben werden.

Es ist möglich, das Dateisystem über das Netzwerk auf einem anderen Rechner zu sichern, oder auf einem Bandlaufwerk eines anderen Rechners. Obwohl die Programme rdump(8) und rrestore(8) für diese Zwecke benutzt werden können, gelten sie als nicht sicher.

Verwenden Sie stattdessen `dump` und `restore` in einer sichereren Weise über eine SSH-Verbindung. In diesem Beispiel wird eine vollständige, komprimierte Sicherung von /usr erstellt, das anschließend an einen bestimmten Host über eine SSH-Verbindung gesendet wird.

*Beispiel 36. `dump` mit ssh benutzen*

```
# /sbin/dump -0uan -f - /usr | gzip -2 | ssh -c blowfish \
      targetuser@targetmachine.example.com dd of=/mybigfiles/dump-usr-l0.gz
```

In diesem Beispiel wird RSH gesetzt, um über eine SSH-Verbindung eine Sicherung auf ein Bandlaufwerk eines entfernten Systems zu schreiben:

*Beispiel 37. dump über ssh mit gesetzter RSH benutzen*

```
# env RSH=/usr/bin/ssh /sbin/dump -0uan -f
tatargetuser@targetmachine.example.com:/dev/sa0 /usr
```

### 35.8.2. Sicherung von Verzeichnissen

Einige integrierte Werkzeuge stehen zur Sicherung und Wiederherstellung von bestimmten Dateien und Verzeichnissen bei Bedarf zur Verfügung.

Wenn es um die Sicherung von Dateien in einem Verzeichnis geht, ist tar(1) eine gute Wahl. Dieses Werkzeug stammt aus Version 6 von AT&T UNIX® und erwartet standardmäßig eine rekursive Sicherung auf ein lokales Band. Es können jedoch Optionen angegeben werden, um den Namen einer Sicherungsdatei zu bestimmen.

In diesem Beispiel wird eine komprimierte Sicherung des aktuellen Verzeichnisses nach /tmp/mybackup.tgz gespeichert. Achten Sie bei der Sicherungsdatei darauf, dass sie nicht in dem Verzeichnis gespeichert wird, welches gesichert werden soll.

*Beispiel 38. Das aktuelle Verzeichnis mit tar sichern*

```
# tar czvf /tmp/mybackup.tgz .
```

Um eine komplette Sicherung wiederherzustellen, wechseln Sie mit cd in das Verzeichnis, in dem Sie die Daten wiederherstellen möchten und geben Sie den Namen der Sicherungsdatei an. Beachten Sie, dass dabei alle Dateien in dem Verzeichnis überschrieben werden. Im Zweifel sichern Sie besser in einem temporären Verzeichnis, oder geben Sie den Verzeichnisnamen bei der Wiederherstellung an.

*Beispiel 39. Wiederherstellung mit tar in das aktuelle Verzeichnis*

```
# tar xzvf /tmp/mybackup.tgz
```

Es gibt dutzende Optionen, die in tar(1) beschrieben werden. Das Programm unterstützt auch die Verwendung von Ausschlußmustern, um bestimmte Dateien von der Sicherung oder Wiederherstellung von Verzeichnissen auszuschließen.

Um bestimmte, aufgelistete Dateien und Verzeichnisse zu sichern, ist cpio(1) eine gute Wahl. Im Gegensatz zu tar weiß cpio nicht wie ein Verzeichnisbaum durchlaufen wird. Daher ist es auf eine Liste von zu sichernden Dateien angewiesen.

So kann beispielsweise eine Liste von Dateien mit `ls` oder `find` erzeugt werden. Dieses Beispiel erstellt eine rekursive Liste des aktuellen Verzeichnisses, die dann über eine Pipe an `cpio` übergeben wird, um eine Sicherung namens /tmp/mybackup.cpio zu erstellen.

*Beispiel 40. Rekursive Sicherung des aktuellen Verzeichnisses mit `ls` und `cpio`*

```
# ls -R | cpio -ovF /tmp/mybackup.cpio
```

pax(1) ist ein Programm, welches versucht die Funktionen von `tar` und `cpio` zu kombinieren. Über die Jahre hinweg sind die verschiedenen Versionen von `tar` und `cpio` leicht inkompatibel geworden. Daher hat POSIX® `pax` geschaffen, welches versucht viele der unterschiedlichen `cpio`- und `tar` -Formate zu lesen und zu schreiben, außerdem einige neue, eigene Formate.

Für die vorangegangenen Beispiele wäre ein äquivalenter Aufruf von `pax`:

*Beispiel 41. Das aktuelle Verzeichnis mit `pax` sichern*

```
# pax -wf /tmp/mybackup.pax .
```

### 35.8.3. Bandmedien benutzen

Obwohl sich Bandmedien mit der Zeit weiterentwickelt haben, verwenden moderne Backup-Systeme in der Regel Offsite-Backups in Verbindung mit lokalen Wechseldatenträgern. FreeBSD unterstützt alle SCSI-Bandlaufwerke, wie etwa LTO und DAT. Zusätzlich gibt es begrenzte Unterstützung für SATA- und USB-Bandlaufwerke.

Für SCSI-Bandlaufwerke nutzt FreeBSD den sa(4) Treiber, der die Schnittstellen /dev/sa0, /dev/nsa0 und /dev/esa0 bereitstellt. Der Name des physikalischen Geräts ist /dev/sa0. Wird /dev/nsa0 benutzt, dann wird die Backup-Anwendung nach dem Schreibvorgang das Band nicht zurückspulen, was es ermöglicht, mehr als eine Datei auf das Band zu schreiben. Die Verwendung von /dev/esa0 wirft das Band aus, nachdem das Gerät geschlossen wurde.

FreeBSD nutzt `mt` für die Steuerung der Operationen des Bandlaufwerks, wie die Suche nach Dateien auf einem Band, oder um Kontrollmarkierungen auf ein Band zu schreiben. Beispielsweise können die ersten drei Dateien auf einem Band erhalten bleiben, indem sie übersprungen werden, bevor eine neue Datei auf das Band geschrieben wird

```
# mt -f /dev/nsa0 fsf 3
```

Dieses Werkzeug unterstützt viele Operationen. Weitere Einzelheiten finden Sie in mt(1).

Um eine Datei mit `tar` auf ein Band zu schreiben, geben Sie den Namen des Bandlaufwerks und den Dateinamen an:

```
# tar cvf /dev/sa0 file
```

Wiederherstellung von Dateien aus dem `tar`-Archiv von Band in das aktuelle Verzeichnis:

```
# tar xvf /dev/sa0
```

Benutzen Sie `dump`, um ein UFS-Dateisystem zu sichern. Dieses Beispiel sichert /usr, ohne danach das Band zurückzuspulen:

```
# dump -0aL -b64 -f /dev/nsa0 /usr
```

Interaktive Wiederherstellung von Dateien aus einer dump(8)-Datei von Band in das aktuelle Verzeichnis:

```
# restore -i -f /dev/nsa0
```

### 35.8.4. Backup-Software von Drittanbietern

Die FreeBSD Ports-Sammlung enthält viele Programme von Drittanbietern, die verwendet werden können um die zeitliche Erstellung von Sicherungen zu planen, zu vereinfachen und bequemer zu machen. Viele dieser Programme basieren auf dem Client-Server-Modell und können benutzt werden, um die Sicherung von einzelnen Systemen oder allen Rechnern in einem Netzwerk zu automatisieren.

Zu den bekannten Programmen gehören Amanda, Bacula, rsync und duplicity.

### 35.8.5. Die Wiederherstellung in einem Notfall

Zusätzlich zu den regelmäßigen Sicherungen empfiehlt es sich, die folgenden Schritte im Rahmen eines Notfallplans durchzuführen.

Erstellen Sie einen Ausdruck der Ausgabe der folgenden Kommandos:

- `gpart show`
- `more /etc/fstab`
- `dmesg`

Bewahren Sie diesen Ausdruck und eine Kopie des Installationsmediums an einem sicheren Ort auf. Im Falle einer Wiederherstellung im Notfall, starten Sie von dem Installationsmedium und wählen Sie `Live CD`, um eine Rettungs-Shell zu starten. Dieser Rettungsmodus kann verwendet werden, um den aktuellen Stand des Systems anzuzeigen, und wenn nötig, Festplatten zu formatieren und Daten aus den Sicherungen wiederherzustellen.



Das Installationsmedium für FreeBSD/i386 11.2-RELEASE enthält keine Rettungs-

> Shell. Laden Sie für diese Version ein Abbild der Livefs CD von ftp://ftp.FreeBSD.org/pub/FreeBSD/releases/i386/ISO-IMAGES/11.2/FreeBSD-11.2-RELEASE-i386-livefs.iso.

Als nächstes testen Sie die Rettungs-Shell und die Sicherungen. Dokumentieren Sie diesen Ablauf. Bewahren Sie diese Notizen zusammen mit den Medien, den Ausdrucken und den Sicherungen auf. Diese Notizen können Ihnen im Notfall helfen eine versehentliche Zerstörung der Sicherungen zu verhindern, während Sie unter Stress eine Wiederherstellung durchführen.

Als zusätzliche Sicherheitsvorkehrung kann jeweils die letzte Sicherung an einem entfernten Standort aufbewahrt werden. Dieser Standort sollte räumlich von den Computern und Festplatten durch eine erhebliche Entfernung getrennt sein.

# 35.9. Speicherbasierte Laufwerke

Neben physikalischen Laufwerken unterstützt FreeBSD auch speicherbasierte Laufwerke. Eine mögliche Verwendung für ein speicherbasiertes Laufwerk ist der Zugriff auf ein ISO-Dateisystem, jedoch ohne vorher die Daten auf eine CD oder DVD zu brennen und dann das Medium einzuhängen.

FreeBSD verwendet den md(4) Treiber um Unterstützung für speicherbasierte Laufwerke bereitzustellen. Dieser Treiber ist bereits im GENERIC-Kernel enthalten. Wenn Sie eine angepasste Kernelkonfigurationsdatei verwenden, stellen Sie sicher, dass folgende Zeile enthalten ist:

```
device md
```

## 35.9.1. Ein- und Aushängen von bestehenden Abbildern

Um ein bestehendes Abbild eines Dateisystems einzuhängen, verwenden Sie `mdconfig` zusammen mit dem Namen der ISO-Datei und einer freien Gerätenummer. Benutzen Sie dann diese Gerätenummer, um das Abbild in einen existierenden Mountpunkt einzuhängen. Sobald dies erledigt ist, erscheinen die Dateien des Abbildes unterhalb des Mountpunktes. Dieses Beispiel wird diskimage.iso an das speicherbasierte Laufwerk /dev/md0 binden und dann in /mnt einhängen:

```
# mdconfig -f diskimage.iso -u 0
# mount -t cd9660 /dev/md0 /mnt
```

Beachten Sie, dass `-t cd9660` benutzt wurde, um ein ISO-Format einzuhängen. Wenn keine Gerätenummer mit `-u` angegeben ist, wird von md(4) automatisch eine ungenutzte Gerätenummer zugewiesen. Das zugewiesene Gerät wird auf der Standardausgabe ausgegeben (zum Beispiel md4). Weitere Informationen zu diesem Kommando und dessen Optionen finden Sie in mdconfig(8).

Wenn ein speicherbasiertes Laufwerk nicht mehr in Gebrauch ist, sollten seine belegten Ressourcen wieder an das System zurückgegeben werden. Hängen Sie zuerst das Dateisystem aus, dann verwenden Sie `mdconfig`, um die Platte vom System zu trennen und die Ressourcen freizugeben.

```
# umount /mnt
# mdconfig -d -u 0
```

Um festzustellen, ob noch irgendwelche speicherbasierten Laufwerke am System angeschlossen sind, benutzen Sie `mdconfig -l`.

### 35.9.2. Ein datei- oder speicherbasiertes Laufwerk erzeugen

FreeBSD unterstützt auch speicherbasierte Laufwerke, bei denen der verwendete Speicher entweder einer Festplatte, oder einem Bereich im Arbeitsspeicher zugewiesen wird. Die erste Methode ist gemeinhin als dateibasiertes Dateisystem, die zweite als speicherbasiertes Dateisystem bekannt. Beide Typen können mit `mdconfig` erzeugt werden.

Um ein speicherbasiertes Dateisystem zu erstellen, geben Sie den Typ `swap` sowie die gewünschte Größe des Laufwerks an. Dieses Beispiel erzeugt ein 5 MB großes Laufwerk an der Gerätenummer 1. Das Laufwerk wird mit dem UFS-Dateisystem formatiert, bevor es eingehängt wird:

```
# mdconfig -a -t swap -s 5m -u 1
# newfs -U md1
/dev/md1: 5.0MB (10240 sectors) block size 16384, fragment size 2048
        using 4 cylinder groups of 1.27MB, 81 blks, 192 inodes.
        with soft updates
super-block backups (for fsck -b #) at:
 160, 2752, 5344, 7936
# mount /dev/md1 /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md1        4718    4  4338     0%    /mnt
```

Um ein dateibasiertes Dateisystem zu erstellen, muss zunächst ein Stück Speicher auf der Festplatte reserviert werden. Dieses Beispiel erzeugt eine 5 MB große Datei namens newimage:

```
# dd if=/dev/zero of=newimage bs=1k count=5k
5120+0 records in
5120+0 records out
```

Als nächstes muss diese Datei an ein speicherbasiertes Laufwerk gebunden, gelabelt und mit dem UFS-Dateisystem formatiert werden. Danach können Sie das Laufwerk einhängen und die Größe überprüfen:

```
# mdconfig -f newimage -u 0
# bsdlabel -w md0 auto
# newfs -U md0a
/dev/md0a: 5.0MB (10224 sectors) block size 16384, fragment size 2048
        using 4 cylinder groups of 1.25MB, 80 blks, 192 inodes.
super-block backups (for fsck -b #) at:
```

```
 160, 2720, 5280, 7840
# mount /dev/md0a /mnt
# df /mnt
Filesystem 1K-blocks Used Avail Capacity  Mounted on
/dev/md0a       4710    4  4330     0%    /mnt
```

Es benötigt mehrere Befehle, um ein datei- oder speicherbasiertes Dateisystem mit `mdconfig` zu erstellen. FreeBSD enthält auch `mdmfs`, das ein speicherbasiertes Laufwerk automatisch konfigurieren, formatieren und einhängen kann. Nachdem beispielsweise newimage mit `dd` erstellt wurde, hätte auch der folgende Befehl benutzt werden können, anstelle der oben verwendeten Kommandos `bsdlabel`, `newfs` und `mount`:

```
# mdmfs -F newimage -s 5m md0 /mnt
```

Um hingegen ein speicherbasiertes Laufwerk mit `mdmfs` zu erstellen, wird dieser Befehl benutzt:

```
# mdmfs -s 5m md1 /mnt
```

Wenn die Gerätenummer nicht angegeben wird, wählt `mdmfs` automatisch ein ungenutztes Gerät aus. Weitere Einzelheiten über `mdmfs` finden Sie in mdmfs(8).

## 35.10. Schnappschüsse von Dateisystemen

Zusammen mit Soft Updates bietet FreeBSD eine weitere Funktion: Schnappschüsse von Dateisystemen.

UFS-Schnappschüsse sind Dateien, die ein Abbild eines Dateisystems enthalten und müssen auf dem jeweiligen Dateisystem erstellt werden. Pro Dateisystem darf es maximal 20 Schnappschüsse, die im Superblock vermerkt werden, geben. Schnappschüsse bleiben erhalten, wenn das Dateisystem abgehangen, neu eingehangen oder das System neu gestartet wird. Wenn ein Schnappschuss nicht mehr benötigt wird, kann er mit rm(1) gelöscht werden. Es ist egal, in welcher Reihenfolge Schnappschüsse gelöscht werden. Es kann allerdings vorkommen, dass nicht der gesamte Speicherplatz wieder freigegeben wird, da ein anderer Schnappschuss einen Teil der entfernten Blöcke für sich beanspruchen kann.

Das unveränderliche `Snapshot`-Dateiflag wird nach der Erstellung des Snapshots von mksnap_ffs(8) gesetzt. Durch die Verwendung von unlink(1) ist es allerdings möglich, einen Schnappschuss zu löschen.

Schnappschüsse werden mit mount(8) erstellt. Das folgende Kommando legt einen Schnappschuss von /var in /var/snapshot/snap ab:

```
# mount -u -o snapshot /var/snapshot/snap /var
```

Alternativ kann der Schnappschuss auch mit mksnap_ffs(8) erstellt werden.

```
# mksnap_ffs /var /var/snapshot/snap
```

Um Schnappschüsse auf einem Dateisystem, beispielsweise /var zu finden, kann man find(1) verwenden:

```
# find /var -flags snapshot
```

Nachdem ein Schnappschuss erstellt wurde, können Sie ihn für verschiedene Zwecke benutzen:

- Sie können den Schnappschuss für die Datensicherung benutzen und ihn auf eine CD oder ein Band schreiben.
- Die Integrität des Schnappschusses kann mit fsck(8) geprüft werden. Wenn das Dateisystem zum Zeitpunkt der Erstellung des Schnappschusses in Ordnung war, sollte fsck(8) immer erfolgreich durchlaufen.
- Sie können den Schnappschuss mit dump(8) sichern. Sie erhalten dann eine konsistente Sicherung des Dateisystems zu dem Zeitpunkt, der durch den Zeitstempel des Schnappschusses gegeben ist. Der Schalter `-L` von dump(8) erstellt für die Sicherung einen Schnappschuss und entfernt diesen am Ende der Sicherung wieder.
- Sie können einen Schnappschuss in den Verzeichnisbaum einhängen und sich dann den Zustand des Dateisystems zu dem Zeitpunkt ansehen, an dem der Schnappschuss erstellt wurde. Der folgende Befehl hängt den Schnappschuss /var/snapshot/snap ein:

  ```
  # mdconfig -a -t vnode -o readonly -f /var/snapshot/snap -u 4
  # mount -r /dev/md4 /mnt
  ```

Der eingefrorene Stand des /var-Dateisystems ist nun unterhalb von /mnt verfügbar. Mit Ausnahme der früheren Schnappschüsse, die als leere Dateien auftauchen, wird zu Beginn alles so aussehen, wie zum Zeitpunkt der Erstellung des Schnappschusses. Der Schnappschuss kann wie folgt abgehängt werden:

```
# umount /mnt
# mdconfig -d -u 4
```

Weitere Informationen über Soft Updates und Schnappschüsse von Dateisystemen sowie technische Artikel finden Sie auf der Webseite von Marshall Kirk McKusick.

## 35.11. Disk Quotas

Disk Quotas erlauben dem Administrator, den Plattenplatz und/oder die Anzahl der Dateien eines Benutzers oder der Mitglieder einer Gruppe, auf Dateisystemebene zu beschränken. Dadurch wird verhindert, dass ein Benutzer oder eine Gruppe von Benutzern den ganzen verfügbaren Plattenplatz belegt.

Dieser Abschnitt beschreibt die Konfiguration von Disk Quotas für UFS-Dateisysteme. Lesen Sie Dataset-, Benutzer- und Gruppenquotas, wenn Sie Disk Quotas auf einem ZFS-Dateisystem einrichten möchten.

### 35.11.1. Disk Quotas aktivieren

Prüfen Sie zunächst, ob der FreeBSD-Kernel Disk Quotas unterstützt:

```
% sysctl kern.features.ufs_quota
kern.features.ufs_quota: 1
```

In diesem Beispiel zeigt die `1` an, das Quotas unterstützt werden. Falls `0` ausgegeben wird, fügen Sie folgende Zeile in die Kernelkonfigurationsdatei ein, und folgen Sie den Anweisungen in Konfiguration des FreeBSD-Kernels um den Kernel zu aktualisieren:

```
options QUOTA
```

Als nächstes aktivieren Sie Disk Quotas in /etc/rc.conf:

```
quota_enable="YES"
```

Normalerweise wird beim Booten die Integrität der Quotas auf allen Dateisystemen mit quotacheck(8) überprüft. Dieses Programm stellt sicher, dass die Quota-Datenbank mit den Daten auf einem Dateisystem übereinstimmt. Dies ist allerdings ein zeitraubender Prozess, der die Zeit, die das System zum Booten braucht, signifikant beeinflusst. Eine Variable in /etc/rc.config erlaubt es, diesen Schritt zu überspringen:

```
check_quotas="NO"
```

Zuletzt muss noch /etc/fstab bearbeitet werden, um die Plattenquotas auf Dateisystemebene zu aktivieren. Um Quotas pro Benutzer für ein Dateisystem zu aktivieren, geben Sie für dieses Dateisystem `userquota` im Feld Optionen von /etc/fstab an. Zum Beispiel:

```
/dev/da1s2g   /home    ufs rw,userquota 1 2
```

Um Quotas für Gruppen einzurichten, verwenden Sie `groupquota`. Um Quotas für Benutzer und Gruppen einzurichten, trennen Sie die Optionen durch Kommata:

```
/dev/da1s2g    /home    ufs rw,userquota,groupquota 1 2
```

Quota-Dateien werden standardmäßig im Rootverzeichnis des Dateisystems unter quota.user und quota.group abgelegt. Weitere Informationen finden Sie in fstab(5). Es wird nicht empfohlen, Quota-Dateien an anderen Stellen zu speichern.

Sobald die Konfiguration abgeschlossen ist, starten Sie das System neu. /etc/rc wird dann automatisch die richtigen Kommandos aufrufen, um die Quota-Dateien für alle in /etc/rc.conf definierten Quotas anzulegen.

Normalerweise brauchen die Kommandos quotacheck(8), quotaon(8) oder quotaoff(8) nicht von Hand aufgerufen werden, obwohl man die entsprechenden Seiten im Manual lesen sollte, um sich mit ihnen vertraut zu machen.

### 35.11.2. Setzen von Quota-Limits

Stellen Sie sicher, dass Quotas auch tatsächlich aktiviert sind:

```
# quota -v
```

Für jedes Dateisystem, auf dem Quotas aktiviert sind, sollte eine Zeile mit der Plattenauslastung und den aktuellen Quota-Limits zu sehen sein.

Mit `edquota` können nun Quota-Limits zugewiesen werden.

Mehrere Möglichkeiten stehen zur Verfügung, um Limits für den Plattenplatz, den ein Benutzer oder eine Gruppe verbrauchen kann, oder die Anzahl der Dateien, die angelegt werden dürfen, festzulegen. Die Limits können auf dem Plattenplatz (Block-Quotas), der Anzahl der Dateien (Inode-Quotas) oder einer Kombination von beiden basieren. Jedes Limit wird weiterhin in zwei Kategorien geteilt: Hardlimits und Softlimits.

Ein Hardlimit kann nicht überschritten werden. Hat der Benutzer einmal ein Hardlimit erreicht, so kann er auf dem betreffenden Dateisystem keinen weiteren Platz mehr beanspruchen. Hat ein Benutzer beispielsweise ein Hardlimit von 500 Kilobytes auf einem Dateisystem und benutzt davon 490 Kilobyte, so kann er nur noch 10 weitere Kilobytes beanspruchen. Der Versuch, weitere 11 Kilobytes zu beanspruchen, wird fehlschlagen.

Softlimits können für eine befristete Zeit überschritten werden. Diese Frist beträgt in der Grundeinstellung eine Woche. Hat der Benutzer das Softlimit über die Frist hinaus überschritten, so wird das Softlimit in ein Hardlimit umgewandelt und der Benutzer kann keinen weiteren Platz mehr beanspruchen. Wenn er einmal das Softlimit unterschreitet, wird die Frist wieder zurückgesetzt.

Im folgenden Beispiel wird das Quota des Benutzerkonto `test` bearbeitet. Wenn `edquota` aufgerufen wird, wird der in `EDITOR` definierte Editor aufgerufen, um die Quota-Limts zu konfigurieren. Der Standard-Editor ist vi.

```
# edquota -u test

Quotas for user test:

/usr: kbytes in use: 65, limits (soft = 50, hard = 75)
        inodes in use: 7, limits (soft = 50, hard = 60)
/usr/var: kbytes in use: 0, limits (soft = 50, hard = 75)
```

```
        inodes in use: 0, limits (soft = 50, hard = 60)
```

Für jedes Dateisystem, auf dem Quotas aktiv sind, sind zwei Zeilen zu sehen. Eine repräsentiert die Block-Quotas und die andere die Inode-Quotas. Um ein Limit zu modifizieren, ändern Sie einfach den angezeigten Wert. Um beispielsweise das Blocklimit von /usr auf ein Softlimit von `500` und ein Hardlimit von `600` zu erhöhen, ändern Sie die Zeile wie folgt:

```
/usr: kbytes in use: 65, limits (soft = 500, hard = 600)
```

Die neuen Limits sind wirksam, sobald der Editor verlassen wird.

Manchmal ist es wünschenswert, die Limits für eine Reihe von Benutzern zu setzen. Dazu weisen Sie zunächst einem Benutzer das gewünschte Quota-Limit zu. Anschließend benutzen Sie `-p`, um das Quota auf einen bestimmten Bereich von Benutzer-IDs (UID) zu duplizieren. Der folgende Befehl dupliziert die Quota-Limits auf die UIDs `10000` bis `19999`:

```
# edquota -p test 10000-19999
```

Weitere Informationen finden Sie in edquota(8).

### 35.11.3. Überprüfen von Quota-Limits und Plattennutzung

Um die Limits oder die Plattennutzung individueller Benutzer und Gruppen zu überprüfen, kann quota(1) benutzt werden. Ein Benutzer kann nur die eigenen Quotas und die Quotas der Gruppe, der er angehört untersuchen. Nur der Superuser darf sich alle Limits ansehen. Mit repquota(8) erhalten Sie eine Zusammenfassung von allen Limits und der Plattenausnutzung für alle Dateisysteme, auf denen Quotas aktiv sind.

In der Ausgabe von quota(1) werden Dateisysteme, auf denen ein Benutzer keinen Platz verbraucht, nicht angezeigt, auch wenn diesem Quotas zugewiesen wurden. Benutzen Sie `-v` um solche Dateisysteme ebenfalls anzuzeigen. Das folgende Beispiel zeigt die Ausgabe von `quota -v` für einen Benutzer, der Quota-Limits auf zwei Dateisystemen besitzt:

```
Disk quotas for user test (uid 1002):
     Filesystem  usage   quota   limit   grace   files   quota   limit   grace
           /usr      65*     50      75   5days       7      50      60
      /usr/var       0      50      75               0      50      60
```

Im Dateisystem /usr liegt der Benutzer momentan 15 Kilobytes über dem Softlimit von 50 Kilobytes und hat noch 5 Tage seiner Frist übrig. Der Stern `*` zeigt an, dass der Benutzer sein Limit überschritten hat.

### 35.11.4. Quotas über NFS

Quotas werden von dem Quota-Subsystem auf dem NFS-Server erzwungen. Der rpc.rquotad(8) Daemon stellt `quota` die Quota Informationen auf dem NFS-Client zur Verfügung, so dass Benutzer

auf diesen Systemen ihre Quotas abfragen können.

Sie aktivieren `rpc.rquotad` auf dem NFS-Server, indem Sie das Zeichen `#` auf folgender Zeile in /etc/inetd.conf entfernen:

```
rquotad/1      dgram rpc/udp wait root /usr/libexec/rpc.rquotad rpc.rquotad
```

Anschließend starten Sie `inetd` neu:

```
# service inetd restart
```

# 35.12. Partitionen verschlüsseln

FreeBSD bietet ausgezeichnete Möglichkeiten, Daten vor unberechtigten Zugriffen zu schützen. Wenn das Betriebssystem läuft, schützen Zugriffsrechte und vorgeschriebene Zugriffskontrollen (MAC) (siehe Verbindliche Zugriffskontrolle) die Daten. Die Zugriffskontrollen des Betriebssystems schützen allerdings nicht vor einem Angreifer, der Zugriff auf den Rechner hat. Der Angreifer kann eine Festplatte in ein anderes System einbauen und dort die Daten analysieren.

Die für FreeBSD verfügbaren kryptografischen Subsysteme, GEOM Based Disk Encryption (`gbde`) und `geli` sind in der Lage, Daten auf Dateisystemen auch vor hoch motivierten Angreifern zu schützen, die über erhebliche Mittel verfügen. Dieser Schutz ist unabhängig von der Art und Weise, durch die ein Angreifer Zugang zu einer Festplatte oder zu einem Rechner erlangt hat. Im Gegensatz zu anderen Verschlüsselungsmethoden, bei denen einzelne Dateien verschlüsselt werden, verschlüsseln gbde und `geli` transparent ganze Dateisysteme. Auf der Festplatte werden dabei keine Daten im Klartext gespeichert.

Dieses Kapitel zeigt, wie ein verschlüsseltes Dateisystem unter FreeBSD erstellt wird. Zunächst wird der Ablauf für gbde beschrieben und anschließend das gleiche Beispiel für geli.

## 35.12.1. Plattenverschlüsselung mit gbde

Das Ziel von gbde(4) ist es, einen Angreifer vor eine große Herausforderung zu stellen, um an die Daten einer Festplatte zu gelangen. Falls jedoch der Rechner kompromittiert wurde, während er im Betrieb war und das Speichergerät aktiv verbunden war, oder wenn der Angreifer eine gültige Passphrase kennt, bietet dieses System keinen Schutz für die Daten der Festplatte. Daher ist es wichtig, für die physische Sicherheit zu sorgen, während das System im Betrieb ist. Außerdem muss die Passphrase für den Verschlüsselungsmechanismus geschützt werden.

gbde(4) besitzt einige Funktionen um die Daten, die in einem Sektor gespeichert sind, zu schützen. Es benutzt 128-Bit AES im CBC-Modus, um die Daten eines Sektors zu verschlüsseln. Jeder Sektor einer Festplatte wird mit einem anderen AES-Schlüssel verschlüsselt. Weitere Informationen zum kryptographischen Design und wie die Schlüssel für einen Sektor aus der gegebenen Passphrase ermittelt werden, finden Sie in gbde(4).

FreeBSD enthält ein Kernelmodul für gbde, das wie folgt geladen werden kann:

```
# kldload geom_bde
```

Wenn Sie einen angepassten Kernel verwenden, stellen Sie sicher, dass folgende Zeile in der Kernelkonfigurationsdatei enthalten ist:

```
options GEOM_BDE
```

Das folgende Beispiel beschreibt, wie eine Partition auf einer neuen Festplatte verschlüsselt wird. Die Partition wird in /private eingehangen.

*Procedure: Eine Partition mit gbde verschlüsseln*

1. Installieren der Festplatte

   Installieren Sie die Festplatte wie in Hinzufügen von Laufwerken beschrieben. Im Beispiel wird die Partition /dev/ad4s1c verwendet. Die Gerätedateien /dev/ad0s1* sind Standard-Partitionen des FreeBSD-Systems.

   ```
   # ls /dev/ad*
   /dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
   /dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
   /dev/ad0s1a     /dev/ad0s1d     /dev/ad4
   ```

2. Verzeichnis für gbde-Lock-Dateien anlegen

   ```
   # mkdir /etc/gbde
   ```

   Die Lock-Dateien sind für den Zugriff von gbde auf verschlüsselte Partitionen notwendig. Ohne die Lock-Dateien können die Daten nur mit erheblichem manuellen Aufwand wieder entschlüsselt werden (dies wird auch von der Software nicht unterstützt). Jede verschlüsselte Partition benötigt eine gesonderte Lock-Datei.

3. Vorbereiten der gbde-Partition

   Eine von gbde benutzte Partition muss einmalig initialisiert werden, bevor sie benutzt werden kann. Das Programm öffnet eine Vorlage im Standard-Editor, um verschiedene Optionen zu konfigurieren. Setzen Sie `sector_size` auf `2048`, wenn Sie UFS benutzen:

   ```
   # gbde init /dev/ad4s1c -i -L /etc/gbde/ad4s1c.lock
   $FreeBSD: src/sbin/gbde/template.txt,v 1.1.36.1 2009/08/03 08:13:06 kensmith Exp $
   #
   # Sector size is the smallest unit of data which can be read or written.
   # Making it too small decreases performance and decreases available space.
   # Making it too large may prevent filesystems from working.  512 is the
   # minimum and always safe.  For UFS, use the fragment size
   #
   sector_size     =       2048
   ```

```
[...]
```

Sobald die Änderungen gespeichert werden, wird der Benutzer zweimal aufgefordert, die zum Schutz der Daten verwendete Passphrase einzugeben. Die Passphrase muss beide Mal gleich eingegeben werden. Die Sicherheit der Daten hängt allein von der Qualität der gewählten Passphrase ab. Die Auswahl einer sicheren und leicht zu merkenden Passphrase wird auf der Webseite http://world.std.com/~reinhold/diceware.html beschrieben.

Bei der Initialisierung wird eine Lock-Datei für die gbde-Partition erstellt. In diesem Beispiel /etc/gbde/ad4s1c.lock. Lock-Dateien müssen die Dateiendung ".lock" aufweisen, damit sie von /etc/rc.d/gbde, dem Startskript von gbde, erkannt werden.



Lock-Dateien müssen immer zusammen mit den verschlüsselten Dateisystemen gesichert werden. Ohne die Lock-Datei können Sie allerdings nicht auf die verschlüsselten Daten zugreifen.

4. Einbinden der verschlüsselten Partition in den Kernel

```
# gbde attach /dev/ad4s1c -l /etc/gbde/ad4s1c.lock
```

Dieses Kommando fragt die Passphrase ab, die bei der Initialisierung der verschlüsselten Partition eingegeben wurde. Das neue verschlüsselte Gerät erscheint danach in /dev als /dev/device_name.bde:

```
# ls /dev/ad*
/dev/ad0        /dev/ad0s1b     /dev/ad0s1e     /dev/ad4s1
/dev/ad0s1      /dev/ad0s1c     /dev/ad0s1f     /dev/ad4s1c
/dev/ad0s1a     /dev/ad0s1d     /dev/ad4        /dev/ad4s1c.bde
```

5. Dateisystem auf dem verschlüsselten Gerät anlegen

Nachdem die verschlüsselte Partition im Kernel eingebunden ist, kann ein Dateisystem erstellt werden. Dieses Beispiel erstellt ein UFS-Dateisystem mit aktivierten Soft Updates. Achten Sie darauf, die Partition mit der Erweiterung *.bde zu benutzen:

```
# newfs -U -O2 /dev/ad4s1c.bde
```

6. Einhängen der verschlüsselten Partition

Legen Sie einen Mountpunkt für das verschlüsselte Dateisystem an. Hängen Sie anschließend das Dateisystem ein:

```
# mkdir /private
# mount /dev/ad4s1c.bde /private
```

7. Überprüfen des verschlüsselten Dateisystems

   Das verschlüsselte Dateisystem sollte jetzt erkannt und benutzt werden können:

```
% df -H
Filesystem         Size    Used   Avail Capacity  Mounted on
/dev/ad0s1a       1037M     72M    883M     8%    /
/devfs             1.0K    1.0K      0B   100%    /dev
/dev/ad0s1f        8.1G     55K    7.5G     0%    /home
/dev/ad0s1e       1037M    1.1M    953M     0%    /tmp
/dev/ad0s1d        6.1G    1.9G    3.7G    35%    /usr
/dev/ad4s1c.bde    150G    4.1K    138G     0%    /private
```

Nach jedem Neustart müssen verschlüsselte Dateisysteme dem Kernel wieder bekannt gemacht werden, auf Fehler überprüft werden und eingehangen werden. Für die dazu nötigen Schritte fügen Sie folgende Zeilen in /etc/rc.conf hinzu:

```
gbde_autoattach_all="YES"
gbde_devices="ad4s1c"
gbde_lockdir="/etc/gbde"
```

Durch diese Argumente muss beim Systemstart auf der Konsole die Passphrase eingegeben werden. Erst nach Eingabe der korrekten Passphrase wird die verschlüsselte Partition automatisch in den Verzeichnisbaum eingehängt. Weitere Bootoptionen von gbde finden Sie in rc.conf(5).



sysinstall ist nicht kompatibel mit gbde-verschlüsselten Geräten. Bevor sysinstall gestartet wird, müssen alle *.bde Geräte vom Kernel getrennt werden, da sonst der Kernel bei der ersten Suche nach Geräten abstürzt. Um das verschlüsselte Gerät aus dem Beispiel zu trennen, benutzen Sie das folgende Kommando:

```
# gbde detach /dev/ad4s1c
```

### 35.12.2. Plattenverschlüsselung mit `geli`

Mit `geli` steht eine alternative kryptografische GEOM-Klasse zur Verfügung. Dieses Werkzeug unterstützt unterschiedliche Fähigkeiten und verfolgt einen anderen Ansatz für die Verschlüsselung. geli bietet die folgenden Funktionen:

- Die Nutzung des crypto(9)-Frameworks. Wenn das System über kryptografische Hardware verfügt, wird diese von `geli` automatisch verwendet.
- Die Unterstützung verschiedener kryptografischer Algorithmen, wie AES, Blowfish, und 3DES.
- Die Möglichkeit, die root-Partition zu verschlüsseln. Um auf die verschlüsselte root-Partition zugreifen zu können, muss beim Systemstart die Passphrase eingegeben werden.
- Erlaubt den Einsatz von zwei voneinander unabhängigen Schlüsseln.

- Es ist durch einfache Sektor-zu-Sektor-Verschlüsselung sehr schnell.
- Die Möglichkeit, Master-Keys zu sichern und wiederherzustellen. Wenn ein Benutzer seinen Schlüssel zerstört, kann er über seinen zuvor gesicherten Schlüssel wieder auf seine Daten zugreifen.
- `geli` erlaubt es, Platten mit einem zufälligen Einmal-Schlüssel einzusetzen, was für Swap-Partitionen und temporäre Dateisysteme interessant ist.

Weitere Funktionen und Anwendungsbeispiele finden Sie in geli(8).

Das folgende Beispiel beschreibt, wie eine Schlüsseldatei erzeugt wird, die als Teil des Master-Keys für den Verschlüsselungs-Provider verwendet wird, der unter /private in den Verzeichnisbaum eingehängt wird. Die Schlüsseldatei liefert zufällige Daten, die für die Verschlüsselung des Master-Keys benutzt werden. Zusätzlich wird der Master-Key durch eine Passphrase geschützt. Die Sektorgröße des Providers beträgt 4 KB. Das Beispiel beschreibt, wie Sie einen `geli`-Provider aktivieren, ein vom ihm verwaltetes Dateisystem erzeugen, es mounten, mit ihm arbeiten und wie Sie es schließlich wieder unmounten und den Provider deaktivieren.

*Procedure: Eine Partition mit `geli` verschlüsseln*

1. Laden der `geli`-Unterstützung

   Die Unterstützung für `geli` wird über ein ladbares Kernelmodul zur Verfügung gestellt. Damit das Modul automatisch beim Booten geladen wird, fügen Sie folgende Zeile in /boot/loader.conf ein:

   ```
   geom_eli_load="YES"
   ```

   Um das Modul direkt zu laden:

   ```
   # kldload geom_eli
   ```

   Stellen Sie bei einer angepassten Kernelkonfigurationsdatei sicher, dass diese Zeilen enthalten sind:

   ```
   options GEOM_ELI
   device crypto
   ```

2. Erzeugen des Master-Keys

   Die folgenden Befehle erzeugen einen Master-Key, mit dem alle Daten verschlüsselt werden. Dieser Schlüssel kann niemals geändert werden. Anstatt ihn direkt zu benutzen, wird er mit einem oder mehrere Schlüsseln verschlüsselt. Die Schlüssel bestehen aus einer optionalen Kombination von zufälligen Bytes aus einer Datei, /root/da2.key, und/oder einer Passphrase. In diesem Fall ist die Datenquelle der Schlüsseldatei /dev/random. Dieser Befehl konfiguriert auch die Sektorgröße des Providers (/dev/da2.eli) mit 4 KB, um eine bessere Leistung zu erzielen:

```
# dd if=/dev/random of=/root/da2.key bs=64 count=1
# geli init -K /root/da2.key -s 4096 /dev/da2
Enter new passphrase:
Reenter new passphrase:
```

Es ist nicht zwingend nötig, sowohl eine Passphrase als auch eine Schlüsseldatei zu verwenden. Die einzelnen Methoden können auch unabhängig voneinander eingesetzt werden.

Wird für die Schlüsseldatei "-" angegeben, wird dafür die Standardeingabe verwendet. Das folgende Kommando erzeugt beispielsweise drei Schlüsseldateien:

```
# cat keyfile1 keyfile2 keyfile3 | geli init -K - /dev/da2
```

3. Aktivieren des Providers mit dem erzeugten Schlüssel

   Um den Provider zu aktivieren, geben Sie die Schlüsseldatei, den Namen des Laufwerks und die Passphrase an:

```
# geli attach -k /root/da2.key /dev/da2
Enter passphrase:
```

   Dadurch wird ein neues Gerät mit der Erweiterung .eli angelegt:

```
# ls /dev/da2*
/dev/da2  /dev/da2.eli
```

4. Das neue Dateisystem erzeugen

   Als nächstes muss das Gerät mit dem UFS-Dateisystem formatiert und an einen vorhandenen Mountpunkt eingehängt werden:

```
# dd if=/dev/random of=/dev/da2.eli bs=1m
# newfs /dev/da2.eli
# mount /dev/da2.eli /private
```

   Das verschlüsselte Dateisystem sollte jetzt erkannt und benutzt werden können:

```
# df -H
Filesystem     Size   Used  Avail Capacity  Mounted on
/dev/ad0s1a    248M    89M   139M    38%    /
/devfs         1.0K   1.0K     0B   100%    /dev
/dev/ad0s1f    7.7G   2.3G   4.9G    32%    /usr
/dev/ad0s1d    989M   1.5M   909M     0%    /tmp
/dev/ad0s1e    3.9G   1.3G   2.3G    35%    /var
```

```
/dev/da2.eli    150G    4.1K    138G     0%    /private
```

Wenn Sie nicht mehr mit dem verschlüsselten Dateisystem arbeiten und die unter /private eingehängte Partition daher nicht mehr benötigen, sollten Sie diese unmounten und den `geli` -Verschlüsselungs-Provider wieder deaktivieren:

```
# umount /private
# geli detach da2.eli
```

FreeBSD verfügt über ein rc.d-Skript, das dass Einhängen von verschlüsselten Geräten beim Booten deutlich vereinfacht. Für dieses Beispiel, fügen Sie folgende Zeilen in /etc/rc.conf hinzu:

```
geli_devices="da2"
geli_da2_flags="-p -k /root/da2.key"
```

Dies konfiguriert /dev/da2 als `geli`-Provider mit dem Master-Key /root/da2.key. Das System wird den Provider automatisch deaktivieren, bevor es heruntergefahren wird. Während des Startvorgangs fordert das Skript die Passphrase an, bevor der Provider aktiviert wird. Vor und nach der Eingabeaufforderung für die Passphrase werden noch weitere Kernelmeldungen angezeigt. Achten Sie sorgfältig auf die Eingabeaufforderung zwischen den anderen Meldungen, falls es zu Problemen beim Startvorgang kommt. Sobald die richtige Passphrase eingegeben wurde, wird der Provider aktiviert. Anschließend werden die Dateisysteme gemäß /etc/fstab eingehängt. Lesen Sie “Anhängen und Abhängen von Dateisystemen” wenn Sie wissen möchten, wie Sie ein Dateisystem konfigurieren, sodass es beim booten automatisch gestartet wird.

# 35.13. Den Auslagerungsspeicher verschlüsseln

Wie die Verschlüsselung von Partitionen, wird auch der Auslagerungsspeicher verschlüsselt, um sensible Informationen zu schützen. Stellen Sie sich eine Anwendung vor, die mit Passwörtern umgeht. Solange sich diese Passwörter im Arbeitsspeicher befinden, werden sie nicht auf die Festplatte geschrieben und nach einem Neustart gelöscht. Falls FreeBSD jedoch damit beginnt Speicher auszulagern, um Platz für andere Anwendungen zu schaffen, können die Passwörter unverschlüsselt auf die Festplatte geschrieben werden. Die Verschlüsselung des Auslagerungsspeichers kann in solchen Situationen Abhilfe schaffen.

Dieser Abschnitt zeigt die Konfiguration eines verschlüsselten Auslagerungsspeichers mittels gbde(8) oder geli(8). In den Beispielen repräsentiert /dev/ada0s1b die Swap-Partition.

## 35.13.1. Konfiguration eines verschlüsselten Auslagerungsspeichers

Swap-Partitionen werden standardmäßig nicht verschlüsselt. Sie sollten daher alle sensiblen Daten im Auslagerungsspeicher löschen, bevor Sie fortfahren. Führen Sie folgenden Befehl aus, um die Swap-Partition mit Zufallsdaten zu überschreiben:

```
# dd if=/dev/random of=/dev/ada0s1b bs=1m
```

Um den Auslagerungsspeicher mit gbde(8) zu verschlüsseln, fügen Sie in /etc/fstab das Suffix `.bde` an den Gerätenamen der Swap-Partition hinzu:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ada0s1b.bde         none            swap    sw              0       0
```

Wenn Sie geli(8) benutzen, verwenden Sie stattdessen das Suffix `.eli`, um den Auslagerungsspeicher zu verschlüsseln:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/ada0s1b.eli         none            swap    sw              0       0
```

In der Voreinstellung verschlüsselt geli(8) mit dem AES-Algorithmus und einer Schlüssellänge von 128 Bit. Diese Voreinstellungen sind in der Regel ausreichend, können jedoch im Options-Feld in /etc/fstab angepasst werden. Mögliche Optionen sind:

**aalgo**

Der Algorithmus für die Prüfung der Datenintegrität. Dieser wird benutzt um sicherzustellen, dass die verschlüsselten Daten nicht manipuliert wurden. Eine Liste der unterstützten Algorithmen finden Sie in geli(8).

**ealgo**

Der Verschlüsselungsalgorithmus, der verwendet wird um die Daten zu schützen. Eine Liste der unterstützten Algorithmen finden Sie in geli(8).

**keylen**

Die Länge des Schlüssels für den Verschlüsselungsalgorithmus. In geli(8) können Sie lesen, welche Schlüssellängen von welchem Algorithmus unterstützt werden.

**sectorsize**

Die Größe, in der die Datenblöcke aufgeteilt werden, bevor sie verschlüsselt werden. Größere Blöcke erhöhen die Leistung auf Kosten des Speicherverbrauchs. Die empfohlene Größe beträgt 4096 Byte.

Dieses Beispiel konfiguriert eine verschlüsselte Swap-Partition mit dem Blowfish-Algorithmus, einer Schlüssellänge von 128 Bit und einer Sektorgröße von 4 KB:

```
# Device        Mountpoint  FStype  Options             Dump    Pass#
/dev/ada0s1b.eli    none        swap    sw,ealgo=blowfish,keylen=128,sectorsize=4096
0   0
```

## 35.13.2. Überprüfung des verschlüsselten Auslagerungsspeichers

Nachdem das System neu gestartet wurde, kann die korrekte Funktion des verschlüsselten Auslagerungsspeichers mit `swapinfo` geprüft werden.

Wenn Sie gbde(8) einsetzen, erhalten Sie eine Meldung ähnlich der folgenden:

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ada0s1b.bde    542720        0   542720     0%
```

Wenn Sie geli(8) einsetzen, erhalten Sie hingegen eine Ausgabe ähnlich der folgenden:

```
% swapinfo
Device          1K-blocks     Used    Avail Capacity
/dev/ada0s1b.eli    542720        0   542720     0%
```

## 35.14. Highly Available Storage (HAST)

Hochverfügbarkeit ist eine der Hauptanforderungen von ernsthaften Geschäftsanwendungen und hochverfügbarer Speicher ist eine Schlüsselkomponente in solchen Umgebungen. Highly Available STorage (HAST) ist ein Framework in FreeBSD, welches die transparente Speicherung der gleichen Daten über mehrere physikalisch getrennte Maschinen ermöglicht, die über ein TCP/IP-Netzwerk verbunden sind. HAST kann als ein netzbasiertes RAID1 (Spiegel) verstanden werden und ist dem DRBD®-Speichersystem der GNU/Linux®-Plattform ähnlich. In Kombination mit anderen Hochverfügbarkeitseigenschaften von FreeBSD wie CARP, ermöglicht es HAST, hochverfügbare Speichercluster zu bauen, die in der Lage sind, Hardwareausfällen zu widerstehen.

Die Hauptmerkmale von HAST sind:

- Es kann zur Maskierung von I/O-Fehlern auf lokalen Festplatten eingesetzt werden.
- Dateisystem-unabhängig, was es erlaubt, jedes von FreeBSD unterstützte Dateisystem zu verwenden.
- Effiziente und schnelle Resynchronisation: es werden nur die Blöcke synchronisiert, die während der Ausfallzeit eines Knotens geändert wurden.
- Es kann in einer bereits bestehenden Umgebung eingesetzt werden, um zusätzliche Redundanz zu erreichen.
- Zusammen mit CARP, Heartbeat, oder anderen Werkzeugen, ist es möglich, ein robustes und dauerhaftes Speichersystem zu bauen.

Nachdem Sie diesen Abschnitt gelesen haben, werden Sie folgendes wissen:

- Was HAST ist, wie es funktioniert und welche Eigenschaften es besitzt.
- Wie man HAST unter FreeBSD aufsetzt und verwendet.
- Wie man CARP und devd(8) kombiniert, um ein robustes Speichersystem zu bauen.

Bevor Sie diesen Abschnitt lesen, sollten Sie:

- die Grundlagen von UNIX® und FreeBSD verstanden haben (Grundlagen des FreeBSD Betriebssystems).

- wissen, wie man Netzwerkschnittstellen und andere Kernsysteme von FreeBSD konfiguriert (Konfiguration und Tuning).
- ein gutes Verständnis der FreeBSD-Netzwerkfunktionalität besitzen (Netzwerke).

Das HAST-Projekt wurde von der FreeBSD Foundation mit Unterstützung der OMCnet Internet Service GmbH und TransIP BV gesponsert.

### 35.14.1. HAST im Einsatz

HAST bietet eine synchrone Replikation auf Blockebene zwischen zwei Maschinen: einem `primary`, auch bekannt als `master` Knoten, sowie dem `secondary`, oder `slave` Knoten. Diese beiden Maschinen zusammen werden als Cluster bezeichnet.

Da HAST in einer primär-sekundär-Konfiguration funktioniert, ist immer nur ein Knoten des Clusters zu jeder Zeit aktiv. Der primäre Knoten, auch *active* genannt, ist derjenige, der alle I/O-Anfragen verarbeitet, die an die HAST-Schnittstelle gesendet werden. Der sekundäre Knoten wird automatisch vom primären Knoten aus synchronisiert.

Die physischen Komponenten des HAST-Systems sind die lokale Platte am Primärknoten und die entfernte Platte am Sekundärknoten.

HAST arbeitet synchron auf Blockebene, was es für Dateisysteme und Anwendungen transparent macht. HAST stellt gewöhnliche GEOM-Provider in /dev/hast/ für die Verwendung durch andere Werkzeuge oder Anwendungen zur Verfügung. Es gibt keinen Unterschied zwischen dem Einsatz von HAST bereitgestellten Geräten und herkömmlichen Platten oder Partitionen.

Jede Schreib-, Lösch- oder Entleerungsoperation wird an die lokale und über TCP/IP zu der entfernt liegenden Platte gesendet. Jede Leseoperation wird von der lokalen Platte durchgeführt, es sei denn, die lokale Platte ist nicht aktuell oder es tritt ein I/O-Fehler auf. In solchen Fällen wird die Leseoperation an den Sekundärknoten geschickt.

HAST versucht, eine schnelle Fehlerbereinigung zu gewährleisten. Aus diesem Grund ist es wichtig, die Synchronisationszeit nach dem Ausfall eines Knotens zu reduzieren. Um eine schnelle Synchronisation zu ermöglichen, verwaltet HAST eine Bitmap von unsauberen Bereichen auf der Platte und synchronisiert nur diese während einer regulären Synchronisation (mit Ausnahme der initialen Synchronisation).

Es gibt viele Wege, diese Synchronisation zu behandeln. HAST implementiert mehrere Replikationsarten, um unterschiedliche Methoden der Synchronisation zu realisieren:

- *memsync*: Dieser Modus meldet Schreiboperationen als vollständig, wenn die lokale Schreiboperation beendet ist und der entfernt liegende Knoten die Ankunft der Daten bestätigt hat, jedoch bevor die Daten wirklich gespeichert wurden. Die Daten werden auf dem entfernt liegenden Knoten direkt nach dem Senden der Bestätigung gespeichert. Dieser Modus ist dafür gedacht, Latenzen zu verringern und zusätzlich eine gute Verlässlichkeit zu bieten. In der Voreinstellung wird dieser Modus benutzt.
- *fullsync*: Dieser Modus meldet Schreiboperationen als vollständig, wenn sowohl die lokale, als auch die entfernte Schreiboperation abgeschlossen wurde. Dies ist der sicherste und zugleich der langsamste Replikationsmodus.

- *async*: Dieser Modus meldet Schreiboperationen als vollständig, wenn lokale Schreibvorgänge abgeschlossen wurden. Dies ist der schnellste und gefährlichste Replikationsmodus. Er sollte nur verwendet werden, wenn die Latenz zu einem entfernten Knoten bei einer Replikation zu hoch ist für andere Modi.

### 35.14.2. HAST-Konfiguration

Das HAST-Framework besteht aus mehreren Komponenten:

- Dem hastd(8)-Daemon, welcher für Datensynchronisation verantwortlich ist. Wenn dieser Daemon gestartet wird, wird automatisch `geom_gate.ko` geladen.
- Dem hastctl(8) Management-Werkzeug.
- Der Konfigurationsdatei hast.conf(5). Diese Datei muss vorhanden sein, bevor hastd gestartet wird.

Alternativ lässt sich die `GEOM_GATE`-Unterstützung in den Kernel statisch einbauen, indem folgende Zeile zur Kernelkonfigurationsdatei hinzugefügt wird. Anschließend muss der Kernel, wie in Konfiguration des FreeBSD-Kernels beschrieben, neu gebaut werden:

```
options GEOM_GATE
```

Das folgende Beispiel beschreibt, wie man zwei Knoten als master-slave / primary-secondary mittels HAST konfiguriert, um Daten zwischen diesen beiden auszutauschen. Die Knoten werden als `hasta` mit der IP-Adresse `172.16.0.1` und `hastb` mit der IP-Adresse `172.16.0.2` bezeichnet. Beide Knoten besitzen eine dedizierte Festplatte /dev/ad6 mit der gleichen Größe für den HAST-Betrieb. Der HAST-Pool, manchmal auch Ressource genannt, oder der GEOM-Provider in /dev/hast/ wird als test bezeichnet.

Die Konfiguration von HAST wird in /etc/hast.conf vorgenommen. Diese Datei sollte auf beiden Knoten gleich sein. Die einfachste Konfiguration ist folgende:

```
resource test {
    on hasta {
        local /dev/ad6
        remote 172.16.0.2
    }
    on hastb {
        local /dev/ad6
        remote 172.16.0.1
    }
}
```

Fortgeschrittene Konfigurationsmöglichkeiten finden Sie in hast.conf(5).



Es ist ebenfalls möglich, den Hostnamen in den `remote`-Anweisungen zu verwenden, falls die Rechner aufgelöst werden können und in /etc/hosts, oder im

lokalen DNS definiert sind.

Sobald die Konfiguration auf beiden Rechnern vorhanden ist, kann ein HAST-Pool erstellt werden. Lassen Sie diese Kommandos auf beiden Knoten ablaufen, um die initialen Metadaten auf die lokale Platte zu schreiben und starten Sie anschließend hastd(8):

```
# hastctl create test
# service hastd onestart
```



Es ist *nicht* möglich, GEOM-Provider mit einem bereits bestehenden Dateisystem zu verwenden, um beispielsweise einen bestehenden Speicher in einen von HAST verwalteten Pool zu konvertieren. Dieses Verfahren muss einige Metadaten auf den Provider schreiben und dafür würde nicht genug freier Platz zur Verfügung stehen.

Die Rolle eines HAST Knotens, `primary` oder `secondary`, wird vom einem Administrator, oder einer Software wie Heartbeat, mittels hastctl(8) festgelegt. Auf dem primären Knoten `hasta` geben Sie diesen Befehl ein:

```
# hastctl role primary test
```

Geben Sie folgendes Kommando auf dem sekundären Knoten `hastb` ein:

```
# hastctl role secondary test
```

Überprüfen Sie das Ergebnis mit `hastctl` auf beiden Knoten:

```
# hastctl status test
```

Überprüfen Sie die `status`-Zeile. Wird hier `degraded` angezeigt, dann ist etwas mit der Konfigurationsdatei nicht in Ordnung. Auf jedem Konten sollte `complete` angezeigt werden, was bedeutet, dass die Synchronisation zwischen den beiden Knoten gestartet wurde. Die Synchronisierung ist abgeschlossen, wenn `hastctl status` meldet, dass die `dirty`-Bereiche 0 Bytes betragen.

Der nächste Schritt ist, ein Dateisystem auf dem GEOM-Provider anzulegen und dieses ins System einzuhängen. Dies muss auf dem `primary`-Knoten durchgeführt werden. Die Erstellung des Dateisystems kann ein paar Minuten dauern, abhängig von der Größe der Festplatte. Dieses Beispiel erstellt ein UFS-Dateisystem auf /dev/hast/test:

```
# newfs -U /dev/hast/test
# mkdir /hast/test
# mount /dev/hast/test /hast/test
```

Sobald das HAST-Framework richtig konfiguriert wurde, besteht der letzte Schritt nun darin, sicherzustellen, dass HAST während des Systemstarts automatisch gestartet wird. Fügen Sie diese Zeile in /etc/rc.conf hinzu:

```
hastd_enable="YES"
```

**Failover-Konfiguration**

Das Ziel dieses Beispiels ist, ein robustes Speichersystem zu bauen, welches Fehlern auf einem beliebigen Knoten widerstehen kann. Wenn der `primary`-Knoten ausfällt, ist der `secondary`-Knoten da, um nahtlos einzuspringen, das Dateisystem zu prüfen, einzuhängen und mit der Arbeit fortzufahren, ohne dass auch nur ein einzelnes Bit an Daten verloren geht.

Um diese Aufgabe zu bewerkstelligen, wird das Common Address Redundancy Protocol (CARP) benutzt, welches ein automatisches Failover auf der IP-Schicht ermöglicht. CARP erlaubt es mehreren Rechnern im gleichen Netzsegment, die gleiche IP-Adresse zu verwenden. Setzen Sie CARP auf beiden Knoten des Clusters anhand der Dokumentation in “Common Address Redundancy Protocol (CARP)” auf. In diesem Beispiel hat jeder Knoten seine eigene Management IP-Adresse und die geteilte IP-Adresse *172.16.0.254*. Der primäre HAST-Knoten des Clusters muss der CARP-Masterknoten sein.

Der HAST-Pool, welcher im vorherigen Abschnitt erstellt wurde, ist nun bereit für den Export über das Netzwerk auf den anderen Rechner. Dies kann durch den Export über NFS oder Samba erreicht werden, indem die geteilte IP-Adresse *172.16.0.254* verwendet wird. Das einzige ungelöste Problem ist der automatische Failover, sollte der primäre Knoten einmal ausfallen.

Falls die CARP-Schnittstelle aktiviert oder deaktiviert wird, generiert das FreeBSD-Betriebssystem ein devd(8)-Ereignis, was es ermöglicht, Zustandsänderungen auf den CARP-Schnittstellen zu überwachen. Eine Zustandsänderung auf der CARP-Schnittstelle ist ein Indiz dafür, dass einer der Knoten gerade ausgefallen oder wieder verfügbar ist. Diese Zustandsänderungen machen es möglich, ein Skript zu starten, welches automatisch den HAST-Failover durchführt.

Um Zustandsänderungen auf der CARP-Schnittstelle abzufangen, müssen diese Zeilen in /etc/devd.conf auf jedem Knoten hinzugefügt werden:

```
notify 30 {
    match "system" "IFNET";
    match "subsystem" "carp0";
    match "type" "LINK_UP";
    action "/usr/local/sbin/carp-hast-switch master";
};

notify 30 {
    match "system" "IFNET";
    match "subsystem" "carp0";
    match "type" "LINK_DOWN";
    action "/usr/local/sbin/carp-hast-switch slave";
};
```



Wenn auf dem System FreeBSD 10 oder höher eingesetzt wird, ersetzen Sie carp0 durch den Namen der konfigurierten Schnittstelle für CARP.

Starten Sie devd(8) auf beiden Knoten neu, um die neue Konfiguration wirksam werden zu lassen:

```
# service devd restart
```

Wenn die Schnittstelle aktiviert oder deaktiviert wird, erzeugt das System eine Meldung, was es dem devd(8)-Subsystem ermöglicht, ein automatisches Failover-Skript zu starten, /usr/local/sbin/carp-hast-switch. Weitere Informationen zu dieser Konfiguration finden Sie in devd.conf(5).

Es folgt ein Beispiel für ein automatisches Failover-Skript:

```
#!/bin/sh

# Original script by Freddie Cash <fjwcash@gmail.com>
# Modified by Michael W. Lucas <mwlucas@BlackHelicopters.org>
# and Viktor Petersson <vpetersson@wireload.net>

# The names of the HAST resources, as listed in /etc/hast.conf
resources="test"

# delay in mounting HAST resource after becoming master
# make your best guess
delay=3

# logging
log="local0.debug"
name="carp-hast"

# end of user configurable stuff

case "$1" in
    master)
        logger -p $log -t $name "Switching to primary provider for ${resources}."
        sleep ${delay}

        # Wait for any "hastd secondary" processes to stop
        for disk in ${resources}; do
            while $( pgrep -lf "hastd: ${disk} \(secondary\)" > /dev/null 2>&1 ); do
                sleep 1
            done

            # Switch role for each disk
            hastctl role primary ${disk}
            if [ $? -ne 0 ]; then
                logger -p $log -t $name "Unable to change role to primary for resource
${disk}."
```

```
                exit 1
            fi
        done

        # Wait for the /dev/hast/* devices to appear
        for disk in ${resources}; do
            for I in $( jot 60 ); do
                [ -c "/dev/hast/${disk}" ] && break
                sleep 0.5
            done

            if [ ! -c "/dev/hast/${disk}" ]; then
                logger -p $log -t $name "GEOM provider /dev/hast/${disk} did not
appear."
                exit 1
            fi
        done

        logger -p $log -t $name "Role for HAST resources ${resources} switched to
primary."

        logger -p $log -t $name "Mounting disks."
        for disk in ${resources}; do
            mkdir -p /hast/${disk}
            fsck -p -y -t ufs /dev/hast/${disk}
            mount /dev/hast/${disk} /hast/${disk}
        done

    ;;

    slave)
        logger -p $log -t $name "Switching to secondary provider for ${resources}."

        # Switch roles for the HAST resources
        for disk in ${resources}; do
            if ! mount | grep -q "^/dev/hast/${disk} on "
            then
            else
                umount -f /hast/${disk}
            fi
            sleep $delay
            hastctl role secondary ${disk} 2>&1
            if [ $? -ne 0 ]; then
                logger -p $log -t $name "Unable to switch role to secondary for
resource ${disk}."
                exit 1
            fi
            logger -p $log -t $name "Role switched to secondary for resource ${disk}."
        done
    ;;
```

```
esac
```

Im Kern führt das Skript die folgenden Aktionen durch, sobald ein Knoten zum Master wird:

- Es ernennt den HAST-Pool als den primären für einen gegebenen Knoten.
- Es prüft das Dateisystem, dass auf dem HAST-Pool erstellt wurde.
- Es hängt den Pool ins System ein.

Wenn ein Knoten zum Sekundären ernannt wird:

- Hängt es den HAST-Pool aus dem Dateisystem aus.
- Degradiert es den HAST-Pool zum sekundären.



Dieses Skript ist nur ein Beispiel für eine mögliche Lösung. Es behandelt nicht alle möglichen Szenarien, die auftreten können und sollte erweitert bzw. abgeändert werden, so dass z.B. benötigte Dienste gestartet oder gestoppt werden.



Für dieses Beispiel wurde ein UFS-Dateisystem verwendet. Um die Zeit für die Wiederherstellung zu verringern, kann ein UFS mit Journal oder ein ZFS-Dateisystem benutzt werden.

Weitere detaillierte Informationen mit zusätzlichen Beispielen können unter http://wiki.FreeBSD.org/HAST abgerufen werden.

### 35.14.3. Fehlerbehebung

HAST sollte generell ohne Probleme funktionieren. Jedoch kann es, wie bei jeder anderen Software auch, zu gewissen Zeiten sein, dass sie sich nicht so verhält wie angegeben. Die Quelle dieser Probleme kann unterschiedlich sein, jedoch sollte als Faustregel gewährleistet werden, dass die Zeit für alle Knoten im Cluster synchron läuft.

Für die Fehlersuche bei HAST sollte die Anzahl an Debugging-Meldungen von hastd(8) erhöht werden. Dies kann durch das Starten von `hastd` mit `-d` erreicht werden. Diese Option kann mehrfach angegeben werden, um die Anzahl an Meldungen weiter zu erhöhen. Sie sollten ebenfalls die Verwendung von `-F` in Erwägung ziehen, was `hastd` im Vordergrund startet.

#### Auflösung des Split-brain-Zustands

`split-brain` bezeichnet eine Situation, in der beide Knoten des Clusters nicht in der Lage sind, miteinander zu kommunizieren und dadurch beide als primäre Knoten fungieren. Dies ist ein gefährlicher Zustand, weil es beiden Knoten erlaubt ist, Änderungen an den Daten vorzunehmen, die miteinander nicht in Einklang gebracht werden können. Diese Situation muss vom Systemadministrator manuell bereinigt werden.

Der Administrator muss entscheiden, welcher Knoten die wichtigeren Änderungen besitzt, oder die Zusammenführung manuell durchführen. Anschließend kann HAST die volle Synchronisation mit dem Knoten durchführen, der die beschädigten Daten enthält. Um dies zu tun, geben Sie folgende

Befehle auf dem Knoten ein, der neu synchronisiert werden muss:

```
# hastctl role init test
# hastctl create test
# hastctl role secondary test
```

# Kapitel 36. GEOM: Modulares Framework zur Plattentransformation

## 36.1. Übersicht

GEOM erlaubt den Zugriff und die Kontrolle von Klassen, wie beispielsweise Master Boot Records und BSD-Label, durch die Nutzung von Datenträgern (Providern) oder den besonderen Dateien in /dev. Verschiedene Software RAID-Konfigurationen unterstützend, gewährt GEOM transparenten Zugriff auf das Betriebssystem und die System-Dienstprogramme.

Dieses Kapitel behandelt den Einsatz von Laufwerken mit dem GEOM-Framework in FreeBSD. Dies beinhaltet auch die wichtigen RAID-Überwachungswerkzeuge, welche das Framework zur Konfiguration nutzen. Dieses Kapitel ist kein ausführlicher Leitfaden für RAID-Konfigurationen. Nur die von GEOM unterstützten RAID-Klassen werden erörtert.

Nach Lesen dieses Kapitels werden Sie folgendes wissen:

- Welche Art von RAID-Unterstützung durch GEOM verfügbar ist.
- Wie man die Basis-Dienstprogramme nutzt, um verschiedene RAID-Stufen zu konfigurieren, zu manipulieren und zu warten.
- Wie man mittels GEOM spiegelt, striped, verschlüsselt und entfernte Laufwerke verbindet.
- Wie man an Laufwerken, welche an das GEOM-Framework angeschlossen sind, Fehler behebt.

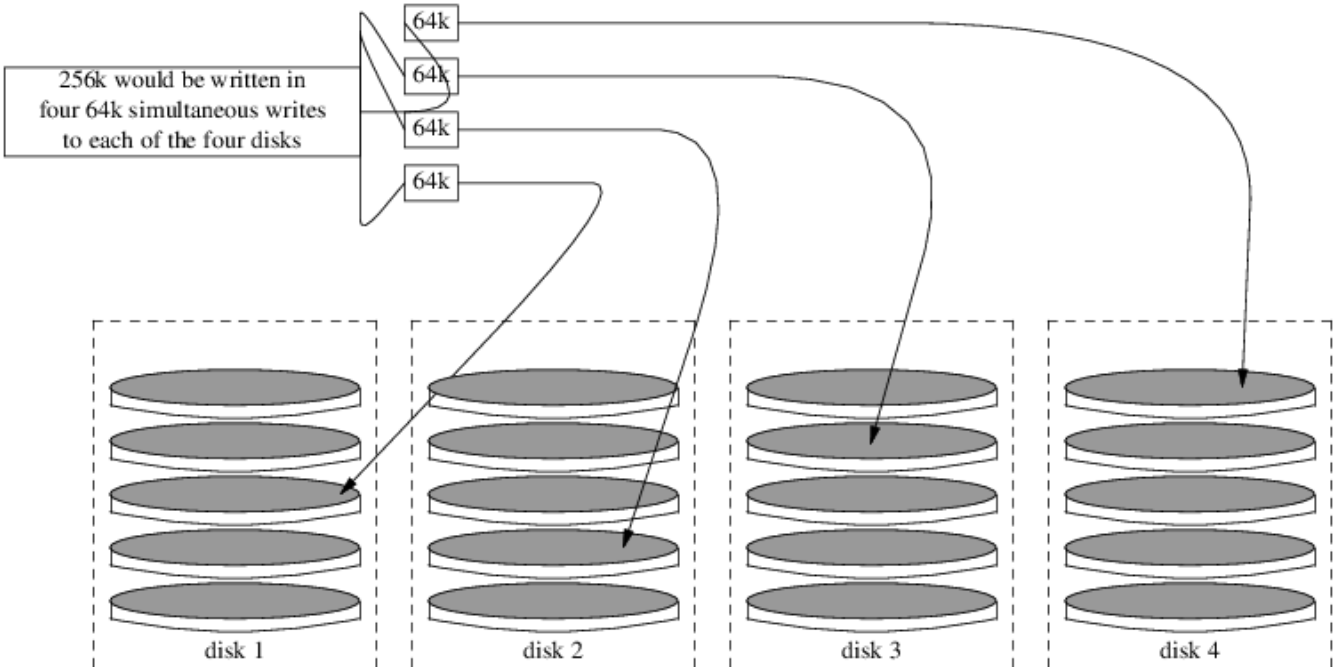
Bevor Sie dieses Kapitel lesen, sollten Sie:

- Verstehen, wie FreeBSD Laufwerke behandelt (Speichermedien).
- Wissen wie man einen neuen FreeBSD-Kernel konfiguriert und installiert (Konfiguration des FreeBSD-Kernels).

## 36.2. RAID0 - Striping

Striping (stripe = Streifen) fasst verschiedene Laufwerke in einem einzigen Datenträger zusammen. Dies wird durch die Nutzung von Hardware-Controllern bewerkstelligt. Das GEOM-Subsystem unterstützt Software-RAID0, welches auch als Striping bekannt ist. Bei dieser Technik wird kein RAID-Controller benötigt.

In einem RAID0-System werden die Daten in einzelne Blöcke aufgeteilt, welche über alle angeschlossenen Laufwerke in einem Datenfeld (Array) geschrieben werden. Anstatt darauf warten zu müssen, dass 256K auf ein einzelnes Laufwerk geschrieben werden, kann ein RAID0-System gleichzeitig 64K auf jedes von vier Laufwerken schreiben mit entsprechend besserer I/O-Leistung. Dieser Durchsatz kann durch die Verwendung mehrerer Controller noch zusätzlich gesteigert werden.




Jedes Laufwerk in einem RAID0-Stripe muss die gleiche Größe haben, da I/O-Anforderungen für das Lesen und Schreiben abwechselnd auf mehrere Laufwerke parallel erfolgen.



RAID0 bietet keine Redundanz. Das bedeutet, dass wenn eine Platte im Array ausfällt, die gesamten Daten auf den Platten verloren gehen. Wenn es sich um wichtige Daten handelt, sollten Sie eine Backup-Strategie entwickeln, die regelmäßig Sicherungen auf einem entferntem System speichert.

Die Erstellung eines GEOM-basierten RAID0 auf einem FreeBSD-System wird im folgenden beschrieben. Nachdem das Stripe erzeugt wurde, finden Sie in gstripe(8) weitere Informationen zur Verwaltung der vorhandenen Stripes.

**Procedure: Ein Stripe aus unformatierten ATA-Platten erzeugen**

1. Laden Sie das geom_stripe.ko-Modul:

```
# kldload geom_stripe
```

2. Stellen Sie sicher, dass ein geeigneter Mountpunkt existiert. Falls dieser Datenträger eine Root-Partition werden soll, dann nutzen Sie zeitweise einen anderen Mountpunkt, beispielsweise /mnt.
3. Bestimmen Sie die Gerätenamen derjenigen Platten, welche gestriped werden sollen, und erzeugen Sie ein neues Stripe-Gerät. Das folgende Beispiel verwendet zwei unbenutzte und unpartitionierte ATA-Platten, die gestriped werden sollen. Die Gerätenamen lauten /dev/ad2 und /dev/ad3:

```
# gstripe label -v st0 /dev/ad2 /dev/ad3
Metadata value stored on /dev/ad2.
Metadata value stored on /dev/ad3.
```

```
Done.
```

4. Schreiben Sie einen Standard-Label (auch als Partitions-Tabelle bekannt) auf den neuen Datenträger und installieren Sie den normalen Bootstrap-Code:

```
# bsdlabel -wB /dev/stripe/st0
```

5. Dieser Prozess sollte zwei weitere Geräte im Verzeichnis /dev/stripe (zusätzlich zum Gerät st0) erzeugt haben. Diese schliessen st0a und st0c ein. Nun kann mit `newfs` ein UFS-Dateisystem auf dem Gerät st0a erzeugt werden:

```
# newfs -U /dev/stripe/st0a
```

   Viele Zahlen rauschen nun über den Bildschirm und nach ein paar Sekunden wird der Prozess abgeschlossen sein. Der Datenträger wurde erzeugt und kann in den Verzeichnisbaum eingehängt werden.

6. Um das erzeugte Stripe manuell zu mounten:

```
# mount /dev/stripe/st0a /mnt
```

7. Um das erzeugte Dateisystem automatisch während des Startvorgangs zu mounten, muss die Datenträgerinformation in /etc/fstab eingetragen werden. In diesem Beispiel wird ein permanenter Mountpunkt namens stripe erstellt:

```
# mkdir /stripe
# echo "/dev/stripe/st0a /stripe ufs rw 2 2" \
>> /etc/fstab
```

8. Das geom_stripe.ko-Modul muss ebenfalls automatisch beim Systemstart geladen werden (durch die Aufnahme der folgenden Zeile in die Datei /boot/loader.conf):

```
# echo 'geom_stripe_load="YES"' >> /boot/loader.conf
```

## 36.3. RAID1 - Spiegelung

Spiegelung (RAID1 / *Mirroring*) ist eine Technik, bei der identische Daten auf mehr als ein Laufwerk geschrieben werden. Spiegel werden in der Regel zum Schutz vor Datenverlust aufgrund von Festplattenausfällen verwendet. Jedes Laufwerk in einem Spiegel enthält eine identische Kopie der Daten. Wenn ein einzelnes Laufwerk ausfällt, funktioniert der Spiegel weiterhin und die Daten werden von den restlichen Festplatten bereit gestellt. Der Rechner läuft einfach weiter und der Administrator hat die Gelegentheit, das defekte Laufwerk auszutauschen.

Zwei häufige Situationen werden in diesem Beispiel erläutert. Im ersten Beispiel wird ein Spiegel aus zwei neuen Laufwerken erstellt, der die existierende Platte ersetzt. Das zweite Beispiel erzeugt ein Spiegel mit einem einzigen Laufwerk, kopiert dann die Daten von der alten Platte und fügt die alte Platte zum Spiegel hinzu. Obwohl dieses Verfahren etwas komplizierter ist, wird nur ein neues Laufwerk benötigt.

Traditionell sind die Laufwerke in einem Spiegel vom gleichen Modell und besitzen die gleiche Kapazität. Dies ist jedoch keine Voraussetzung für gmirror(8). Hier können Spiegel mit unterschiedlichen Kapazitäten verwendet werden. Die Kapazität richtet sich dann nach dem kleinsten Laufwerk im Spiegel. Zusätzlicher Speicherplatz auf größeren Laufwerken bleibt dann ungenutzt. Werden später weitere Laufwerke zum Spiegel hinzugefügt, müssen diese mindestens so viel Kapazität haben wie das kleinste Laufwerk im Spiegel.



Die hier gezeigten Verfahren löschen keine Daten. Dennoch sollte, wie bei jeder größeren Operation, zuerst eine vollständige Sicherung erstellt werden.



Obwohl in diesem Abschnitt dump(8) zum Kopieren der Dateisysteme verwendet wird, funktioniert es nicht auf Dateisystemen mit aktiviertem Soft-Updates Journaling. In tunefs(8) finden Sie Informationen, wie Sie Soft-Updates Journaling erkennen und deaktivieren.

### 36.3.1. Probleme mit Metadaten

Viele Plattensysteme speichern Metadaten am Ende der Platte. Alte Metadaten sollten vor der Wiederverwendung in einem Spiegel gelöscht werden, da die meisten Probleme aus zwei Arten von übrig gebliebenen Metadaten resultieren: GPT-Partitionstabellen und alte Metadaten aus einem vorherigen Spiegel.

GPT-Metadaten können mit gpart(8) gelöscht werden. Dieses Beispiel löscht sowohl die primären, als auch die GPT-Partitionstabelle von der Festplatte ada8:

```
# gpart destroy -F ada8
```

Mit gmirror(8) kann eine Platte aus einem aktiven Spiegel entfernt und gleichzeitig die Metadaten gelöscht werden. In diesem Beispiel wird die Platte ada8 aus dem aktiven Spiegel gm4 entfernt:

```
# gmirror remove gm4 ada8
```

Wenn der Spiegel nicht aktiv ist, sich jedoch noch alte Metadaten auf der Festplatte befinden, benutzen Sie `gmirror clear`, um die Metadaten zu entfernen:

```
# gmirror clear ada8
```

gmirror(8) speichert einen Datenblock an Metadaten am Ende der Festplatte. Da das GPT-Partitionschema die Metadaten auch am Ende der Platte speichert, wird es nicht empfohlen, mit

gmirror(8) einen Spiegel aus einem gesamten GPT-Datenträger zu erstellen. In diesen Fällen sollte eine MBR-Partitionierung benutzt werden, weil hier nur eine Partitionstabelle am Anfang der Platte gespeichert wird und somit nicht mit den Metadaten des Spiegels im Konflikt steht.

### 36.3.2. Einen Spiegel mit zwei neuen Festplatten erstellen

In diesem Beispiel wurde FreeBSD bereits auf der vorhandenen Festplatte ada0 installiert. Zwei neue Platten, ada1 und ada2, wurden bereits mit dem System verbunden. Ein neuer Spiegel soll mit diesen beiden Platten erzeugt und verwendet werden, um die alte vorhandene Platte zu ersetzen.

Das Kernelmodul geom_mirror.ko muss entweder in den Kernel eingebaut, oder zur Laufzeit geladen werden. Sie können das Modul manuell laden:

```
# gmirror load
```

Erstellen Sie den Spiegel mit den beiden neuen Festplatten:

```
# gmirror label -v gm0 /dev/ada1 /dev/ada2
```

gm0 ist ein vom Benutzer gewählter Name, der dem neuen Spiegel zugeordnet wird. Nachdem der Spiegel gestartet wurde, erscheint dieser Gerätename in /dev/mirror/.

MBR- und bsdlabel-Partitionstabellen können jetzt auf dem neuen Spiegel erzeugt werden. Dieses Beispiel verwendet das herkömmliche Dateisystem-Layout für /, swap, /var, /tmp und /usr. Eine einzelne Root- und Swap-Partition würde ebenfalls funktionieren.

Die Partitionen auf dem Spiegel müssen nicht zwingend die gleiche Größe wie die auf der Festplatte haben, aber sie müssen groß genug sein, um alle Daten aufnehmen zu können, die bereits auf ada0 gespeichert sind.

```
# gpart create -s MBR mirror/gm0
# gpart add -t freebsd -a 4k mirror/gm0
# gpart show mirror/gm0
=>       63  156301423  mirror/gm0  MBR  (74G)
         63         63                  - free -  (31k)
        126  156301299                1  freebsd  (74G)
  156301425         61                  - free -  (30k)
```

```
# gpart create -s BSD mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-swap -a 4k -s 4g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 2g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k -s 1g mirror/gm0s1
# gpart add -t freebsd-ufs  -a 4k mirror/gm0s1
# gpart show mirror/gm0s1
=>        0  156301299  mirror/gm0s1  BSD  (74G)
```

```
          0          2                    - free -  (1.0k)
          2    4194304                 1  freebsd-ufs  (2.0G)
    4194306    8388608                 2  freebsd-swap (4.0G)
   12582914    4194304                 4  freebsd-ufs  (2.0G)
   16777218    2097152                 5  freebsd-ufs  (1.0G)
   18874370  137426928                 6  freebsd-ufs  (65G)
  156301298          1                    - free -  (512B)
```

Damit von dem Spiegel gebootet werden kann, muss der Bootcode in den MBR installiert, ein bsdlabel erstellt und die aktive Partition gesetzt werden:

```
# gpart bootcode -b /boot/mbr mirror/gm0
# gpart set -a active -i 1 mirror/gm0
# gpart bootcode -b /boot/boot mirror/gm0s1
```

Erstellen Sie die Dateisysteme auf dem neuen Spiegel und aktivieren Sie Soft-Updates:

```
# newfs -U /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
```

Die Dateisysteme der vorhandenen Platte ada0 können jetzt mit dump(8) und restore(8) auf den Spiegel kopiert werden.

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - / | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/tmp
# mount /dev/mirror/gm0s1f /mnt/usr
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /tmp | (cd /mnt/tmp && restore -rf -)
# dump -C16 -b64 -0aL -f - /usr | (cd /mnt/usr && restore -rf -)
```

Fügen Sie die Dateisysteme für den Spiegel in /etc/rc.conf hinzu:

```
# Device          Mountpoint       FStype  Options Dump    Pass#
/dev/mirror/gm0s1a      /               ufs     rw       1       1
/dev/mirror/gm0s1b      none            swap    sw       0       0
/dev/mirror/gm0s1d      /var            ufs     rw       2       2
/dev/mirror/gm0s1e      /tmp            ufs     rw       2       2
/dev/mirror/gm0s1f      /usr            ufs     rw       2       2
```

Wenn das Modul geom_mirror.ko nicht im Kernel enthalten ist, können Sie /mnt/boot/loader.conf bearbeiten, damit das Modul beim Systemstart geladen wird:

```
geom_mirror_load="YES"
```

Starten Sie das System neu und überprüfen Sie, ob alle Daten erfolgreich kopiert wurden. Das BIOS wird den Spiegel vermutlich als zwei einzelne Laufwerke erkennen. Da beide Laufwerke jedoch identisch sind, spielt es keine Rolle, welches Laufwerk zum Booten ausgewählt wird.

Falls es Probleme beim Booten gibt, lesen Sie den Fehlerbehebung. Die alte Festplatte ada0 kann vom System getrennt und als Offline-Sicherung aufbewahrt werden.

Im laufenden Betrieb verhält sich der Spiegel genau wie ein einzelnes Laufwerk.

### 36.3.3. Einen Spiegel mit einem vorhandenen Laufwerk erstellen

In diesem Beispiel wurde FreeBSD bereits auf der Festplatte ada0 installiert und eine weitere Platte, ada1, wurde an das System angeschlossen. Zunächst wird ein Spiegel mit einer Festplatte erstellt, dann das vorhandene System auf den Spiegel kopiert. Zuletzt wird die alte Festplatte in den Spiegel eingefügt. Diese etwas komplexere Vorgehensweise ist erforderlich, da `gmirror` 512 Byte an Metadaten am Ende der Festplatte speichert, und die bestehende Platte, ada0, in der Regel den Platz bereits belegt hat.

Laden Sie das Kernelmodul geom_mirror.ko:

```
# gmirror load
```

Prüfen Sie mit `diskinfo` die Mediengröße der vorhandenen Festplatte:

```
# diskinfo -v ada0 | head -n3
/dev/ada0
        512             # sectorsize
        1000204821504   # mediasize in bytes (931G)
```

Jetzt können Sie den Spiegel auf der neuen Festplatte erzeugen. Um sicherzustellen, dass die Kapazität nicht größer ist, als die Kapazität der vorhandenen Platte ada0, benutzen Sie gnop(8) um eine Platte mit der exakt gleichen Größe zu imitieren. Diese Platte speichert keine Daten und wird nur verwendet, um die Größe des Spiegels zu begrenzen. gmirror(8) wird die Kapazität des Spiegels auf die Größe von gzero.nop beschränken, auch wenn die neue Festplatte ada1 mehr Platz zur Verfügung hätte. Beachten Sie, dass *1000204821504* in der zweiten Zeile der ermittelten Mediengröße von `diskinfo` entspricht.

```
# geom zero load
# gnop create -s 1000204821504 gzero
# gmirror label -v gm0 gzero.nop ada1
# gmirror forget gm0
```

Da gzero.nop keine Daten speichert, sieht der Spiegel sie als nicht verbunden an. Der Spiegel ist so

konfiguriert, dass er nicht verbundene Komponenten einfach "vergisst". Das Ergebnis ist ein Spiegel mit nur einer einzigen Platte, ada1.

Sehen Sie sich nach der Erstellung von gm0 die Partitionstabelle von ada0 an. Diese Ausgabe stammt von einer 1 TB Festplatte. Falls am Ende der Platte noch freier Speicherplatz ist, kann der Inhalt von ada0 direkt auf den Spiegel kopiert werden.

Falls jedoch der gesamte Speicherplatz auf der Platte zugeordnet ist, dann gibt es keinen Platz mehr für die 512 Byte Metadaten für den Spiegel am Ende der Platte, wie in dieser Auflistung zu sehen.

```
# gpart show ada0
=>        63  1953525105        ada0  MBR  (931G)
          63  1953525105           1  freebsd  [active]  (931G)
```

In diesem Fall muss die Partitionstabelle bearbeitet werden, um die Kapazität von mirror/gm0 um einen Sektor zu reduzieren. Dieses Verfahren wird später erläutert.

In beiden Fällen sollte die Partitionstabelle der primären Platte mit `gpart backup` gesichert werden.

```
# gpart backup ada0 > table.ada0
# gpart backup ada0s1 > table.ada0s1
```

Diese Kommandos erstellen zwei Dateien, table.ada0 und table.ada0s1. Das Beispiel verwendet eine 1 TB Festplatte:

```
# cat table.ada0
MBR 4
1 freebsd         63 1953525105   [active]
```

```
# cat table.ada0s1
BSD 8
1  freebsd-ufs          0    4194304
2 freebsd-swap    4194304   33554432
4  freebsd-ufs   37748736   50331648
5  freebsd-ufs   88080384   41943040
6  freebsd-ufs  130023424  838860800
7  freebsd-ufs  968884224  984640881
```

Wenn am Ende der Platte kein Platz vorhanden ist, muss die Größe des Slice und der letzten Partition verringert werden. Bearbeiten Sie die beiden Dateien, und verringern Sie die Größe der Slice und der Partition jeweils um eins. Dies bezieht sich auf die letzten Zahlen in der Liste.

```
# cat table.ada0
MBR 4
1 freebsd         63 1953525104   [active]
```

```
# cat table.ada0s1
BSD 8
1  freebsd-ufs          0    4194304
2 freebsd-swap    4194304   33554432
4  freebsd-ufs   37748736   50331648
5  freebsd-ufs   88080384   41943040
6  freebsd-ufs  130023424  838860800
7  freebsd-ufs  968884224  984640880
```

Wenn mindestens ein Sektor der Platte nicht zugewiesen wurde, kann die Platte ohne Modifikation verwendet werden.

Jetzt kann die Partitionstabelle auf mirror/gm0 wiederhergestellt werden:

```
# gpart restore mirror/gm0 < table.ada0
# gpart restore mirror/gm0s1 < table.ada0s1
```

Prüfen Sie die Partitionstabellen mit `gpart show`. Dieses Beispiel nutzt gm0s1a für /, gm0s1d für /var, gm0s1e für /usr, gm0s1f für /data1 und gm0s1g für /data2.

```
# gpart show mirror/gm0
=>        63  1953525104  mirror/gm0  MBR  (931G)
          63  1953525042           1  freebsd  [active]  (931G)
  1953525105          62              - free -  (31k)

# gpart show mirror/gm0s1
=>         0  1953525042  mirror/gm0s1  BSD  (931G)
           0     2097152             1  freebsd-ufs  (1.0G)
     2097152    16777216             2  freebsd-swap  (8.0G)
    18874368    41943040             4  freebsd-ufs  (20G)
    60817408    20971520             5  freebsd-ufs  (10G)
    81788928   629145600             6  freebsd-ufs  (300G)
   710934528  1242590514             7  freebsd-ufs  (592G)
  1953525042          63                - free -  (31k)
```

Sowohl die Slice, als auch die letzte Partition, muss mindestens einen freien Block am Ende der Platte haben.

Erstellen Sie Dateisysteme auf diesen neuen Partitionen:

```
# newfs -U /dev/mirror/gm0s1a
# newfs -U /dev/mirror/gm0s1d
# newfs -U /dev/mirror/gm0s1e
# newfs -U /dev/mirror/gm0s1f
# newfs -U /dev/mirror/gm0s1g
```

Damit Sie von dem Spiegel booten können, müssen Sie den Bootcode in den MBR installieren, ein bsdlabel anlegen und das aktive Slice setzen:

```
# gpart bootcode -b /boot/mbr mirror/gm0
# gpart set -a active -i 1 mirror/gm0
# gpart bootcode -b /boot/boot mirror/gm0s1
```

Bearbeiten Sie /etc/fstab, um die neuen Partitionen auf dem Spiegel nutzen zu können. Speichern Sie zunächst eine Kopie der Datei unter /etc/fstab.orig:

```
# cp /etc/fstab /etc/fstab.orig
```

Ersetzen Sie in /etc/fstab /dev/ada0 durch mirror/gm0.

```
# Device        Mountpoint  FStype  Options Dump    Pass#
/dev/mirror/gm0s1a  /       ufs rw  1   1
/dev/mirror/gm0s1b  none        swap    sw  0   0
/dev/mirror/gm0s1d  /var        ufs rw  2   2
/dev/mirror/gm0s1e  /usr        ufs rw  2   2
/dev/mirror/gm0s1f  /data1      ufs rw  2   2
/dev/mirror/gm0s1g  /data2      ufs rw  2   2
```

Wenn das Modul geom_mirror.ko nicht im Kernel enthalten ist, können Sie /boot/loader.conf bearbeiten, damit das Modul beim Systemstart geladen wird:

```
geom_mirror_load="YES"
```

Die Dateisysteme der ursprünglichen Platte können jetzt mit dump(8) und restore(8) auf den Spiegel kopiert werden. Wenn Sie das Dateisystem mit `dump -L` sichern, wird zunächst ein Snapshot des Dateisystems erstellt, was einige Zeit dauern kann.

```
# mount /dev/mirror/gm0s1a /mnt
# dump -C16 -b64 -0aL -f - / | (cd /mnt && restore -rf -)
# mount /dev/mirror/gm0s1d /mnt/var
# mount /dev/mirror/gm0s1e /mnt/usr
# mount /dev/mirror/gm0s1f /mnt/data1
# mount /dev/mirror/gm0s1g /mnt/data2
# dump -C16 -b64 -0aL -f - /usr | (cd /mnt/usr && restore -rf -)
# dump -C16 -b64 -0aL -f - /var | (cd /mnt/var && restore -rf -)
# dump -C16 -b64 -0aL -f - /data1 | (cd /mnt/data1 && restore -rf -)
# dump -C16 -b64 -0aL -f - /data2 | (cd /mnt/data2 && restore -rf -)
```

Starten Sie das System neu und booten Sie von ada1. Wenn alles funktioniert, wird das System von mirror/gm0 booten, welches jetzt die gleichen Daten enthält wie ada0. Lesen Sie Fehlerbehebung,

falls es Probleme beim Booten gibt.

An dieser Stelle besteht der Spiegel immer noch aus der einzelnen Platte ada1.

Nachdem erfolgreich von mirror/gm0 gebootet wurde, besteht der letzte Schritt darin, ada0 in den Spiegel einzufügen.



Wenn Sie ada0 in den Spiegel einfügen, wird der Inhalt der Platte mit den Daten aus dem Spiegel überschrieben. Sie müssen sicherstellen, das mirror/gm0 den gleichen Inhalt wie ada0 hat, bevor Sie ada0 zum Spiegel hinzufügen. Falls der zuvor mit dump(8) und restore(8) kopierte Inhalt nicht mit dem von ada0 identisch ist, machen Sie die Änderungen an /etc/fstab rückgängig, starten Sie das System neu und beginnen Sie die Prozedur von vorn.

```
# gmirror insert gm0 ada0
GEOM_MIRROR: Device gm0: rebuilding provider ada0
```

Die Synchronisation zwischen den beiden Platten wird direkt gestartet. Verwenden Sie `gmirror status` um den Fortschritt zu beobachten.

```
# gmirror status
      Name    Status  Components
girror/gm0  DEGRADED  ada1 (ACTIVE)
                      ada0 (SYNCHRONIZING, 64%)
```

Nach einer Weile wird die Wiederherstellung abgeschlossen sein.

```
GEOM_MIRROR: Device gm0: rebuilding provider ada0 finished.
# gmirror status
      Name    Status  Components
mirror/gm0  COMPLETE  ada1 (ACTIVE)
                      ada0 (ACTIVE)
```

mirror/gm0 besteht nun aus den beiden Platten ada0 und ada1. Der Inhalt der beiden Platten wird automatisch miteinander synchronisiert. Im laufenden Betrieb verhält sich mirror/gm0 wie eine einzelne Festplatte.

### 36.3.4. Fehlerbehebung

Falls das System nicht mehr startet, müssen möglicherweise die BIOS-Einstellungen geändert werden, um von dem neuen gespiegelten Laufwerk zu booten. Beide Platten des Spiegels können zum Booten verwendet werden, da sie als Komponenten des Spiegels identische Daten enthalten.

Wenn der Bootvorgang mit der folgenden Meldung abbricht, ist irgendwas mit dem Spiegel nicht in Ordnung:

```
Mounting from ufs:/dev/mirror/gm0s1a failed with error 19.

Loader variables:
  vfs.root.mountfrom=ufs:/dev/mirror/gm0s1a
  vfs.root.mountfrom.options=rw

Manual root filesystem specification:
  <fstype>:<device> [options]
      Mount <device> using filesystem <fstype>
      and with the specified (optional) option list.

    eg. ufs:/dev/da0s1a
        zfs:tank
        cd9660:/dev/acd0 ro
          (which is equivalent to: mount -t cd9660 -o ro /dev/acd0 /)

  ?               List valid disk boot devices
  .               Yield 1 second (for background tasks)
  <empty line>    Abort manual input

mountroot>
```

Dieses Problem kann durch ein nicht geladenes Kernelmodul geom_mirror.ko in /boot/loader.conf verursacht werden. Um das Problem zu beheben, booten Sie von einem FreeBSD-Installationsmedium und wählen Sie `Shell` an der Eingabeaufforderung. Laden Sie dann das Modul und hängen Sie den Spiegel ein:

```
# gmirror load
# mount /dev/mirror/gm0s1a /mnt
```

Bearbeiten Sie dann /mnt/boot/loader.conf und fügen Sie eine Zeile für das Kernelmodul hinzu:

```
geom_mirror_load="YES"
```

Speichern Sie die Datei und starten Sie das System neu.

Andere Probleme, die `error 19` verursachen können, sind nur mit mehr Aufwand zu beheben. Obwohl das System von ada0 booten sollte, wird ein weiterer Prompt erscheinen, wenn /etc/fstab fehlerhaft ist. Geben Sie am Loader-Prompt `ufs:/dev/ada0s1a` ein und drücken Sie `Enter`. Machen Sie die Änderungen an /etc/fstab rückgängig und hängen Sie anstelle des Spiegels die originale Festplatte (ada0) ein. Starten Sie dann das System neu und versuchen Sie den Vorgang erneut.

```
Enter full pathname of shell or RETURN for /bin/sh:
# cp /etc/fstab.orig /etc/fstab
# reboot
```

### 36.3.5. Wiederherstellung des Systems nach einem Plattenausfall

Der Vorteil der Plattenspiegelung ist, dass eine Platte ausfallen kann, ohne dass Sie dabei Daten verlieren. Falls ada0 aus dem obigen Beispiel ausfällt, steht der Spiegel weiterhin zur Verfügung und bietet die Daten von der verbleibenden Platte ada1 an.

Um das ausgefallene Laufwerk zu ersetzen, muss das System heruntergefahren werden und das ausgefallene Laufwerk durch ein neues Laufwerk von gleicher oder größerer Kapazität ersetzt werden. Hersteller verwenden oft etwas willkürliche Werte für die Kapazität. Der einzige Weg, um wirklich sicher zu sein, ist die Gesamtzahl der Sektoren von `diskinfo -V` zu vergleichen. Ein Laufwerk mit größerer Kapazität wird funktionieren, allerdings wird der zusätzliche Platz ungenutzt bleiben.

Nachdem der Rechner wieder eingeschaltet ist, wird der Spiegel im "degraded" Modus ausgeführt werden. Der Spiegel wird angewiesen, Laufwerke zu vergessen, die noch nicht verbunden sind:

```
# gmirror forget gm0
```

Alte Metadaten sollten von der Ersatzfestplatte nach den Anweisungen in Probleme mit Metadaten gelöscht werden. Anschließend kann die Ersatzfestplatte, in diesem Beispiel ada4, in den Spiegel eingefügt werden:

```
# gmirror insert gm0 /dev/ada4
```

Die Wiederherstellung beginnt, sobald das neue Laufwerk in den Spiegel eingesetzt wird. Das Kopieren der Daten vom Spiegel auf das neue Laufwerk kann eine Weile dauern. Die Leistung des Spiegels ist während dieser Zeit stark reduziert, deswegen sollten neue Laufwerke idealerweise dann eingefügt werden, wenn der Rechner nicht benötigt wird.

Der Fortschritt der Wiederherstellung kann mit `gmirror status` überwacht werden. Während der Wiederherstellung ist der Status `DEGRADED`. Wenn der Vorgang abgeschlossen ist, wechselt der Status zu `COMPLETE`.

## 36.4. RAID3 - Byte-Level Striping mit dedizierter Parität

RAID3 ist eine Methode, die mehrere Festplatten zu einem einzigen Volume mit einer dedizierten Paritätsfestplatte kombiniert. In einem RAID3-System werden die Daten in einzelne Bytes aufgeteilt und dann über alle Laufwerke, mit Ausnahme der Paritätsfestplatte, geschrieben. Beim Lesen von Daten in einer RAID3 Implementierung werden alle Festplatten im Array parallel genutzt. Die Leistung kann durch den Einsatz von mehreren Controllern weiter erhöht werden. Ein RAID3-Array hat eine Fehlertoleranz von 1 Laufwerk und bietet dabei eine Kapazität von 1 - 1/n der Gesamtkapazität der Laufwerke im Array, wobei n die Anzahl der Festplatten im Array darstellt. So eine Konfiguration ist meistens für die Speicherung von größeren Dateien geeignet, wie beispielsweise Multimediadateien.

Mindestens 3 Festplatten sind erforderlich, um ein RAID3 zu erstellen. Jede Festplatte muss von der gleichen Größe sein, da die I/O-Anfragen für Lesen oder Schreiben auf mehreren Festplatten parallel stattfinden. Aufgrund der Beschaffenheit von RAID3, muss die Anzahl der Laufwerke 3, 5, 9, 17 bzw. 2^n + 1 sein.

Dieser Abschnitt beschreibt, wie ein Software RAID3 auf einem FreeBSD-System erstellt wird.



Obwohl es theoretisch möglich ist FreeBSD von einem RAID3-Array zu booten, wird von solch einer ungewöhnlichen Konfiguration dringend abgeraten.

### 36.4.1. Ein dediziertes RAID3-Array erstellen

In FreeBSD wird die Unterstützung für RAID3 über die GEOM-Klasse graid3(8) implementiert. Zum Erstellen eines dedizierten RAID3-Arrays sind folgende Schritte erforderlich.

1. Laden Sie zunächst das Modul geom_raid3.ko mit einem der folgenden Befehle:

   ```
   # graid3 load
   ```

   oder:

   ```
   # kldload geom_raid3
   ```

2. Stellen Sie sicher, dass ein geeigneter Mountpunkt existiert. Dieser Befehl erstellt ein neues Verzeichnis, welches als Mountpunkt verwendet werden kann:

   ```
   # mkdir /multimedia
   ```

3. Bestimmen Sie die Gerätenamen der Festplatten, die dem Array hinzugefügt werden und erstellen Sie ein neues RAID3 Gerät. Das letzte aufgeführte Gerät wird als dediziertes Paritätslaufwerk verwendet. Dieses Beispiel verwendet drei unpartionierte ATA-Platten: ada1 und ada2 für die Daten, sowie ada3 für die Parität.

   ```
   # graid3 label -v gr0 /dev/ada1 /dev/ada2 /dev/ada3
   Metadata value stored on /dev/ada1.
   Metadata value stored on /dev/ada2.
   Metadata value stored on /dev/ada3.
   Done.
   ```

4. Partitionieren Sie das neu erstelle Gerät gr0 und erstellen Sie darauf ein UFS-Dateisystem:

   ```
   # gpart create -s GPT /dev/raid3/gr0
   # gpart add -t freebsd-ufs /dev/raid3/gr0
   # newfs -j /dev/raid3/gr0p1
   ```

Viele Zahlen rauschen nun über den Bildschirm und nach einer gewissen Zeit ist der Vorgang abgeschlossen. Das Volume wurde erstellt und kann jetzt in den Verzeichnisbaum eingehangen werden:

```
# mount /dev/raid3/gr0p1 /multimedia/
```

Das RAID3-Array ist nun einsatzbereit.

Weitere Konfigurationsschritte sind erforderlich, um die Einstellungen nach einem Systemneustart zu erhalten.

1. Das Modul geom_raid3.ko muss geladen werden, bevor das Array eingehangen werden kann. Damit das Kernelmodul automatisch beim Systemstart geladen wird, muss die folgende Zeile in /boot/loader.conf hinzugefügt werden:

```
geom_raid3_load="YES"
```

2. Die folgenden Informationen über das Volume müssen in /etc/fstab hinzugefügt werden, um das Dateisystem des Arrays automatisch beim Systemstart zu aktivieren:

```
/dev/raid3/gr0p1    /multimedia ufs rw  2   2
```

## 36.5. Software RAID

Einige Motherboards und Erweiterungskarten besitzen ein ROM, das dem Rechner erlaubt von einem RAID-Array zu booten. Nach dem Booten wird der Zugriff auf das RAID-Array durch die Software auf dem Prozessor des Rechners abgewickelt. Dieses "Hardware-unterstützte Software-RAID" ist nicht abhängig von einem bestimmten Betriebssystem. Sie funktionieren bereits, noch bevor das Betriebssystem geladen wird.

Abhängig von der verwendeten Hardware werden mehrere Arten von RAID unterstützt. Eine vollständige Liste finden Sie in graid(8).

graid(8) benötigt das geom_raid.ko Kernelmodul, welches beginnend mit FreeBSD 9.1 im GENERIC -Kernel enthalten ist. Bei Bedarf kann es manuell mit `graid load` geladen werden.

### 36.5.1. Ein Array erstellen

Geräte mit Software-RAID haben oft ein Menü, das über eine bestimmte Tastenkombination beim Booten aufgerufen werden kann. Das Menü kann verwendet werden, um RAID-Arrays zu erstellen und zu löschen. Mit graid(8) können Arrays auch direkt von der Kommandozeile erstellt werden.

`graid label` wird verwendet, um ein neues Array zu erstellen. Das Motherboard in diesem Beispiel besitzt einen Intel® Software-RAID Chipsatz, so dass das Metadatenformat von Intel® angegeben wird. Das neue Array bekommt den Namen (Label) gm0, verhält sich als Spiegel (RAID1) und verwendet die Laufwerke ada0 und ada1.



Bei der Erstellung des Arrays wird etwas Platz auf den Laufwerken überschrieben. Sichern Sie zuvor alle vorhandenen Daten!

```
# graid label Intel gm0 RAID1 ada0 ada1
GEOM_RAID: Intel-a29ea104: Array Intel-a29ea104 created.
GEOM_RAID: Intel-a29ea104: Disk ada0 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:0-ada0 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Array started.
GEOM_RAID: Intel-a29ea104: Volume gm0 state changed from STARTING to OPTIMAL.
Intel-a29ea104 created$
GEOM_RAID: Intel-a29ea104: Provider raid/r0 for volume gm0 created.
```

Eine Statusabfrage zeigt, dass der neue Spiegel einsatzbereit ist:

```
# graid status
   Name   Status  Components
raid/r0  OPTIMAL  ada0 (ACTIVE (ACTIVE))
                  ada1 (ACTIVE (ACTIVE))
```

Das Array-Gerät erscheint in /dev/raid/. Das erste Gerät heißt r0. Falls weitere Geräte vorhanden sind heißen diese r1, r2 und so weiter.

Das BIOS-Menü einiger Geräte erstellt Arrays mit Sonderzeichen im Namen. Um Probleme mit diesen Sonderzeichen zu vermeiden, werden einfache numerische Namen wie r0 vergeben. Um das tatsächliche Label anzuzeigen, wie gm0 im obigen Beispiel, benutzen Sie sysctl(8):

```
# sysctl kern.geom.raid.name_format=1
```

### 36.5.2. Mehrere Volumes

Einige Software-RAID Geräte unterstützen mehr als ein *Volume* pro Array. Volumes funktionieren wie Festplatten, dass heißt der Platz auf den Laufwerken kann auf unterschiedliche Weise geteilt und genutzt werden. Intels Software-RAID Geräte unterstützen beispielsweise zwei Volumes. In diesem Beispiel wird ein 40 GB Spiegel verwendet um das Betriebssystem zu speichern, gefolgt von einem 20 GB RAID0 (Stripe) Volume für die schnelle Speicherung von temporären Daten.

```
# graid label -S 40G Intel gm0 RAID1 ada0 ada1
# graid add -S 20G gm0 RAID0
```

Volumes erscheinen unter /dev/raid/ als zusätzliche Einträge rX. Ein Array mit Volumes wird als r0 und r1.

Lesen Sie graid(8) um die Anzahl der Volumes zu ermitteln, die von den verschiedenen Software-

RAID Geräten unterstützt wird.

### 36.5.3. Ein einzelnes Laufwerk zu einem Spiegel konvertieren

Unter bestimmten Umständen ist es möglich, ein bestehendes Laufwerk ohne Neuformatierung zu einem graid(8) Array zu konvertieren. Um Datenverlust bei der Konvertierung zu vermeiden, müssen die vorhandenen Laufwerke folgende Mindestanforderungen erfüllen:

- Das Laufwerk muss mit MBR partitioniert werden. GPT oder andere Partitionierungsschemata funktionieren nicht, da durch graid(8) die Metadaten am Ende des Laufwerks überschieben und beschädigt werden.
- Am Ende des Laufwerks muss genügend freier Platz zur Verfügung stehen, um die graid(8) Metadaten zu speichern. Die Metadaten variieren in der Größe, es werden jedoch mindestens 64 MB freier Speicherplatz empfohlen.

Wenn das Laufwerk diese Anforderungen erfüllt, erstellen Sie zuerst eine vollständige Sicherung. Erzeugen Sie dann einen Spiegel mit diesem einen Laufwerk:

```
# graid label Intel gm0 RAID1 ada0 NONE
```

Die Metadaten von graid(8) werden in den ungenutzten Raum am Ende des Laufwerks geschrieben. Ein zweites Laufwerk kann nun in den Spiegel eingefügt werden:

```
# graid insert raid/r0 ada1
```

Die Daten von dem ersten Laufwerk werden direkt auf das zweite Laufwerk kopiert. Der Spiegel wird im eingeschränkten Zustand laufen, bis der Kopiervorgang abgeschlossen ist.

### 36.5.4. Neue Laufwerke zum Array hinzufügen

Laufwerke in einem Array können für ausgefallene oder fehlende Laufwerke eingesetzt werden. Falls es keine ausgefallenen oder fehlenden Laufwerke gibt, wird das neue Laufwerk als Ersatz (Spare) verwendet.

Das Array in diesem Beispiel beginnt sofort damit, die Daten auf das neu hinzugefügte Laufwerk zu kopieren. Alle vorhandenen Daten auf dem neuen Laufwerk werden überschrieben.

```
# graid insert raid/r0 ada1
GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from NONE to ACTIVE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NONE to NEW.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 state changed from NEW to REBUILD.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-ada1 rebuild start at 0.
```

### 36.5.5. Laufwerke aus dem Array entfernen

Einzelne Laufwerke können permanent aus dem Array entfernt werden. Die Metadaten werden

dabei gelöscht:

```
# graid remove raid/r0 ada1
GEOM_RAID: Intel-a29ea104: Disk ada1 state changed from ACTIVE to OFFLINE.
GEOM_RAID: Intel-a29ea104: Subdisk gm0:1-[unknown] state changed from ACTIVE to NONE.
GEOM_RAID: Intel-a29ea104: Volume gm0 state changed from OPTIMAL to DEGRADED.
```

### 36.5.6. Das Array anhalten

Ein Array kann angehalten werden, ohne die Metadaten von den Laufwerken zu löschen. Das Array wird wieder anlaufen, wenn das System neu gestartet wird.

```
# graid stop raid/r0
```

### 36.5.7. Den Status des Arrays überprüfen

Der Status des Arrays kann jederzeit überprüft werden. Nachdem ein Laufwerk zum Array hinzugefügt wurde, werden die Daten vom ursprünglichen Laufwerk auf das neue Laufwerk kopiert:

```
# graid status
   Name    Status  Components
raid/r0  DEGRADED  ada0 (ACTIVE (ACTIVE))
                   ada1 (ACTIVE (REBUILD 28%))
```

Andere Arten von Arrays, wie `RAID0` oder `CONCAT`, werden den Status eines fehlgeschlagenen Laufwerks vielleicht nicht anzeigen. Um diese teilweise ausgefallenen Arrays anzuzeigen, fügen Sie `-ga` hinzu:

```
# graid status -ga
          Name  Status  Components
Intel-e2d07d9a  BROKEN  ada6 (ACTIVE (ACTIVE))
```

### 36.5.8. Arrays löschen

Arrays werden zerstört, indem alle Volumes gelöscht werden. Wenn das letzte Volume gelöscht wird, wird das Array gestoppt und die Metadaten von den Laufwerken entfernt:

```
# graid delete raid/r0
```

### 36.5.9. Unerwartete Arrays löschen

Laufwerke können unerwartete graid(8) Metadaten enthalten, entweder aus früherer Nutzung

oder aus Tests des Herstellers. graid(8) würde diese Metadaten erkennen und daraus ein Array erstellen, was den Zugriff auf die einzelnen Laufwerke beeinträchtigen würde. Um die unerwünschten Metadaten zu entfernen:

1. Booten Sie das System. Im Boot-Menü wählen Sie `2` für den Loader-Prompt. Geben Sie dann folgendes ein:

   ```
   OK set kern.geom.raid.enable=0
   OK boot
   ```

   Das System wird nun mit deaktiviertem graid(8) starten.

2. Sichern Sie alle Daten auf dem betroffenen Laufwerk.
3. Zur Abhilfe kann auch die Array-Erkennung von graid(8) deaktiviert werden, indem

   ```
   kern.geom.raid.enable=0
   ```

   in /boot/loader.conf hinzugefügt wird.

   Um die graid(8) Metadaten von dem entsprechenden Laufwerk zu entfernen, booten Sie vom FreeBSD Installationsmedium und wählen Sie `Shell` aus. Benutzen Sie `status`, um den Namen des Arrays zu bestimmten, typischerweise `raid/r0`:

   ```
   # graid status
      Name   Status  Components
   raid/r0  OPTIMAL  ada0 (ACTIVE (ACTIVE))
                     ada1 (ACTIVE (ACTIVE))
   ```

   Löschen Sie das Volume:

   ```
   # graid delete raid/r0
   ```

   Wiederholen Sie den Vorgang für jedes Volume. Nachdem das letzte Volume gelöscht wurde, wird das Volume zerstört.

   Starten Sie das System neu und prüfen die Vollständigkeit der Daten. Falls erforderlich, müssen die Daten aus der Sicherung wiederhergestellt werden. Nachdem die Metadaten entfernt wurden, kann auch der Eintrag `kern.geom.raid.enable=0` aus /boot/loader.conf entfernt werden.

## 36.6. GEOM Gate Netzwerk

GEOM unterstützt einen einfachen Mechanismus für den Zugriff auf entfernte Geräte wie Festplatten, CDs und Dateien, durch die Verwendung des GEOM Gate Netzwerk Daemons, ggated. Der Server-Dameon läuft auf dem System, welches ein Gerät anbietet und bearbeitet die ggatec-Anfragen der Clients. Die Geräte sollten keine sensiblen Daten enthalten, da die Verbindung

zwischen Client und Server nicht verschlüsselt ist.

Ähnlich wie bei NFS, das in Network File System (NFS) beschrieben ist, wird für die Konfiguration von ggated eine Exportdatei verwendet. Diese Datei legt fest, welche Systeme auf die exportierten Ressourcen zugreifen können und in welchem Umfang der Zugriff gestattet wird. Um dem Client `192.168.1.5` Lese- und Schreibzugriff auf die vierte Slice der ersten SCSI-Platte zu geben, erstellen Sie /etc/gg.exports mit folgender Zeile:

```
192.168.1.5 RW /dev/da0s4d
```

Bevor das Gerät exportiert werden kann, müssen Sie sicherstellen, dass es nicht bereits gemountet ist. Anschließend starten Sie ggated.

```
# ggated
```

Es stehen mehrere Optionen bereit, mit denen zum Beispiel ein alternativer Port oder eine alternative Exportdatei festgelegt werden kann. Weitere Einzelheiten finden Sie in ggated(8).

Damit ein Client auf das exportierte Gerät zugreifen kann, benutzten Sie ggatec zusammen mit der IP-Adresse des Servers und dem entsprechenden Gerätenamen. Wenn dies erfolgreich ist, zeigt dieser Befehl einen `ggate`-Gerätenamen. Hängen Sie dieses Gerät in einen freien Mountpunkt ein. Dieses Beispiel verbindet sich mit der Partition /dev/da0s4d auf `192.168.1.1` und hängt /dev/ggate0 in /mnt ein:

```
# ggatec create -o rw 192.168.1.1 /dev/da0s4d
ggate0
# mount /dev/ggate0 /mnt
```

Auf das Gerät des Servers kann jetzt über den Mountpunkt /mnt des Clients zugegriffen werden. Weitere Informationen über `ggatec` und einige Anwendungsbeispiele finden Sie in ggatec(8).



Das Einhängen des Gerätes wird scheitern, falls das Gerät momentan entweder auf dem Server oder einem Client im Netzwerk gemountet ist. Wenn ein gleichzeitiger Zugriff auf die Netzwerkressourcen benötigt wird, verwenden Sie stattdessen NFS.

Wenn das Gerät nicht länger gebraucht wird, kann es mit umount(8) ausgehängt werden, so dass die Ressourcen für andere Client wieder verfügbar sind.

## 36.7. Das Labeln von Laufwerken

Während der Initialisierung des Systems legt der FreeBSD-Kernel für jedes gefundene Gerät Knotenpunkte an. Diese Methode für die Überprüfung auf vorhandene Geräte wirft einige Fragen auf. Was passiert beispielsweise, wenn ein neues USB-Laufwerk hinzugefügt wird? Es ist sehr wahrscheinlich, dass ein Flash-Speicher-Gerät den Gerätenamen da0 erhält, während gleichzeitig das bisherige da0 zu da1 wird. Dies verursacht Probleme beim Einhängen von Dateisystemen,

wenn diese in /etc/fstab aufgeführt sind und kann dazu führen, dass das System nicht mehr startet.

Eine Lösung für dieses Problem ist das Aneinanderketten der SCSI-Geräte, damit ein neues Gerät, welches der SCSI-Karte hinzugefügt wird, unbenutzte Gerätenummern erhält. Aber was geschieht, wenn ein USB-Gerät möglicherweise die primäre SCSI-Platte ersetzt? Dies kann passieren, weil USB-Geräte normalerweise vor der SCSI-Karte geprüft werden. Eine Lösung ist das Hinzufügen dieser Geräte, nachdem das System gestartet ist. Eine andere Lösung könnte sein, nur ein einzelnes ATA-Laufwerk zu nutzen und die SCSI-Geräte niemals in der /etc/fstab aufzuführen.

Eine bessere Lösung ist die Verwendung von `glabel`, um die Laufwerke zu mit Labeln zu versehen und diese in /etc/fstab zu nutzen. Da `glabel` seine Label im letzten Sektor jedes vorhandenen Datenträgers speichert, wird das Label persistent bleiben (auch über Neustarts hinweg). Durch Nutzung dieses Labels als Gerät kann das Dateisystem immer gemountet sein, unabhängig davon, durch welchen Geräte-Knotenpunkt auf ihn zugegriffen wird.



`glabel` kann permanente (dauerhaft) und vorübergehende Label erstellen. Aber nur dauerhafte Label bleiben konsistent über Neustarts hinweg. Lesen Sie die glabel(8) für weitere Unterschiede zwischen den Label-Typen.

### 36.7.1. Label-Typen und Beispiele

Permanente Label können generische Label oder Dateisystem-Label sein. Permanente Dateisystem-Label können mit tunefs(8) oder newfs(8) erzeugt werden. Dieser Typ von Label wird in einem Unterverzeichnis von /dev angelegt und wird dem Dateisystem entsprechend benannt. UFS2-Dateisystem-Label werden zum Beispiel in /dev/ufs angelegt. Permanente Label können außerdem durch den Befehl `glabel label` erzeugt werden. Diese Label sind nicht dateisystemspezisch und werden im Unterverzeichnis /dev/label erzeugt.

Temporäre Label werden beim nächsten Systemstart zerstört. Diese Label werden im Verzeichnis /dev/label erzeugt und sind ideal für Testzwecke. Ein temporäres Label kann mit `glabel create` erzeugt werden.

Um ein permanentes Label auf einem UFS2-Dateisystem ohne Löschung von Daten zu erzeugen, kann man folgenden Befehl verwenden:

```
# tunefs -L home /dev/da3
```

In /dev/ufs sollte nun ein Label vorhanden sein, welches zu /etc/fstab hinzugefügt werden kann:

```
/dev/ufs/home       /home           ufs     rw              2       2
```



Das Dateisystem darf nicht gemountet sein beim Versuch, `tunefs` auszuführen.

Nun kann das Dateisystem eingehängt werden:

```
# mount /home
```

Von nun an kann der Geräte-Knotenpunkt sich ohne negative Effekte auf das System ändern, solange das Kernelmodul geom_label.ko beim Systemstart mittels /boot/loader.conf geladen wird oder die GEOM_LABEL-Kernel-Option aktiv ist.

Dateisysteme können auch mit einem Standard-Label erzeugt werden (mittels des Flags -L in newfs). Lesen Sie newfs(8) für weitere Informationen.

Der folgende Befehl kann genutzt werden, um das Label zu beseitigen:

```
# glabel destroy home
```

Das folgende Beispiel zeigt Ihnen, wie Sie Label für die Partitionen einer Bootplatte erzeugen.

*Beispiel 42. Die Partitionen einer Bootplatte labeln*

Durch das Erstellen von permanenten Labeln für die Partitionen einer Bootplatte sollte das System selbst dann noch normal starten können, wenn Sie die Platte an einen anderen Controller anschließen oder in ein anderes System installieren. In diesem Beispiel nehmen wir an, dass nur eine einzige ATA-Platte verwendet wird, die das System derzeit als ad0 erkennt. Weiters nehmen wir an, dass Sie das Standard-Partionierungsschema von FreeBSD vewendet haben und die Platte daher die Dateisysteme /, /var, /usr sowie /tmp aufweist. Zusätzlich wurde eine Swap-Partition angelegt.

Starten Sie das System neu. Am loader(8)-Prompt drücken Sie die Taste `4`, um in den Single-User-Modus zu gelangen. Dort führen Sie die folgenden Befehle aus:

```
# glabel label rootfs /dev/ad0s1a
GEOM_LABEL: Label for provider /dev/ad0s1a is label/rootfs
# glabel label var /dev/ad0s1d
GEOM_LABEL: Label for provider /dev/ad0s1d is label/var
# glabel label usr /dev/ad0s1f
GEOM_LABEL: Label for provider /dev/ad0s1f is label/usr
# glabel label tmp /dev/ad0s1e
GEOM_LABEL: Label for provider /dev/ad0s1e is label/tmp
# glabel label swap /dev/ad0s1b
GEOM_LABEL: Label for provider /dev/ad0s1b is label/swap
# exit
```

Das System startet daraufhin in den Multi-User-Modus. Nachdem der Startvorgang abgeschlossen ist, editieren Sie /etc/fstab und ersetzen die konventionellen Gerätedateien durch die entsprechenden Label. Die modifizierte /etc/fstab sollte wie folgt aussehen:

```
# Device                Mountpoint      FStype  Options         Dump    Pass#
/dev/label/swap         none            swap    sw              0       0
```

```
/dev/label/rootfs       /               ufs     rw              1       1
/dev/label/tmp          /tmp            ufs     rw              2       2
/dev/label/usr          /usr            ufs     rw              2       2
/dev/label/var          /var            ufs     rw              2       2
```

Starten Sie das System neu. Treten keine Probleme auf, wird das System normal hochfahren und Sie erhalten die folgende Ausgabe, wenn Sie den Befehl `mount` ausführen:

```
# mount
/dev/label/rootfs on / (ufs, local)
devfs on /dev (devfs, local)
/dev/label/tmp on /tmp (ufs, local, soft-updates)
/dev/label/usr on /usr (ufs, local, soft-updates)
/dev/label/var on /var (ufs, local, soft-updates)
```

glabel(8) unterstützt einen Labeltyp für UFS-Dateisysteme. Dieser basiert auf der eindeutigen Dateisystem-ID `ufsid`. Derartige Label finden sich in /dev/ufsid und werden während des Systemstarts automatisch erzeugt. Es ist möglich, diese `ufsid`-Label zum automatischen Einhängen von Partitionen in /etc/fstab einzusetzen. Verwenden Sie `glabel status`, um eine Liste aller Dateisysteme und ihrer `ufsid`-Label zu erhalten:

```
% glabel status
                  Name  Status  Components
ufsid/486b6fc38d330916     N/A  ad4s1d
ufsid/486b6fc16926168e     N/A  ad4s1f
```

In diesem Beispiel repräsentiert ad4s1d das /var-Dateisystem, während ad4s1f dem /usr -Dateisystem entspricht. Wenn Sie die angegebenen `ufsid`-Werte verwenden, können diese Dateisysteme durch die folgenden Einträge in der Datei /etc/fstab gemountet werden:

```
/dev/ufsid/486b6fc38d330916        /var        ufs        rw        2      2
/dev/ufsid/486b6fc16926168e        /usr        ufs        rw        2      2
```

Jede Partition, die ein `ufsid`-Label aufweist, kann auf diese Art gemountet werden. Dies hat den Vorteil, dass Sie die permanenten Label nicht manuell anlegen müssen, wobei sich die Platten nach wie vor über geräteunabhängige Namen ansprechen und einhängen lassen.

## 36.8. UFS Journaling in GEOM

FreeBSD unterstützt Journaling für UFS-Dateisysteme. Diese Funktion wird über das GEOM-Subsystem realisiert und kann über das Werkzeug gjournal(8) eingerichtet werden. Im Gegensatz zu anderen Journaling-Dateisystemen arbeitet `gjournal` blockbasiert und wurde nicht als Teil des Dateisystems implementiert, sondern als GEOM-Erweiterung.

Bei Journaling wird ein Protokoll über alle Dateisystemtransaktionen angelegt, inklusive aller

Veränderungen, aus denen ein kompletter Schreibvorgang besteht, bevor diese Änderungen (Metadaten sowie tatsächliche Schreibvorgänge) physisch auf der Festplatte ausgeführt werden. Dieses Protokoll kann später erneut aufgerufen werden, um diese Vorgänge zu wiederholen, damit Systeminkonsistenzen vermieden werden.

Diese Technik bietet eine weitere Möglichkeit, sich vor Datenverlust und Dateisystem-Inkonsistenzen zu schützen. Im Gegensatz zu Soft Updates (die Metadaten-Aktualisierungen verfolgen und erzwingen) und Snapshots (die ein Image eines Dateisystems darstellen) wird bei Journaling ein tatsächliches Protokoll in einem speziell dafür bereitgestellten Bereich der Festplatte gespeichert. Um die Leistung zu optimieren, kann das Journal auf eine externe Platte ausgelagert werden. In einem solchen Fall geben Sie die Gerätedatei der Platte nach dem Gerät an, für das Sie Journaling aktivieren wollen.

Der GENERIC-Kernel bietet Unterstützung für `gjournal`. Damit das Kernelmodul geom_journal.ko beim Booten automatisch geladen wird, fügen Sie folgende Zeile in /boot/loader.conf hinzu:

```
geom_journal_load="YES"
```

Wenn ein angepasster Kernel benutzt wird, stellen Sie sicher, dass folgende Zeile in der Kernelkonfigurationsdatei enthalten ist:

```
options     GEOM_JOURNAL
```

Sobald das Modul geladen ist, kann ein Journal auf einem neuen Dateisystem erstellt werden. In diesem Beispiel ist da4 die neue SCSI-Platte:

```
# gjournal load
# gjournal label /dev/da4
```

Diese Befehle laden das Modul und erstellen die Gerätedatei /dev/da4.journal auf /dev/da4.

Nun kann auf dem neuen Gerät ein UFS-Dateisystem erstellt werden, welches dann in den Verzeichnisbaum eingehängt wird:

```
# newfs -O 2 -J /dev/da4.journal
# mount /dev/da4.journal /mnt
```



Falls auf dem System mehrere Slices angelegt sind (beispielsweise ad4s1 sowie ad4s2), wird `gjournal` für jedes Slice ein Journal anlegen (also ad4s1.journal sowie ad4s2.journal).

Mit `tunefs` ist es auch möglich, Journaling auf bereits existierenden Dateisystemen zu aktivieren. Machen Sie aber *immer* eine Sicherung der Daten, bevor Sie versuchen, ein existierendes Dateisystem zu ändern. `gjournal` wird zwar den Vorgang abbrechen, wenn es das Journal nicht erzeugen kann, allerdings schützt dies nicht vor Datenverlust durch einen fehlerhaften Einsatz von

`tunefs`. Weitere Informationen über diese beiden Werkzeuge finden Sie in gjournal(8) und tunefs(8).

Es ist möglich, Journale auch für die Bootplatte eines FreeBSD-Systems zu verwenden. Der Artikel Implementing UFS Journaling on a Desktop PC enthält eine ausführliche Anleitung zu diesem Thema.

# Kapitel 37. Das Z-Dateisystem (ZFS)

Das *Z-Dateisystem*, oder kurz ZFS, ist ein fortgeschrittenes Dateisystem, das entwickelt wurde, um viele der großen Probleme in vorherigen Entwicklungen zu überwinden.

Ursprünglich von Sun™ entworfen, wird die weitere Entwicklung von ZFS heutzutage als Open Source vom OpenZFS Projekt vorangetrieben.

ZFS hat drei große Entwurfsziele:

- Datenintegrität: Alle Daten enthalten eine Prüfsumme (checksum) der Daten. Wenn Daten geschrieben werden, wird die Prüfsumme berechnet und zusammen mit den Daten gespeichert. Wenn diese Daten später wieder eingelesen werden, wird diese Prüfsumme erneut berechnet. Falls die Prüfsummen nicht übereinstimmen, wurde ein Datenfehler festgestellt. ZFS wird versuchen, diesen Fehler automatisch zu korrigieren, falls genug Datenredundanz vorhanden ist.
- Gepoolter Speicher: physikalische Speichermedien werden zu einem Pool zusammengefasst und der Speicherplatz wird von diesem gemeinsam genutzten Pool allokiert. Der Speicherplatz steht allen Dateisystemen zur Verfügung und kann durch das Hinzufügen von neuen Speichermedien vergrößert werden.
- Geschwindigkeit: mehrere Zwischenspeichermechanismen sorgen für erhöhte Geschwindigkeit. Der ARC ist ein weiterentwickelter, hauptspeicherbasierter Zwischenspeicher für Leseanfragen. Auf einer zweiten Stufe kann ein plattenbasierter L2ARC-Lesezwischenspeicher hinzugefügt werden. Zusätzlich ist auch noch ein plattenbasierter, synchroner Schreibzwischenspeicher verfügbar, der sog. ZIL.

Eine vollständige Liste aller Eigenschaften und der dazugehörigen Terminologie ist in ZFS-Eigenschaften und Terminologie zu sehen.

## 37.1. Was ZFS anders macht

ZFS ist signifikant unterschiedlich zu allen bisherigen Dateisystemen, weil es mehr als nur ein Dateisystem ist. Durch die Kombination von traditionell getrennten Rollen von Volumenmanager und Dateisystem ist ZFS mit einzigartigen Vorteilen ausgestattet. Das Dateisystem besitzt jetzt Kenntnis von der zugrundeliegenden Struktur der Speichermedien. Traditionelle Dateisysteme konnten nur auf einer einzigen Platte gleichzeitig angelegt werden. Falls es zwei Festplatten gab, mussten auch zwei getrennte Dateisysteme erstellt werden. In einer traditionellen Hardware-RAID-Konfiguration wurde dieses Problem umgangen, indem dem Betriebssystem nur eine einzige logische Platte angezeigt wurde, die sich aus dem Speicherplatz von der Anzahl an physischen Platten zusammensetzte, auf dem dann das Betriebssystem ein Dateisystem erstellte. Sogar im Fall von Software-RAID-Lösungen, wie die, die von GEOM bereitgestellt werden, war das UFS-Dateisystem der Ansicht, dass es auf nur einem einzigen Gerät angelegt wurde. ZFS's Kombination eines Volumenmanagers und eines Dateisystems löst dies und erlaubt das Erstellen von vielen Dateisystemen, die sich alle den darunterliegenden Pool aus verfügbarem Speicher teilen. Einer der größten Vorteile von ZFS's Kenntnis des physikalischen Layouts der Platten ist, dass existierende Dateisysteme automatisch wachsen können, wenn zusätzliche Platten zum Pool hinzugefügt werden. Dieser neue Speicherplatz wird dann allen Dateisystemen zur Verfügung gestellt. ZFS

besitzt ebenfalls eine Menge an unterschiedlichen Eigenschaften, die für jedes Dateisystem angepasst werden können, was viele Vorteile bringt, wenn man unterschiedliche Dateisysteme und Datasets anlegt, anstatt ein einziges, monolithisches Dateisystem zu erzeugen.

## 37.2. Schnellstartanleitung

Es existiert ein Startmechanismus, der es FreeBSD erlaubt, ZFS-Pools während der Systeminitialisierung einzubinden. Um diesen zu aktivieren, fügen Sie diese Zeile in /etc/rc.conf ein:

```
zfs_enable="YES"
```

Starten Sie dann den Dienst:

```
# service zfs start
```

Die Beispiele in diesem Abschnitt gehen von drei SCSI-Platten mit den Gerätenamen da0, da1 und da2 aus. Nutzer von SATA-Hardware sollten stattdessen die Bezeichnung ada als Gerätenamen verwenden.

### 37.2.1. Pools mit einer Platte

Um einen einfachen, nicht-redundanten Pool mit einem einzigen Gerät anzulegen, geben Sie folgendes ein:

```
# zpool create example /dev/da0
```

Um den neuen Pool anzuzeigen, prüfen Sie die Ausgabe von `df`:

```
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235230  1628718    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032846 48737598     2%    /usr
example      17547136       0 17547136     0%    /example
```

Diese Ausgabe zeigt, dass der `example`-Pool erstellt und eingehängt wurde. Er ist nun als Dateisystem verfügbar. Dateien können darauf angelegt werden und Anwender können sich den Inhalt ansehen:

```
# cd /example
# ls
# touch testfile
# ls -al
total 4
```

```
drwxr-xr-x   2 root  wheel    3 Aug 29 23:15 .
drwxr-xr-x  21 root  wheel  512 Aug 29 23:12 ..
-rw-r--r--   1 root  wheel    0 Aug 29 23:15 testfile
```

Allerdings nutzt dieser Pool noch keine der Vorteile von ZFS. Um ein Dataset auf diesem Pool mit aktivierter Komprimierung zu erzeugen, geben Sie ein:

```
# zfs create example/compressed
# zfs set compression=gzip example/compressed
```

Das `example/compressed`-Dataset ist nun ein komprimiertes ZFS-Dateisystem. Versuchen Sie, ein paar große Dateien auf /example/compressed zu kopieren.

Deaktivieren lässt sich die Komprimierung durch:

```
# zfs set compression=off example/compressed
```

Um ein Dateisystem abzuhängen, verwenden Sie `zfs umount` und überprüfen Sie dies anschließend mit `df`:

```
# zfs umount example/compressed
# df
Filesystem  1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a   2026030  235232  1628716    13%    /
devfs               1       1        0   100%    /dev
/dev/ad0s1d  54098308 1032864 48737580     2%    /usr
example      17547008       0 17547008     0%    /example
```

Um das Dateisystem wieder einzubinden und erneut verfügbar zu machen, verwenden Sie `zfs mount` und prüfen Sie erneut mit `df`:

```
# zfs mount example/compressed
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
```

Den Pool und die Dateisysteme können Sie auch über die Ausgabe von `mount` prüfen:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
```

```
/dev/ad0s1d on /usr (ufs, local, soft-updates)
example on /example (zfs, local)
example/compressed on /example/compressed (zfs, local)
```

Nach der Erstellung können ZFS-Datasets wie jedes andere Dateisystem verwendet werden. Jedoch sind jede Menge andere Besonderheiten verfügbar, die individuell auf Dataset-Basis eingestellt sein können. Im Beispiel unten wird ein neues Dateisystem namens `data` angelegt. Wichtige Dateien werden dort abgespeichert, deshalb wird es so konfiguriert, dass zwei Kopien jedes Datenblocks vorgehalten werden.

```
# zfs create example/data
# zfs set copies=2 example/data
```

Es ist jetzt möglich, den Speicherplatzverbrauch der Daten durch die Eingabe von `df` zu sehen:

```
# df
Filesystem         1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a          2026030  235234  1628714    13%    /
devfs                      1       1        0   100%    /dev
/dev/ad0s1d         54098308 1032864 48737580     2%    /usr
example             17547008       0 17547008     0%    /example
example/compressed  17547008       0 17547008     0%    /example/compressed
example/data        17547008       0 17547008     0%    /example/data
```

Sie haben vermutlich bemerkt, dass jedes Dateisystem auf dem Pool die gleiche Menge an verfügbarem Speicherplatz besitzt. Das ist der Grund dafür, dass in diesen Beispielen `df` verwendet wird, um zu zeigen, dass die Dateisysteme nur die Menge an Speicher verbrauchen, den sie benötigen und alle den gleichen Pool verwenden. ZFS eliminiert Konzepte wie Volumen und Partitionen und erlaubt es mehreren Dateisystemen den gleichen Pool zu belegen.

Um das Dateisystem und anschließend den Pool zu zerstören, wenn dieser nicht mehr benötigt wird, geben Sie ein:

```
# zfs destroy example/compressed
# zfs destroy example/data
# zpool destroy example
```

### 37.2.2. RAID-Z

Platten fallen aus. Eine Methode, um Datenverlust durch Festplattenausfall zu vermeiden, ist die Verwendung von RAID. ZFS unterstützt dies in seiner Poolgestaltung. Pools mit RAID-Z benötigen drei oder mehr Platten, bieten aber auch mehr nutzbaren Speicher als gespiegelte Pools.

Dieses Beispiel erstellt einen RAID-Z-Pool, indem es die Platten angibt, die dem Pool hinzugefügt werden sollen:

```
# zpool create storage raidz da0 da1 da2
```



Sun™ empfiehlt, dass die Anzahl der Geräte in einer RAID-Z Konfiguration zwischen drei und neun beträgt. Für Umgebungen, die einen einzelnen Pool benötigen, der aus 10 oder mehr Platten besteht, sollten Sie in Erwägung ziehen, diesen in kleinere RAID-Z-Gruppen aufzuteilen. Falls nur zwei Platten verfügbar sind und Redundanz benötigt wird, ziehen Sie die Verwendung eines ZFS-Spiegels (mirror) in Betracht. Lesen Sie dazu zpool(8), um weitere Details zu erhalten.

Das vorherige Beispiel erstellte einen ZPool namens `storage`. Dieses Beispiel erzeugt ein neues Dateisystem, genannt `home`, in diesem Pool:

```
# zfs create storage/home
```

Komprimierung und das Vorhalten von mehreren Kopien von Dateien und Verzeichnissen kann aktiviert werden:

```
# zfs set copies=2 storage/home
# zfs set compression=gzip storage/home
```

Um dies als das neue Heimatverzeichnis für Anwender zu setzen, kopieren Sie die Benutzerdaten in dieses Verzeichnis und erstellen passende symbolische Verknüpfungen:

```
# cp -rp /home/* /storage/home
# rm -rf /home /usr/home
# ln -s /storage/home /home
# ln -s /storage/home /usr/home
```

Daten von Anwendern werden nun auf dem frisch erstellten /storage/home abgelegt. Überprüfen Sie dies durch das Anlegen eines neuen Benutzers und das anschließende Anmelden als dieser Benutzer.

Versuchen Sie, einen Dateisystemschnappschuss anzulegen, den Sie später wieder zurückrollen können:

```
# zfs snapshot storage/home@08-30-08
```

Schnappschüsse können nur auf einem Dateisystem angelegt werden, nicht auf einem einzelnen Verzeichnis oder einer Datei.

Das Zeichen `@` ist der Trenner zwischen dem Dateisystem- oder dem Volumennamen. Wenn ein wichtiges Verzeichnis aus Versehen gelöscht wurde, kann das Dateisystem gesichert und dann zu einem früheren Schnappschuss zurückgerollt werden, in welchem das Verzeichnis noch existiert:

```
# zfs rollback storage/home@08-30-08
```

Um all verfügbaren Schnappschüsse aufzulisten, geben Sie `ls` im Verzeichnis .zfs/snapshot dieses Dateisystems ein. Beispielsweise lässt sich der zuvor angelegte Schnappschuss wie folgt anzeigen:

```
# ls /storage/home/.zfs/snapshot
```

Es ist möglich, ein Skript zu schreiben, um regelmäßig Schnappschüsse von Benutzerdaten anzufertigen. Allerdings verbrauchen Schnappschüsse über lange Zeit eine große Menge an Speicherplatz. Der zuvor angelegte Schnappschuss kann durch folgendes Kommando wieder entfernt werden:

```
# zfs destroy storage/home@08-30-08
```

Nach erfolgreichen Tests kann /storage/home zum echten /home-Verzeichnis werden, mittels:

```
# zfs set mountpoint=/home storage/home
```

Prüfen Sie mit `df` und `mount`, um zu bestätigen, dass das System das Dateisystem nun als /home verwendet:

```
# mount
/dev/ad0s1a on / (ufs, local)
devfs on /dev (devfs, local)
/dev/ad0s1d on /usr (ufs, local, soft-updates)
storage on /storage (zfs, local)
storage/home on /home (zfs, local)
# df
Filesystem   1K-blocks    Used    Avail Capacity  Mounted on
/dev/ad0s1a    2026030  235240  1628708    13%    /
devfs                1       1        0   100%    /dev
/dev/ad0s1d   54098308 1032826 48737618     2%    /usr
storage       26320512       0 26320512     0%    /storage
storage/home  26320512       0 26320512     0%    /home
```

Damit ist die RAID-Z Konfiguration abgeschlossen. Tägliche Informationen über den Status der erstellten Dateisysteme können als Teil des nächtlichen periodic(8)-Berichts generiert werden. Fügen Sie dazu die folgende Zeile in /etc/periodic.conf ein:

```
daily_status_zfs_enable="YES"
```

### 37.2.3. RAID-Z wiederherstellen

Jedes Software-RAID besitzt eine Methode, um den Zustand (`state`) zu überprüfen. Der Status von RAID-Z Geräten wird mit diesem Befehl angezeigt:

```
# zpool status -x
```

Wenn alle Pools Online sind und alles normal ist, zeigt die Meldung folgendes an:

```
all pools are healthy
```

Wenn es ein Problem gibt, womöglich ist eine Platte im Zustand Offline, dann wird der Poolzustand ähnlich wie dieser aussehen:

```
  pool: storage
 state: DEGRADED
status: One or more devices has been taken offline by the administrator.
    Sufficient replicas exist for the pool to continue functioning in a
    degraded state.
action: Online the device using 'zpool online' or replace the device with
    'zpool replace'.
 scrub: none requested
config:

    NAME        STATE     READ WRITE CKSUM
    storage     DEGRADED     0     0     0
      raidz1    DEGRADED     0     0     0
        da0     ONLINE       0     0     0
        da1     OFFLINE      0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

Dies zeigt an, dass das Gerät zuvor vom Administrator mit diesem Befehl abgeschaltet wurde:

```
# zpool offline storage da1
```

Jetzt kann das System heruntergefahren werden, um da1 zu ersetzen. Wenn das System wieder eingeschaltet wird, kann die fehlerhafte Platte im Pool ersetzt werden:

```
# zpool replace storage da1
```

Von diesem Punkt an kann der Status erneut geprüft werden. Dieses Mal ohne die Option `-x`, damit alle Pools angezeigt werden:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: resilver completed with 0 errors on Sat Aug 30 19:44:11 2008
config:

    NAME        STATE     READ WRITE CKSUM
    storage     ONLINE       0     0     0
      raidz1    ONLINE       0     0     0
        da0     ONLINE       0     0     0
        da1     ONLINE       0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

In diesem Beispiel ist alles normal.

### 37.2.4. Daten verifizieren

ZFS verwendet Prüfsummen, um die Integrität der gespeicherten Daten zu gewährleisten. Dies wird automatisch beim Erstellen von Dateisystemen aktiviert.



Prüfsummen können deaktiviert werden, dies wird jedoch *nicht* empfohlen! Prüfsummen verbrauchen nur sehr wenig Speicherplatz und sichern die Integrität der Daten. Viele Eigenschaften vom ZFS werden nicht richtig funktionieren, wenn Prüfsummen deaktiviert sind. Es gibt keinen merklichen Geschwindigkeitsunterschied durch das Deaktivieren dieser Prüfsummen.

Prüfsummenverifikation ist unter der Bezeichnung *scrubbing* bekannt. Verifizieren Sie die Integrität der Daten des `storage`-Pools mit diesem Befehl:

```
# zpool scrub storage
```

Die Laufzeit einer Überprüfung hängt ab von der Menge an Daten, die gespeichert sind. Größere Mengen an Daten benötigen proportional mehr Zeit zum überprüfen. Diese Überprüfungen sind sehr I/O-intensiv und es kann auch nur eine Überprüfung zur gleichen Zeit durchgeführt werden. Nachdem eine Prüfung beendet ist, kann der Status mit dem Unterkommando `status` angezeigt werden:

```
# zpool status storage
 pool: storage
 state: ONLINE
 scrub: scrub completed with 0 errors on Sat Jan 26 19:57:37 2013
config:

    NAME        STATE     READ WRITE CKSUM
    storage     ONLINE       0     0     0
```

```
      raidz1    ONLINE       0     0     0
        da0     ONLINE       0     0     0
        da1     ONLINE       0     0     0
        da2     ONLINE       0     0     0

errors: No known data errors
```

Das Datum der letzten Prüfoperation wird angezeigt, um zu verfolgen, wann die nächste Prüfung benötigt wird. Routinemässige Überprüfungen helfen dabei, Daten vor stiller Korrumpierung zu schützen und die Integrität des Pools sicher zu stellen.

Lesen Sie zfs(8) und zpool(8), um über weitere ZFS-Optionen zu erfahren.

## 37.3. `zpool` Administration

Administration von ZFS ist unterteilt zwischen zwei Hauptkommandos. Das `zpool`-Werkzeug steuert die Operationen des Pools und kümmert sich um das Hinzufügen, entfernen, ersetzen und verwalten von Platten. Mit dem `zfs`-Befehl können Datasets erstellt, zerstört und verwaltet werden, sowohl Dateisysteme als auch Volumes.

### 37.3.1. Pools anlegen und zerstören

Einen ZFS-Pool (*zpool*) anzulegen beinhaltet das Treffen von einer Reihe von Entscheidungen, die relativ dauerhaft sind, weil die Struktur des Pools nachdem er angelegt wurde, nicht mehr geändert werden kann. Die wichtigste Entscheidung ist, welche Arten von vdevs als physische Platten zusammengefasst werden soll. Sehen Sie sich dazu die Liste von vdev-Arten an, um Details zu möglichen Optionen zu bekommen. Nachdem der Pool angelegt wurde, erlauben die meisten vdev-Arten es nicht mehr, weitere Geräte zu diesem vdev hinzuzufügen. Die Ausnahme sind Spiegel, die das Hinzufügen von weiteren Platten zum vdev gestatten, sowie stripes, die zu Spiegeln umgewandelt werden können, indem man zusätzliche Platten zum vdev anhängt. Obwohl weitere vdevs eingefügt werden können, um einen Pool zu vergrößern, kann das Layout des Pools nach dem Anlegen nicht mehr verändert werden. Stattdessen müssen die Daten gesichert, der Pool zerstört und danach neu erstellt werden.

Erstellen eines einfachen gespiegelten Pools:

```
# zpool create mypool mirror /dev/ada1 /dev/ada2
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada1    ONLINE       0     0     0
            ada2    ONLINE       0     0     0
```

```
errors: No known data errors
```

Mehrere vdevs können gleichzeitig angelegt werden. Geben Sie zusätzliche Gruppen von Platten, getrennt durch das vdev-Typ Schlüsselwort, in diesem Beispiel `mirror`, an:

```
# zpool create mypool mirror /dev/ada1 /dev/ada2 mirror /dev/ada3 /dev/ada4
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada1    ONLINE       0     0     0
            ada2    ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            ada3    ONLINE       0     0     0
            ada4    ONLINE       0     0     0

errors: No known data errors
```

Pools lassen sich auch durch die Angabe von Partitionen anstatt von ganzen Platten erzeugen. Durch die Verwendung von ZFS in einer separaten Partition ist es möglich, dass die gleiche Platte andere Partitionen für andere Zwecke besitzen kann. Dies ist besonders von Interesse, wenn Partitionen mit Bootcode und Dateisysteme, die zum starten benötigt werden, hinzugefügt werden können. Das erlaubt es, von Platten zu booten, die auch Teil eines Pools sind. Es gibt keinen Geschwindigkeitsnachteil unter FreeBSD wenn eine Partition anstatt einer ganzen Platte verwendet wird. Durch den Einsatz von Partitionen kann der Administrator die Platten *unter provisionieren*, indem weniger als die volle Kapazität Verwendung findet. Wenn in Zukunft eine Ersatzfestplatte mit der gleichen Größe als die Originalplatte eine kleinere Kapazität aufweist, passt die kleinere Partition immer noch und die Ersatzplatte kann immer noch verwendet werden.

Erstellen eines RAID-Z2-Pools mit Partitionen:

```
# zpool create mypool raidz2 /dev/ada0p3 /dev/ada1p3 /dev/ada2p3 /dev/ada3p3
/dev/ada4p3 /dev/ada5p3
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          raidz2-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
```

```
            ada1p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0
            ada3p3  ONLINE       0     0     0
            ada4p3  ONLINE       0     0     0
            ada5p3  ONLINE       0     0     0

errors: No known data errors
```

Ein Pool, der nicht länger benötigt wird, kann zerstört werden, so dass die Platten für einen anderen Einsatzzweck Verwendung finden können. Um einen Pool zu zerstören, müssen zuerst alle Datasets in diesem Pool abgehängt werden. Wenn die Datasets verwendet werden, wird das Abhängen fehlschlagen und der Pool nicht zerstört. Die Zerstörung des Pools kann erzwungen werden durch die Angabe der Option `-f`, jedoch kann dies undefiniertes Verhalten in den Anwendungen auslösen, die noch offene Dateien auf diesen Datasets hatten.

### 37.3.2. Hinzufügen und Löschen von Geräten

Es gibt zwei Fälle für das Hinzufügen von Platten zu einem Pool: einhängen einer Platte zu einem existierenden vdev mit `zpool attach` oder einbinden von vdevs zum Pool mit `zpool add`. Nur manche vdev-Arten gestatten es, Platten zum vdev hinzuzufügen, nachdem diese angelegt wurden.

Ein Pool mit nur einer einzigen Platte besitzt keine Redundanz. Datenverfälschung kann erkannt, aber nicht repariert werden, weil es keine weiteren Kopien der Daten gibt. Die Eigenschaft copies kann genutzt werden, um einen geringen Fehler wie einen beschädigtem Sektor auszumerzen, enthält aber nicht die gleiche Art von Schutz, die Spiegelung oder RAID-Z bieten. Wenn man mit einem Pool startet, der nur aus einer einzigen vdev-Platte besteht, kann mit dem Kommando `zpool attach` eine zustätzliche Platte dem vdev hinzugefügt werden, um einen Spiegel zu erzeugen. Mit `zpool attach` können auch zusätzliche Platten zu einer Spiegelgruppe eingefügt werden, was die Redundanz und Lesegeschwindigkeit steigert. Wenn die Platten, aus denen der Pool besteht, partitioniert sind, replizieren Sie das Layout der ersten Platte auf die Zweite. Verwenden Sie dazu `gpart backup` und `gpart restore`, um diesen Vorgang einfacher zu gestalten.

Umwandeln eines (stripe) vdevs namens *ada0p3* mit einer einzelnen Platte zu einem Spiegel durch das Einhängen von *ada1p3*:

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          ada0p3    ONLINE       0     0     0

errors: No known data errors
# zpool attach mypool ada0p3 ada1p3
Make sure to wait until resilver is done before rebooting.
```

```
If you boot from pool 'mypool', you may need to update
boot code on newly attached disk 'ada1p3'.

Assuming you use GPT partitioning und 'da0' is your new boot disk
you may use the following command:

        gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 da0
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada1
bootcode written to ada1
# zpool status
  pool: mypool
 state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
  scan: resilver in progress since Fri May 30 08:19:19 2014
        527M scanned out of 781M at 47.9M/s, 0h0m to go
        527M resilvered, 67.53% done
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0  (resilvering)

errors: No known data errors
# zpool status
  pool: mypool
 state: ONLINE
  scan: resilvered 781M in 0h0m with 0 errors on Fri May 30 08:15:58 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0

errors: No known data errors
```

Wenn das Hinzufügen von Platten zu einem vdev keine Option wie für RAID-Z ist, gibt es eine Alternative, nämlich einen anderen vdev zum Pool hinzuzufügen. Zusätzliche vdevs bieten höhere Geschwindigkeit, indem Schreibvorgänge über die vdevs verteilt werden. Jedes vdev ist dafür verantwortlich, seine eigene Redundanz sicherzustellen. Es ist möglich, aber nicht empfehlenswert, vdev-Arten zu mischen, wie zum Beispiel `mirror` und `RAID-Z`. Durch das Einfügen eines nicht-redundanten vdev zu einem gespiegelten Pool oder einem RAID-Z vdev riskiert man die Daten des gesamten Pools. Schreibvorgänge werden verteilt, deshalb ist der Ausfall einer nicht-redundanten Platte mit dem Verlust eines Teils von jedem Block verbunden, der auf den Pool geschrieben wird.

Daten werden über jedes vdev gestriped. Beispielsweise sind zwei Spiegel-vdevs effektiv ein RAID 10, dass über zwei Sets von Spiegeln die Daten schreibt. Speicherplatz wird so allokiert, dass jedes vdev zur gleichen Zeit vollgeschrieben wird. Es gibt einen Geschwindigkeitsnachteil wenn die vdevs unterschiedliche Menge von freiem Speicher aufweisen, wenn eine unproportionale Menge an Daten auf das weniger volle vdev geschrieben wird.

Wenn zusätzliche Geräte zu einem Pool, von dem gebootet wird, hinzugefügt werden, muss der Bootcode aktualisiert werden.

Einbinden einer zweiten Spiegelgruppe (ada2p3 und ada3p3) zu einem bestehenden Spiegel:

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: resilvered 781M in 0h0m with 0 errors on Fri May 30 08:19:35 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0

errors: No known data errors
# zpool add mypool mirror ada2p3 ada3p3
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada2
bootcode written to ada2
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada3
bootcode written to ada3
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub repaired 0 in 0h0m with 0 errors on Fri May 30 08:29:51 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
          mirror-1  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0
            ada3p3  ONLINE       0     0     0

errors: No known data errors
```

Momentan können vdevs nicht von einem Pool entfernt und Platten nur von einem Spiegel ausgehängt werden, wenn genug Redundanz übrig bleibt. Wenn auch nur eine Platte in einer Spiegelgruppe bestehen bleibt, hört der Spiegel auf zu existieren und wird zu einem stripe, was den

gesamten Pool riskiert, falls diese letzte Platte ausfällt.

Entfernen einer Platte aus einem Spiegel mit drei Platten:

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub repaired 0 in 0h0m with 0 errors on Fri May 30 08:29:51 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0

errors: No known data errors
# zpool detach mypool ada2p3
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub repaired 0 in 0h0m with 0 errors on Fri May 30 08:29:51 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0

errors: No known data errors
```

### 37.3.3. Den Status eines Pools überprüfen

Der Status eines Pools ist wichtig. Wenn ein Gerät sich abschaltet oder ein Lese-, Schreib- oder Prüfsummenfehler festgestellt wird, wird der dazugehörige Fehlerzähler erhöht. Die `status` -Ausgabe zeigt die Konfiguration und den Status von jedem Gerät im Pool und den Gesamtstatus des Pools. Aktionen, die durchgeführt werden sollten und Details zum letzten `scrub` werden ebenfalls angezeigt.

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub repaired 0 in 2h25m with 0 errors on Sat Sep 14 04:25:50 2013
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
```

```
          raidz2-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0
            ada3p3  ONLINE       0     0     0
            ada4p3  ONLINE       0     0     0
            ada5p3  ONLINE       0     0     0

errors: No known data errors
```

### 37.3.4. Fehler beseitigen

Wenn ein Fehler erkannt wurde, werden die Lese-, Schreib- oder Prüfsummenzähler erhöht. Die Fehlermeldung kann beseitigt und der Zähler mit `zpool clear mypool` zurückgesetzt werden. Den Fehlerzustand zurückzusetzen kann wichtig sein, wenn automatisierte Skripte ablaufen, die den Administrator informieren, sobald der Pool Fehler anzeigt. Weitere Fehler werden nicht gemeldet, wenn der alte Fehlerbericht nicht entfernt wurde.

### 37.3.5. Ein funktionierendes Gerät ersetzen

Es gibt eine Reihe von Situationen, in denen es nötig ist, eine Platte mit einer anderen auszutauschen. Wenn eine funktionierende Platte ersetzt wird, hält der Prozess die alte Platte während des Ersetzungsvorganges noch aktiv. Der Pool wird nie den Zustand degraded erhalten, was das Risiko eines Datenverlustes minimiert. Alle Daten der alten Platte werden durch das Kommando `zpool replace` auf die Neue übertragen. Nachdem die Operation abgeschlossen ist, wird die alte Platte vom vdev getrennt. Falls die neue Platte grösser ist als die alte Platte , ist es möglich den Pool zu vergrößern, um den neuen Platz zu nutzen. Lesen Sie dazu Einen Pool vergrößern.

Ersetzen eines funktionierenden Geräts in einem Pool:

```
# zpool status
  pool: mypool
 state: ONLINE
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0

errors: No known data errors
# zpool replace mypool ada1p3 ada2p3
Make sure to wait until resilver is done before rebooting.

If you boot from pool 'zroot', you may need to update
boot code on newly attached disk 'ada2p3'.
```

```
Assuming you use GPT partitioning und 'da0' is your new boot disk
you may use the following command:

        gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 da0
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada2
# zpool status
  pool: mypool
 state: ONLINE
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
  scan: resilver in progress since Mon Jun  2 14:21:35 2014
        604M scanned out of 781M at 46.5M/s, 0h0m to go
        604M resilvered, 77.39% done
config:

        NAME             STATE     READ WRITE CKSUM
        mypool           ONLINE       0     0     0
          mirror-0       ONLINE       0     0     0
            ada0p3       ONLINE       0     0     0
            replacing-1  ONLINE       0     0     0
              ada1p3     ONLINE       0     0     0
              ada2p3     ONLINE       0     0     0  (resilvering)

errors: No known data errors
# zpool status
  pool: mypool
 state: ONLINE
  scan: resilvered 781M in 0h0m with 0 errors on Mon Jun  2 14:21:52 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0

errors: No known data errors
```

### 37.3.6. Behandlung von fehlerhaften Geräten

Wenn eine Platte in einem Pool ausfällt, wird das vdev zu dem diese Platte gehört, den Zustand degraded erhalten. Alle Daten sind immer noch verfügbar, jedoch wird die Geschwindigkeit möglicherweise reduziert, weil die fehlenden Daten aus der verfügbaren Redundanz heraus berechnet werden müssen. Um das vdev in einen funktionierenden Zustand zurück zu versetzen, muss das physikalische Gerät ersetzt werden. ZFS wird dann angewiesen, den resilver-Vorgang zu beginnen. Daten, die sich auf dem defekten Gerät befanden, werden neu aus der vorhandenen Prüfsumme berechnet und auf das Ersatzgerät geschrieben. Nach Beendigung dieses Prozesses kehrt das vdev zum Status online zurück.

Falls das vdev keine Redundanz besitzt oder wenn mehrere Geräte ausgefallen sind und es nicht genug Redundanz gibt, um dies zu kompensieren, geht der Pool in den Zustand faulted über. Wenn keine ausreichende Anzahl von Geräten wieder an den Pool angeschlossen wird, fällt der Pool aus und die Daten müssen von Sicherungen wieder eingespielt werden.

Wenn eine defekte Platte ausgewechselt wird, wird der Name dieser defekten Platte mit der GUID des Geräts ersetzt. Ein neuer Gerätename als Parameter für `zpool replace` wird nicht benötigt, falls das Ersatzgerät den gleichen Gerätenamen besitzt.

Ersetzen einer defekten Platte durch `zpool replace`:

```
# zpool status
  pool: mypool
 state: DEGRADED
status: One or more devices could not be opened.  Sufficient replicas exist for
        the pool to continue functioning in a degraded state.
action: Attach the missing device und online it using 'zpool online'.
   see: http://illumos.org/msg/ZFS-8000-2Q
  scan: none requested
config:

        NAME                    STATE     READ WRITE CKSUM
        mypool                  DEGRADED     0     0     0
          mirror-0              DEGRADED     0     0     0
            ada0p3              ONLINE       0     0     0
            316502962686821739  UNAVAIL      0     0     0  was /dev/ada1p3

errors: No known data errors
# zpool replace mypool 316502962686821739 ada2p3
# zpool status
  pool: mypool
 state: DEGRADED
status: One or more devices is currently being resilvered.  The pool will
        continue to function, possibly in a degraded state.
action: Wait for the resilver to complete.
  scan: resilver in progress since Mon Jun  2 14:52:21 2014
        641M scanned out of 781M at 49.3M/s, 0h0m to go
        640M resilvered, 82.04% done
config:

        NAME                        STATE     READ WRITE CKSUM
        mypool                      DEGRADED     0     0     0
          mirror-0                  DEGRADED     0     0     0
            ada0p3                  ONLINE       0     0     0
            replacing-1             UNAVAIL      0     0     0
              15732067398082357289  UNAVAIL      0     0     0  was /dev/ada1p3/old
              ada2p3                ONLINE       0     0     0  (resilvering)

errors: No known data errors
# zpool status
```

```
  pool: mypool
 state: ONLINE
  scan: resilvered 781M in 0h0m with 0 errors on Mon Jun  2 14:52:38 2014
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0

errors: No known data errors
```

### 37.3.7. Einen Pool überprüfen

Es wird empfohlen, dass Pools regelmäßig geprüft (scrubbed) werden, idealerweise mindestens einmal pro Monat. Der `scrub`-Vorgang ist beansprucht die Platte sehr und reduziert die Geschwindigkeit während er läuft. Vermeiden Sie Zeiten, in denen großer Bedarf besteht, wenn Sie `scrub` starten oder benutzen Sie `vfs.zfs.scrub_delay`, um die relative Priorität vom `scrub` einzustellen, um zu verhindern, dass es mit anderen Aufgaben kollidiert.

```
# zpool scrub mypool
# zpool status
  pool: mypool
 state: ONLINE
  scan: scrub in progress since Wed Feb 19 20:52:54 2014
        116G scanned out of 8.60T at 649M/s, 3h48m to go
        0 repaired, 1.32% done
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          raidz2-0  ONLINE       0     0     0
            ada0p3  ONLINE       0     0     0
            ada1p3  ONLINE       0     0     0
            ada2p3  ONLINE       0     0     0
            ada3p3  ONLINE       0     0     0
            ada4p3  ONLINE       0     0     0
            ada5p3  ONLINE       0     0     0

errors: No known data errors
```

Falls eine Überrpüfaktion abgebrochen werden muss, geben Sie `zpool scrub -s mypool` ein.

### 37.3.8. Selbstheilung

Die Prüfsummen, welche zusammen mit den Datenblöcken gespeichert werden, ermöglichen dem Dateisystem, sich *selbst zu heilen*. Diese Eigenschaft wird automatisch Daten korrigieren, deren

Prüfsumme nicht mit der Gespeicherten übereinstimmt, die auf einem anderen Gerät, das Teil des Pools ist, vorhanden ist. Beispielsweise bei einem Spiegel aus zwei Platten, von denen eine anfängt, Fehler zu produzieren und nicht mehr länger Daten speichern kann. Dieser Fall ist sogar noch schlimmer, wenn auf die Daten seit einiger Zeit nicht mehr zugegriffen wurde, zum Beispiel bei einem Langzeit-Archivspeicher. Traditionelle Dateisysteme müssen dann Algorithmen wie fsck(8) ablaufen lassen, welche die Daten überprüfen und reparieren. Diese Kommandos benötigen einige Zeit und in gravierenden Fällen muss ein Administrator manuelle Entscheidungen treffen, welche Reparaturoperation vorgenommen werden soll. Wenn ZFS einen defekten Datenblock mit einer Prüfsumme erkennt, die nicht übereinstimmt, versucht es die Daten von der gespiegelten Platte zu lesen. Wenn diese Platte die korrekten Daten liefern kann, wird nicht nur dieser Datenblock an die anfordernde Applikation geschickt, sondern auch die falschen Daten auf der Disk reparieren, welche die falsche Prüfsumme erzeugt hat. Dies passiert während des normalen Betriebs des Pools, ohne dass eine Interaktion vom Systemadministrator notwendig wäre.

Das nächste Beispiel demonstriert dieses Verhalten zur Selbstheilung. Ein gespiegelter Pool mit den beiden Platten /dev/ada0 und /dev/ada1 wird angelegt.

```
# zpool create healer mirror /dev/ada0 /dev/ada1
# zpool status healer
  pool: healer
 state: ONLINE
  scan: none requested
config:

    NAME        STATE     READ WRITE CKSUM
    healer      ONLINE       0     0     0
      mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0     0

errors: No known data errors
# zpool list
NAME     SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG   CAP  DEDUP  HEALTH  ALTROOT
healer   960M  92.5K   960M         -         -     0%    0%  1.00x  ONLINE  -
```

Ein paar wichtige Daten, die es vor Datenfehlern mittels der Selbstheilungsfunktion zu schützen gilt, werden auf den Pool kopiert. Eine Prüfsumme wird zum späteren Vergleich berechnet.

```
# cp /some/important/data /healer
# zfs list
NAME     SIZE  ALLOC   FREE    CAP  DEDUP  HEALTH  ALTROOT
healer   960M  67.7M   892M     7%  1.00x  ONLINE  -
# sha1 /healer > checksum.txt
# cat checksum.txt
SHA1 (/healer) = 2753eff56d77d9a536ece6694bf0a82740344d1f
```

Datenfehler werden durch das Schreiben von zufälligen Daten an den Anfang einer Platte des Spiegels simuliert. Um ZFS daran zu hindern, die Daten so schnell zu reparieren, wie es diese

entdeckt, wird der Pool vor der Veränderung exportiert und anschließend wieder importiert.



Dies ist eine gefährliche Operation, die wichtige Daten zerstören kann. Es wird hier nur zu Demonstrationszwecken gezeigt und sollte nicht während des normalen Betriebs des Pools versucht werden. Dieses vorsätzliche Korrumpierungsbeispiel sollte auf gar keinen Fall auf einer Platte mit einem anderen Dateisystem durchgeführt werden. Verwenden Sie keine anderen Gerätenamen als diejenigen, die hier gezeigt werden, die Teil des Pools sind. Stellen Sie sicher, dass die passende Sicherungen angefertigt haben, bevor Sie dieses Kommando ausführen!

```
# zpool export healer
# dd if=/dev/random of=/dev/ada1 bs=1m count=200
200+0 records in
200+0 records out
209715200 bytes transferred in 62.992162 secs (3329227 bytes/sec)
# zpool import healer
```

Der Status des Pools zeigt an, dass bei einem Gerät ein Fehler aufgetreten ist. Wichtig zu wissen ist, dass Anwendungen, die Daten vom Pool lesen keine ungültigen Daten erhalten haben. ZFS lieferte Daten vom ada0-Gerät mit der korrekten Prüfsumme aus. Das Gerät mit der fehlerhaften Prüfsumme kann sehr einfach gefunden werden, da die Spalte `CKSUM` einen Wert ungleich Null enthält.

```
# zpool status healer
    pool: healer
   state: ONLINE
  status: One or more devices has experienced an unrecoverable error.  An
          attempt was made to correct the error.  Applications are unaffected.
  action: Determine if the device needs to be replaced, und clear the errors
          using 'zpool clear' or replace the device with 'zpool replace'.
     see: http://illumos.org/msg/ZFS-8000-4J
    scan: none requested
  config:

      NAME        STATE     READ WRITE CKSUM
      healer      ONLINE       0     0     0
        mirror-0  ONLINE       0     0     0
          ada0    ONLINE       0     0     0
          ada1    ONLINE       0     0     1

errors: No known data errors
```

Der Fehler wurde erkannt und korrigiert durch die vorhandene Redundanz, welche aus der nicht betroffenen Platte ada0 des Spiegels gewonnen wurde. Ein Vergleich der Prüfsumme mit dem Original wird zeigen, ob sich der Pool wieder in einem konsistenten Zustand befindet.

```
# sha1 /healer >> checksum.txt
# cat checksum.txt
SHA1 (/healer) = 2753eff56d77d9a536ece6694bf0a82740344d1f
SHA1 (/healer) = 2753eff56d77d9a536ece6694bf0a82740344d1f
```

Die beiden Prüfsummen, die vor und nach der vorsätzlichen Korrumpierung der Daten des Pools angelegt wurden, stimmen immer noch überein. Dies zeigt wie ZFS in der Lage ist, Fehler automatisch zu erkennen und zu korrigieren, wenn die Prüfsummen nicht übereinstimmen. Beachten Sie, dass dies nur möglich ist, wenn genug Redundanz im Pool vorhanden ist. Ein Pool, der nur aus einer einzigen Platte besteht besitzt keine Selbstheilungsfunktion. Dies ist auch der Grund warum Prüfsummen bei ZFS so wichtig sind und deshalb aus keinem Grund deaktiviert werden sollten. Kein fsck(8) ist nötig, um diese Fehler zu erkennen und zu korrigieren und der Pool war während der gesamten Zeit, in der das Problem bestand, verfügbar. Eine scrub-Aktion ist nun nötig, um die fehlerhaften Daten auf ada1 zu beheben.

```
# zpool scrub healer
# zpool status healer
  pool: healer
 state: ONLINE
status: One or more devices has experienced an unrecoverable error.  An
            attempt was made to correct the error.  Applications are unaffected.
action: Determine if the device needs to be replaced, und clear the errors
            using 'zpool clear' or replace the device with 'zpool replace'.
   see: http://illumos.org/msg/ZFS-8000-4J
  scan: scrub in progress since Mon Dec 10 12:23:30 2012
        10.4M scanned out of 67.0M at 267K/s, 0h3m to go
        9.63M repaired, 15.56% done
config:

    NAME        STATE     READ WRITE CKSUM
    healer      ONLINE       0     0     0
      mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0   627  (repairing)

errors: No known data errors
```

Durch das scrub werden die Daten von ada0 gelesen und alle Daten mit einer falschen durch diejenigen mit der richtigen Prüfsumme auf ada1 ersetzt. Dies wird durch die Ausgabe `(repairing)` des Kommandos `zpool status` angezeigt. Nachdem die Operation abgeschlossen ist, ändert sich der Poolstatus zu:

```
# zpool status healer
  pool: healer
 state: ONLINE
status: One or more devices has experienced an unrecoverable error.  An
        attempt was made to correct the error.  Applications are unaffected.
```

```
action: Determine if the device needs to be replaced, und clear the errors
             using 'zpool clear' or replace the device with 'zpool replace'.
   see: http://illumos.org/msg/ZFS-8000-4J
  scan: scrub repaired 66.5M in 0h2m with 0 errors on Mon Dec 10 12:26:25 2012
config:

    NAME        STATE     READ WRITE CKSUM
    healer      ONLINE       0     0     0
      mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0 2.72K

errors: No known data errors
```

Nach der scrub-Operation und der anschliessenden Synchronisation der Daten von ada0 nach ada1, kann die Fehlermeldung vom Poolstatus durch die Eingabe von `zpool clear`bereinigt werden.

```
# zpool clear healer
# zpool status healer
  pool: healer
 state: ONLINE
  scan: scrub repaired 66.5M in 0h2m with 0 errors on Mon Dec 10 12:26:25 2012
config:

    NAME        STATE     READ WRITE CKSUM
    healer      ONLINE       0     0     0
      mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0     0

errors: No known data errors
```

Der Pool ist jetzt wieder in einem voll funktionsfähigen Zustand versetzt worden und alle Fehler wurden beseitigt.

### 37.3.9. Einen Pool vergrössern

Die verwendbare Größe eines redundant ausgelegten Pools ist durch die Kapazität des kleinsten Geräts in jedem vdev begrenzt. Das kleinste Gerät kann durch ein größeres Gerät ersetzt werden. Nachdem eine replace oder resilver-Operation abgeschlossen wurde, kann der Pool anwachsen, um die Kapazität des neuen Geräts zu nutzen. Nehmen wir als Beispiel einen Spiegel mit einer 1 TB und einer 2 TB Platte. Der verwendbare Plattenplatz beträgt 1 TB. Wenn die 1 TB Platte mit einer anderen 2 TB Platte ersetzt wird, kopiert der resilver-Prozess die existierenden Daten auf die neue Platte. Da beide Geräte nun 2 TB Kapazität besitzen, kann auch der verfügbare Plattenplatz auf die Größe von 2 TB anwachsen.

Die Erweiterung wird durch das Kommando `zpool online -e` auf jedem Gerät ausgelöst. Nachdem alle Geräte expandiert wurden, wird der Speicher im Pool zur Verfügung gestellt.

### 37.3.10. Importieren und Exportieren von Pools

Pools werden *exportiert* bevor diese an ein anderes System angeschlossen werden. Alle Datasets werden abgehängt und jedes Gerät wird als exportiert markiert, ist jedoch immer noch gesperrt, so dass es nicht von anderen Festplattensubsystemen verwendet werden kann. Dadurch können Pools auf anderen Maschinen *importiert* werden, die ZFS und sogar andere Hardwarearchitekturen (bis auf ein paar Ausnahmen, siehe zpool(8)) unterstützen. Besitzt ein Dataset offene Dateien, kann `zpool export -f` den Export des Pools erzwingen. Verwenden Sie dies mit Vorsicht. Die Datasets werden dadurch gewaltsam abgehängt, was bei Anwendungen, die noch offene Dateien auf diesem Dataset hatten, möglicherweise zu unerwartetem Verhalten führen kann.

Einen nichtverwendeten Pool exportieren:

```
# zpool export mypool
```

Beim Importieren eines Pool werden auch automatisch alle Datasets eingehängt. Dies ist möglicherweise nicht das bevorzugte Verhalten und wird durch `zpool import -N` verhindert. Durch `zpool import -o` temporäre Eigenschaften nur für diesen Import gesetzt. Mit dem Befehl `zpool import altroot=` ist es möglich, einen Pool mit einem anderen Basiseinhängepunkt anstatt der Wurzel des Dateisystems einzubinden. Wenn der Pool zuletzt auf einem anderen System verwendet und nicht korrekt exportiert wurde, muss unter Umständen ein Import erzwungen werden durch `zpool import -f`. Alle Pools, die momentan nicht durch ein anderes System verwendet werden, lassen sich mit `zpool import -a` importieren.

Alle zum Import verfügbaren Pools auflisten:

```
# zpool import
   pool: mypool
     id: 9930174748043525076
  state: ONLINE
 action: The pool can be imported using its name or numeric identifier.
 config:

        mypool      ONLINE
          ada2p3    ONLINE
```

Den Pool mit einem anderen Wurzelverzeichnis importieren:

```
# zpool import -o altroot=/mnt mypool
# zfs list
zfs list
NAME                 USED  AVAIL  REFER  MOUNTPOINT
mypool               110K  47.0G    31K  /mnt/mypool
```

### 37.3.11. Einen Pool aktualisieren

Nachdem FreeBSD aktualisiert wurde oder wenn der Pool von einem anderen System, das eine ältere Version von ZFS einsetzt, lässt sich der Pool manuell auf den aktuellen Stand von ZFS bringen, um die neuesten Eigenschaften zu unterstützen. Bedenken Sie, ob der Pool jemals wieder von einem älteren System eingebunden werden muss, bevor Sie die Aktualisierung durchführen. Das aktualisieren eines Pools ist ein nicht umkehrbarer Prozess. ältere Pools lassen sich aktualisieren, jedoch lassen sich Pools mit neueren Eigenschaften nicht wieder auf eine ältere Version bringen.

Aktualisierung eines v28-Pools, um `Feature Flags` zu unterstützen:

```
# zpool status
  pool: mypool
 state: ONLINE
status: The pool is formatted using a legacy on-disk format.  The pool can
        still be used, but some features are unavailable.
action: Upgrade the pool using 'zpool upgrade'.  Once this is done, the
        pool will no longer be accessible on software that does not support feat
        flags.
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0     0

errors: No known data errors
# zpool upgrade
This system supports ZFS pool feature flags.

The following pools are formatted with legacy version numbers und can
be upgraded to use feature flags.  After being upgraded, these pools
will no longer be accessible by software that does not support feature
flags.

VER  POOL
---  ------------
28   mypool

Use 'zpool upgrade -v' for a list of available legacy versions.
Every feature flags pool has all supported features enabled.
# zpool upgrade mypool
This system supports ZFS pool feature flags.

Successfully upgraded 'mypool' from version 28 to feature flags.
Enabled the following features on 'mypool':
  async_destroy
```

```
  empty_bpobj
  lz4_compress
  multi_vdev_crash_dump
```

Die neueren Eigenschaften von ZFS werden nicht verfügbar sein, bis `zpool upgrade` abgeschlossen ist. `zpool upgrade -v` kann verwendet werden, um zu sehen, welche neuen Eigenschaften durch die Aktualisierung bereitgestellt werden, genauso wie diejenigen, die momentan schon verfügbar sind.

Einen Pool um zusätzliche Feature Flags erweitern:

```
# zpool status
  pool: mypool
 state: ONLINE
status: Some supported features are not enabled on the pool. The pool can
        still be used, but some features are unavailable.
action: Enable all features using 'zpool upgrade'. Once this is done,
        the pool may no longer be accessible by software that does not support
        the features. See zpool-features(7) for details.
  scan: none requested
config:

        NAME        STATE     READ WRITE CKSUM
        mypool      ONLINE       0     0     0
          mirror-0  ONLINE       0     0     0
        ada0    ONLINE       0     0     0
        ada1    ONLINE       0     0     0

errors: No known data errors
# zpool upgrade
This system supports ZFS pool feature flags.

All pools are formatted using feature flags.

Some supported features are not enabled on the following pools. Once a
feature is enabled the pool may become incompatible with software
that does not support the feature. See zpool-features(7) for details.

POOL  FEATURE
---------------
zstore
      multi_vdev_crash_dump
      spacemap_histogram
      enabled_txg
      hole_birth
      extensible_dataset
      bookmarks
      filesystem_limits
# zpool upgrade mypool
This system supports ZFS pool feature flags.
```

```
Enabled the following features on 'mypool':
  spacemap_histogram
  enabled_txg
  hole_birth
  extensible_dataset
  bookmarks
  filesystem_limits
```



Der Bootcode muss auf Systemen, die von dem Pool starten, aktualisiert werden, um diese neue Version zu unterstützen. Verwenden Sie `gpart bootcode` auf der Partition, die den Bootcode enthält. Es gibt zwei Arten von Bootcode, je nachdem, wie das System bootet: GPT (die häufigste Option) und EFI (für moderne Systeme).

Benutzen Sie für GPT den folgenden Befehl:

```
# gpart bootcode -b /boot/pmbr -p /boot/gptzfsboot -i 1 ada1
```

Für Systeme, die EFI zum Booten benutzen, führen Sie folgenden Befehl aus:

```
# gpart bootcode -p /boot/boot1.efifat -i 1 ada1
```

Installieren Sie den Bootcode auf allen bootfähigen Platten im Pool. Lesen Sie gpart(8) für weitere Informationen.

### 37.3.12. Aufgezeichnete Historie des Pools anzeigen

Befehle, die den Pool in irgendeiner Form verändern, werden aufgezeichnet. Diese Befehle beinhalten das Erstellen von Datasets, verändern von Eigenschaften oder das Ersetzen einer Platte. Diese Historie ist nützlich um nachzuvollziehen, wie ein Pool aufgebaut ist und welcher Benutzer eine bestimmte Aktion wann und wie getätigt hat. Die aufgezeichnete Historie wird nicht in einer Logdatei festgehalten, sondern ist Teil des Pools selbst. Das Kommando zum darstellen dieser Historie lautet passenderweise `zpool history`:

```
# zpool history
History for 'tank':
2013-02-26.23:02:35 zpool create tank mirror /dev/ada0 /dev/ada1
2013-02-27.18:50:58 zfs set atime=off tank
2013-02-27.18:51:09 zfs set checksum=fletcher4 tank
2013-02-27.18:51:18 zfs create tank/backup
```

Die Ausgabe zeigt `zpool` und `zfs`-Befehle, die ausgeführt wurden zusammen mit einem Zeitstempel. Nur Befehle, die den Pool verändern werden aufgezeichnet. Befehle wie `zfs list` sind dabei nicht enthalten. Wenn kein Name angegeben wird, erscheint die gesamte Historie aller Pools.

Der Befehl `zpool history` kann sogar noch mehr Informationen ausgeben, wenn die Optionen `-i`

oder `-l` angegeben werden. Durch `-i` zeigt ZFS vom Benutzer eingegebene, als auch interne Ereignisse an.

```
# zpool history -i
History for 'tank':
2013-02-26.23:02:35 [internal pool create txg:5] pool spa 28; zfs spa 28; zpl 5;uts
9.1-RELEASE 901000 amd64
2013-02-27.18:50:53 [internal property set txg:50] atime=0 dataset = 21
2013-02-27.18:50:58 zfs set atime=off tank
2013-02-27.18:51:04 [internal property set txg:53] checksum=7 dataset = 21
2013-02-27.18:51:09 zfs set checksum=fletcher4 tank
2013-02-27.18:51:13 [internal create txg:55] dataset = 39
2013-02-27.18:51:18 zfs create tank/backup
```

Weitere Details lassen sich durch die Angabe von `-l` entlocken. Historische Einträge werden in einem langen Format ausgegeben, einschließlich Informationen wie der Name des Benutzers, welcher das Kommando eingegeben hat und der Hostname, auf dem die Änderung erfolgte.

```
# zpool history -l
History for 'tank':
2013-02-26.23:02:35 zpool create tank mirror /dev/ada0 /dev/ada1 [user 0 (root) on
:global]
2013-02-27.18:50:58 zfs set atime=off tank [user 0 (root) on myzfsbox:global]
2013-02-27.18:51:09 zfs set checksum=fletcher4 tank [user 0 (root) on myzfsbox:global]
2013-02-27.18:51:18 zfs create tank/backup [user 0 (root) on myzfsbox:global]
```

Die Ausgabe zeigt, dass der Benutzer `root` den gespiegelten Pool mit den beiden Platten /dev/ada0 und /dev/ada1 angelegt hat. Der Hostname `myzfsbox` wird ebenfalls in den Kommandos angezeigt, nachdem der Pool erzeugt wurde. Die Anzeige des Hostnamens wird wichtig, sobald der Pool von einem System exportiert und auf einem anderen importiert wird. Die Befehle, welche auf dem anderen System verwendet werden, können klar durch den Hostnamen, der bei jedem Kommando mit verzeichnet wird, unterschieden werden.

Beide Optionen für `zpool history` lassen sich auch kombinieren, um die meisten Details zur Historie eines Pools auszugeben. Die Pool Historie liefert wertvolle Informationen, wenn Aktionen nachverfolgt werden müssen oder zur Fehlerbeseitigung mehr Informationen gebraucht werden.

### 37.3.13. Geschwindigkeitsüberwachung

Ein eingebautes Überwachungssystem kann I/O-Statistiken in Echtzeit liefern. Es zeigt die Menge von freiem und belegtem Speicherplatz auf dem Pool an, wieviele Lese- und Schreiboperationen pro Sekunde durchgeführt werden und die aktuell verwendete I/O-Bandbreite. Standardmäßig werden alle Pools in einem System überwacht und angezeigt. Ein Poolname kann angegeben werden, um die Anzeige auf diesen Pool zu beschränken. Ein einfaches Beispiel:

```
# zpool iostat
               capacity     operations    bundwidth
```

```
pool        alloc   free   read  write   read  write
----------  -----  -----  -----  -----  -----  -----
data         288G  1.53T      2     11  11.3K  57.1K
```

Um kontinuierlich die I/O-Aktivität zu überprüfen, kann eine Zahl als letzter Parameter angegeben werden, die ein Intervall in Sekunden angibt, die zwischen den Aktualisierungen vergehen soll. Die nächste Zeile mit Statistikinformationen wird dann nach jedem Intervall ausgegeben. Drücken Sie `Ctrl` + `C`, um diese kontinuierliche Überwachung zu stoppen. Alternativ lässt sich auch eine zweite Zahl nach dem Intervall auf der Kommandozeile angeben, welche die Obergrenze von Statistikausgaben darstellt, die angezeigt werden sollen.

Noch mehr Informationen zu I/O-Statistiken können durch Angabe der Option `-v` angezeigt werden. Jedes Gerät im Pool wird dann mit einer eigenen Statistikzeile aufgeführt. Dies ist hilfreich um zu sehen, wieviele Lese- und Schreiboperationen von jedem Gerät durchgeführt werden und kann bei der Diagnose eines langsamen Geräts, das den Pool ausbremst, hilfreich sein. Dieses Beispiel zeigt einen gespiegelten Pool mit zwei Geräten:

```
# zpool iostat -v
                           capacity     operations     bundwidth
pool                     alloc   free   read  write   read  write
-----------------------  -----  -----  -----  -----  -----  -----
data                      288G  1.53T      2     12  9.23K  61.5K
  mirror                  288G  1.53T      2     12  9.23K  61.5K
    ada1                     -      -      0      4  5.61K  61.7K
    ada2                     -      -      1      4  5.04K  61.7K
-----------------------  -----  -----  -----  -----  -----  -----
```

### 37.3.14. Einen Pool aufteilen

Ein Pool, der aus einem oder mehreren gespiegelten vdevs besteht, kann in zwei Pools aufgespalten werden. Falls nicht anders angegeben, wird das letzte Mitglied eines Spiegels abgehängt und dazu verwendet, einen neuen Pool mit den gleichen Daten zu erstellen. Die Operation sollte zuerst mit der Option `-n` versucht werden. Die Details der vorgeschlagenen Option werden dargestellt, ohne die Aktion in Wirklichkeit durchzuführen. Das hilft dabei zu bestätigen, ob die Aktion das tut, was der Benutzer damit vor hatte.

## 37.4. `zfs` Administration

Das `zfs`-Werkzeug ist dafür verantwortlich, alle ZFS Datasets innerhalb eines Pools zu erstellen, zerstören und zu verwalten. Der Pool selbst wird durch `zpool` verwaltet.

### 37.4.1. Datasets erstellen und zerstören

Anders als in traditionellen Festplatten- und Volumenmanagern wird der Plattenplatz in ZFS *nicht* vorher allokiert. Bei traditionellen Dateisystemen gibt es, nachdem der Plattenplatz partitioniert und zugeteilt wurde, keine Möglichkeit, ein zusätzliches Dateisystem hinzuzufügen, ohne eine neue Platte anzuschließen. Mit ZFS lassen sich neue Dateisysteme zu jeder Zeit anlegen. Jedes *Dataset*

besitzt Eigenschaften wie Komprimierung, Deduplizierung, Zwischenspeicher (caching), Quotas, genauso wie andere nützliche Einstellungen wie Schreibschutz, Unterscheidung zwischen Groß- und Kleinschreibung, Netzwerkfreigaben und einen Einhängepunkt. Datasets können ineinander verschachtelt werden und Kind-Datasets erben die Eigenschaften ihrer Eltern. Jedes Dataset kann als eine Einheit verwaltet, delegiert, repliziert, mit Schnappschüssen versehen, in Jails gesteckt und zerstört werden. Es gibt viele Vorteile, ein separates Dataset für jede Art von Dateien anzulegen. Der einzige Nachteil einer großen Menge an Datasets ist, dass manche Befehle wie `zfs list` langsamer sind und dass das Einhängen von hunderten oder hunderttausenden von Datasets den FreeBSD-Bootvorgang verzögert.

Erstellen eines neuen Datasets und aktivieren von LZ4 Komprimierung:

```
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
mypool                781M  93.2G   144K  none
mypool/ROOT           777M  93.2G   144K  none
mypool/ROOT/default   777M  93.2G   777M  /
mypool/tmp            176K  93.2G   176K  /tmp
mypool/usr            616K  93.2G   144K  /usr
mypool/usr/home       184K  93.2G   184K  /usr/home
mypool/usr/ports      144K  93.2G   144K  /usr/ports
mypool/usr/src        144K  93.2G   144K  /usr/src
mypool/var           1.20M  93.2G   608K  /var
mypool/var/crash      148K  93.2G   148K  /var/crash
mypool/var/log        178K  93.2G   178K  /var/log
mypool/var/mail       144K  93.2G   144K  /var/mail
mypool/var/tmp        152K  93.2G   152K  /var/tmp
# zfs create -o compress=lz4 mypool/usr/mydataset
# zfs list
NAME                   USED  AVAIL  REFER  MOUNTPOINT
mypool                 781M  93.2G   144K  none
mypool/ROOT            777M  93.2G   144K  none
mypool/ROOT/default    777M  93.2G   777M  /
mypool/tmp             176K  93.2G   176K  /tmp
mypool/usr             704K  93.2G   144K  /usr
mypool/usr/home        184K  93.2G   184K  /usr/home
mypool/usr/mydataset  87.5K  93.2G  87.5K  /usr/mydataset
mypool/usr/ports       144K  93.2G   144K  /usr/ports
mypool/usr/src         144K  93.2G   144K  /usr/src
mypool/var            1.20M  93.2G   610K  /var
mypool/var/crash       148K  93.2G   148K  /var/crash
mypool/var/log         178K  93.2G   178K  /var/log
mypool/var/mail        144K  93.2G   144K  /var/mail
mypool/var/tmp         152K  93.2G   152K  /var/tmp
```

Ein Dataset zu zerstören ist viel schneller, als alle Dateien zu löschen, die sich in dem Dataset befindet, da es keinen Scan aller Dateien und aktualisieren der dazugehörigen Metadaten erfordert.

Zerstören des zuvor angelegten Datasets:

```
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
mypool                880M  93.1G   144K  none
mypool/ROOT           777M  93.1G   144K  none
mypool/ROOT/default   777M  93.1G   777M  /
mypool/tmp            176K  93.1G   176K  /tmp
mypool/usr            101M  93.1G   144K  /usr
mypool/usr/home       184K  93.1G   184K  /usr/home
mypool/usr/mydataset  100M  93.1G   100M  /usr/mydataset
mypool/usr/ports      144K  93.1G   144K  /usr/ports
mypool/usr/src        144K  93.1G   144K  /usr/src
mypool/var           1.20M  93.1G   610K  /var
mypool/var/crash      148K  93.1G   148K  /var/crash
mypool/var/log        178K  93.1G   178K  /var/log
mypool/var/mail       144K  93.1G   144K  /var/mail
mypool/var/tmp        152K  93.1G   152K  /var/tmp
# zfs destroy mypool/usr/mydataset
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
mypool                781M  93.2G   144K  none
mypool/ROOT           777M  93.2G   144K  none
mypool/ROOT/default   777M  93.2G   777M  /
mypool/tmp            176K  93.2G   176K  /tmp
mypool/usr            616K  93.2G   144K  /usr
mypool/usr/home       184K  93.2G   184K  /usr/home
mypool/usr/ports      144K  93.2G   144K  /usr/ports
mypool/usr/src        144K  93.2G   144K  /usr/src
mypool/var           1.21M  93.2G   612K  /var
mypool/var/crash      148K  93.2G   148K  /var/crash
mypool/var/log        178K  93.2G   178K  /var/log
mypool/var/mail       144K  93.2G   144K  /var/mail
mypool/var/tmp        152K  93.2G   152K  /var/tmp
```

In modernen Versionen von ZFS ist `zfs destroy` asynchron und der freie Speicherplatz kann erst nach ein paar Minuten im Pool auftauchen. Verwenden Sie `zpool get freeing poolname`, um die Eigenschaft `freeing` aufzulisten, die angibt, bei wievielen Datasets die Blöcke im Hintergrund freigegeben werden. Sollte es Kind-Datasets geben, Schnappschüsse oder andere Datasets, dann lässt sich der Elternknoten nicht zerstören. Um ein Dataset und all seine Kinder zu zerstören, verwenden Sie die Option `-r`, um das Dataset und all seine Kinder rekursiv zu entfernen. Benutzen Sie die Option `-n` und `-v`, um Datasets und Snapshots anzuzeigen, die durch diese Aktion zerstört werden würden, dies jedoch nur zu simulieren und nicht wirklich durchzuführen. Speicherplatz, der dadurch freigegeben würde, wird ebenfalls angezeigt.

### 37.4.2. Volumes erstellen und zerstören

Ein Volume ist ein spezieller Typ von Dataset. Anstatt dass es als Dateisystem eingehängt wird, stellt es ein Block-Gerät unter /dev/zvol/poolname/dataset dar. Dies erlaubt es, das Volume für andere Dateisysteme zu verwenden, die Festplatten einer virtuellen Maschine bereitzustellen oder über Protokolle wie iSCSI oder HAST exportiert zu werden.

Ein Volume kann mit einem beliebigen Dateisystem formatiert werden oder auch ohne ein Dateisystem als reiner Datenspeicher fungieren. Für den Benutzer erscheint ein Volume als eine gewöhnliche Platte. Indem gewöhnliche Dateisysteme auf diesen *zvols* angelegt werden, ist es möglich, diese mit Eigenschaften auszustatten, welche diese normalerweise nicht besitzen. Beispielsweise wird durch Verwendung der Komprimierungseigenschaft auf einem 250 MB Volume das Erstellen eines komprimierten FAT Dateisystems möglich.

```
# zfs create -V 250m -o compression=on tank/fat32
# zfs list tank
NAME USED AVAIL REFER MOUNTPOINT
tank 258M  670M   31K /tank
# newfs_msdos -F32 /dev/zvol/tank/fat32
# mount -t msdosfs /dev/zvol/tank/fat32 /mnt
# df -h /mnt | grep fat32
Filesystem           Size Used Avail Capacity Mounted on
/dev/zvol/tank/fat32 249M  24k  249M     0%   /mnt
# mount | grep fat32
/dev/zvol/tank/fat32 on /mnt (msdosfs, local)
```

Ein Volume zu zerstören ist sehr ähnlich wie ein herkömmliches Dataset zu entfernen. Die Operation wird beinahe sofort durchgeführt, jedoch kann es mehrere Minuten dauern, bis der freie Speicherplatz im Hintergrund wieder freigegeben ist.

### 37.4.3. Umbenennen eines Datasets

Der Name eines Datasets lässt sich durch `zfs rename` ändern. Das Eltern-Dataset kann ebenfalls mit diesem Kommando umbenannt werden. Ein Dataset unter einem anderen Elternteil umzubenennen wird den Wert dieser Eigenschaft verändern, die vom Elternteil vererbt wurden. Wird ein Dataset umbenannt, wird es abgehängt und dann erneut unter der neuen Stelle eingehängt (welche vom neuen Elternteil geerbt wird). Dieses Verhalten kann durch die Option `-u` verhindert werden.

Ein Dataset umbenennen und unter einem anderen Elterndataset verschieben:

```
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
mypool                780M  93.2G   144K  none
mypool/ROOT           777M  93.2G   144K  none
mypool/ROOT/default   777M  93.2G   777M  /
mypool/tmp            176K  93.2G   176K  /tmp
mypool/usr            704K  93.2G   144K  /usr
mypool/usr/home       184K  93.2G   184K  /usr/home
mypool/usr/mydataset 87.5K  93.2G  87.5K  /usr/mydataset
mypool/usr/ports      144K  93.2G   144K  /usr/ports
mypool/usr/src        144K  93.2G   144K  /usr/src
mypool/var           1.21M  93.2G   614K  /var
mypool/var/crash      148K  93.2G   148K  /var/crash
mypool/var/log        178K  93.2G   178K  /var/log
```

```
mypool/var/mail        144K  93.2G   144K  /var/mail
mypool/var/tmp         152K  93.2G   152K  /var/tmp
# zfs rename mypool/usr/mydataset mypool/var/newname
# zfs list
NAME                  USED  AVAIL  REFER  MOUNTPOINT
mypool                780M  93.2G   144K  none
mypool/ROOT           777M  93.2G   144K  none
mypool/ROOT/default   777M  93.2G   777M  /
mypool/tmp            176K  93.2G   176K  /tmp
mypool/usr            616K  93.2G   144K  /usr
mypool/usr/home       184K  93.2G   184K  /usr/home
mypool/usr/ports      144K  93.2G   144K  /usr/ports
mypool/usr/src        144K  93.2G   144K  /usr/src
mypool/var           1.29M  93.2G   614K  /var
mypool/var/crash      148K  93.2G   148K  /var/crash
mypool/var/log        178K  93.2G   178K  /var/log
mypool/var/mail       144K  93.2G   144K  /var/mail
mypool/var/newname   87.5K  93.2G  87.5K  /var/newname
mypool/var/tmp        152K  93.2G   152K  /var/tmp
```

Schnappschüsse können auf diese Weise ebenfalls umbenannt werden. Aufgrund der Art von Schnappschüssen können diese nicht unter einem anderen Elterndataset eingehängt werden. Um einen rekursiven Schnappschuss umzubenennen, geben Sie die Option `-r` an, um alle Schnappschüsse mit dem gleichen Namen im Kind-Dataset ebenfalls umzubenennen.

```
# zfs list -t snapshot
NAME                                 USED  AVAIL  REFER  MOUNTPOINT
mypool/var/newname@first_snapshot       0      -  87.5K  -
# zfs rename mypool/var/newname@first_snapshot new_snapshot_name
# zfs list -t snapshot
NAME                                    USED  AVAIL  REFER  MOUNTPOINT
mypool/var/newname@new_snapshot_name       0      -  87.5K  -
```

### 37.4.4. Festlegen von Dataset-Eigenschaften

Jedes ZFS-Dataset besitzt eine Menge von Eigenschaften, die sein Verhalten beeinflussen. Die meisten Eigenschaften werden automatisch vom Eltern-Dataset vererbt, können jedoch lokal überschrieben werden. Sie legen eine Eigenschaft durch `zfs set property=value dataset` fest. Die meisten Eigenschaften haben eine begrenzte Menge von gültigen Werten. `zfs get` stellt diese dar und zeigt jede mögliche Eigenschaft und gültige Werte an. Die meisten Eigenschaften können über `zfs inherit` wieder auf ihren Ausgangswert zurückgesetzt werden.

Benutzerdefinierte Eigenschaften lassen sich ebenfalls setzen. Diese werden Teil der Konfiguration des Datasets und können dazu verwendet werden, zusätzliche Informationen über das Dataset oder seine Bestandteile zu speichern. Um diese benutzerdefinierten Eigenschaften von den ZFS-eigenen zu unterscheiden, wird ein Doppelpunkt (`:`) verwendet, um einen eigenen Namensraum für diese Eigenschaft zu erstellen.

```
# zfs set custom:costcenter=1234 tank
# zfs get custom:costcenter tank
NAME PROPERTY           VALUE SOURCE
tank custom:costcenter  1234  local
```

Um eine selbstdefinierte Eigenschaft umzubenennen, verwenden Sie `zfs inherit` mit der Option `-r`. Wenn die benutzerdefinierte Eigenschaft nicht in einem der Eltern-Datasets definiert ist, wird diese komplett entfernt (obwohl diese Änderungen natürlich in der Historie des Pools noch aufgezeichnet sind).

```
# zfs inherit -r custom:costcenter tank
# zfs get custom:costcenter tank
NAME    PROPERTY           VALUE              SOURCE
tank    custom:costcenter  -                  -
# zfs get all tank | grep custom:costcenter
#
```

#### Festlegen und Abfragen von Eigenschaften für Freigaben

Zwei häufig verwendete und nützliche Dataset-Eigenschaften sind die Freigabeoptionen von NFS und SMB. Diese Optionen legen fest, ob und wie ZFS-Datasets im Netzwerk freigegeben werden. Derzeit unterstützt FreeBSD nur Freigaben von Datasets über NFS. Um den Status einer Freigabe zu erhalten, geben Sie folgendes ein:

```
# zfs get sharenfs mypool/usr/home
NAME              PROPERTY   VALUE   SOURCE
mypool/usr/home   sharenfs   on      local
# zfs get sharesmb mypool/usr/home
NAME              PROPERTY   VALUE   SOURCE
mypool/usr/home   sharesmb   off     local
```

Um ein Dataset freizugeben, geben Sie ein:

```
# zfs set sharenfs=on mypool/usr/home
```

Es ist auch möglich, weitere Optionen für die Verwendung von Datasets über NFS zu definieren, wie etwa `-alldirs`, `-maproot` und `-network`. Um zusätzliche Optionen auf ein durch NFS freigegebenes Dataset festzulegen, geben Sie ein:

```
# zfs set sharenfs="-alldirs,maproot=root,-network=192.168.1.0/24" mypool/usr/home
```

### 37.4.5. Verwalten von Schnappschüssen

Schnappschüsse sind eine der mächtigen Eigenschaften von ZFS. Ein Schnappschuss bietet einen

nur-Lese Zustand eines Datasets zu einem bestimmten Zeitpunkt. Mit Kopieren-beim-Schreiben (Copy-On-Write COW), können Schnappschüsse schnell erstellt werden durch das Aufheben der älteren Version der Daten auf der Platte. Falls kein Snapshot existiert, wird der Speicherplatz wieder für zukünftige Verwendung freigegeben wenn Daten geschrieben oder gelöscht werden. Schnappschüsse sparen Speicherplatz, indem diese nur die Unterschiede zwischen dem momentanen Dataset und der vorherigen Version aufzeichnen. Schnappschüsse sind nur auf ganzen Datasets erlaubt, nicht auf individuellen Dateien oder Verzeichnissen. Wenn ein Schnappschuss eines Datasets erstellt wird, wird alles was darin enthalten ist, dupliziert. Das beinhaltet Dateisystemeigenschaften, Dateien, Verzeichnisse, Rechte und so weiter. Schnappschüsse benötigen keinen zusätzlichen Speicherplatz wenn diese erstmals angelegt werden, nur wenn Blöcke, die diese referenzieren, geändert werden. Rekursive Schnappschüsse, die mit der Option `-r` erstellt, erzeugen einen mit dem gleichen Namen des Datasets und all seinen Kindern, was eine konsistente Momentaufnahme aller Dateisysteme darstellt. Dies kann wichtig sein, wenn eine Anwendung Dateien auf mehreren Datasets ablegt, die miteinander in Verbindung stehen oder voneinander abhängig sind. Ohne Schnappschüsse würde ein Backup Kopien dieser Dateien zu unterschiedlichen Zeitpunkten enthalten.

Schnappschüsse in ZFS bieten eine Vielzahl von Eigenschaften, die selbst in anderen Dateisystemen mit Schnappschussfunktion nicht vorhanden sind. Ein typisches Beispiel zur Verwendung von Schnappschüssen ist, den momentanen Stand des Dateisystems zu sichern, wenn eine riskante Aktion wie das Installieren von Software oder eine Systemaktualisierung durchgeführt wird. Wenn diese Aktion fehlschlägt, so kann der Schnappschuss zurückgerollt werden und das System befindet sich wieder in dem gleichen Zustand, wie zu dem, als der Schnappschuss erstellt wurde. Wenn die Aktualisierung jedoch erfolgreich war, kann der Schnappschuss gelöscht werden, um Speicherplatz frei zu geben. Ohne Schnappschüsse, wird durch ein fehlgeschlagenes Update eine Wiederherstellung der Sicherung fällig, was oft mühsam und zeitaufwändig ist, außerdem ist währenddessen das System nicht verwendbar. Schnappschüsse lassen sich schnell und mit wenig bis gar keiner Ausfallzeit zurückrollen, selbst wenn das System im normalen Betrieb läuft. Die Zeitersparnis ist enorm, wenn mehrere Terabyte große Speichersysteme eingesetzt werden und viel Zeit für das Kopieren der Daten vom Sicherungssystem benötigt wird. Schnappschüsse sind jedoch keine Ersatz für eine Vollsicherung des Pools, können jedoch als eine schnelle und einfache Sicherungsmethode verwendet werden, um eine Kopie eines Datasets zu einem bestimmten Zeitpunkt zu sichern.

#### Schnappschüsse erstellen

Schnappschüsse werden durch das Kommando `zfs snapshot dataset@snapshotname` angelegt. Durch Angabe der Option `-r` werden Schnappschüsse rekursive angelegt, mit dem gleichen Namen auf allen Datasets.

Einen rekursiven Schnappschuss des gesamten Pools erzeugen:

```
# zfs list -t all
NAME                                 USED  AVAIL  REFER  MOUNTPOINT
mypool                               780M  93.2G   144K  none
mypool/ROOT                          777M  93.2G   144K  none
mypool/ROOT/default                  777M  93.2G   777M  /
mypool/tmp                           176K  93.2G   176K  /tmp
```

```
mypool/usr                              616K  93.2G   144K  /usr
mypool/usr/home                         184K  93.2G   184K  /usr/home
mypool/usr/ports                        144K  93.2G   144K  /usr/ports
mypool/usr/src                          144K  93.2G   144K  /usr/src
mypool/var                             1.29M  93.2G   616K  /var
mypool/var/crash                        148K  93.2G   148K  /var/crash
mypool/var/log                          178K  93.2G   178K  /var/log
mypool/var/mail                         144K  93.2G   144K  /var/mail
mypool/var/newname                     87.5K  93.2G  87.5K  /var/newname
mypool/var/newname@new_snapshot_name       0      -  87.5K  -
mypool/var/tmp                          152K  93.2G   152K  /var/tmp
# zfs snapshot -r mypool@my_recursive_snapshot
# zfs list -t snapshot
NAME                                        USED  AVAIL  REFER  MOUNTPOINT
mypool@my_recursive_snapshot                   0      -   144K  -
mypool/ROOT@my_recursive_snapshot              0      -   144K  -
mypool/ROOT/default@my_recursive_snapshot      0      -   777M  -
mypool/tmp@my_recursive_snapshot               0      -   176K  -
mypool/usr@my_recursive_snapshot               0      -   144K  -
mypool/usr/home@my_recursive_snapshot          0      -   184K  -
mypool/usr/ports@my_recursive_snapshot         0      -   144K  -
mypool/usr/src@my_recursive_snapshot           0      -   144K  -
mypool/var@my_recursive_snapshot               0      -   616K  -
mypool/var/crash@my_recursive_snapshot         0      -   148K  -
mypool/var/log@my_recursive_snapshot           0      -   178K  -
mypool/var/mail@my_recursive_snapshot          0      -   144K  -
mypool/var/newname@new_snapshot_name           0      -  87.5K  -
mypool/var/newname@my_recursive_snapshot       0      -  87.5K  -
mypool/var/tmp@my_recursive_snapshot           0      -   152K  -
```

Schnappschüsse werden nicht durch einen `zfs list`-Befehl angezeigt. Um Schnappschüsse mit aufzulisten, muss `-t snapshot` an das Kommando `zfs list` angehängt werden. Durch `-t all` werden sowohl Dateisysteme als auch Schnappschüsse nebeneinander angezeigt.

Schnappschüsse werden nicht direkt eingehängt, deshalb wird auch kein Pfad in der Spalte `MOUNTPOINT` angezeigt. Ebenso wird kein freier Speicherplatz in der Spalte `AVAIL` aufgelistet, da Schnappschüsse nicht mehr geschrieben werden können, nachdem diese angelegt wurden. Vergleichen Sie den Schnappschuss mit dem ursprünglichen Dataset von dem es abstammt:

```
# zfs list -rt all mypool/usr/home
NAME                                    USED  AVAIL  REFER  MOUNTPOINT
mypool/usr/home                         184K  93.2G   184K  /usr/home
mypool/usr/home@my_recursive_snapshot      0      -   184K  -
```

Durch das Darstellen des Datasets und des Schnappschusses nebeneinander zeigt deutlich, wie Schnappschüsse in COW Manier funktionieren. Sie zeichnen nur die Änderungen (*delta*) auf, die währenddessen entstanden sind und nicht noch einmal den gesamten Inhalt des Dateisystems. Das bedeutet, dass Schnappschüsse nur wenig Speicherplatz benötigen, wenn nur kleine Änderungen vorgenommen werden. Der Speicherverbrauch kann sogar noch deutlicher gemacht werden, wenn

eine Datei auf das Dataset kopiert wird und anschließend ein zweiter Schnappschuss angelegt wird:

```
# cp /etc/passwd /var/tmp
# zfs snapshot mypool/var/tmp@after_cp
# zfs list -rt all mypool/var/tmp
NAME                                  USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp                        206K  93.2G   118K  /var/tmp
mypool/var/tmp@my_recursive_snapshot   88K      -   152K  -
mypool/var/tmp@after_cp                 0      -   118K  -
```

Der zweite Schnappschuss enthält nur die Änderungen am Dataset, die nach der Kopieraktion gemacht wurden. Dies bedeutet enorme Einsparungen von Speicherplatz. Beachten Sie, dass sich die Größe des Schnappschusses `mypool/var/tmp@my_recursive_snapshot` in der Spalte `USED` ebenfalls geändert hat, um die Änderungen von sich selbst und dem Schnappschuss, der im Anschluss angelegt wurde, anzuzeigen.

### Schnappschüsse vergleichen

ZFS enthält ein eingebautes Kommando, um die Unterschiede zwischen zwei Schnappschüssen miteinander zu vergleichen. Das ist hilfreich, wenn viele Schnappschüsse über längere Zeit angelegt wurden und der Benutzer sehen will, wie sich das Dateisystem über diesen Zeitraum verändert hat. Beispielsweise kann `zfs diff` den letzten Schnappschuss finden, der noch eine Datei enthält, die aus Versehen gelöscht wurde. Wenn dies für die letzten beiden Schnappschüsse aus dem vorherigen Abschnitt durchgeführt wird, ergibt sich folgende Ausgabe:

```
# zfs list -rt all mypool/var/tmp
NAME                                  USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp                        206K  93.2G   118K  /var/tmp
mypool/var/tmp@my_recursive_snapshot   88K      -   152K  -
mypool/var/tmp@after_cp                 0      -   118K  -
# zfs diff mypool/var/tmp@my_recursive_snapshot
M       /var/tmp/
+       /var/tmp/passwd
```

Das Kommando zeigt alle Änderungen zwischen dem angegebenen Schnappschuss (in diesem Fall `mypool/var/tmp@my_recursive_snapshot`) und dem momentan aktuellen Dateisystem. Die erste Spalte zeigt die Art der Änderung an:

| | |
|---|---|
| + | Das Verzeichnis oder die Datei wurde hinzugefügt. |
| - | Das Verzeichnis oder die Datei wurde gelöscht. |
| M | Das Verzeichnis oder die Datei wurde geändert. |
| R | Das Verzeichnis oder die Datei wurde umbenannt. |

Vergleicht man die Ausgabe mit der Tabelle, wird klar, dass passwd hinzugefügt wurde, nachdem der Schnappschuss `mypool/var/tmp@my_recursive_snapshot` erstellt wurde. Das resultierte ebenfalls in einer Änderung am darüberliegenden Verzeichnis, das unter `/var/tmp` eingehängt ist.

Zwei Schnappschüsse zu vergleichen ist hilfreich, wenn die Replikationseigenschaft von ZFS verwendet wird, um ein Dataset auf einen anderen Host zu Sicherungszwecken übertragen.

Zwei Schnappschüsse durch die Angabe des kompletten Namens des Datasets und dem Namen des Schnappschusses beider Datasets vergleichen:

```
# cp /var/tmp/passwd /var/tmp/passwd.copy
# zfs snapshot mypool/var/tmp@diff_snapshot
# zfs diff mypool/var/tmp@my_recursive_snapshot mypool/var/tmp@diff_snapshot
M       /var/tmp/
+       /var/tmp/passwd
+       /var/tmp/passwd.copy
# zfs diff mypool/var/tmp@my_recursive_snapshot mypool/var/tmp@after_cp
M       /var/tmp/
+       /var/tmp/passwd
```

Ein Administrator, der für die Sicherung zuständig ist, kann zwei Schnappschüsse miteinander vergleichen, die vom sendenden Host empfangen wurden, um festzustellen, welche Änderungen am Dataset vorgenommen wurden. Lesen Sie dazu den Abschnitt Replication um weitere Informationen zu erhalten.

#### Schnappschüsse zurückrollen

Wenn zumindest ein Schnappschuss vorhanden ist, kann dieser zu einem beliebigen Zeitpunkt zurückgerollt werden. In den meisten Fällen passiert dies, wenn der aktuelle Zustand des Datasets nicht mehr benötigt wird und eine ältere Version bevorzugt wird. Szenarien wie lokale Entwicklungstests, die fehlgeschlagen sind, defekte Systemaktualisierungen, welche die Funktionalität des Gesamtsystems einschränken oder die Anforderung, versehentlich gelöschte Dateien oder Verzeichnisse wiederherzustellen, sind allgegenwärtig. Glücklicherweise ist das zurückrollen eines Schnappschusses so leicht wie die Eingabe von `zfs rollback snapshotname`. Abhängig davon, wie viele Änderungen betroffen sind, wird diese Operation innerhalb einer gewissen Zeit abgeschlossen sein. Während dieser Zeit bleibt das Dataset in einem konsistenten Zustand, sehr ähnlich den ACID-Prinzipien, die eine Datenbank beim Zurückrollen entspricht. Während all dies passiert, ist das Dataset immer noch aktiv und erreichbar ohne dass eine Ausfallzeit nötig wäre. Sobald der Schnappschuss zurückgerollt wurde, besitzt das Dataset den gleichen Zustand, den es besaß, als der Schnappschuss angelegt wurde. Alle anderen Daten in diesem Dataset, die nicht Teil des Schnappschusses sind, werden verworfen. Einen Schnappschuss des aktuellen Zustandes des Datasets vor dem Zurückrollen anzulegen ist eine gute Idee, wenn hinterher noch Daten benötigt werden. Auf diese Weise kann der Benutzer vor und zurück zwischen den Schnappschüssen springen, ohne wertvolle Daten zu verlieren.

Im ersten Beispiel wird ein Schnappschuss aufgrund eines unvorsichtigen `rm`-Befehls zurückgerollt, der mehr Daten gelöscht hat, als vorgesehen.

```
# zfs list -rt all mypool/var/tmp
NAME                                 USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp                       262K  93.2G   120K  /var/tmp
mypool/var/tmp@my_recursive_snapshot    88K      -   152K  -
```

```
mypool/var/tmp@after_cp               53.5K      -   118K  -
mypool/var/tmp@diff_snapshot              0      -   120K  -
# ls /var/tmp
passwd          passwd.copy     vi.recover
# rm /var/tmp/passwd*
# ls /var/tmp
vi.recover
#
```

Zu diesem Zeitpunkt bemerkt der Benutzer, dass zuviele Dateien gelöscht wurden und möchte diese zurück haben. ZFS bietet eine einfache Möglichkeit, diese durch zurückrollen zurück zu bekommen, allerdings nur, wenn Schnappschüsse von wichtigen Daten regelmäßig angelegt werden. Um die Dateien zurückzuerhalten und vom letzten Schnappschuss wieder zu beginnen, geben Sie ein:

```
# zfs rollback mypool/var/tmp@diff_snapshot
# ls /var/tmp
passwd          passwd.copy     vi.recover
```

Die Operation zum Zurückrollen versetzt das Dataset in den Zustand des letzten Schnappschusses zurück. Es ist ebenfalls möglich, zu einem Schnappschuss zurückzurollen, der viel früher angelegt wurde und es noch Schnappschüsse nach diesem gibt. Wenn Sie dies versuchen, gibt ZFS die folgende Warnung aus:

```
# zfs list -rt snapshot mypool/var/tmp
AME                                     USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp@my_recursive_snapshot     88K      -   152K  -
mypool/var/tmp@after_cp               53.5K      -   118K  -
mypool/var/tmp@diff_snapshot              0      -   120K  -
# zfs rollback mypool/var/tmp@my_recursive_snapshot
cannot rollback to 'mypool/var/tmp@my_recursive_snapshot': more recent snapshots exist
use '-r' to force deletion of the following snapshots:
mypool/var/tmp@after_cp
mypool/var/tmp@diff_snapshot
```

Diese Warnung bedeutet, dass noch Schnappschüsse zwischen dem momentanen Stand des Datasets und dem Schnappschuss, zu dem der Benutzer zurückrollen möchte, existieren. Um das Zurückrollen durchzuführen, müssen die Schnappschüsse gelöscht werden. ZFS kann nicht alle Änderungen zwischen verschiedenen Zuständen eines Datasets verfolgen, da Schnappschüsse nur gelesen werden können. ZFS wird nicht die betroffenen Schnappschüsse löschen, es sei denn, der Benutzer verwendet die Option `-r`, um anzugeben, dass dies die gewünschte Aktion ist. Falls dies der Fall ist und die Konsequenzen alle dazwischenliegenden Schnappschüsse zu verlieren verstanden wurden, kann der Befehl abgesetzt werden:

```
# zfs rollback -r mypool/var/tmp@my_recursive_snapshot
# zfs list -rt snapshot mypool/var/tmp
```

```
NAME                                    USED  AVAIL  REFER  MOUNTPOINT
mypool/var/tmp@my_recursive_snapshot     8K      -   152K  -
# ls /var/tmp
vi.recover
```

Die Ausgabe von `zfs list -t snapshot` bestätigt, dass die dazwischenliegenden Schnappschüsse als Ergebnis von `zfs rollback -r` entfernt wurden.

#### Individuelle Dateien aus Schnappschüssen wiederherstellen

Schnappschüsse sind unter einem versteckten Verzeichnis unter dem Eltern-Dataset eingehängt: .zfs/snapshots/snapshotname. Standardmäßig werden diese Verzeichnisse nicht von einem gewöhnlichen `ls -a` angezeigt. Obwohl diese Verzeichnisse nicht angezeigt werden, sind diese trotzdem vorhanden und der Zugriff darauf erfolgt wie auf jedes andere Verzeichnis. Die Eigenschaft `snapdir` steuert, ob diese Verzeichnisse beim Auflisten eines Verzeichnisses angezeigt werden oder nicht. Das Einstellen der Eigenschaft auf den Wert `visible` erlaubt es, diese in der Ausgabe von `ls` und anderen Kommandos, die mit Verzeichnisinhalten umgehen können, anzuzeigen.

```
# zfs get snapdir mypool/var/tmp
NAME            PROPERTY  VALUE    SOURCE
mypool/var/tmp  snapdir   hidden   default
# ls -a /var/tmp
.               ..              passwd          vi.recover
# zfs set snapdir=visible mypool/var/tmp
# ls -a /var/tmp
.               ..              .zfs            passwd          vi.recover
```

Einzelne Dateien lassen sich einfach auf einen vorherigen Stand wiederherstellen, indem diese aus dem Schnappschuss zurück in das Eltern-Dataset kopiert werden. Die Verzeichnisstruktur unterhalb von .zfs/snapshot enthält ein Verzeichnis, das exakt wie der Schnappschuss benannt ist, der zuvor angelegt wurde, um es einfacher zu machen, diese zu identifizieren. Im nächsten Beispiel wird angenommen, dass eine Datei aus dem versteckten .zfs Verzeichnis durch kopieren aus dem Schnappschuss, der die letzte Version dieser Datei enthielt, wiederhergestellt wird:

```
# rm /var/tmp/passwd
# ls -a /var/tmp
.               ..              .zfs            vi.recover
# ls /var/tmp/.zfs/snapshot
after_cp                my_recursive_snapshot
# ls /var/tmp/.zfs/snapshot/after_cp
passwd          vi.recover
# cp /var/tmp/.zfs/snapshot/after_cp/passwd /var/tmp
```

Als `ls .zfs/snapshot` ausgeführt wurde, war die `snapdir`-Eigenschaft möglicherweise nicht auf hidden gesetzt, trotzdem ist es immer noch möglich, den Inhalt dieses Verzeichnisses aufzulisten. Es liegt am Administrator zu entscheiden, ob diese Verzeichnisse angezeigt werden soll. Es ist

möglich, diese für bestimmte Datasets anzuzeigen und für andere zu verstecken. Das Kopieren von Dateien oder Verzeichnissen aus diesem versteckten .zfs/snapshot Verzeichnis ist einfach genug. Jedoch führt der umgekehrte Weg zu einem Fehler:

```
# cp /etc/rc.conf /var/tmp/.zfs/snapshot/after_cp/
cp: /var/tmp/.zfs/snapshot/after_cp/rc.conf: Read-only file system
```

Der Fehler erinnert den Benutzer daran, dass Schnappschüsse nur gelesen aber nicht mehr geändert werden können, nachdem diese angelegt wurden. Es können keine Dateien in diese Schnappschuss-Verzeichnisse kopiert oder daraus gelöscht werden, da dies sonst den Zustand des Datasets verändern würde, den sie repräsentieren.

Schnappschüsse verbrauchen Speicherplatz basierend auf der Menge an Änderungen, die am Eltern-Dataset durchgeführt wurden, seit der Zeit als der Schnappschuss erstellt wurde. Die Eigenschaft `written` eines Schnappschusses verfolgt, wieviel Speicherplatz vom Schnappschuss belegt wird.

Schnappschüsse werden zerstört und der belegte Platz wieder freigegeben durch den Befehl `zfs destroy dataset@snapshot`. Durch hinzufügen von `-r` werden alle Schnappschüsse rekursiv gelöscht, die den gleichen Namen wie das Eltern-Dataset besitzen. Mit der Option `-n -v` wird eine Liste von Schnappschüssen, die gelöscht werden würden, zusammen mit einer geschätzten Menge an zurückgewonnenem Speicherplatz angezeigt, ohne die eigentliche Zerstöroperation wirklich durchzuführen.

### 37.4.6. Klone verwalten

Ein Klon ist eine Kopie eines Schnappschusses, der mehr wie ein reguläres Dataset behandelt wird. Im Gegensatz zu Schnappschüssen kann man von einem Klon nicht nur lesen, er ist eingehängt und kann seine eigenen Eigenschaften haben. Sobald ein Klon mittels `zfs clone` erstellt wurde, lässt sich der zugrundeliegende Schnappschuss nicht mehr zerstören. Die Eltern-/Kindbeziehung zwischen dem Klon und dem Schnappschuss kann über `zfs promote` aufgelöst werden. Nachdem ein Klon auf diese Weise befördert wurde, wird der Schnappschuss zum Kind des Klons, anstatt des ursprünglichen Datasets. Dies wird die Art und Weise, wie der Speicherplatz berechnet wird, verändern, jedoch nicht den bereits belegten Speicher anpassen. Der Klon kann an einem beliebigen Punkt innerhalb der ZFS-Dateisystemhierarchie eingehängt werden, nur nicht unterhalb der ursprünglichen Stelle des Schnappschusses.

Um diese Klon-Funktionalität zu demonstrieren, wird dieses Beispiel-Dataset verwendet:

```
# zfs list -rt all camino/home/joe
NAME                    USED  AVAIL  REFER  MOUNTPOINT
camino/home/joe         108K   1.3G    87K  /usr/home/joe
camino/home/joe@plans    21K      -  85.5K  -
camino/home/joe@backup    0K      -    87K  -
```

Ein typischer Einsatzzweck für Klone ist das experimentieren mit einem bestimmten Dataset, während der Schnappschuss beibehalten wird für den Fall, dass etwas schiefgeht. Da

Schnappschüsse nicht verändert werden können, wird ein Lese-/Schreibklon des Schnappschusses angelegt. Nachdem das gewünschte Ergebnis im Klon erreicht wurde, kann der Klon zu einem Dataset ernannt und das alte Dateisystem entfernt werden. Streng genommen ist das nicht nötig, da der Klon und das Dataset ohne Probleme miteinander koexistieren können.

```
# zfs clone camino/home/joe@backup camino/home/joenew
# ls /usr/home/joe*
/usr/home/joe:
backup.txz     plans.txt

/usr/home/joenew:
backup.txz     plans.txt
# df -h /usr/home
Filesystem          Size    Used   Avail Capacity  Mounted on
usr/home/joe        1.3G     31k    1.3G     0%    /usr/home/joe
usr/home/joenew     1.3G     31k    1.3G     0%    /usr/home/joenew
```

Nachdem ein Klon erstellt wurde, stellt er eine exakte Kopie des Datasets zu dem Zeitpunkt dar, als der Schnappschuss angelegt wurde. Der Klon kann nun unabhängig vom ursprünglichen Dataset geändert werden. Die einzige Verbindung zwischen den beiden ist der Schnappschuss. ZFS zeichnet diese Verbindung in der Eigenschaft namens `origin` auf. Sobald die Abhängigkeit zwischen dem Schnappschuss und dem Klon durch das Befördern des Klons mittels `zfs promote` entfernt wurde, wird auch die `origin`-Eigenschaft des Klons entfernt, da es sich nun um ein eigenständiges Dataset handelt. Dieses Beispiel demonstriert dies:

```
# zfs get origin camino/home/joenew
NAME                 PROPERTY  VALUE                    SOURCE
camino/home/joenew   origin    camino/home/joe@backup   -
# zfs promote camino/home/joenew
# zfs get origin camino/home/joenew
NAME                 PROPERTY  VALUE   SOURCE
camino/home/joenew   origin    -       -
```

Nachdem ein paar Änderungen, wie beispielsweise das Kopieren von loader.conf auf den beförderten Klon vorgenommen wurden, wird das alte Verzeichnis in diesem Fall überflüssig. Stattdessen kann der beförderte Klon diesen ersetzen. Dies kann durch zwei aufeinanderfolgende Befehl geschehen: `zfs destroy` auf dem alten Dataset und `zfs rename` auf dem Klon, um diesen genauso wie das alte Dataset zu benennen (es kann auch einen ganz anderen Namen erhalten).

```
# cp /boot/defaults/loader.conf /usr/home/joenew
# zfs destroy -f camino/home/joe
# zfs rename camino/home/joenew camino/home/joe
# ls /usr/home/joe
backup.txz     loader.conf     plans.txt
# df -h /usr/home
Filesystem          Size    Used   Avail Capacity  Mounted on
```

```
usr/home/joe        1.3G    128k    1.3G     0%    /usr/home/joe
```

Der geklonte Schnappschuss wird jetzt wie ein gewöhnliches Dataset behandelt. Es enthält alle Daten aus dem ursprünglichen Schnappschuss inklusive der Dateien, die anschließend hinzugefügt wurden, wie loader.conf. Klone können in unterschiedlichen Szenarien eingesetzt werden, um nützliche Eigenschaften für ZFS-Anwender zur Verfügung zu stellen. Zum Beispiel können Jails als Schnappschüsse bereitgestellt werden, die verschiedene Arten von installierten Anwendungen anbieten. Anwender können diese Schnappschüsse klonen und ihre eigenen Anwendungen nach Belieben hinzufügen. Sobald sie mit den Änderungen zufrieden sind, können die Klone zu vollständigen Datasets ernannt werden und dem Anwender zur Verfügung gestellt werden, als würde es sich um echte Datasets handeln. Das spart Zeit und Administrationsaufwand, wenn diese Jails auf diese Weise zur Verfügung gestellt werden.

### 37.4.7. Replikation

Daten auf einem einzigen Pool an einem Platz aufzubewahren, setzt diese dem Risiko aus, gestohlen oder Opfer von Naturgewalten zu werden, sowie menschlichem Versagen auszusetzen. Regelmäßige Sicherungen des gesamten Pools ist daher unerlässlich. ZFS bietet eine Reihe von eingebauten Serialisierungsfunktionen an, die in der Lage ist, eine Repräsentation der Daten als Datenstrom auf die Standardausgabe zu schreiben. Mit dieser Methode ist es nicht nur möglich, die Daten auf einen anderen Pool zu schicken, der an das lokale System angeschlossen ist, sondern ihn auch über ein Netzwerk an ein anderes System zu senden. Schnappschüsse stellen dafür die Replikationsbasis bereit (lesen Sie dazu den Abschnitt zu ZFS snapshots). Die Befehle, die für die Replikation verwendet werden, sind `zfs send` und `zfs receive`.

Diese Beispiele demonstrieren die Replikation von ZFS anhand dieser beiden Pools:

```
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG   CAP  DEDUP  HEALTH  ALTROOT
backup  960M    77K   896M         -         -     0%    0%  1.00x  ONLINE  -
mypool  984M  43.7M   940M         -         -     0%    4%  1.00x  ONLINE  -
```

Der Pool namens *mypool* ist der primäre Pool, auf den regelmäßig Daten geschrieben und auch wieder gelesen werden. Ein zweiter Pool, genannt *backup* wird verwendet, um als Reserve zu dienen im Falle, dass der primäre Pool nicht zur Verfügung steht. Beachten Sie, dass diese Ausfallsicherung nicht automatisch von ZFS durchgeführt wird, sondern manuell von einem Systemadministrator bei Bedarf eingerichtet werden muss. Ein Schnappschuss wird verwendet, um einen konsistenten Zustand des Dateisystems, das repliziert werden soll, zu erzeugen. Sobald ein Schnappschuss von *mypool* angelegt wurde, kann er auf den *backup*-Pool abgelegt werden. Nur Schnappschüsse lassen sich auf diese Weise replizieren. Änderungen, die seit dem letzten Schnappschuss entstanden sind, werden nicht mit repliziert.

```
# zfs snapshot mypool@backup1
# zfs list -t snapshot
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool@backup1             0      -  43.6M  -
```

Da nun ein Schnappschuss existiert, kann mit `zfs send` ein Datenstrom, der den Inhalt des Schnappschusses repräsentiert, erstellt werden. Dieser Datenstrom kann als Datei gespeichert oder von einem anderen Pool empfangen werden. Der Datenstrom wird auf die Standardausgabe geschrieben, muss jedoch in eine Datei oder in eine Pipe umgeleitet werden, sonst wird ein Fehler produziert:

```
# zfs send mypool@backup1
Error: Stream can not be written to a terminal.
You must redirect standard output.
```

Um ein Dataset mit `zfs send` zu replizieren, leiten Sie dieses in eine Datei auf dem eingehängten Backup-Pool um. Stellen Sie sicher, dass der Pool genug freien Speicherplatz besitzt, um die Größe des gesendeten Schnappschusses aufzunehmen. Das beinhaltet alle Daten im Schnappschuss, nicht nur die Änderungen zum vorherigen Schnappschuss.

```
# zfs send mypool@backup1 > /backup/backup1
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  63.7M   896M         -         -     0%     6%  1.00x  ONLINE  -
mypool  984M  43.7M   940M         -         -     0%     4%  1.00x  ONLINE  -
```

Das Kommando `zfs send` transferierte alle Daten im *backup1*-Schnappschuss auf den Pool namens *backup*. Erstellen und senden eines Schnappschusses kann automatisch von cron(8) durchgeführt werden.

Anstatt die Sicherungen als Archivdateien zu speichern, kann ZFS diese auch als aktives Dateisystem empfangen, was es erlaubt, direkt auf die gesicherten Daten zuzugreifen. Um an die eigentlichen Daten in diesem Strom zu gelangen, wird `zfs receive` benutzt, um den Strom wieder in Dateien und Verzeichnisse umzuwandeln. Das Beispiel unten kombiniert `zfs send` und `zfs receive` durch eine Pipe, um die Daten von einem Pool auf den anderen zu kopieren. Die Daten können direkt auf dem empfangenden Pool verwendet werden, nachdem der Transfer abgeschlossen ist. Ein Dataset kann nur auf ein leeres Dataset repliziert werden.

```
# zfs snapshot mypool@replica1
# zfs send -v mypool@replica1 | zfs receive backup/mypool
send from @ to mypool@replica1 estimated size is 50.1M
total estimated size is 50.1M
TIME        SENT   SNAPSHOT

# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  63.7M   896M         -         -     0%     6%  1.00x  ONLINE  -
mypool  984M  43.7M   940M         -         -     0%     4%  1.00x  ONLINE  -
```

### Inkrementelle Sicherungen

Die Unterschiede zwischen zwei Schnappschüssen kann `zfs send` ebenfalls erkennen und nur diese übertragen. Dies spart Speicherplatz und Übertragungszeit. Beispielsweise:

```
# zfs snapshot mypool@replica2
# zfs list -t snapshot
NAME                    USED  AVAIL  REFER  MOUNTPOINT
mypool@replica1         5.72M      -  43.6M  -
mypool@replica2             0      -  44.1M  -
# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  61.7M   898M         -         -     0%     6%  1.00x  ONLINE  -
mypool  960M  50.2M   910M         -         -     0%     5%  1.00x  ONLINE  -
```

Ein zweiter Schnappschuss genannt *replica2* wurde angelegt. Dieser zweite Schnappschuss enthält nur die Änderungen, die zwischen dem jetzigen Stand des Dateisystems und dem vorherigen Schnappschuss, *replica1*, vorgenommen wurden. Durch `zfs send -i` und die Angabe des Schnappschusspaares wird ein inkrementeller Replikationsstrom erzeugt, welcher nur die Daten enthält, die sich geändert haben. Das kann nur erfolgreich sein, wenn der initiale Schnappschuss bereits auf der Empfängerseite vorhanden ist.

```
# zfs send -v -i mypool@replica1 mypool@replica2 | zfs receive /backup/mypool
send from @replica1 to mypool@replica2 estimated size is 5.02M
total estimated size is 5.02M
TIME        SENT   SNAPSHOT

# zpool list
NAME    SIZE  ALLOC   FREE   CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
backup  960M  80.8M   879M         -         -     0%     8%  1.00x  ONLINE  -
mypool  960M  50.2M   910M         -         -     0%     5%  1.00x  ONLINE  -

# zfs list
NAME                         USED  AVAIL  REFER  MOUNTPOINT
backup                      55.4M   240G   152K  /backup
backup/mypool               55.3M   240G  55.2M  /backup/mypool
mypool                      55.6M  11.6G  55.0M  /mypool

# zfs list -t snapshot
NAME                                    USED  AVAIL  REFER  MOUNTPOINT
backup/mypool@replica1                  104K      -  50.2M  -
backup/mypool@replica2                     0      -  55.2M  -
mypool@replica1                         29.9K      -  50.0M  -
mypool@replica2                            0      -  55.0M  -
```

Der inkrementelle Datenstrom wurde erfolgreich übertragen. Nur die Daten, die verändert wurden, sind übertragen worden, anstatt das komplette *replica1*. Nur die Unterschiede wurden gesendet, was weniger Zeit und Speicherplatz in Anspruch genommen hat, statt jedesmal den gesamten Pool

zu kopieren. Das ist hilfreich wenn langsame Netzwerke oder Kosten für die übertragene Menge Bytes in Erwägung gezogen werden müssen.

Ein neues Dateisystem, *backup/mypool*, ist mit allen Dateien und Daten vom Pool *mypool* verfügbar. Wenn die Option `-P` angegeben wird, werden die Eigenschaften des Datasets kopiert, einschließlich der Komprimierungseinstellungen, Quotas und Einhängepunkte. Wird die Option `-R` verwendet, so werden alle Kind-Datasets des angegebenen Datasets kopiert, zusammen mit ihren Eigenschaften. Senden und Empfangen kann automatisiert werden, so dass regelmäßig Sicherungen auf dem zweiten Pool angelegt werden.

#### Sicherungen verschlüsselt über SSH senden

Datenströme über das Netzwerk zu schicken ist eine gute Methode, um Sicherungen außerhalb des Systems anzulegen. Jedoch ist dies auch mit einem Nachteil verbunden. Daten, die über die Leitung verschickt werden, sind nicht verschlüsselt, was es jedem erlaubt, die Daten abzufangen und die Ströme wieder zurück in Daten umzuwandeln, ohne dass der sendende Benutzer davon etwas merkt. Dies ist eine unerwünschte Situation, besonders wenn die Datenströme über das Internet auf ein entferntes System gesendet werden. SSH kann benutzt werden, um durch Verschlüsselung geschützte Daten über eine Netzwerkverbindung zu übertragen. Da ZFS nur die Anforderung hat, dass der Strom von der Standardausgabe umgeleitet wird, ist es relativ einfach, diesen durch SSH zu leiten. Um den Inhalt des Dateisystems während der Übertragung und auf dem entfernten System weiterhin verschlüsselt zu lassen, denken Sie über den Einsatz von PEFS nach.

Ein paar Einstellungen und Sicherheitsvorkehrungen müssen zuvor abgeschlossen sein. Es werden hier nur die nötigen Schritte für die `zfs send`-Aktion gezeigt. Weiterführende Informationen zu SSH, gibt es im Kapitel OpenSSH.

Die folgende Konfiguration wird benötigt:

- Passwortloser SSH-Zugang zwischen dem sendenden und dem empfangenden Host durch den Einsatz von SSH-Schlüsseln.
- Normalerweise werden die Privilegien des `root`-Benutzers gebraucht, um Strom zu senden und zu empfangen. Das beinhaltet das Anmelden auf dem empfangenden System als `root`. Allerdings ist das Anmelden als `root` aus Sicherheitsgründen standardmäßig deaktiviert. Mit ZFS Delegation lassen sich nicht-`root`-Benutzer auf jedem System einrichten, welche die nötigen Rechte besitzen, um die Sende- und Empfangsoperation durchzuführen.
- Auf dem sendenden System:

```
# zfs allow -u someuser send,snapshot mypool
```

- Um den Pool einzuhängen, muss der unprivilegierte Benutzer das Verzeichnis besitzen und gewöhnliche Benutzern muss die Erlaubnis gegeben werden, das Dateisystem einzuhängen. Auf dem empfangenden System nehmen Sie dazu die folgenden Einstellungen vor:

```
# sysctl vfs.usermount=1
vfs.usermount: 0 -> 1
# echo vfs.usermount=1 >> /etc/sysctl.conf
```

```
# zfs create recvpool/backup
# zfs allow -u someuser create,mount,receive recvpool/backup
# chown someuser /recvpool/backup
```

Der unprivilegierte Benutzer hat jetzt die Fähigkeit, Datasets zu empfangen und einzuhängen und das *home*-Dataset auf das entfernte System zu replizieren:

```
% zfs snapshot -r mypool/home@monday
% zfs send -R mypool/home@monday | ssh someuser@backuphost zfs recv -dvu
recvpool/backup
```

Ein rekursiver Schnappschuss namens *monday* wird aus dem Dataset *home* erstellt, dass auf dem Pool *mypool* liegt. Es wird dann mit `zfs send -R` gesendet, um das Dataset, alle seine Kinder, Schnappschüsse, Klone und Einstellungen in den Strom mit aufzunehmen. Die Ausgabe wird an das wartende System *backuphost* mittels `zfs receive` durch SSH umgeleitet. Die Verwendung des Fully Qulified Domänennamens oder der IP-Adresse wird empfohlen. Die empfangende Maschine schreibt die Daten auf das *backup*-Dataset im *recvpool*-Pool. Hinzufügen der Option `-d` zu `zfs recv` überschreibt den Namen des Pools auf der empfangenden Seite mit dem Namen des Schnappschusses. Durch Angabe von `-u` wird das Dateisystem nicht auf der Empfängerseite eingehängt. Wenn `-v` enthalten ist, werden mehr Details zum Transfer angezeigt werden, einschließlich der vergangenen Zeit und der Menge an übertragenen Daten.

### 37.4.8. Dataset-, Benutzer- und Gruppenquotas

Dataset-Quotas werden eingesetzt, um den Speicherplatz einzuschränken, den ein bestimmtes Dataset verbrauchen kann. Referenz-Quotas funktionieren auf eine ähnliche Weise, jedoch wird dabei der Speicherplatz des Datasets selbst gezählt, wobei Schnappschüsse und Kind-Datasets dabei ausgenommen sind. Ähnlich dazu werden Benutzer- und Gruppen-Quotas dazu verwendet, um Benutzer oder Gruppen daran zu hindern, den gesamten Speicherplatz im Pool oder auf dem Dataset zu verbrauchen.

Die folgenden Beispiele gehen davon aus, dass die Benutzer bereits im System vorhanden sind. Bevor Sie einen Benutzer hinzufügen, stellen Sie sicher, dass Sie zuerst ein Dataset für das Heimatverzeichnis anlegen und den `mountpoint` auf `/home/bob` festlegen. Legen Sie dann den Benutzer an und stellen Sie sicher, dass das Heimatverzeichnis auf den auf den `mountpoint` des Datasets verweist. Auf diese Weise werden die Eigentümer- und Gruppenberechtigungen richtig gesetzt, ohne dass bereits vorhandene Heimatverzeichnisse verschleiert werden.

Um ein 10 GB großes Quota auf dem Dataset storage/home/bob zu erzwingen, verwenden Sie folgenden Befehl:

```
# zfs set quota=10G storage/home/bob
```

Um ein Referenzquota von 10 GB für storage/home/bob festzulegen, geben Sie ein:

```
# zfs set refquota=10G storage/home/bob
```

Um das Quota für storage/home/bob wieder zu entfernen:

```
# zfs set quota=none storage/home/bob
```

Das generelle Format ist `userquota@user=size` und der Name des Benutzers muss in einem der folgenden Formate vorliegen:

- POSIX-kompatibler Name wie *joe*.
- POSIX-numerische ID wie *789*.
- SID-Name wie *joe.bloggs@example.com*.
- SID-numerische ID wie *S-1-123-456-789*.

Um beispielsweise ein Benutzerquota von 50 GB für den Benutzer names *joe* zu erzwingen:

```
# zfs set userquota@joe=50G
```

Um jegliche Quotas zu entfernen:

```
# zfs set userquota@joe=none
```



Benutzerquota-Eigenschaften werden nicht von `zfs get all` dargestellt. Nicht-`root`-Benutzer können nur ihre eigenen Quotas sehen, ausser ihnen wurde das `userquota`-Privileg zugeteilt. Benutzer mit diesem Privileg sind in der Lage, jedermanns Quota zu sehen und zu verändern.

Das generelle Format zum Festlegen einer Gruppenquota lautet: `groupquota@group=size`.

Um ein Quota für die Gruppe *firstgroup* von 50 GB zu setzen, geben Sie ein:

```
# zfs set groupquota@firstgroup=50G
```

Um eine Quota für die Gruppe *firstgroup* zu setzen oder sicherzustellen, dass diese nicht gesetzt ist, verwenden Sie stattdessen:

```
# zfs set groupquota@firstgroup=none
```

Genau wie mit der Gruppenquota-Eigenschaft, werden nicht-`root`-Benutzer nur die Quotas sehen, die den Gruppen zugeordnet ist, in denen Sie Mitglied sind. Allerdings ist `root` oder ein Benutzer mit dem `groupquota`-Privileg in der Lage, die Quotas aller Gruppen zu sehen und festzusetzen.

Um die Menge an Speicherplatz zusammen mit der Quota anzuzeigen, die von jedem Benutzer auf dem Dateisystem oder Schnappschuss verbraucht wird, verwenden Sie `zfs userspace`. Für Gruppeninformationen, nutzen Sie `zfs groupspace`. Für weitere Informationen zu unterstützten Optionen oder wie sich nur bestimmte Optionen anzeigen lassen, lesen Sie zfs(1).

Benutzer mit ausreichenden Rechten sowie `root` können das Quota für storage/home/bob anzeigen lassen:

```
# zfs get quota storage/home/bob
```

### 37.4.9. Reservierungen

Reservierungen garantieren ein Minimum an Speicherplatz, der immer auf dem Dataset verfügbar sein wird. Der reservierte Platz wird nicht für andere Datasets zur Verfügung stehen. Diese Eigenschaft kann besonders nützlich sein, um zu gewährleisten, dass freier Speicherplatz für ein wichtiges Dataset oder für Logdateien bereit steht.

Das generelle Format der `reservation`-Eigenschaft ist `reservation=size`. Um also eine Reservierung von 10 GB auf storage/home/bob festzulegen, geben Sie Folgendes ein:

```
# zfs set reservation=10G storage/home/bob
```

Um die Reservierung zu beseitigen:

```
# zfs set reservation=none storage/home/bob
```

Das gleiche Prinzip kann auf die `refreservation`-Eigenschaft angewendet werden, um eine Referenzreservierung mit dem generellen Format `refreservation=size` festzulegen.

Dieser Befehl zeigt die Reservierungen oder Referenzreservierungen an, die auf storage/home/bob existieren:

```
# zfs get reservation storage/home/bob
# zfs get refreservation storage/home/bob
```

### 37.4.10. Komprimierung

ZFS bietet transparente Komprimierung. Datenkomprimierung auf Blockebene während diese gerade geschrieben werden, spart nicht nur Plattenplatz ein, sondern kann auch den Durchsatz der Platte steigern. Falls Daten zu 25% komprimiert sind, jedoch die komprimierten Daten im gleichen Tempo wie ihre unkomprimierte Version, resultiert das in einer effektiven Schreibgeschwindigkeit von 125%. Komprimierung kann auch eine Alternative zu Deduplizierung darstellen, da es viel weniger zusätzlichen Hauptspeicher benötigt.

ZFS bietet mehrere verschiedene Kompressionsalgorithmen an, jede mit unterschiedlichen

Kompromissen. Mit der Einführung von LZ4-Komprimierung in ZFS v5000, ist es möglich, Komprimierung für den gesamten Pool zu aktivieren, ohne die großen Geschwindigkeitseinbußen der anderen Algorithmen. Der größte Vorteil von LZ4 ist die Eigenschaft *früher Abbruch*. Wenn LZ4 nicht mindestens 12,5% Komprimierung im ersten Teil der Daten erreicht, wird der Block unkomprimiert geschrieben, um die Verschwendung von CPU-Zyklen zu vermeiden, weil die Daten entweder bereits komprimiert sind oder sich nicht komprimieren lassen. Für Details zu den verschiedenen verfügbaren Komprimierungsalgorithmen in ZFS, lesen Sie den Eintrag Komprimierung im Abschnitt Terminologie

Der Administrator kann die Effektivität der Komprimierung über eine Reihe von Dataset-Eigenschaften überwachen.

```
# zfs get used,compressratio,compression,logicalused mypool/compressed_dataset
NAME        PROPERTY          VALUE     SOURCE
mypool/compressed_dataset  used              449G      -
mypool/compressed_dataset  compressratio     1.11x     -
mypool/compressed_dataset  compression       lz4       local
mypool/compressed_dataset  logicalused      496G      -
```

Dieses Dataset verwendet gerade 449 GB Plattenplatz (used-Eigenschaft. Ohne Komprimierung würde es stattdessen 496 GB Plattenplatz belegen (`logicalused`). Das ergibt eine Kompressionsrate von 1,11:1.

Komprimierung kann einen unerwarteten Nebeneffekt haben, wenn diese mit Benutzerquotas kombiniert wird. Benutzerquotas beschränken, wieviel Speicherplatz ein Benutzer auf einem Dataset verbrauchen kann. Jedoch basieren die Berechnungen darauf, wieviel Speicherplatz *nach der Komprimierung* belegt ist. Wenn also ein Benutzer eine Quota von10 GB besitzt und 10 GB von komprimierbaren Daten schreibt, wird dieser immer noch in der Lage sein, zusätzliche Daten zu speichern. Wenn später eine Datei aktualisiert wird, beispielsweise eine Datenbank, mit mehr oder weniger komprimierbaren Daten, wird sich die Menge an verfügbarem Speicherplatz ändern. Das kann in einer merkwürdigen Situation resultieren, in welcher der Benutzer nicht die eigentliche Menge an Daten (die Eigenschaft `logicalused`) überschreitet, jedoch die Änderung in der Komprimierung dazu führt, dass das Quota-Limit erreicht ist.

Kompression kann ebenso unerwartet mit Sicherungen interagieren. Quotas werden oft verwendet, um einzuschränken, wieviele Daten gespeichert werden können um sicherzustellen, dass ausreichend Speicherplatz für die Sicherung vorhanden ist. Wenn jedoch Quotas Komprimierung nicht berücksichtigen, werden womöglich mehr Daten geschrieben als in der unkomprimierten Sicherung Platz ist.

### 37.4.11. Deduplizierung

Wenn aktiviert, verwendet Deduplizierung die Prüfsumme jedes Blocks, um Duplikate dieses Blocks zu ermitteln. Sollte ein neuer Block ein Duplikat eines existierenden Blocks sein, dann schreibt ZFS eine zusätzliche Referenz auf die existierenden Daten anstatt des kompletten duplizierten Blocks. Gewaltige Speicherplatzeinsparungen sind möglich wenn die Daten viele Duplikate von Dateien oder wiederholte Informationen enthalten. Seien Sie gewarnt: Deduplizierung benötigt eine extrem große Menge an Hauptspeicher und die meistens Einsparungen können stattdessen durch das

Aktivieren von Komprimierung erreicht werden.

Um Deduplizierung zu aktivieren, setzen Sie die dedup-Eigenschaft auf dem Zielpool:

```
# zfs set dedup=on pool
```

Nur neu auf den Pool geschriebene Daten werden dedupliziert. Daten, die bereits auf den Pool geschrieben wurden, werden nicht durch das Aktivieren dieser Option dedupliziert. Ein Pool mit einer gerade aktivierten Deduplizierung wird wie in diesem Beispiel aussehen:

```
# zpool list
NAME  SIZE ALLOC  FREE   CKPOINT  EXPANDSZ   FRAG   CAP DEDUP HEALTH ALTROOT
pool 2.84G 2.19M 2.83G         -         -     0%    0% 1.00x ONLINE -
```

Die Spalte DEDUP zeigt das aktuelle Verhältnis der Deduplizierung für diesen Pool an. Ein Wert von 1.00x zeigt an, dass die Daten noch nicht dedupliziert wurden. Im nächsten Beispiel wird die Ports-Sammlung dreimal in verschiedene Verzeichnisse auf dem deduplizierten Pool kopiert.

```
# for d in dir1 dir2 dir3; do
> mkdir $d && cp -R /usr/ports $d &
> done
```

Redundante Daten werden erkannt und dedupliziert:

```
# zpool list
NAME SIZE  ALLOC FREE CAP DEDUP HEALTH ALTROOT
pool 2.84G 20.9M 2.82G 0% 3.00x ONLINE -
```

Die DEDUP-Spalte zeigt einen Faktor von 3.00x. Mehrere Kopien der Ports-Sammlung wurden erkannt und dedupliziert, was nur ein Drittel des Speicherplatzes benötigt. Das Potential für Einsparungen beim Speicherplatz ist enorm, wird jedoch damit erkauft, dass genügend Speicher zur Verfügung stehen muss, um die deduplizierten Blöcke zu verwalten.

Deduplizierung ist nicht immer gewinnbringend, besonders wenn die Daten auf dem Pool nicht redundant sind. ZFS kann potentielle Speicherplatzeinsparungen durch Deduplizierung auf einem Pool simulieren:

```
# zdb -S pool
Simulated DDT histogram:

bucket              allocated                       referenced
______   ______________________________   ______________________________
refcnt   blocks   LSIZE   PSIZE   DSIZE   blocks   LSIZE   PSIZE   DSIZE
------   ------   -----   -----   -----   ------   -----   -----   -----
     1    2.58M    289G    264G    264G    2.58M    289G    264G    264G
```

```
     2    206K   12.6G   10.4G   10.4G     430K   26.4G   21.6G   21.6G
     4   37.6K    692M    276M    276M     170K   3.04G   1.26G   1.26G
     8   2.18K   45.2M   19.4M   19.4M    20.0K    425M    176M    176M
    16     174   2.83M   1.20M   1.20M    3.33K   48.4M   20.4M   20.4M
    32      40   2.17M    222K    222K    1.70K   97.2M   9.91M   9.91M
    64       9     56K   10.5K   10.5K      865   4.96M    948K    948K
   128       2   9.50K      2K      2K      419   2.11M    438K    438K
   256       5   61.5K     12K     12K    1.90K   23.0M   4.47M   4.47M
    1K       2      1K      1K      1K    2.98K   1.49M   1.49M   1.49M
 Total    2.82M    303G    275G    275G    3.20M    319G    287G    287G

dedup = 1.05, compress = 1.11, copies = 1.00, dedup * compress / copies = 1.16
```

Nachdem `zdb -S` die Analyse des Pool abgeschlossen hat, zeigt es die Speicherplatzeinsparungen, die durch aktivierte Deduplizierung erreichbar sind, an. In diesem Fall ist `1.16` ein sehr schlechter Faktor, der größtenteils von Einsparungen durch Komprimierung beeinflusst wird. Aktivierung von Deduplizierung auf diesem Pool würde also keine signifikante Menge an Speicherplatz einsparen und ist daher nicht die Menge an Speicher wert, die nötig sind, um zu deduplizieren. Über die Formel *ratio = dedup * compress / copies* kann ein Systemadministrator die Speicherplatzbelegung planen und entscheiden, ob es sich lohnt, den zusätzlichen Hauptspeicher für die Deduplizierung anhand des späteren Workloads aufzuwenden. Wenn sich die Daten verhältnismäßig gut komprimieren lassen, sind die Speicherplatzeinsparungen sehr gut. Es wird empfohlen, in dieser Situation zuerst die Komprimierung zu aktivieren, da diese auch erhöhte Geschwindigkeit mit sich bringt. Aktivieren Sie Deduplizierung nur in solchen Fällen, bei denen die Einsparungen beträchtlich sind und genug Hauptspeicher zur Verfügung steht, um die DDT aufzunehmen.

### 37.4.12. ZFS und Jails

Um ein ZFS-Dataset einem Jail zuzuweisen, wird der Befehl `zfs jail` und die dazugehörige Eigenschaft `jailed` verwendet. Durch Angabe von `zfs jail jailid` wird ein Dataset dem spezifizierten Jail zugewiesen und kann mit `zfs unjail` wieder abgehängt werden. Damit das Dataset innerhalb der Jail kontrolliert werden kann, muss die Eigenschaft `jailed` gesetzt sein. Sobald ein Dataset sich im Jail befindet, kann es nicht mehr länger auf dem Hostsystem eingehängt werden, da es Einhängepunkte aufweisen könnte, welche die Sicherheit des Systems gefährden.

## 37.5. Delegierbare Administration

Ein umfassendes System zur Berechtigungsübertragung erlaubt unprivilegierten Benutzern, ZFS-Administrationsaufgaben durchzuführen. Beispielsweise, wenn jedes Heimatverzeichnis eines Benutzers ein Dataset ist, können Benutzer das Recht darin erhalten, Schnappschüsse zu erstellen und zu zerstören. Einem Benutzer für die Sicherung kann die Erlaubnis eingeräumt werden, die Replikationseigenschaft zu verwenden. Einem Skript zum Sammeln von Speicherplatzverbrauch kann die Berechtigung gegeben werden, nur auf die Verbrauchsdaten aller Benutzer zuzugreifen. Es ist sogar möglich, die Möglichkeit zum Delegieren zu delegieren. Die Berechtigung zur Delegation ist für jedes Unterkommando und die meisten Eigenschaften möglich.

### 37.5.1. Delegieren, ein Dataset zu erstellen

`zfs allow someuser create mydataset` gibt dem angegebenen Benutzer die Berechtigung, Kind-Datasets unter dem ausgewählten Elterndataset anzulegen. Es gibt einen Haken: ein neues Dataset anzulegen beinhaltet, dass es eingehängt wird. Dies bedeutet, dass FreeBSDs `vfs.usermount` sysctl(8) auf `1` gesetzt wird, um nicht-root Benutzern zu erlauben, Dateisysteme einzubinden. Es gibt eine weitere Einschränkung um Missbrauch zu verhindern: nicht-`root` Benutzer müssen Besitzer des Einhängepunktes sein, an dem das Dateisystem eingebunden werden soll.

### 37.5.2. Delegationsberechtigung delegieren

`zfs allow someuser allow mydataset` gibt dem angegebenen Benutzer die Fähigkeit, jede Berechtigung, die er selbst auf dem Dataset oder dessen Kindern besitzt, an andere Benutzer weiterzugeben. Wenn ein Benutzer die `snapshot`- und die `allow`-Berechtigung besitzt, kann dieser dann die `snapshot`-Berechtigung an andere Benutzer delegieren.

## 37.6. Themen für Fortgeschrittene

### 37.6.1. Anpassungen

Eine Reihe von Anpassungen können vorgenommen werden, um ZFS unter verschiedenen Belastungen während des Betriebs bestmöglich einzustellen.

- `vfs.zfs.arc_max` - Maximale Größe des ARC. Die Voreinstellung ist der gesamte RAM weniger 1 GB oder 5/8 vom RAM, je nachdem, was mehr ist. Allerdings sollte ein niedriger Wert verwendet werden, wenn das System weitere Dienste oder Prozesse laufen lässt, welche Hauptspeicher benötigen. Dieser Wert kann zur Laufzeit mit sysctl(8) eingestellt und in /boot/loader.conf permanent gespeichert werden.
- `vfs.zfs.arc_meta_limit` - Schränkt die Menge des ARC ein, welche für die Speicherung von Metadaten verwendet wird. Die Voreinstellung ist ein Viertel von `vfs.zfs.arc_max`. Diesen Wert zu erhöhen steigert die Geschwindigkeit, wenn die Arbeitslast Operationen auf einer großen Menge an Dateien und Verzeichnissen oder häufigen Metadatenoperationen beinhaltet. Jedoch bedeutet dies auch weniger Dateidaten, die in den ARC passen. Dieser Wert kann zur Laufzeit mit sysctl(8) eingestellt und in /boot/loader.conf oder /etc/sysctl.conf dauerhaft gespeichert werden.
- `vfs.zfs.arc_min` - Minimale Größe des ARC. Der Standard beträgt die Hälfte von `vfs.zfs.arc_meta_limit`. Passen Sie diesen Wert an, um zu verhindern, dass andere Anwendungen den gesamten ARC verdrängen. Dieser Wert kann zur Laufzeit mit sysctl(8) geändert und in /boot/loader.conf oder /etc/sysctl.conf dauerhaft gespeichert werden.
- `vfs.zfs.vdev.cache.size` - Eine vorallokierte Menge von Speicher, die als Cache für jedes Gerät im Pool reserviert wird. Die Gesamtgröße von verwendetem Speicher ist dieser Wert multipliziert mit der Anzahl an Geräten. Nur zur Bootzeit kann dieser Wert angepasst werden und wird in /boot/loader.conf eingestellt.
- `vfs.zfs.min_auto_ashift` - Minimaler `ashift`-Wert (Sektorgröße), welche zur Erstellungszeit des Pools automatisch verwendet wird. Der Wert ist ein Vielfaches zur Basis Zwei. Der Standardwert von `9` repräsentiert `2^9 = 512`, eine Sektorgröße von 512 Bytes. Um *write*

*amplification* zu vermeiden und die bestmögliche Geschwindigkeit zu erhalten, setzen Sie diesen Wert auf die größte Sektorgröße, die bei einem Gerät im Pool vorhanden ist.

Viele Geräte besitzen 4 KB große Sektoren. Die Verwendung der Voreinstellung `9` bei `ashift` mit diesen Geräten resultiert in einer write amplification auf diesen Geräten. Daten, welche in einem einzelnen 4 KB Schreibvorgang Platz finden würden, müssen stattdessen in acht 512-byte Schreibvorgänge aufgeteilt werden. ZFS versucht, die allen Geräten zugrundeliegende Sektorgröße während der Poolerstellung zu lesen, jedoch melden viele Geräte mit 4 KB Sektoren, dass ihre Sektoren aus Kompatibilitätsgründen 512 Bytes betragen. Durch das Setzen von `vfs.zfs.min_auto_ashift` auf `12` (`2^12 = 4096`) bevor der Pool erstellt wird, zwingt ZFS dazu, für diese Geräte 4 KB Blöcke für bessere Geschwindigkeit zu nutzen.

Erzwingen von 4 KB Blöcken ist ebenfalls hilfreich auf Pools bei denen Plattenaufrüstungen geplant sind. Zukünftige Platten werden wahrscheinlich 4 KB große Sektoren und der Wert von `ashift` lässt sich nach dem Erstellen des Pools nicht mehr ändern.

In besonderen Fällen ist die kleinere Blockgröße von 512-Byte vorzuziehen. Weniger Daten werden bei kleinen, zufälligen Leseoperationen übertragen, was besonders bei 512-Byte großen Platten für Datenbanken oder Plattenplatz für virtuelle Maschinen der Fall ist. Dies kann bessere Geschwindigkeit bringen, ganz besonders wenn eine kleinere ZFS record size verwendet wird.

- `vfs.zfs.prefetch_disable` - Prefetch deaktivieren. Ein Wert von `0` bedeutet aktiviert und `1` heißt deaktiviert. Die Voreinstellung ist `0`, außer, das System besitzt weniger als 4 GB RAM. Prefetch funktioniert durch das Lesen von grösseren Blöcken in den ARC als angefordert wurden, in der Hoffnung, dass diese Daten ebenfalls bald benötigt werden. Wenn die I/O-Last viele große Mengen von zufälligen Leseoperationen beinhaltet, ist das Deaktivieren von prefetch eine Geschwindigkeitssteigerung durch die Reduzierung von unnötigen Leseoperationen. Dieser Wert kann zu jeder Zeit über sysctl(8) angepasst werden.
- `vfs.zfs.vdev.trim_on_init` - Steuert, ob neue Geräte, die dem Pool hinzugefügt werden, das `TRIM` -Kommando ausführen sollen. Das beinhaltet die beste Geschwindigkeit und Langlebigkeit für SSDs, benötigt jedoch zusätzliche Zeit. Wenn das Gerät bereits sicher gelöscht wurde, kann durch deaktivieren dieser Option das Hinzufügen neuer Geräte schneller geschehen. Über sysctl(8) lässt sich dieser Wert jederzeit einstellen.
- `vfs.zfs.vdev.max_pending` - Begrenzt die Menge von ausstehenden I/O-Anfragen pro Gerät. Ein größerer Wert wird die Gerätewarteschlange für Befehle gefüllt lassen und möglicherweise besseren Durchsatz erzeugen. Ein niedrigerer Wert reduziert die Latenz. Jederzeit kann dieser Wert über sysctl(8) angepasst werden.
- `vfs.zfs.top_maxinflight` - Maximale Anzahl von ausstehenden I/Os pro darüberliegendem vdev. Begrenzt die Tiefe Kommandowarteschlange, um hohe Latenzen zu vermeiden. Das Limit ist pro darüberliegendem vdev, was bedeutet, dass das Limit für jeden mirror, RAID-Z, oder anderes vdev unabhängig gilt. Mit sysctl(8) kann dieser Wert jederzeit angepasst werden.
- `vfs.zfs.l2arc_write_max` - Begrenzt die Menge an Daten, die pro Sekunde in den L2ARC geschrieben wird. Durch diese Einstellung lässt sich die Lebensdauer von SSDs erhöhen, indem die Menge an Daten beschränkt wird, die auf das Gerät geschrieben wird. Dieser Wert ist über sysctl(8) zu einem beliebigen Zeitpunkt änderbar.
- `vfs.zfs.l2arc_write_boost` - Der Wert dieser Einstellung wird zu `vfs.zfs.l2arc_write_max`

addiert und erhöht die Schreibgeschwindigkeit auf die SSD bis der erste Block aus dem L2ARC verdrängt wurde. Diese "Turbo Warmup Phase" wurde entwickelt, um den Geschwindigkeitsverlust eines leeren L2ARC nach einem Neustart zu reduzieren. Jederzeit kann dieser Wert mit sysctl(8) geändert werden.

- `vfs.zfs.scrub_delay` - Anzahl von Ticks an Verzögerung zwischen jedem I/O während eines `scrub`. Um zu gewährleisten, dass ein `scrub` nicht mit die normalen Vorgänge eines Pools beeinträchtigt. Wenn währenddessen andere I/Os durchgeführt werden, wird der `scrub` zwischen jedem Befehl verzögert. Dieser Wert regelt die Gesamtmenge von IOPS (I/Os Per Second), die von `scrub` generiert werden. Die Granularität der Einstellung ist bestimmt durch den Wert von `kern.hz`, welcher standardmäßig auf auf 1000 Ticks pro Sekunde eingestellt ist. Diese Einstellung kann geändert werden, was in einer unterschiedlich effektiven Limitierung der IOPS resultiert. Der Standardwert ist 4, was ein Limit von 1000 ticks/sec / 4 = 250 IOPS ergibt. Ein Wert von *20* würde ein Limit von 1000 ticks/sec / 20 = 50 IOPS ergeben. Die `scrub` -Geschwindigkeit ist nur begrenzt, wenn es kürzlich Aktivität auf dem Pool gab, wie der Wert von `vfs.zfs.scan_idle` verrät. Zu einem beliebigen Zeitpunkt kann über sysctl(8) eine Änderung an diesem Wert erfolgen.
- `vfs.zfs.resilver_delay` - Anzahl an Millisekunden Verzögerung, die zwischen jedem I/O während eines resilver eingefügt wird. Um zu versichern, dass ein resilver nicht die normalen Vorgänge auf dem Pool stört, wird dieser zwischen jedem Kommando verzögert, wenn andere I/Os auf dem Pool passieren. Dieser Wert steuert das Limit der Gesamt-IOPS (I/Os Pro Sekunde), die vom resilver erzeugt werden. Die Granularität der Einstellung wird durch den Wert von `kern.hz` bestimmt, welcher standardmäßig 1000 Ticks pro Sekunde beträgt. Diese Einstellung lässt sich ändern, was in einem unterschiedlich effizienten IOPS-Limit resultiert. Die Voreinstellung ist 2, was ein Limit von 1000 ticks/sec / 2 = 500 IOPS beträgt. Einen Pool wieder in den Zustand Online zu versetzen ist möglicherweise wichtiger wenn eine andere Platte den Pool in den Fault-Zustand versetzt, was Datenverlust zur Folge hat. Ein Wert von 0 wird der resilver-Operation die gleiche Priorität wie anderen Operationen geben, was den Heilungsprozess beschleunigt. Die Geschwindigkeit des resilver wird nur begrenzt, wenn es kürzlich andere Aktivitäten auf dem Pool gab, wie von `vfs.zfs.scan_idle` festgestellt wird. Dieser Wert kann zu jeder Zeit über. sysctl(8) eingestellt werden.
- `vfs.zfs.scan_idle` - Anzahl an Millisekunden seit der letzten Operation bevor der Pool als im Leerlauf befindlich deklariert wird. Wenn sich der Pool im Leerlauf befindet, wird die Begrenzung für `scrub` und resilver deaktiviert. Dieser Wert kann mittels sysctl(8) jederzeit angepasst werden.
- `vfs.zfs.txg.timeout` - Maximale Anzahl von Sekunden zwischen Transaktionsgruppen (transaction group). Die momentane Transaktionsgruppe wird auf den Pool geschrieben und eine frische Transaktionsgruppe begonnen, wenn diese Menge an Zeit seit der vorherigen Transaktionsgruppe abgelaufen ist. Eine Transaktionsgruppe kann verfrüht ausgelöst werden, wenn genug Daten geschrieben werden. Der Standardwert beträgt 5 Sekunden. Ein größerer Wert kann die Lesegeschwindigkeit durch verzögern von asynchronen Schreibvorgängen verbessern, allerdings kann dies ungleiche Geschwindigkeiten hervorrufen, wenn eine Transaktionsgruppe geschrieben wird. Dieser Wert kann zu einem beliebigen Zeitpunkt mit sysctl(8) geändert werden.

### 37.6.2. ZFS auf i386

Manche der Eigenschaften, die von ZFS bereitgestellt werden, sind speicherintensiv und benötigen Anpassungen für die maximale Effizienz auf Systemen mit begrenztem RAM.

#### Hauptspeicher

Als absolutes Minimum sollte der gesamte verfügbare Hauptspeicher mindestens ein Gigabyte betragen. Die vorgeschlagene Menge an RAM ist bedingt durch die Poolgröße und welche Eigenschaften von ZFS verwendet werden. Eine Faustregel besagt, dass 1 GB RAM für jedes 1 TB Storage vorgesehen werden sollte. Wenn Deduplizierung zum Einsatz kommt, besagt die Regel, dass 5 GB RAM pro TB an Speicher, der dedupliziert werden soll, bereitgestellt sein muss. Obwohl manche Anwender ZFS mit weniger RAM einsetzen, stürzen Systeme häufiger wegen unzureichendem Hauptspeicher ab. Weitere Anpassungen sind unter Umständen nötig für Systeme mit weniger als die vorgeschlagene Menge an RAM.

#### Kernel-Konfiguration

Wegen des begrenzten Addressraumes der i386™-Plattform müssen ZFS-Anwendern auf der i386™-Architektur diese Option der Kernelkonfigurationsdatei hinzufügen, den Kernel erneut bauen und das System neu starten:

```
options        KVA_PAGES=512
```

Dies erweitert den Addressraum des Kernels, was es erlaubt, die Einstellung `vm.kvm_size` hinter die momentan vorgegebene Grenze von 1 GB oder das Limit von 2 GB für PAE zu bringen. Um den passenden Wert für diese Option zu finden, teilen Sie den gewünschten Addressraum in Megabyte durch vier. In diesem Beispiel beträgt sie `512` für 2 GB.

#### Loader-Anpassungen

Der kmem-Addressraum kann auf allen FreeBSD-Architekturen erhöht werden. Auf einem Testsystem mit 1 GB physischen Speichers wurden mit diesen Optionen in /boot/loader.conf und einem anschließenden Systemneustart Erfolge erzielt:

```
vm.kmem_size="330M"
vm.kmem_size_max="330M"
vfs.zfs.arc_max="40M"
vfs.zfs.vdev.cache.size="5M"
```

Für eine detailliertere Liste an Empfehlungen für ZFS-bezogene Einstellungen, lesen Sie https://wiki.freebsd.org/ZFSTuningGuide.

## 37.7. Zusätzliche Informationen

- OpenZFS
- FreeBSD Wiki - ZFS Tuning

- Oracle Solaris ZFS Administration Guide
- Calomel Blog - ZFS Raidz Performance, Capacity und Integrity

## 37.8. ZFS-Eigenschaften und Terminologie

ZFS ist ein fundamental anderes Dateisystem aufgrund der Tatsache, dass es mehr als ein Dateisystem ist. ZFS kombiniert die Rolle eines Dateisystems mit dem Volumemanager, was es ermöglicht, zusätzliche Speichermedien zu einem laufenden System hinzuzufügen und diesen neuen Speicher sofort auf allen auf dem Pool existierenden Dateisystemen zur Verfügung zu haben. Durch die Kombination von traditionell getrennten Rollen ist ZFS in der Lage, Einschränkungen, die zuvor RAID-Gruppen daran gehindert hatten, zu wachsen. Jedes Gerät auf höchster Ebene in einem Pool wird ein *vdev* genannt, was eine einfache Platte oder eine RAID-Transformation wie ein Spiegel oder RAID-Z-Verbund sein kann. ZFS-Dateisysteme (*datasets* genannt), haben jeweils Zugriff auf den gesamten freien Speicherplatz des gesamten Pools. Wenn Blöcke aus diesem Pool allokiert werden, verringert sich auch der freie Speicherplatz für jedes Dateisystem. Dieser Ansatz verhindert die allgegenwärtige Falle von umfangreichen Partitionen, bei denen freier Speicherplatz über alle Partitionen hinweg fragmentiert wird.

| | |
|---|---|
| zpool | Ein Speicher-*Pool* ist der grundlegendste Baustein von ZFS. Ein Pool besteht aus einem oder mehreren vdevs, was die zugrundeliegenden Geräte repräsentiert, welche die Daten speichern. Ein Pool wird dann verwendet, um ein oder mehrere Dateisysteme (Datasets) oder Blockgeräte (Volumes) zu erstellen. Diese Datasets und Volumes teilen sich den im Pool verfügbaren Speicherplatz. Jeder Pool wird eindeutig durch einen Namen und eine GUID identifiziert. Die verfügbaren Eigenschaften werden durch die ZFS-Versionsnummer des Pools bestimmt. |

| | |
|---|---|
| vdev Arten | Ein Pool besteht aus einem oder mehreren vdevs, die selbst eine einfache Platte oder im Fall von RAID eine Gruppe von Platten darstellt. Wenn mehrere vdevs eingesetzt werden, verteilt ZFS die Daten über die vdevs, um die Geschwindigkeit zu steigern und den verfügbaren Platz zu maximieren.<br><br>• *Festplatte* - Der einfachste Typ von vdev ist ein Standard-Blockgerät. Dies kann die komplette Platte (wie /dev/ada0 oder /dev/da0) oder auch eine Partition ( /dev/ada0p3) sein. Auf FreeBSD gibt es keine Geschwindigkeitseinbußen bei der Verwendung einer Partition anstatt einer kompletten Platte. Dies unterscheidet sich von den Empfehlungen, welche in der Solaris Dokumentation gegeben werden.<br><br><br>Es wird dringend davon abgeraten, eine ganze Platte für einen bootbaren Pool zu benutzen, da dies dazu führen kann, dass der Pool nicht mehr bootet. Ebenso sollten Sie nicht eine ganze Platte als Teil eines Spiegels oder RAID-Z vdev verwenden, weil es dann nicht mehr möglich ist, die Größe einer nicht partitionierten Platte beim Booten zuverlässig zu bestimmen. Zudem gibt es dann keinen Platz mehr, um Boot-Code einzufügen.<br><br>• *File* - Zusätzlich zu Festplatten können ZFS-Pools aus regulären Dateien aufgebaut sein, was besonders hilfreich ist, um zu testen und zu experimentieren. Verwenden Sie den kompletten Pfad zu der Datei als Gerätepfad im Befehl `zpool create`. Alle vdevs müssen mindestens 128 MB groß sein.<br><br>• *Mirror* - Wenn ein Spiegel erstellt wird, verwenden Sie das Schlüsselwort `mirror`, gefolgt von der Liste an Mitgliedsgeräten für den Spiegel. Ein Spiegel besteht aus zwei oder mehr Geräten und sämtliche Daten werden auf alle Geräte, die Mitglied des Spiegels sind, geschrieben. Ein Spiegel-vdev wird nur soviele Daten speichern, wie das kleinste Gerät im Verbund aufnehmen kann. Ein Spiegel-vdev kann den Verlust von allen Mitgliedsgeräten bis auf eines verkraften, ohne irgendwelche Daten zu verlieren.<br><br><br>Ein reguläre einzelne vdev-Platte kann jederzeit zu einem Spiegel-vdev über das Kommando `zpool attach` aktualisiert werden.<br><br>• *RAID-Z* - ZFS implementiert RAID-Z, eine Varianten des RAID-5-Standards, der bessere Verteilung der Parität bietet und das "RAID-5 write hole" eliminiert, bei dem die Daten und Parität nach einem unerwarteten Neustart inkonsistent werden können. ZFS unterstützt drei Stufen von RAID-Z, die unterschiedliche Arten von Redundanz im Austausch gegen niedrigere Stufen von verwendbarem Speicher. Diese Typen werden RAID-Z1 bis RAID-Z3 genannt, basierend auf der Anzahl der Paritätsgeräte im Verbund und der Anzahl an Platten, die ausfallen können, während der Pool immer noch normal funktioniert.<br><br>In einer RAID-Z1-Konfiguration mit vier Platten, bei der jede 1 TB besitzt, beträgt der verwendbare Plattenplatz 3 TB und der Pool wird immer noch im Modus degraded weiterlaufen, wenn eine Platte davon ausfällt. Wenn eine zusätzliche Platte ausfällt, bevor die defekte Platte ersetzt wird, können alle Daten im Pool verloren gehen. |

| | |
|---|---|
| Transaktionsgruppe (Transaction Group, TXG) | Transaktionsgruppen sind die Art und Weise, wie geänderte Blöcke zusammen gruppiert und letztendlich auf den Pool geschrieben werden. Transaktionsgruppen sind die atomare Einheit, welche ZFS verwendet, um Konsistenz zu gewährleisten. Jeder Transaktionsgruppe wird eine einzigartige, fortlaufende 64-Bit Identifikationsnummer zugewiesen. Es kann bis zu drei aktive Transaktionsgruppen gleichzeitig geben, wobei sich jede davon in einem der folgenden drei Zustände befinden kann:<br><br>* *Open (Offen)* - Wenn eine neue Transaktionsgruppe erstellt wird, befindet diese sich im Zustand offen und akzeptiert neue Schreibvorgänge. Es ist immer eine Transaktionsgruppe in diesem Zustand, jedoch kann die Transaktionsgruppe neue Schreibvorgänge ablehnen, wenn diese ein Limit erreicht hat. Sobald eine offene Transaktionsgruppe an das Limit stößt oder das `vfs.zfs.txg.timeout` wurde erreicht, geht die Transaktionsgruppe in den nächsten Zustand über. * *Quiescing (Stilllegen)* - Ein kurzer Zustand, der es noch ausstehenden Operationen erlaubt, zum Abschluss zu kommen, währenddessen das Erstellen einer neuen Transaktionsgruppe jedoch nicht blockiert wird. Sobald alle Transaktionen in der Gruppe abgeschlossen sind, geht die Transaktionsgruppen in den letzten Zustand über. * *Syncing (Sychronisieren)* - Alle Daten in der Transaktionsgruppe werden auf das Speichermedium geschrieben. Dieser Prozess wird wiederum andere Daten wie Metadaten und space maps verändern, die ebenfalls auf das Speichermedium geschrieben werden müssen. Der Prozess des Synchronisierens beinhaltet mehrere Durchläufe. Der erste Prozess, welches der größte, gefolgt von den Metadaten, ist, beinhaltet alle geänderten Datenblöcke und kann mehrere Durchläufe benötigen, um zum Ende zu gelangen. Da das Allokieren von Speicher für die Datenblöcke neue Metadaten generiert, kann der Synchronisationsprozess nicht beendet werden, bis ein Durchlauf fertig ist, der keinen zusätzlichen Speicher allokiert. Der Synchronisierungszustand ist der Zustand, in dem auch *synctasks* abgeschlossen werden. Synctasks sind administrative Operationen, wie das Erstellen oder zerstören von Schnappschüssen und Datasets, welche den Überblock verändern, wenn sie abgeschlossen sind. Sobald der Synchronisationszustand abgeschlossen ist, geht die Transaktionsgruppe aus dem Stilllegungszustand über in den Synchronisationszustand. Alle administrativen Funktionen, wie `Schnappschüsse` werden als Teil einer Transaktionsgruppe geschrieben. Wenn ein synctask erstellt ist, wird dieser der momentan geöffneten Transaktionsgruppe hinzugefügt und diese Gruppe wird so schnell wie möglich in den Synchronisationszustand versetzt, um die Latenz von administrativen Befehlen zu reduzieren. |

| | |
|---|---|
| Adaptive Replacement Cache (ARC) | ZFS verwendet einen Adaptive Replacement Cache (ARC), anstatt eines traditionellen Least Recently Used (LRU) Caches. Ein LRU-Cache ist eine einfache Liste von Elementen im Cache, sortiert nach der letzten Verwendung jedes Elements in der Liste. Neue Elemente werden an den Anfang der Liste eingefügt. Wenn der Cache voll ist, werden Elemente vom Ende der Liste verdrängt, um Platz für aktivere Objekte zu schaffen. Ein ARC besteht aus vier Listen: derjenigen der Most Recently Used (MRU) und Most Frequently Used (MFU) Objekte, plus einer sogenannten ghost list für jede von beiden. Diese Ghost Lists verfolgen die kürzlich verdrängten Objekte, um zu verhindern, dass diese erneut in den Cache aufgenommen werden. Dies erhöht die Trefferrate (hit ratio) des Caches, indem verhindert wird, dass Elemente, die in der Vergangenheit nur ab und zu benutzt wurden, wieder im Cache landen. Ein weiterer Vorteil der Verwendung sowohl einer MRU und einer MFU ist, dass das Scannen eines gesamten Dateisystems normalerweise alle Daten aus einem MRU- oder LRU-Cache verdrängt, um dem gerade frisch zugegriffenem Inhalt den Vorzug zu geben. Mit ZFS gibt es also eine MFU, die nur die am häufigsten verwendeten Elemente beinhaltet und der Cache von am meisten zugegriffenen Blöcken bleibt erhalten. |
| L2ARC | L2ARC ist die zweite Stufe des Caching-Systems von ZFS. Der Haupt-ARC wird im RAM abgelegt. Da die Menge an verfügbarem RAM meist begrenzt ist, kann ZFS auch cache vdevs verwenden. Solid State Disks (SSDs) werden oft als diese Cache-Geräte eingesetzt, aufgrund ihrer höheren Geschwindigkeit und niedrigeren Latenz im Vergleich zu traditionellen drehenden Speichermedien wie Festplatten. Der Einsatz des L2ARC ist optional, jedoch wird durch die Verwendung eine signifikante Geschwindigkeitssteigerung bei Lesevorgängen bei Dateien erzielt, welche auf der SSD zwischengespeichert sind, anstatt von der regulären Platte gelesen werden zu müssen. L2ARC kann ebenfalls die Deduplizierung beschleunigen, da eine DDT, welche nicht in den RAM passt, jedoch in den L2ARC wesentlich schneller sein wird als eine DDT, die von der Platte gelesen werden muss. Die Häufigkeit, in der Daten zum Cache-Gerät hinzugefügt werden, ist begrenzt, um zu verhindern, dass eine SSD frühzeitig durch zu viele Schreibvorgänge aufgebraucht ist. Bis der Cache voll ist (also der erste Block verdrängt wurde, um Platz zu schaffen), wird das Schreiben auf den L2ARC begrenzt auf die Summe der Schreibbegrenzung und das Bootlimit, sowie hinterher auf das Schreiblimit. Ein paar sysctl(8)-Werte steuert diese Limits. `vfs.zfs.l2arc_write_max` steuert, wie viele Bytes in den Cache pro Sekunde geschrieben werden, während `vfs.zfs.l2arc_write_boost` zu diesem Limit während der "Turbo Warmup Phase" hinzuaddiert wird (Write Boost). |
| ZIL | ZIL beschleunigt synchrone Transaktionen durch die Verwendung von Speichermedien wie SSDs, welche schneller sind als diejenigen, welche Teil des Speicherpools sind. Wenn eine Anwendung einen synchronen Schreibvorgang anfordert (eine Garantie, dass die Daten sicher auf den Platten gespeichert wurden anstatt nur zwischengespeichert zu sein, um später geschrieben zu werden), werden die Daten auf den schnelleren ZIL-Speicher geschrieben und dann später auf die regulären Festplatten. Dies reduziert die Latenz sehr und verbessert die Geschwindigkeit. Nur synchrone Vorgänge wie die von Datenbanken werden durch den Einsatz eines ZIL profitieren. Reguläre, asynchrone Schreibvorgänge wie das Kopieren von Dateien wird den ZIL überhaupt nicht verwenden. |

| | |
|---|---|
| Copy-On-Write | Im Gegensatz zu traditionellen Dateisystemen werden beim Überschreiben von Daten bei ZFS die neuen Daten an einen anderen Block geschrieben, anstatt die alten Daten an der gleichen Stelle zu überschreiben. Nur wenn dieser Schreibvorgang beendet wurde, werden die Metadaten aktualisiert, um auf die neue Position zu verweisen. Im Falle eines kurzen Schreibvorgangs (ein Systemabsturz oder Spannungsverlust während eine Datei geschrieben wird) sind die gesamten Inhalte der Originaldatei noch vorhanden und der unvollständige Schreibvorgang wird verworfen. Das bedeutet auch, dass ZFS nach einem unvorhergesehenen Ausfall keinen fsck(8) benötigt. |
| Dataset | *Dataset* ist der generische Begriff für ein ZFS-Dateisystem, Volume, Schnappschüsse oder Klone. Jedes Dataset besitzt einen eindeutigen Namen in der Form *poolname/path@snapshot* Die Wurzel des Pools ist technisch gesehen auch ein Dataset. Kind-Datasets werden hierarchisch wie Verzeichnisse benannt. Beispielsweise ist *mypool/home* das Heimatdataset, ein Kind von *mypool* und erbt die Eigenschaften von diesem. Dies kann sogar noch erweitert werden durch das Erstellen von *mypool/home/user*. Dieses Enkelkind-Dataset wird alle Eigenschaften von den Eltern und Großeltern erben. Eigenschaften auf einem Kind können die geerbten Standardwerte der Eltern und Großeltern ändern und überschreiben. Die Verwaltung von Datasets und dessen Kindern lässt sich delegieren. |
| Dateisystem | Ein ZFS-Dataset wird meistens als ein Dateisystem verwendet. Wie jedes andere Dateisystem kann auch ein ZFS-Dateisystem irgendwo in der Verzeichnishierarchie eingehängt werden und enthält seine eigenen Dateien und Verzeichnisse mit Berechtigungen, Flags und anderen Metadaten. |
| Volume | Zusätzlich zu regulären Dateisystem-Datasets, kann ZFS auch Volumes erstellen, die Blockgeräte sind. Volumes besitzen viele der gleichen Eigenschaften, inklusive copy-on-write, Schnappschüsse, Klone und Prüfsummen. Volumes sind nützlich, um andere Dateisystemformate auf ZFS aufzusetzen, so wie UFS Virtualisierung, oder das Exportieren von iSCSI-Abschnitten. |

| | |
|---|---|
| Snapshot (Schnappschuss) | Das copy-on-write (COW)-Entwicklung von ZFS erlaubt das Erstellen von beinahe sofortigen, konsistenten Schnappschüssen mit beliebigen Namen. Nachdem ein Schnappschuss von einem Dataset angelegt oder ein rekursiver Schnappschuss eines Elterndatasets, welcher alle Kinddatasets enthält, erstellt wurde, werden neue Daten auf neue Blöcke geschrieben, jedoch die alten Blöcke nicht wieder als freier Speicher zurückgewonnen. Der Schnappschuss enthält die Originalversion des Dateisystems und das aktive Dateisystem besitzt alle Änderungen, die seit dem Schnappschuss erstellt wurden. Kein zusätzlicher Platz wird benötigt. Werden neue Daten auf das aktive Dateisystem geschrieben, werden neue Blöcke allokiert, um diese Daten zu speichern. Die scheinbare Größe des Schnappschusses wird wachsen, da die Blöcke nicht mehr länger im aktiven Dateisystem, sondern nur noch im Schnappschuss Verwendung finden. Diese Schnappschüsse können nur lesend eingehängt werden, um vorherige Versionen von Dateien wiederherzustellen. Ein rollback eines aktiven Dateisystems auf einen bestimmten Schnappschuss ist ebenfalls möglich, was alle Änderungen, die seit dem Anlegen des Schnappschusses vorgenommen wurden, wieder Rückgängig macht. Jeder Block im Pool besitzt einen Referenzzähler, der verfolgt, wieviele Schnappschüsse, Klone, Datasets oder Volumes diesen Block nutzen. Wenn Dateien und Schnappschüsse gelöscht werden, verringert dies auch den Referenzzähler. Wenn ein Block nicht mehr länger referenziert wird, kann er als freier Speicher wieder genutzt werden. Schnappschüsse können auch mit hold markiert werden. Wenn versucht wird, einen solchen Schnappschuss zu zerstören, wird stattdessen ein `EBUSY`-Fehler ausgegeben. Jeder Schnappschuss kann mehrere holds besitzen, jeder mit einem eindeutigen Namen. Das Kommando release entfernt diese, damit der Schnappschuss gelöscht werden kann. Schnappschüsse lassen sich auf Volumes ebenfalls anlegen, allerdings können diese nur geklont oder zurückgerollt werden, nicht jedoch unabhängig eingehängt. |
| Clone (Klone) | Schnappschüsse können auch geklont werden. Ein Klon stellt eine veränderbare Version eines Schnappschusses dar, was es ermöglicht, das Dateisystem als neues Dataset aufzuspalten. Genau wie bei einem Schnappschuss verbraucht ein Klon keinen zusätzlichen Platz. Wenn neue Daten auf einen Klon geschrieben und neue Blöcke allokiert werden, wächst auch die Größe des Klons. Wenn Blöcke im geklonten Dateisystem oder Volume überschrieben werden, verringert sich auch der Referenzzähler im vorherigen Block. Der Schnappschuss, auf dem der Klon basiert kann nicht gelöscht werden, weil der Klon darauf eine Abhängigkeit besitzt. Der Schnappschuss stellt den Elternteil dar und der Klon das Kind. Klone lassen sich *promoted* (befördern), was die Abhängigkeit auflöst und den Klon zum Elternteil macht und den vorherigen Elternteil das Kind. Diese Operation benötigt keinen zusätzlichen Plattenplatz. Da die Menge an verwendetem Speicher vom Elternteil und dem Kind vertauscht wird, betrifft dies eventuell vorhandene Quotas und Reservierungen. |

| | |
|---|---|
| Checksum (Prüfsumme) | Jeder Block, der allokiert wird erhält auch eine Prüfsumme. Der verwendete Prüfsummenalgorithmus ist eine Eigenschaft jedes Datasets, siehe dazu `set`. Die Prüfsumme jedes Blocks wird transparent validiert wenn er gelesen wird, was es ZFS ermöglicht, stille Verfälschung zu entdecken. Wenn die gelesenen Daten nicht mit der erwarteten Prüfsumme übereinstimmen, wird ZFS versuchen, die Daten aus jeglicher verfügbarer Redundanz (wie Spiegel oder RAID-Z) zu rekonstruieren. Eine Überprüfung aller Prüfsummen kann durch das Kommando `scrub` ausgelöst werden. Prüfsummenalgorithmen sind:<br><br>* `fletcher2` * `fletcher4` * `sha256` Die `fletcher`-Algorithmen sind schneller, aber dafür ist `sha256` ein starker kryptographischer Hash und besitzt eine viel niedrigere Chance auf Kollisionen zu stoßen mit dem Nachteil geringerer Geschwindigkeit. Prüfsummen können deaktiviert werden, dies wird aber nicht empfohlen. |
| Compression | Jedes Dataset besitzt eine compression-Eigenschaft, die standardmäßig ausgeschaltet ist. Diese Eigenschaft kann auf eine Reihe von Kompressionsalgorithmen eingestellt werden. Dadurch werden alle neuen Daten, die auf das Dataset geschrieben werden, komprimiert. Neben einer Reduzierung von verbrauchtem Speicher wird oft der Lese- und Schreibdurchsatz erhöht, weil weniger Blöcke gelesen oder geschrieben werden müssen.<br><br>* *LZ4* - Wurde in der ZFS Poolversion 5000 (feature flags) hinzugefügt und LZ4 ist jetzt der empfohlene Kompressionsalgorithmus. LZ4 komprimiert ungefähr 50% schneller als LZJB, wenn er auf komprimierbaren Daten angewendet wird und ist über dreimal schneller, wenn unkomprimierbare Daten vorliegen. LZ4 entkomprimiert auch ungefähr 80% schneller als LZJB. Auf modernen CPUs, kann LZ4 oft über 500 MB/s komprimieren und entkomprimiert (pro einzelnem CPU-Kern) bei über 1.5 GB/s. * *LZJB* - Der Standardkompressionsalgorithmus wurde von Jeff Bonwick, einem der ursprünglichen Entwickler von ZFS, entworfen. LZJB bietet gute Komprimierung mit weniger CPU-Überhang im Vergleich zu GZIP. In der Zukunft wird der Standardkompressionsalgorithmus wahrscheinlich auf LZ4 gewechselt. * *GZIP* - Ein populärer Stromkompressionsalgorithmus ist auch in ZFS verfügbar. Einer der Hauptvorteile von der Verwendung von GZIP ist seine konfigurierbare Komprimierungsstufe. Wenn die Eigenschaft `compress` gesetzt wird, kann der Administrator die Stufe der Komprimierung wählen, die von `gzip1`, der kleinsten Komprimierungsstufe, bis zu `gzip9`, der höchsten Komprimierungsstufe, reicht. Dies erlaubt es dem Administrator zu steuern, wieviel CPU-Zeit für eingesparten Plattenplatz eingetauscht werde soll. * *ZLE* - Zero Length Encoding ist ein besonderer Kompressionsalgorithmus, welcher nur fortlaufende Aneinanderreihungen von Nullen komprimiert. Dieser Komprimierungsalgorithmus ist nur sinnvoll, wenn das Dataset viele große Blöcke von Nullen aufweist. |

| | |
|---|---|
| Copies | Wenn die Eigenschaft `copies` auf einen Wert grösser als 1 gesetzt wird, weist das ZFS an, mehrere Kopien eines Blocks im Dateisystem oder Volume anzulegen. Diese Eigenschaft auf einem wichtigen Dataset einzustellen sorgt für zusätzliche Redundanz, aus der ein Block wiederhergestellt werden kann, der nicht mehr mit seiner Prüfsumme übereinstimmt. In Pools ohne Redundanz ist die copies-Eigenschaft die einzige Form von Redundanz. Die Eigenschaft kann einen einzelnen schlechten Sektor oder andere Formen von kleineren Verfälschungen wiederherstellen, schützt jedoch nicht den Pool vom Verlust einer gesamten Platte. |
| Deduplizierung | Prüfsummen ermöglichen es, Duplikate von Blöcken zu erkennen, wenn diese geschrieben werden. Mit Deduplizierung erhöht sich der Referenzzähler eines existierenden, identischen Blocks, was Speicherplatz einspart. Um Blockduplikate zu erkennen, wird im Speicher eine Deduplizierungstabelle (DDT) geführt. Die Tabelle enthält eine Liste von eindeutigen Prüfsummen, die Position dieser Blöcke und einen Referenzzähler. Werden neue Daten geschrieben, wird die Prüfsumme berechnet und mit der Liste verglichen. Wird eine Übereinstimmung gefunden, wird der existierende Block verwendet. Der SHA256-Prüfsummenalgorithmus wird mit Deduplizierung benutzt, um einen sicheren kryptographischen Hash zu bieten. Deduplizierung lässt sich konfigurieren. Wenn `dedup` auf `on` steht, wird angenommen, dass eine übereinstimmende Prüfsumme bedeutet, dass die Daten identisch sind. Steht `dedup` auf `verify`, werden die Daten in den beiden Blöcken Byte für Byte geprüft, um sicherzustellen, dass diese wirklich identisch sind. Wenn die Daten nicht identisch sind, wird die Kollision im Hash vermerkt und die beiden Blöcke separat gespeichert. Da die DDT den Hash jedes einzigartigen Blocks speichern muss, benötigt sie eine große Menge an Speicher. Eine generelle Faustregel besagt, dass 5-6 GB RAM pro 1 TB deduplizierter Daten benötigt werden. In Situationen, in denen es nicht praktikabel ist, genug RAM vorzuhalten, um die gesamte DDT im Speicher zu belassen, wird die Geschwindigkeit stark darunter leiden, da die DDT von der Platte gelesen werden muss, bevor jeder neue Block geschrieben wird. Deduplizierung kann den L2ARC nutzen, um die DDT zu speichern, was einen guten Mittelweg zwischen schnellem Systemspeicher und langsameren Platten darstellt. Bedenken Sie, dass durch die Verwendung von Komprimierung meistens genauso große Platzersparnis möglich ist, ohne den zusätzlichen Hauptspeicherplatzbedarf. |
| Scrub (Bereinigung) | Anstatt einer Konsistenzprüfung wie fsck(8) verwendet ZFS `scrub`. `scrub` liest alle Datenblöcke, die auf dem Pool gespeichert sind und prüft deren Prüfsumme gegen die als richtig in den Metadaten gespeicherte Prüfsumme. Eine periodische Prüfung aller im Pool gespeicherten Daten versichert, dass verfälschte Blöcke rekonstruiert werden können, bevor dies nötig ist. Ein Scrub wird nicht nach einem unsauberen Herunterfahren benötigt, wird jedoch einmal alle drei Monate angeraten. Die Prüfsumme von jedem Block wird verifiziert, wenn Blöcke während des normalen Betriebs gelesen werden, jedoch stellt ein Scrub sicher, dass sogar weniger häufig verwendete Blöcke auf stille Verfälschungen hin untersucht werden. Datenintegrität wird dadurch erhöht, besonders wenn es sich um Archivspeichersituationen handelt. Die relative Priorität des `scrub` lässt sich mit `vfs.zfs.scrub_delay` anpassen, um zu verhindern, dass der scrub die Geschwindigkeit von anderen Anfragen auf dem Pool beeinträchtigt. |

| | |
|---|---|
| Dataset Quotas | ZFS bietet sehr schnelle und akkurate Dataset-, Benutzer- und Gruppenspeicherplatzbuchhaltung, zusätzlich zu Quotas und Speicherplatzreservierungen. Dies gibt dem Administrator feingranulare Kontrolle darüber, wie Speicherplatz allokiert und die Reservierung für kritische Dateisysteme vorgenommen wird<br><br>ZFS unterstützt verschiedene Arten von Quotas: die Dataset-Quota, die Referenzquota (refquota), die Benutzerquota und die Gruppenquota sind verfügbar.<br><br>Quotas beschränken die Menge an Speicherplatz, welche ein Dataset, seine Kinder, einschließlich Schnappschüsse des Datasets, deren Kinder und die Schnappschüsse von diesen Datasets, verbrauchen können.<br><br> Quotas können nicht auf Volumes gesetzt werden, da die Eigenschaft `volsize` als eine implizite Quota agiert. |
| Referenzquota | Ein Referenzquota beschränkt die Menge an Speicherplatz, die ein Dataset verbrauchen kann durch das Erzwingen einer harten Grenze. Jedoch beinhaltet diese harte Grenze nur Speicherplatz, die das Dataset referenziert und beinhaltet nicht den Speicher, der von Kindern, wie Dateisystemen oder Schnappschüssen, verbraucht wird. |
| Benutzerquota | Benutzerquotas sind hilfreich, um die Menge an Speicherplatz, die ein bestimmter Benutzer verbrauchen kann, einzuschränken. |
| Gruppenquota | Die Gruppenquota beschränkt die Menge an Speicherplatz, die eine bestimmte Gruppe verbrauchen darf. |
| Dataset-Reservierung | Die Eigenschaft `reservation` ermöglicht es, ein Minimum an Speicherplatz für ein bestimmtes Dataset und dessen Kinder zu garantieren. Wenn eine Reservierung von 10 GB auf storage/home/bob gesetzt ist und ein anderes Dataset versucht, allen freien Speicherplatz zu verwenden, bleiben zumindest noch 10 GB an Speicher reserviert. Wenn von storage/home/bob ein Schnappschuss angelegt wird, wird dieser von der Reservierung abgezogen und zählt damit dagegen. Die Eigenschaft `refreservation` funktioniert auf ähnliche Weise, jedoch *exkludiert* diese Kinder wie Schnappschüsse.<br><br>Reservierungen jeder Art sind in vielen Situationen nützlich, so wie bei der Planung und dem Testen der richtigen Speicherplatzallokation in einem neuen System oder durch die Zusicherung, dass genug Speicherplatz auf Dateisystemen für Audio-Logs oder Systemwiederherstellungsprozeduren und Dateien verfügbar ist. |

| | |
|---|---|
| Referenzreservierung | Die Eigenschaft `refreservation` ermöglicht es, ein Minimum an Speicherplatz für die Verwendung eines bestimmten Datasets zu garantieren, *exklusiv* dessen Kinder. Das bedeutet, dass wenn eine 10 GB-Reservierung auf storage/home/bob vorhanden ist und ein anderes Dataset versucht, alle freien Speicherplatz aufzubrauchen, sind zumindest noch 10 GB Speicher reserviert. Im Gegensatz zu einer regulären Reservierung wird der Speicher von Schnappschüssen und Kinddataset nicht gegen die Reservierung gezählt. Beispielsweise, wenn ein Schnappschuss von storage/home/bob angelegt wird, muss genug Plattenplatz außerhalb der Menge an `refreservation` vorhanden sein, damit die Operation erfolgreich durchgeführt wird. Kinder des Hauptdatasets werden nicht in die Menge an `refreservation` gezählt und dringen auf diese Weise auch nicht in den gesetzten Speicher ein. |
| Resilver | Wenn eine Platte ausfällt und ersetzt wird, muss die neue Platte mit den Daten gefüllt werden, die verloren gegangen sind. Der Prozess der Verwendung der Paritätsinformationen, welche über die übrigen Platten verteilt sind, um die fehlenden Daten zu berechnen und auf die neue Platte zu übertragen, wird *resilvering* genannt. |
| Online | Ein Pool oder vdev im Zustand `Online` besitzt alle verbundenen Mitgliedsgeräte und ist voll funktionsfähig. Individuelle Geräte im Zustand `Online` funktionieren normal. |
| Offline | Individuelle Geräte lassen sich vom Administrator in den Zustand `Offline` versetzen, wenn es ausreichend Redundanz gibt, um zu verhindern, dass der Pool oder das vdev in den Zustand Faulted versetzt wird. Ein Administrator kann eine Platte vor einem Austausch offline nehmen oder um es leichter zu machen, diese zu identifizieren. |
| Degraded | Ein Pool oder vdev im Zustand `Degraded` hat eine oder mehrere Platten, welche getrennt wurden oder ausgefallen sind. Der Pool kann immer noch verwendet werden, doch wenn noch weitere Geräte ausfallen, kann der Pool nicht wiederhergestellt werden. Die fehlenden Geräte anzuschließen oder die defekten Platten zu ersetzen wird den Pool wieder in den Zustand Online versetzen, nachdem die angeschlossenen oder neuen Geräte den Resilver-Prozess abgeschlossen haben. |
| Faulted | Ein Pool oder vdev im Zustand `Faulted` funktioniert nicht länger. Die Daten darauf sind nicht mehr länger verfügbar. Ein Pool oder vdev geht in den Zustand `Faulted` über, wenn die Anzahl der fehlenden oder defekten Geräte die Redundanzstufe im vdev überschreiten. Wenn fehlende Geräte angeschlossen werden, geht der Pool wieder in den Zustand Online. Wenn es nicht genügend Redundanz gibt, um die Anzahl an defekten Platten zu kompensieren, sind die Inhalte des Pools verloren und müssen von der Sicherung wiederhergestellt werden. |

# Kapitel 38. Dateisystemunterstützung

## 38.1. Übersicht

Dateisysteme sind ein wesentlicher Bestandteil von Betriebssystemen. Sie erlauben es Benutzern, Dateien zu laden und zu speichern, ermöglichen den Zugriff auf Daten und machen Festplatten überhaupt erst nützlich. Betriebssysteme unterscheiden sich normalerweise bei dem mitgelieferten Dateisystem. Traditionell ist dies unter FreeBSD das Unix File System UFS, welches mit UFS2 modernisiert wurde. Seit FreeBSD 7.0 steht auch das Z-Dateisystem (ZFS) als natives Dateisystem zur Verfügung. Hierzu finden Sie in Das Z-Dateisystem (ZFS) weitere Informationen.

FreeBSD unterstützt auch eine Vielzahl weiterer Dateisysteme, um auf Daten von anderen Betriebssystemen lokal zuzugreifen, beispielsweise Daten auf USB-Speichermedien, Flash-Speichern und Festplatten. Dazu gehört die Unterstützung für das Linux® Extended File System (EXT).

Es gibt verschiedene Stufen der Unterstützung in FreeBSD für diese unterschiedlichen Dateisysteme. Manche benötigen ein geladenes Kernelmodul, andere die Installation bestimmter Werkzeuge. Einige Dateisysteme haben volle Unterstützung für Lese- und Schreibzugriffe, während auf andere nur-lesend zugegriffen werden kann.

Nachdem Sie dieses Kapitel gelesen haben, wissen Sie:

- Den Unterschied zwischen nativen und unterstützten Dateisystemen.
- Welche Dateisysteme von FreeBSD unterstützt werden.
- Wie man fremde Dateisysteme aktiviert, konfiguriert, darauf zugreift und diese verwendet.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Grundlagen von UNIX® und FreeBSD verstehen (Grundlagen des FreeBSD Betriebssystems).
- Mit den Grundlagen der Konfiguration und dem Bauen des Kernels vertraut sein (Konfiguration des FreeBSD-Kernels).
- Problemlos Software von Drittherstellern in FreeBSD installieren können (Installieren von Anwendungen: Pakete und Ports).
- Sich ein wenig mit Festplatten, Speicher und Gerätenamen in FreeBSD auskennen (Speichermedien).

## 38.2. Linux® Dateisysteme

FreeBSD bietet integrierte Unterstützung für einige Linux®-Dateisysteme. Dieser Abschnitt demonstriert, wie der Support aktiviert und die unterstützten Linux®-Dateisysteme eingehangen werden.

### 38.2.1. ext2

Seit FreeBSD 2.2 ist eine Kernel-Unterstützung für das ext2-Dateisystem vorhanden. In FreeBSD 8.x

und früheren Versionen wurde der Code noch unter der GPL lizensiert. Der Code wurde neu geschrieben und steht seit FreeBSD 9.0 unter der BSD-Lizenz.

Der ext2fs(5)-Treiber erlaubt dem FreeBSD Kernel sowohl Lese-, als auch Schreibzugriffe auf ext2-Dateisysteme.



Dieser Treiber kann auch für den Zugriff auf ext3 und ext4 Dateisysteme verwendet werden. Das Dateisystem ext2fs(5) bietet ab FreeBSD 12.0-RELEASE volle Lese- und Schreibunterstützung für ext4. Darüber hinaus werden auch erweiterte Attribute und ACLs unterstützt, jedoch kein Journaling und Verschlüsselung. Ab FreeBSD 12.1-RELEASE ist auch ein DTrace Provider verfügbar. Frühere Versionen von FreeBSD können mit sysutils/fusefs-ext2 auf ext4 im Lese- und Schreibmodus zugreifen.

Um auf ein ext-Dateisystem zuzugreifen, muss zuerst das entsprechende Kernelmodul geladen werden:

```
# kldload ext2fs
```

Mounten Sie anschließend das ext-Volume unter Angabe des FreeBSD Partitionsnamens und eines existierenden Mountpunktes. Dieses Beispiel hängt /dev/ad1s1 nach /mnt ein:

```
# mount -t ext2fs /dev/ad1s1 /mnt
```

# Kapitel 39. Virtualisierung

## 39.1. Übersicht

Virtualisierungssoftware erlaubt es, mehrere Betriebssysteme gleichzeitig auf dem selben Computer laufen zu lassen. Derartige Softwaresysteme für PCs setzen in der Regel ein Host-Betriebssystem voraus, auf dem die Virtualisierungssoftware läuft und unterstützen eine nahezu beliebige Anzahl von Gast-Betriebssystemen.

Nachdem Sie dieses Kapitel gelesen haben,

- Kennen Sie den Unterscheid zwischen einem Host-Betriebssystem und einem Gast-Betriebssystem.
- Können Sie FreeBSD auf einem Intel®-basierenden Apple® Mac® installieren.
- Können Sie FreeBSD unter Microsoft® Windows® und Virtual PC installieren.
- Wissen Sie, wie man ein virtualisiertes FreeBSD-System für optimale Leistung konfiguriert.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Die Grundlagen von UNIX® und FreeBSD verstehen.
- Wissen, wie Sie FreeBSD installieren können.
- Wissen, wie Sie eine Netzwerkverbindung konfigurieren.
- Wissen, wie Sie zusätzliche Software installieren können.

## 39.2. FreeBSD als Gast-Betriebssystem unter Parallels für Mac OS® X

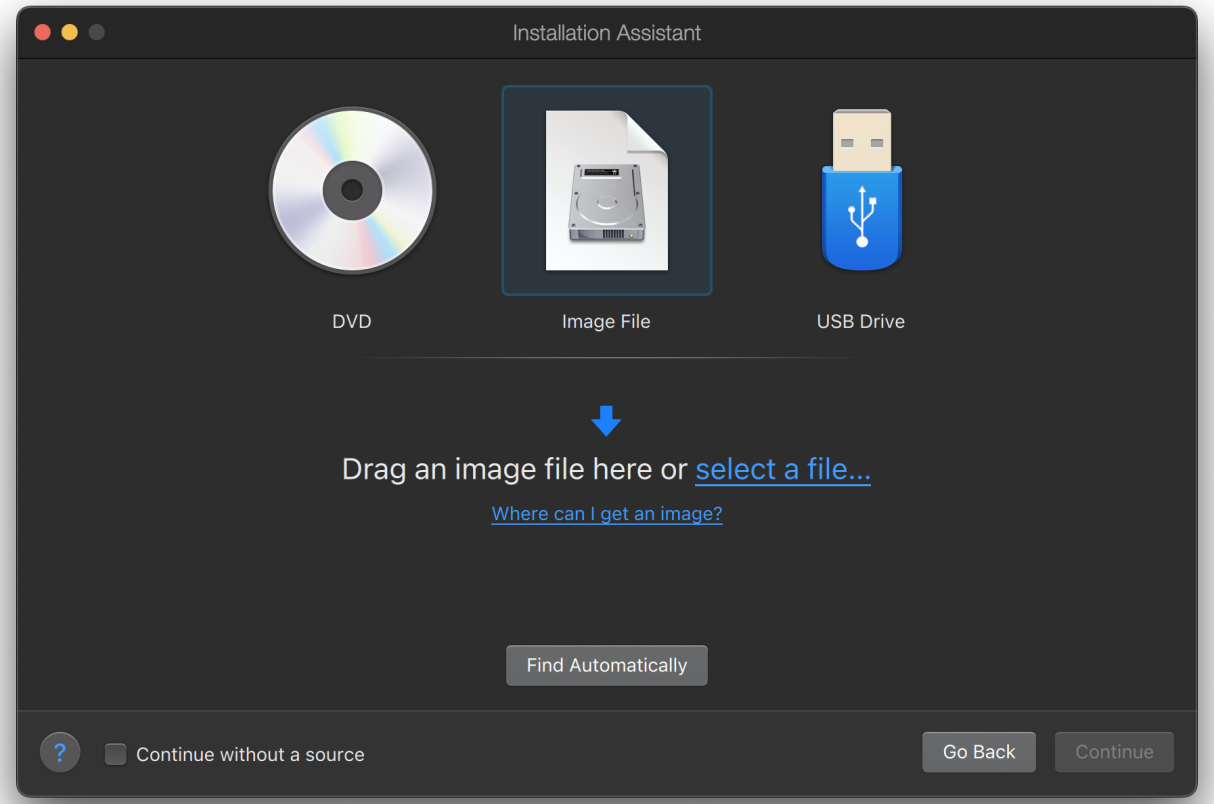
Parallels Desktop für Mac® ist ein kommerzielles Softwareprodukt, welches für Intel®-basierende Apple® Mac®-Computer mit Mac OS® X 10.4.6 oder höher verfügbar ist. FreeBSD wird von diesem Softwarepaket als Gast-Betriebssystem vollständig unterstützt. Nach der Installation von Parallels auf Mac OS® X konfigurieren Sie als erstes eine virtuelle Maschine, in der Sie danach das gewünschte Gast-Betriebssystem (in diesem Fall FreeBSD) installieren.

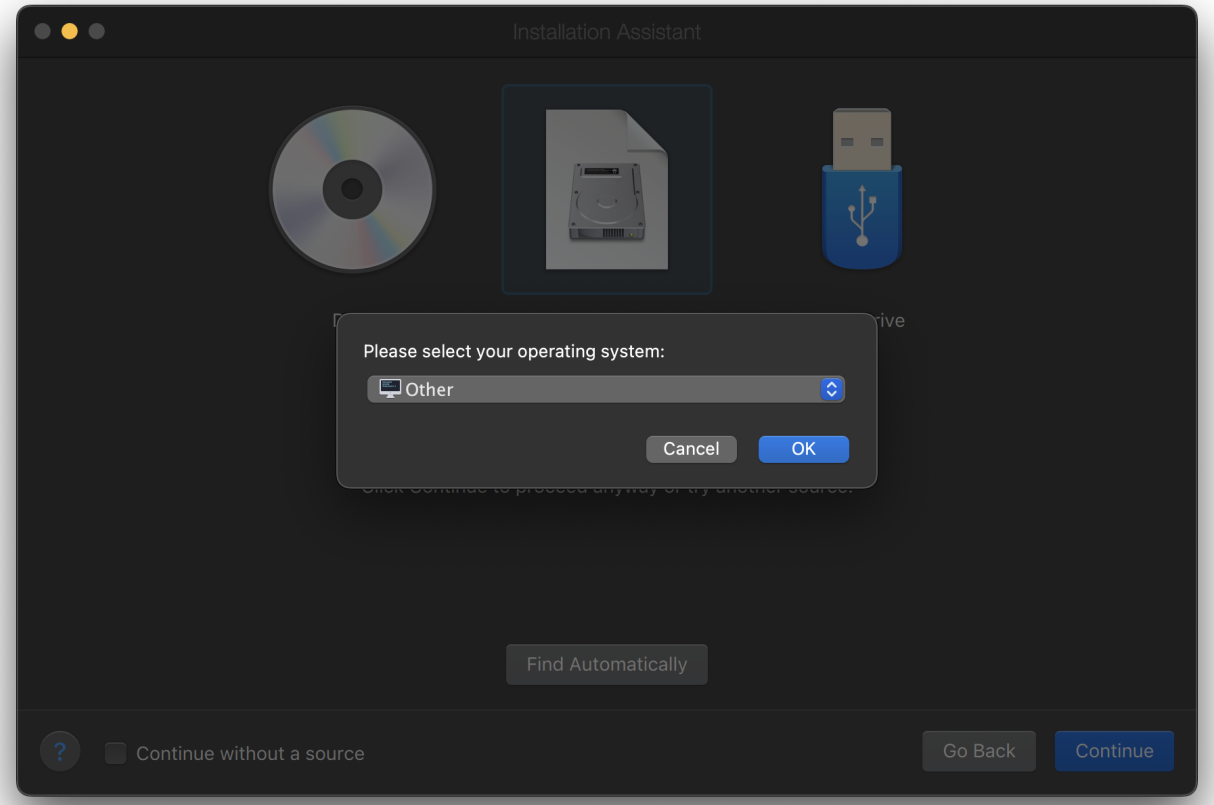
### 39.2.1. Installation von FreeBSD unter Parallels/Mac OS® X

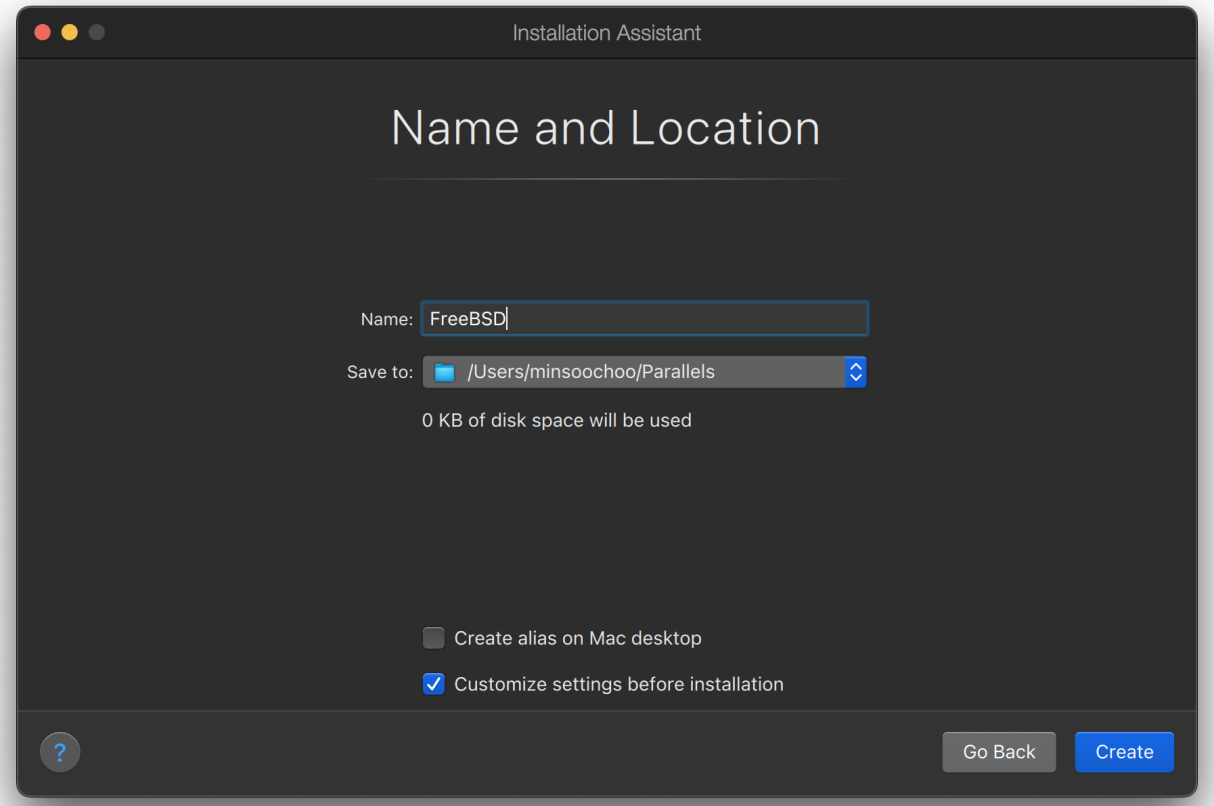
Der erste Schritt bei der Installation von FreeBSD unter Parallels ist es, eine virtuelle Maschine zu konfigurieren, in der Sie FreeBSD installieren können. Dazu wählen Sie bei der Frage nach dem **Guest OS Type**FreeBSD aus:

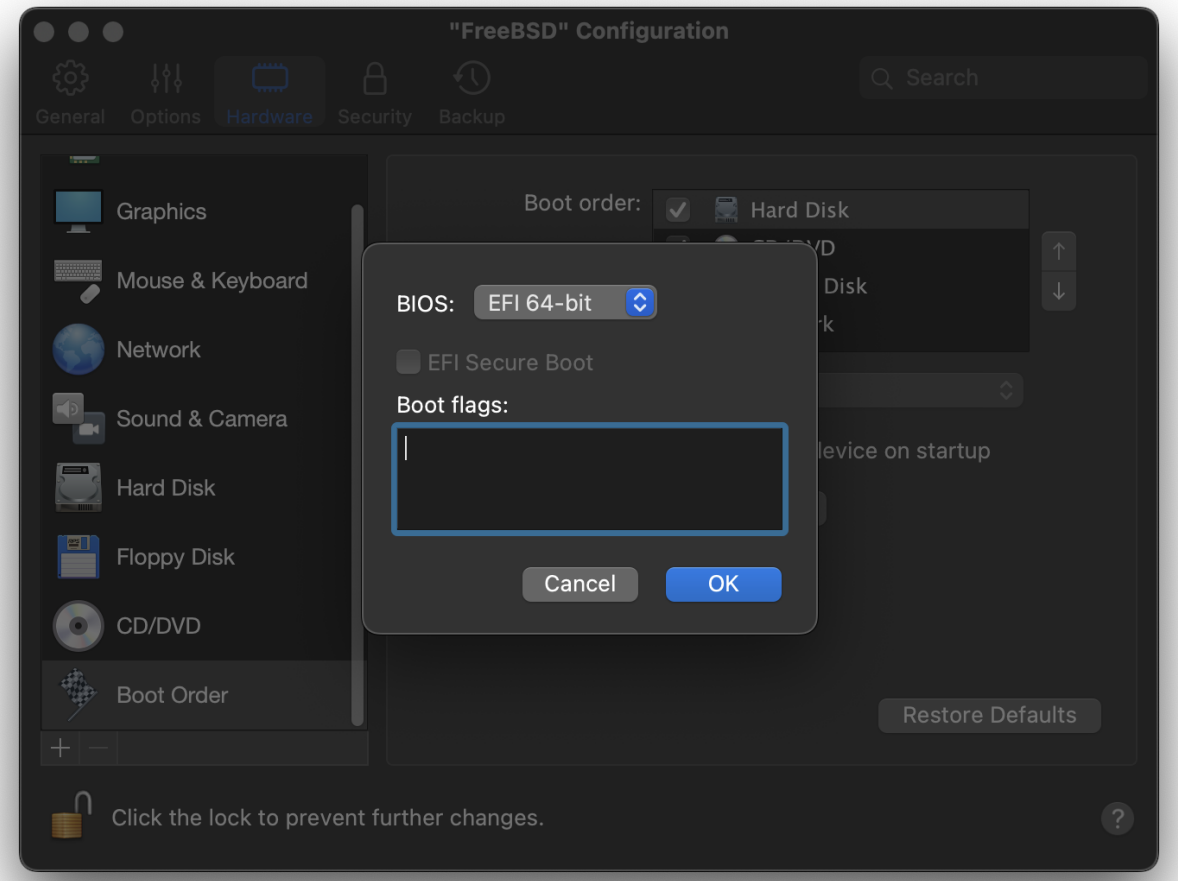



Legen Sie geeignete Größen für Festplatten- und Arbeitsspeicher für die zu erstellende FreeBSD-Instanz fest. 4 GB Plattenplatz sowie 512 MB RAM sind in der Regel für die Arbeit unter Parallels ausreichend:

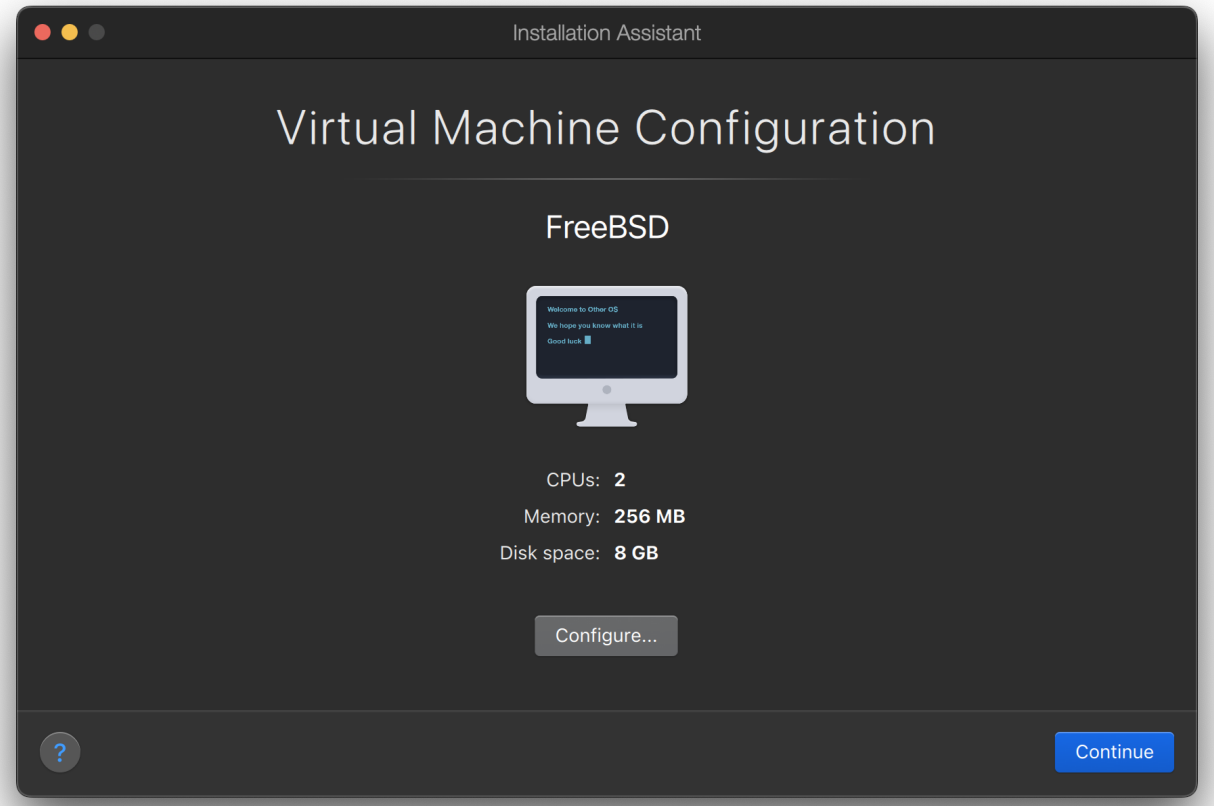

Installation Assistant
DVD
Image File
USB Drive
Drag an image file here or select a file...
Where can I get an image?
Find Automatically
Continue without a source
Go Back
Continue


Installation Assistant
Please select your operating system:
Other
Cancel
OK
Find Automatically
Continue without a source
Go Back
Continue


Installation Assistant
Name and Location
Name:
FreeBSD
Save to:
/Users/minsoochoo/Parallels
0 KB of disk space will be used
Create alias on Mac desktop
Customize settings before installation
Go Back
Create

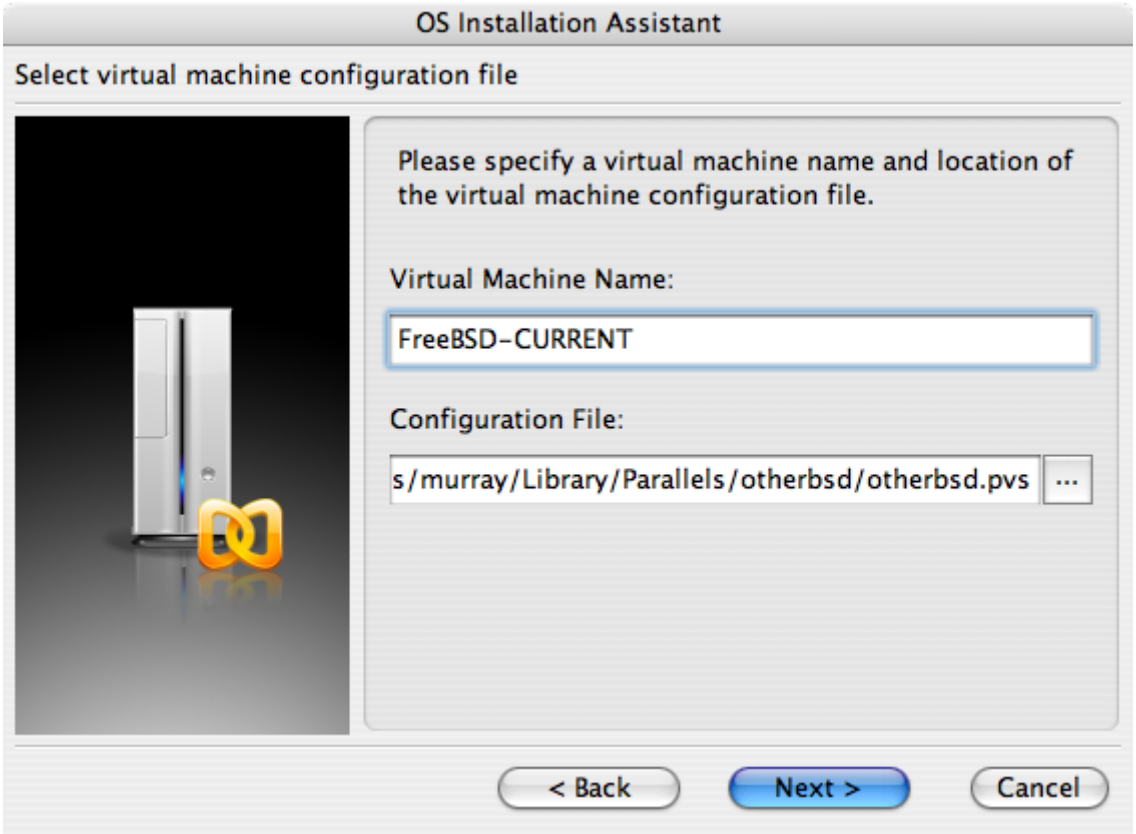


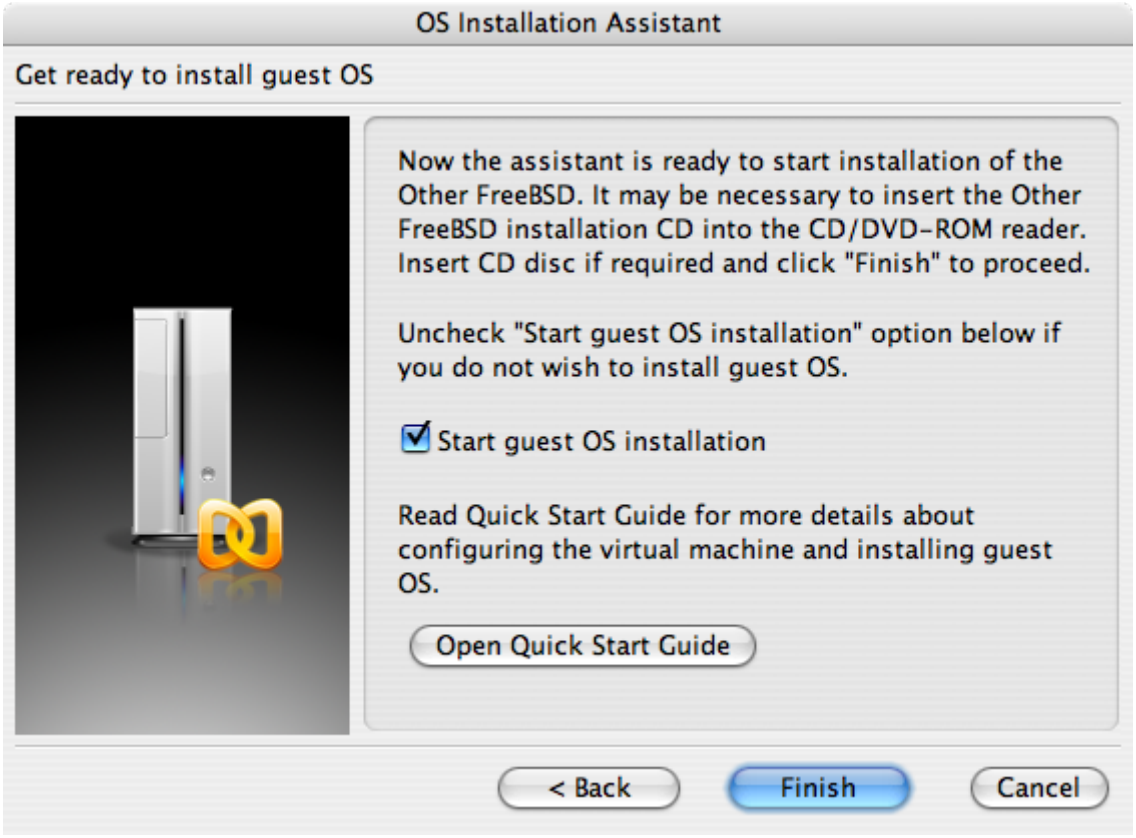

Wählen Sie den gewünschten Netzwerktyp aus und konfigurieren Sie die Netzwerkverbindung:


Installation Assistant
Virtual Machine Configuration
FreeBSD
Welcome to Other OS
We hope you know what it is
Good luck
CPUs: 2
Memory: 256 MB
Disk space: 8 GB
Configure...
?
Continue

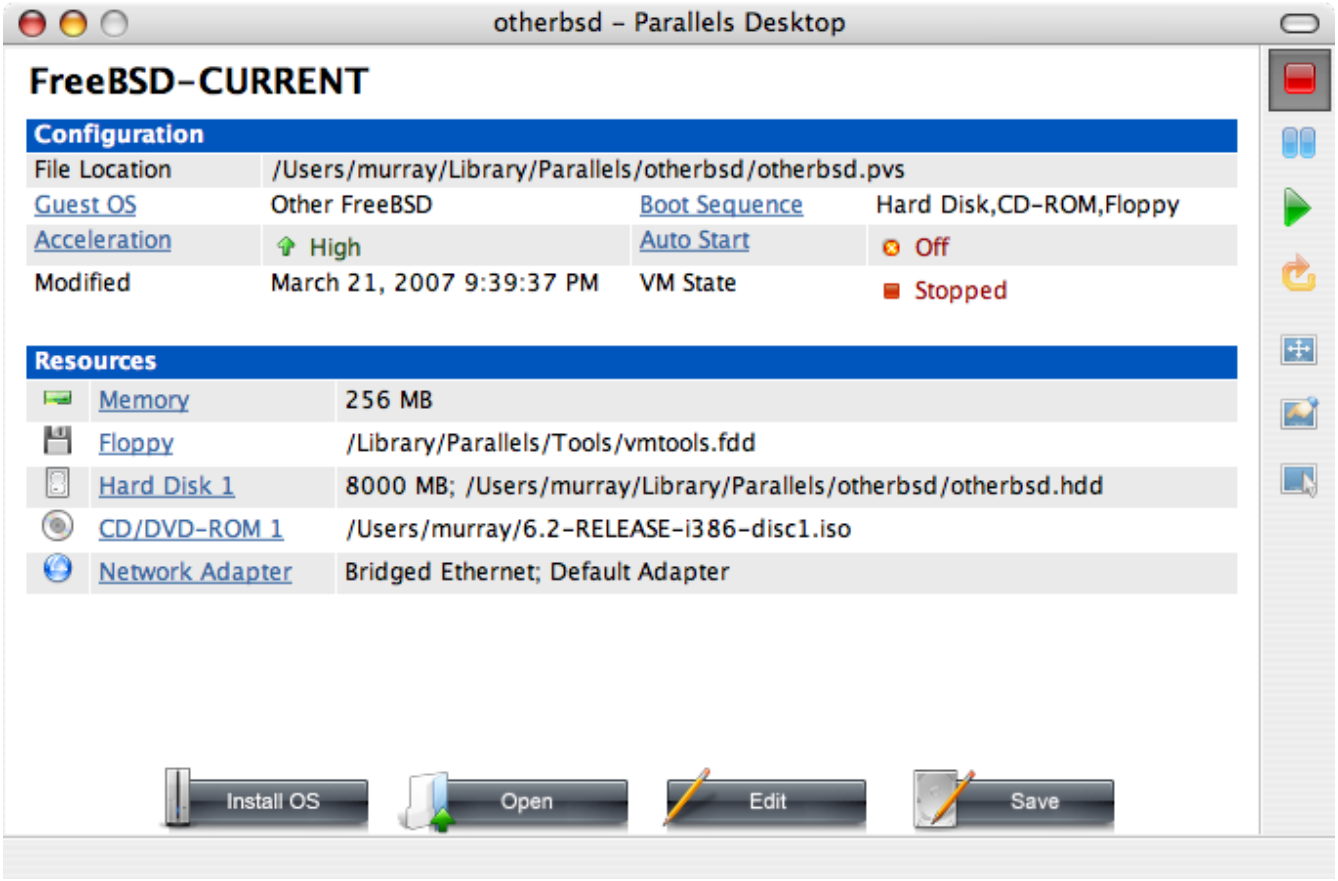



Speichern Sie Ihre Eingaben, um die Konfiguration abzuschließen:

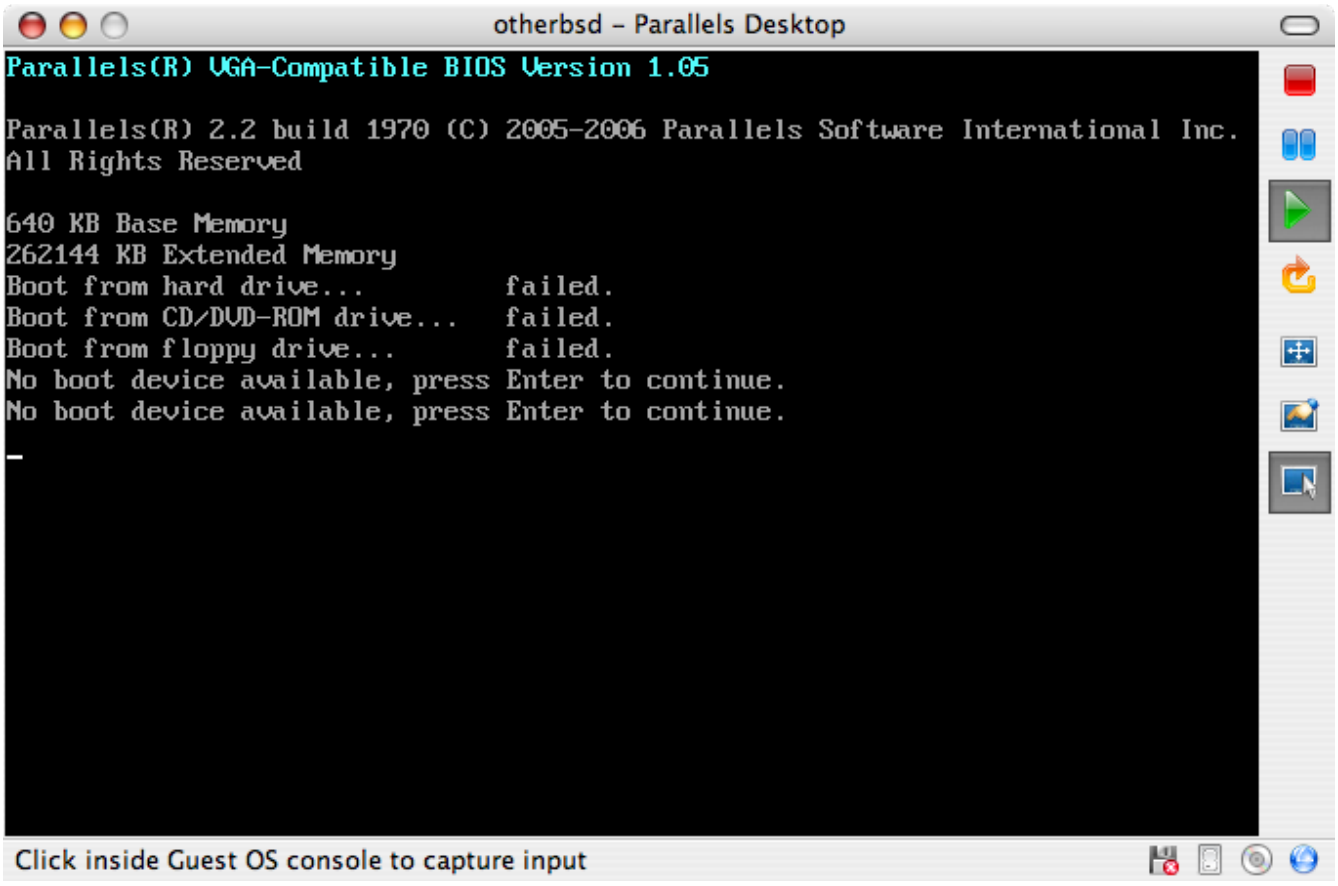





Nachdem Sie die virtuelle Maschine erstellt haben, installieren Sie im nächsten Schritt FreeBSD in dieser virtuellen Maschine. Dazu verwenden Sie am besten eine offizielle FreeBSD-CD/DVD oder Sie laden von einem offiziellen FTP-Server ein ISO-Abbild auf Ihren Mac® herunter. Danach klicken Sie auf das Laufwerksymbol in der rechten unteren Ecke des Parallels-Fensters, um das virtuelles Laufwerk mit dem ISO-Abbild oder mit dem physikalischen CD-ROM-Laufwerk des Computers zu verknüpfen.

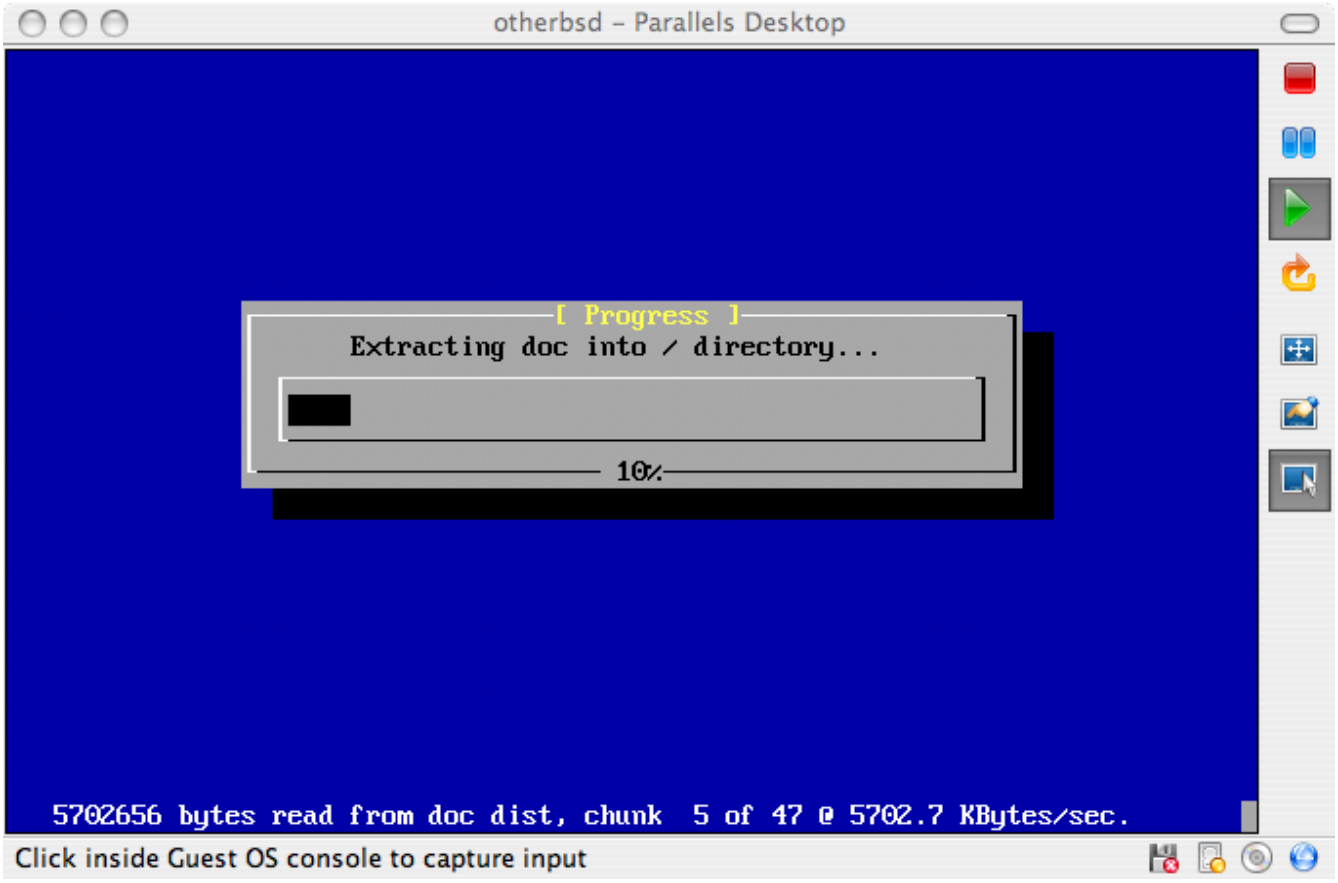



Nachdem Sie diese Verknüpfung hergestellt haben, starten sie die virtuelle FreeBSD-Maschine neu, indem Sie auf das Symbol "Neustarten" klicken. Parallels startet nun ein Spezial-BIOS, das zuerst prüft, ob eine CD-ROM eingelegt wurde.

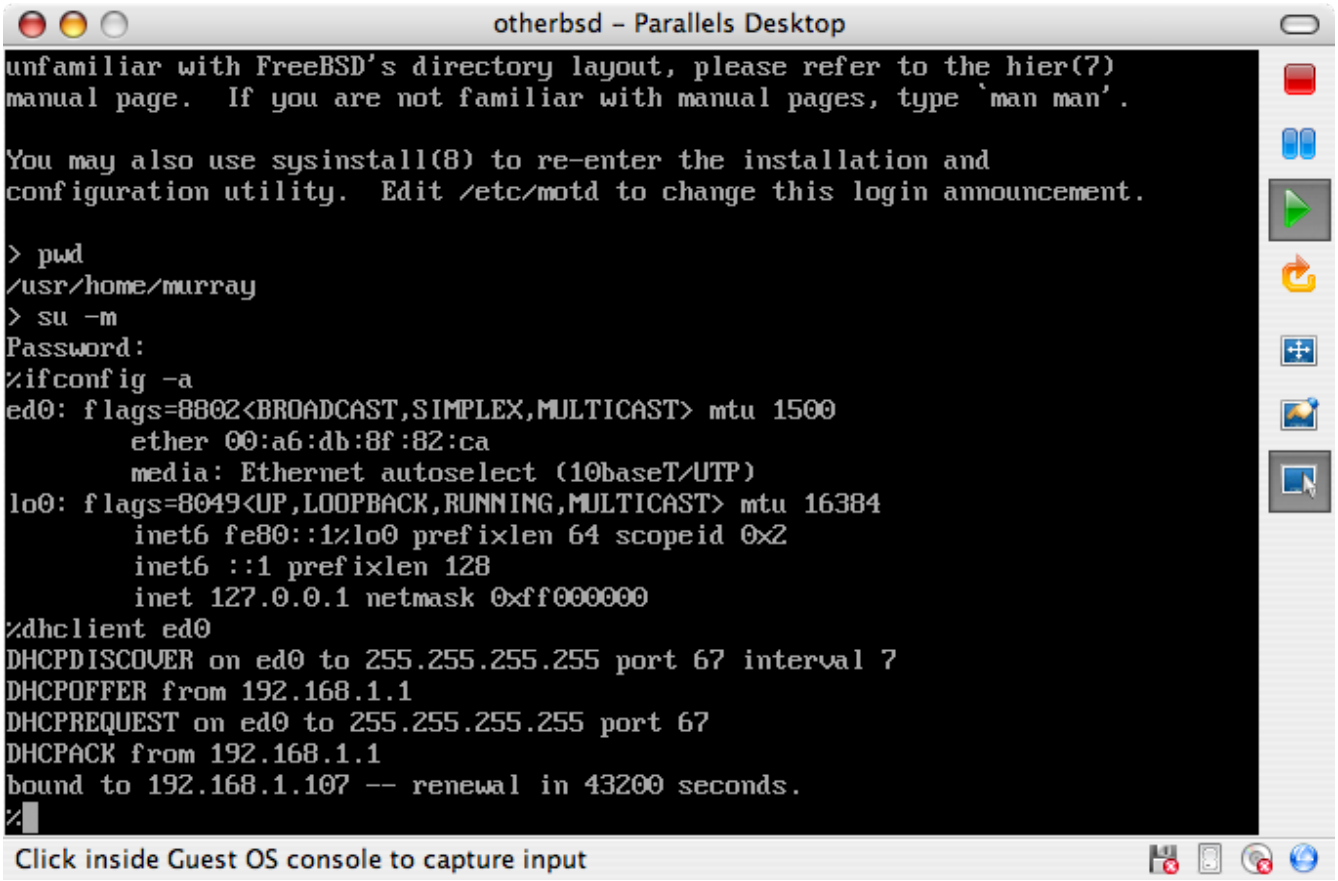



In diesem Fall findet das BIOS ein FreeBSD-Installationsmedium und beginnt eine normale

Installation. Versuchen Sie jetzt noch nicht Xorg zu konfigurieren.




Nachdem die Installation abgeschlossen ist, können Sie die virtuelle FreeBSD-Maschine starten.



### 39.2.2. FreeBSD für den Einsatz unter Parallels konfigurieren

Nachdem FreeBSD erfolgreich unter Mac OS® X mit Parallels installiert wurde, sollten Sie das virtuelle FreeBSD-System für virtualisierte Operationen optimieren:

1. Setzen der Bootloader-Variablen

   Die wichtigste Änderung ist es, die Variable `kern.hz` zu verkleinern, um so die CPU-Auslastung in der Parallels-Umgebung zu verringern.

   ```
   kern.hz=100
   ```

   Ohne diese Einstellung kann ein unbeschäftigtes FreeBSD unter Parallels trotzdem rund 15 Prozent der CPU-Leistung eines Single Prozessor iMac®'s verbrauchen. Nach dieser Änderung reduziert sich dieser Wert auf etwa 5 Prozent.

2. Erstellen einer neuen Kernelkonfigurationsdatei

   Sie können alle SCSI-, FireWire- und USB-Laufwerks-Treiber entfernen. Parallels stellt einen virtuellen Netzwerkadapter bereit, der den ed(4)-Treiber verwendet. Daher können alle Netzwerkgeräte bis auf ed(4) und miibus(4) aus dem Kernel entfernt werden.

3. Netzwerkbetrieb einrichten

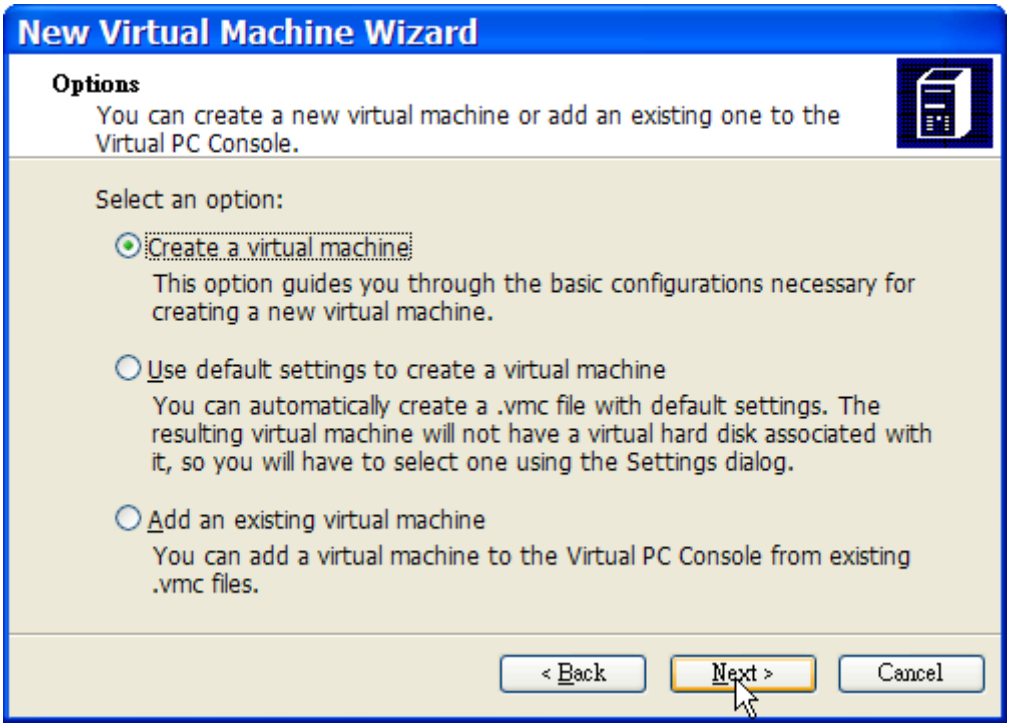
   Die einfachste Netzwerkkonfiguration ist der Einsatz von DHCP, um die virtuelle Maschine mit dem gleichen lokalen Netzwerk, in dem sich der Host-Mac® befindet, zu verbinden. Dazu fügen Sie die Zeile `ifconfig_ed0="DHCP"` in /etc/rc.conf ein. Weitere Informationen zur Konfiguration des Netzwerks unter FreeBSD finden Sie im Netzwerkverbindung konfigurieren.

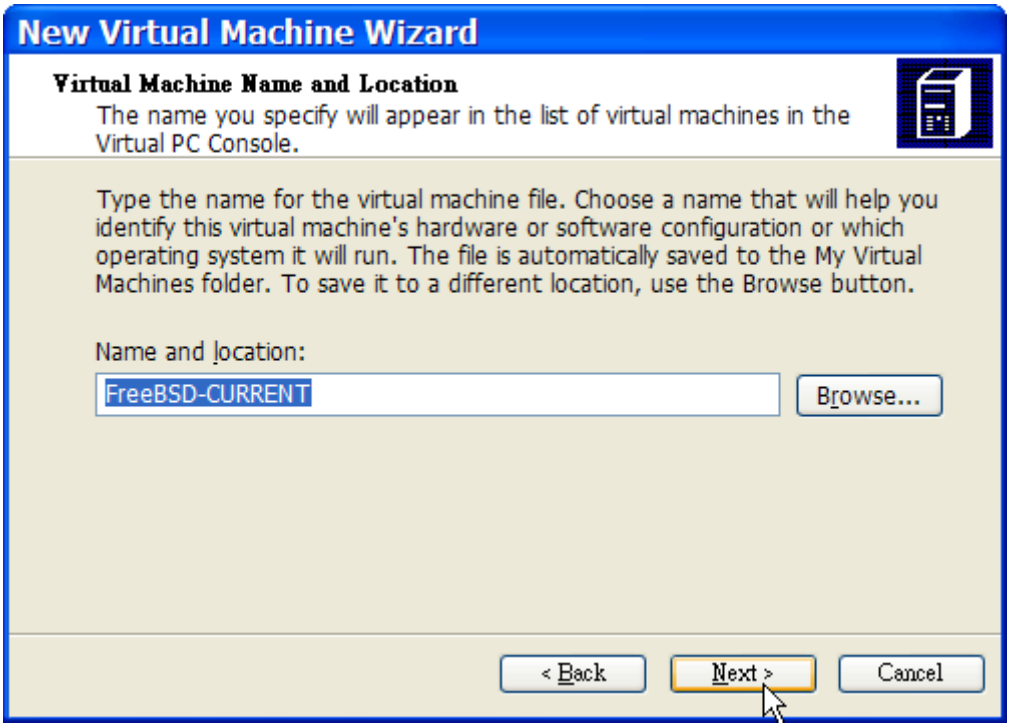
## 39.3. FreeBSD als Gast-Betriebssystem unter Virtual PC für Windows®

Virtual PC für Windows® wird von Microsoft® kostenlos zum Download angeboten. Die Systemanforderungen für dieses Programm finden Sie hier. Nachdem Virtual PC unter Microsoft® Windows® installiert wurde, muss eine virtuelle Maschine konfiguriert und das gewünschte Betriebssystem installiert werden.

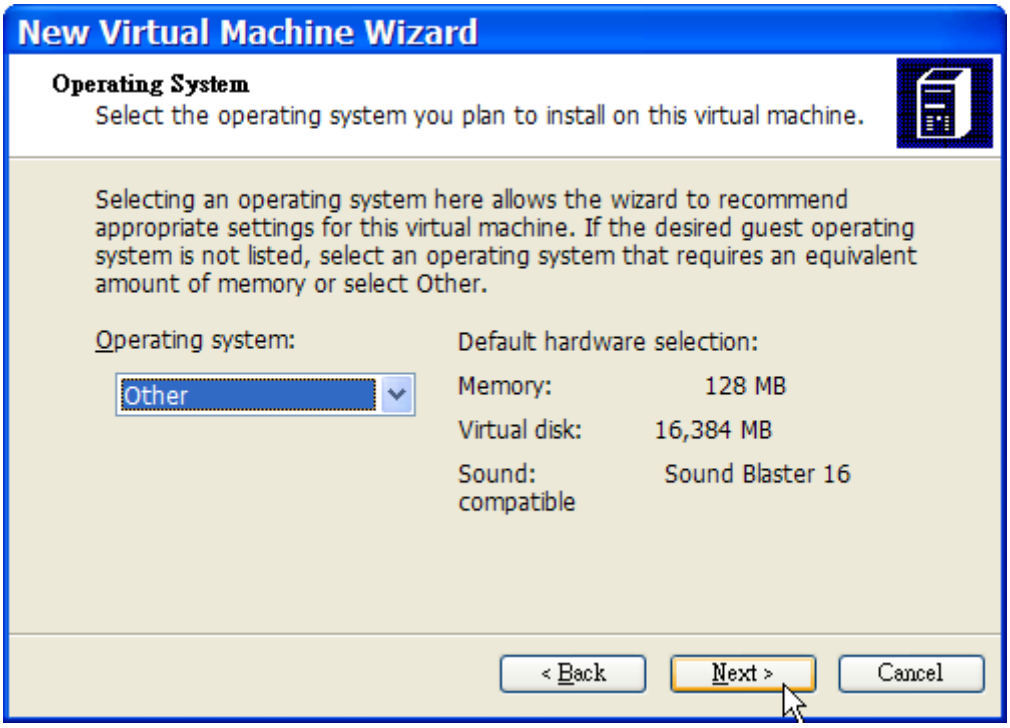
### 39.3.1. FreeBSD in Virtual PC installieren

Der erste Schritt zur Installation von FreeBSD in Virtual PC ist es, eine neue virtuelle Maschine zu erstellen, in die Sie FreeBSD installieren können. Dazu wählen Sie die Option Create a virtual machine, wenn Sie danach gefragt werden:

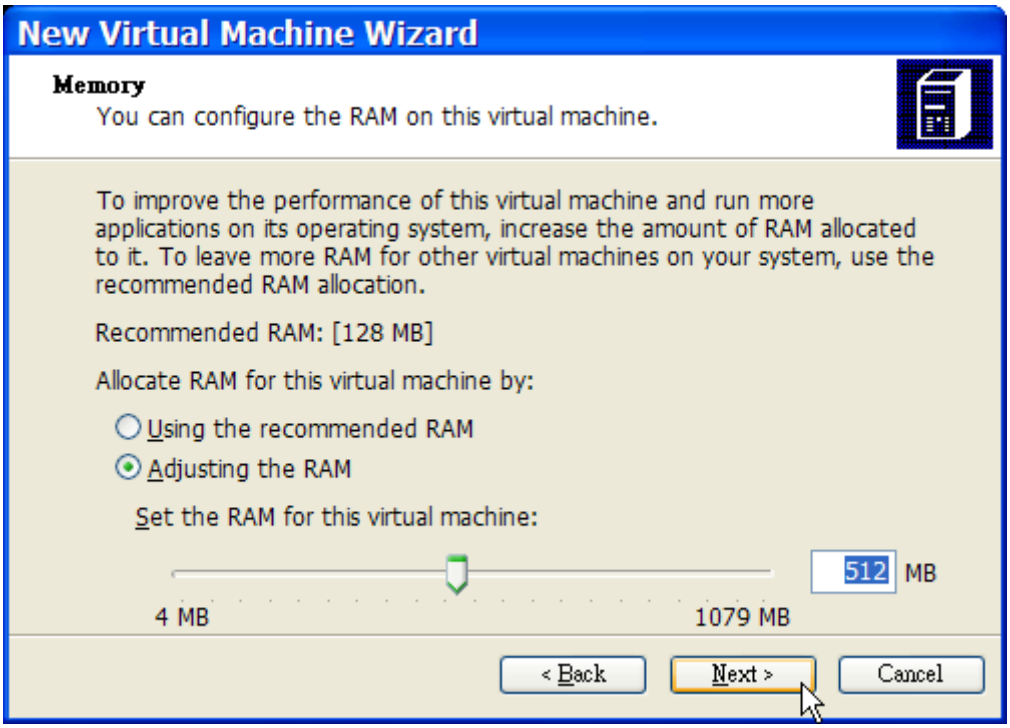


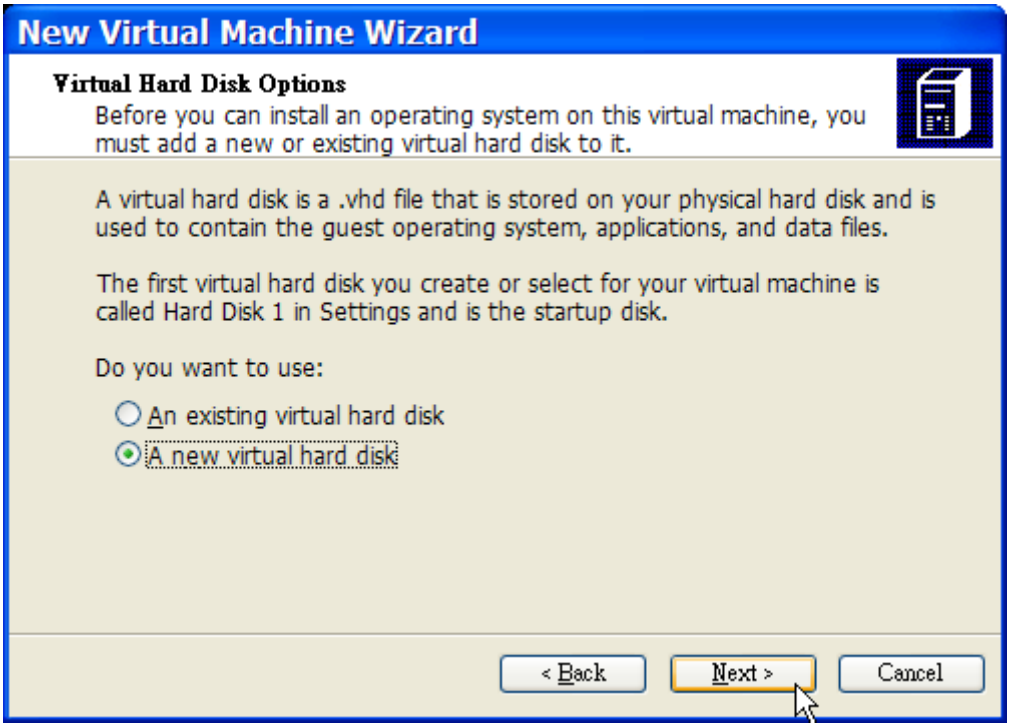




Bei der Frage nach dem Operating system wählen Sie Other:

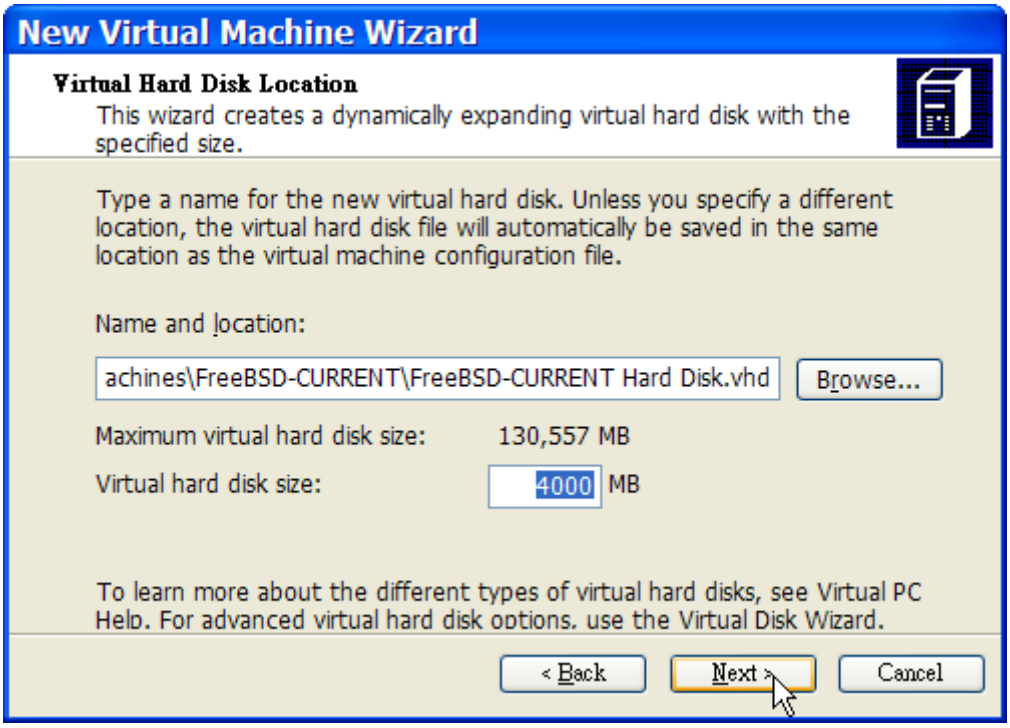



Danach müssen Sie den gewünschten Plattenplatz sowie die Größe des Hauptspeichers angeben. 4 GB Plattenplatz sowie 512 MB RAM sollten für die Installation von FreeBSD in Virtual PC ausreichend sein:

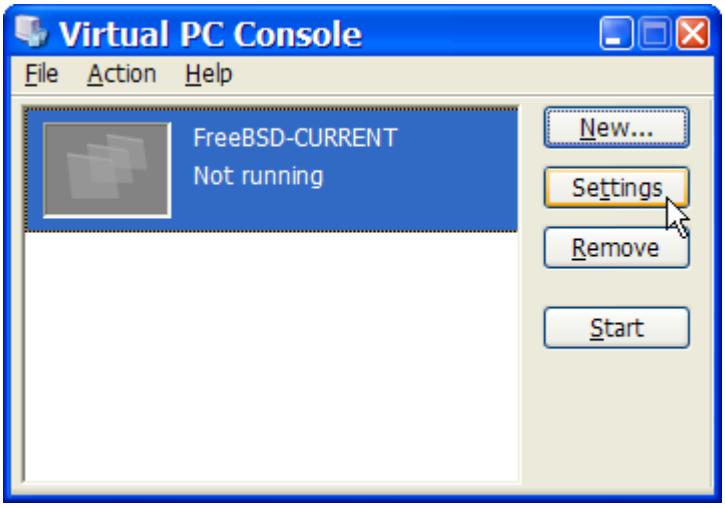


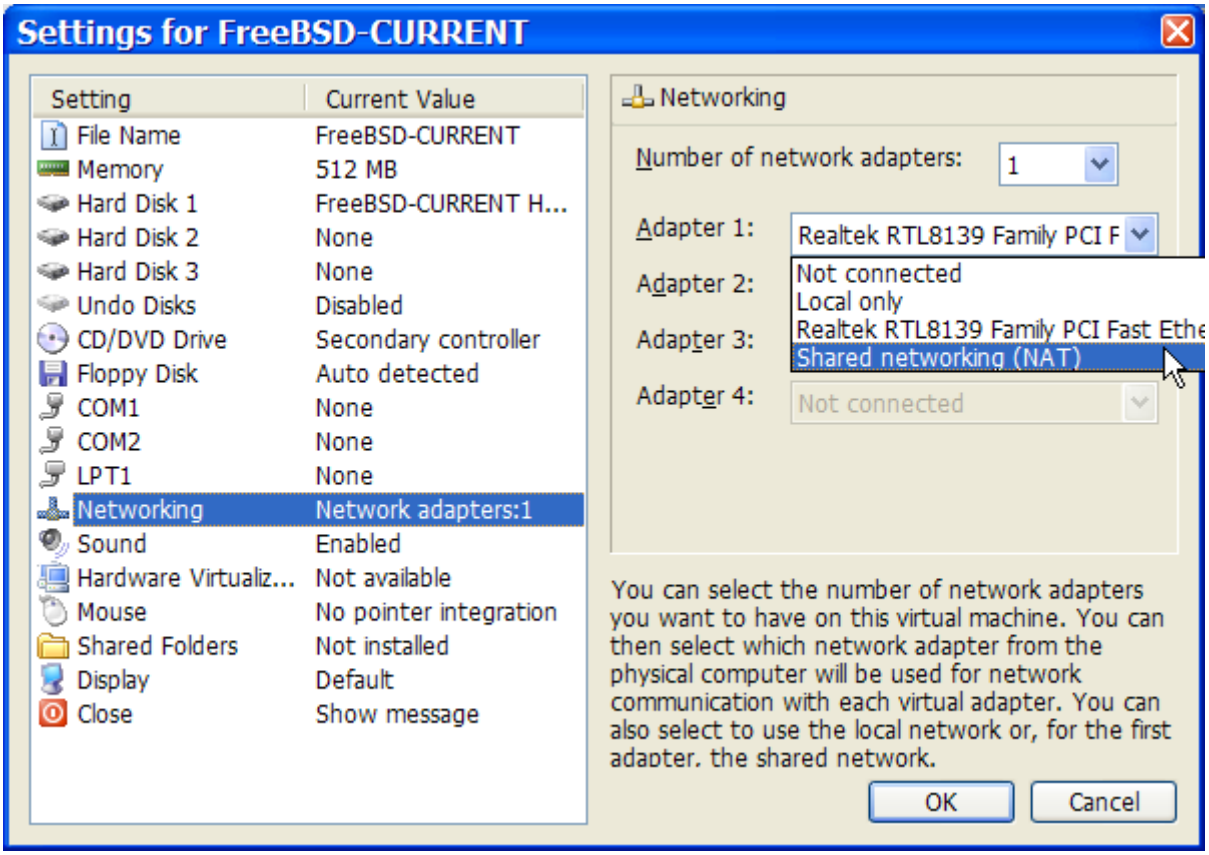



Speichern Sie die Eingaben und beenden Sie die Konfiguration:

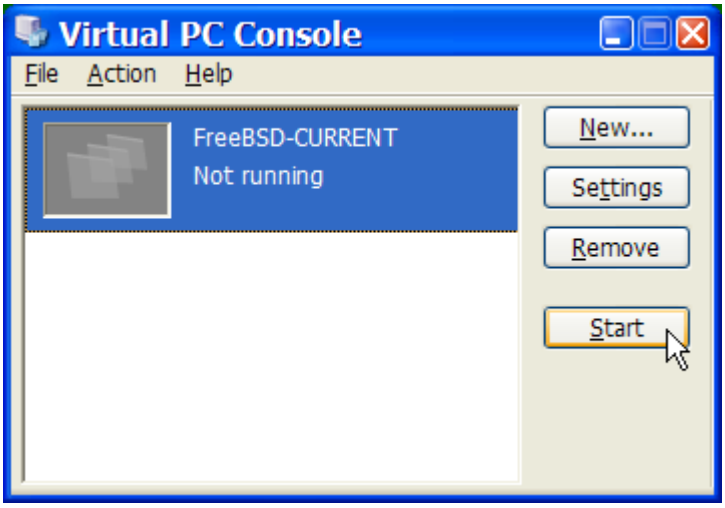


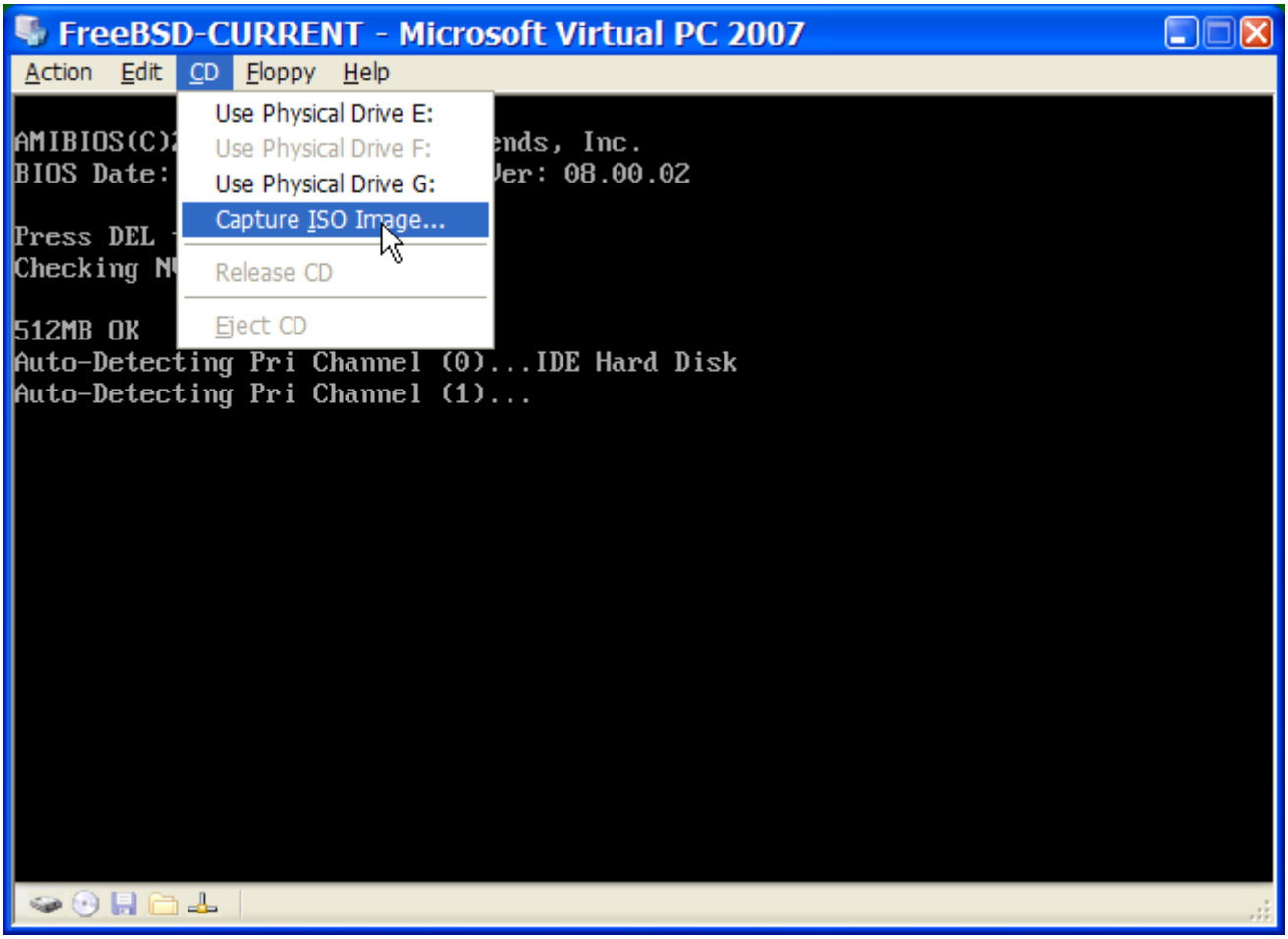

Wählen Sie nun die für FreeBSD erstellte virtuelle Maschine aus und klicken Sie auf **Settings**, um das Netzwerk zu konfigurieren:



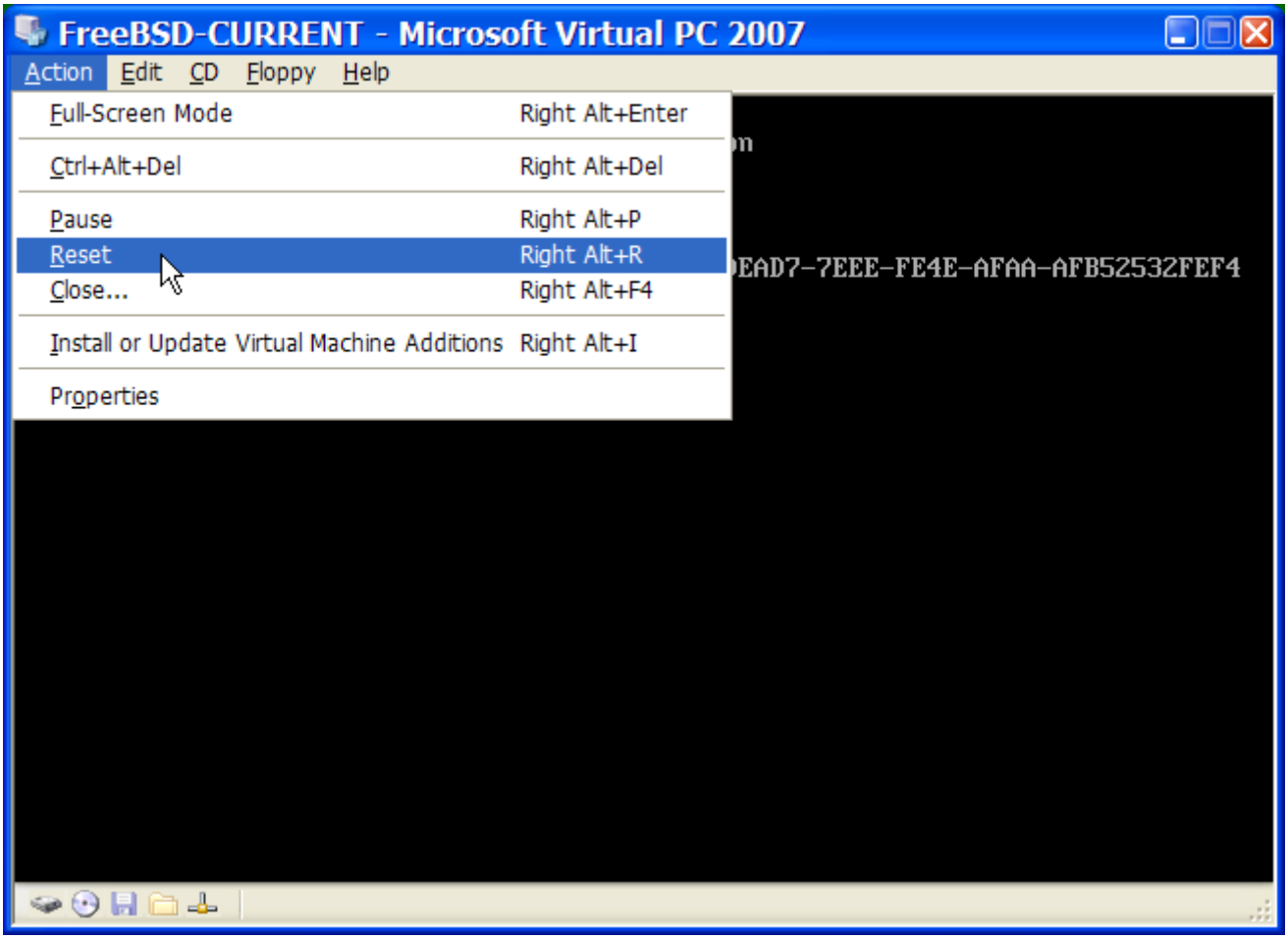




Nachdem die virtuelle Maschine erstellt wurde, können Sie FreeBSD installieren. Dazu verwenden Sie am besten eine offizielle FreeBSD-CD/DVD oder ein ISO-Image, das Sie von einem offiziellen FreeBSD-FTP-Server heruntergeladen haben. Wenn Sie ein ISO-Image auf der Festplatte gespeichert haben, oder eine FreeBSD-CD/DVD in das Laufwerk eingelegt haben, doppelklicken Sie auf die virtuelle Maschine, die Sie für FreeBSD angelegt haben. Danach klicken Sie auf **CD** und wählen die Option **Capture ISO Image...** im Virtual PC-Fenster. Danach können Sie im folgenden Fenster das CD-Laufwerk mit dem physikalischen CD-Laufwerk oder mit dem ISO-Image verknüpfen.

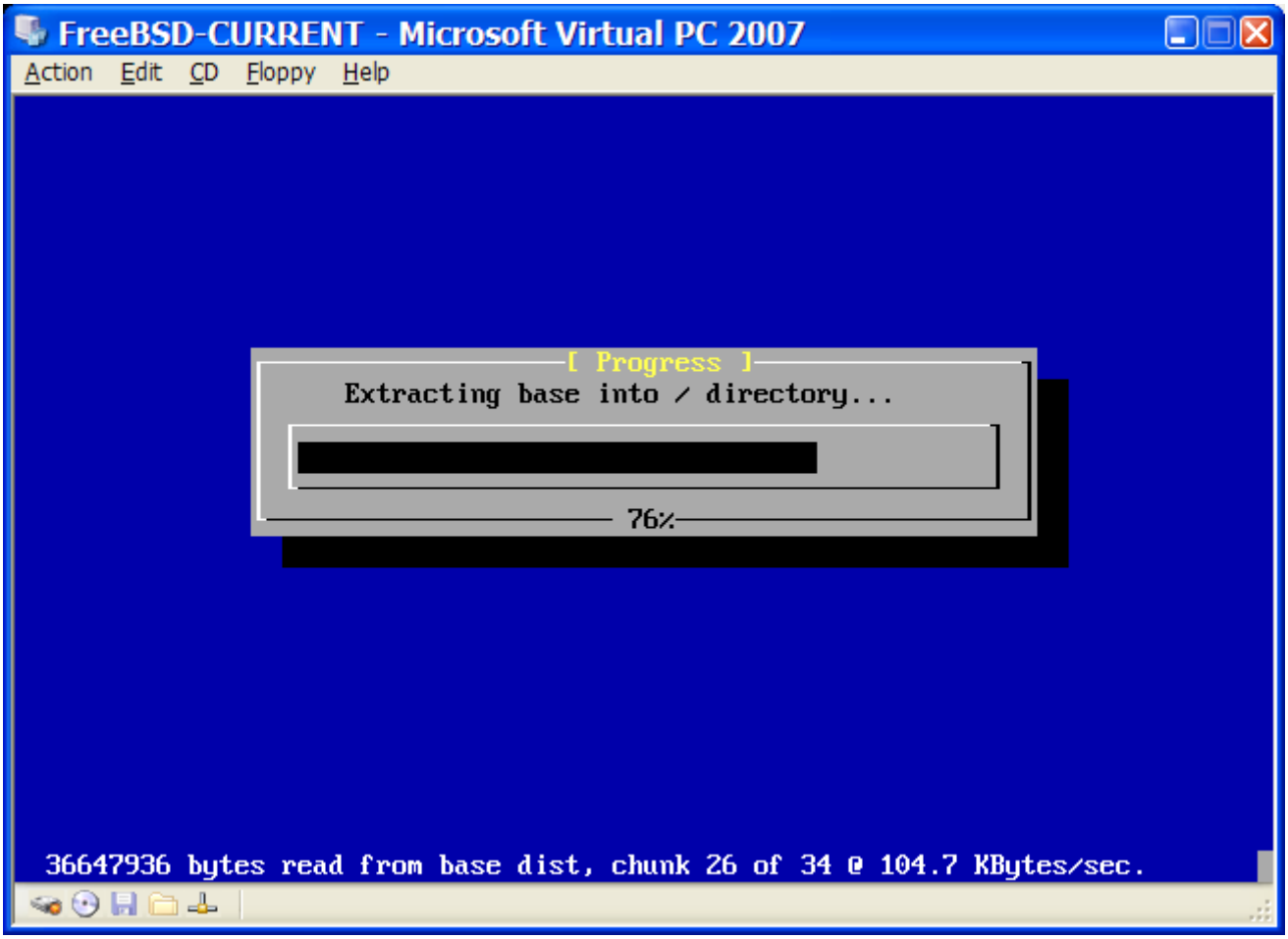


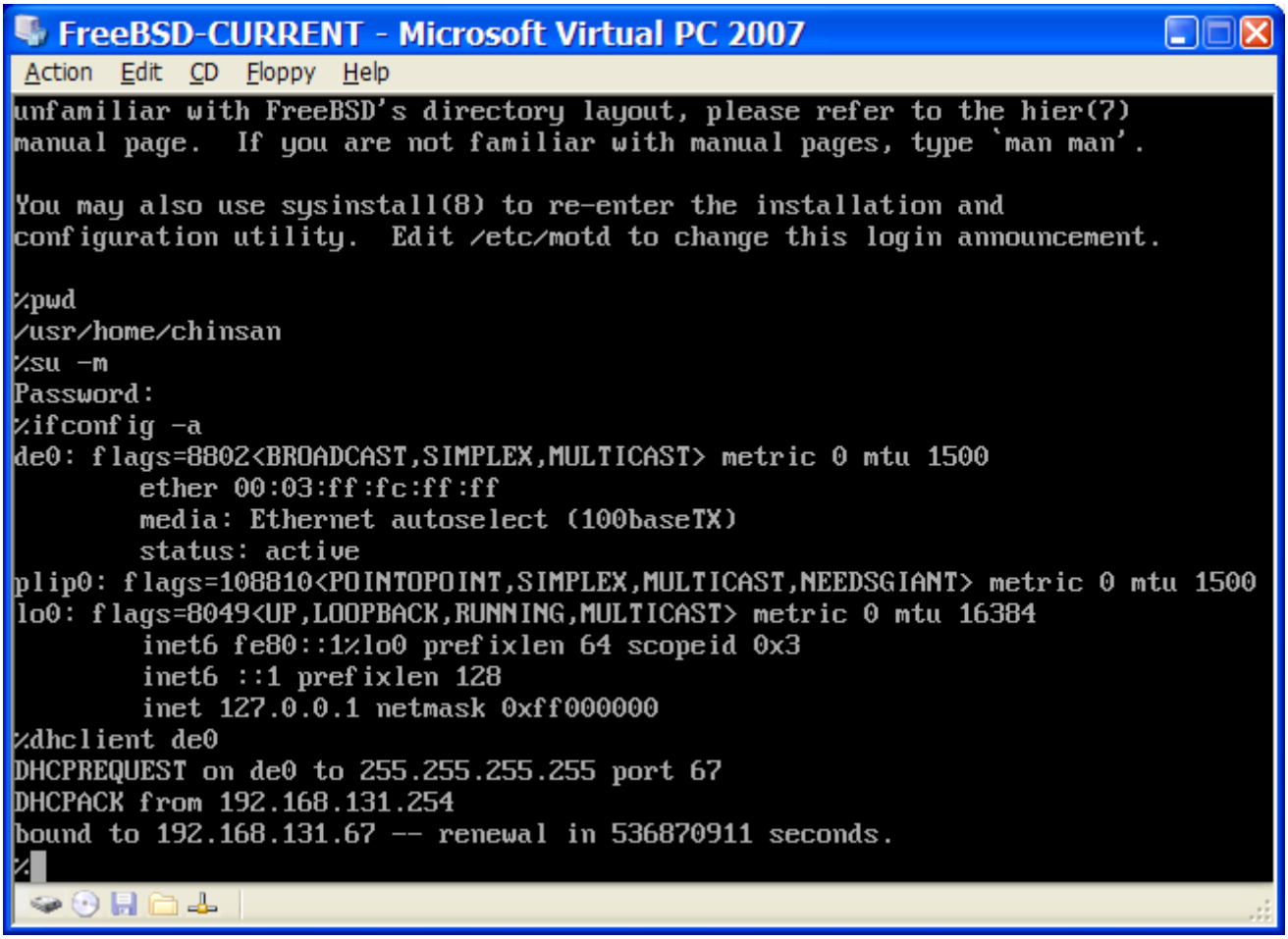

Danach starten Sie die virtuelle Maschine neu, indem Sie zuerst auf **Action** und danach auf **Reset** klicken. Virtual PC startet die virtuelle Maschine nun neu und prüft zuerst, ob die virtuelle Maschine über ein CD-Laufwerk verfügt.




Da dies hier der Fall ist, beginnt nun eine normale FreeBSD-Installation. Sie können FreeBSD nun installieren, aber verzichten Sie an dieser Stelle unbedingt auf die Xorg-Konfiguration.




Nachdem die Installation abgeschlossen ist, entfernen Sie die CD/DVD aus dem Laufwerk (oder lösen die Verknüpfung zum ISO-Image). Danach starten Sie die virtuelle Maschine neu, um FreeBSD zu starten.




### 39.3.2. FreeBSD in Virtual PC konfigurieren

Nachdem FreeBSD auf Microsoft® Windows® mit Virtual PC erfolgreich installiert wurde, sollten Sie das virtuelle FreeBSD noch anpassen, um eine optimale Funktion zu gewährleisten.

1. Setzen der Bootloader-Variablen

   Die wichtigste Änderung ist es, die Variable `kern.hz` zu verkleinern, um so die CPU-Auslastung in der Virtual PC-Umgebung zu verringern. Dazu fügen Sie die folgende Zeile in /boot/loader.conf ein:

   ```
   kern.hz=100
   ```

   Ohne diese Einstellung kann ein unbeschäftigtes FreeBSD unter Virutal PC trotzdem rund 40 Prozent der CPU-Leistung eines Ein-Prozessor-Systems verbrauchen. Nach dieser Änderung reduziert sich dieser Wert auf etwa 3 Prozent.

2. Erstellen einer neuen Kernelkonfigurationsdatei

   Alle SCSI-, FireWire- und USB-Laufwerks-Treiber können aus der Kernelkonfigurationsdatei entfernt werden. Virtual PC stellt einen virtuellen Netzwerkadapter bereit, der den de(4) -Treiber verwendet. Daher können alle Netzwerkgeräte bis auf de(4) und miibus(4) aus dem Kernel entfernt werden.

3. Das Netzwerk einrichten

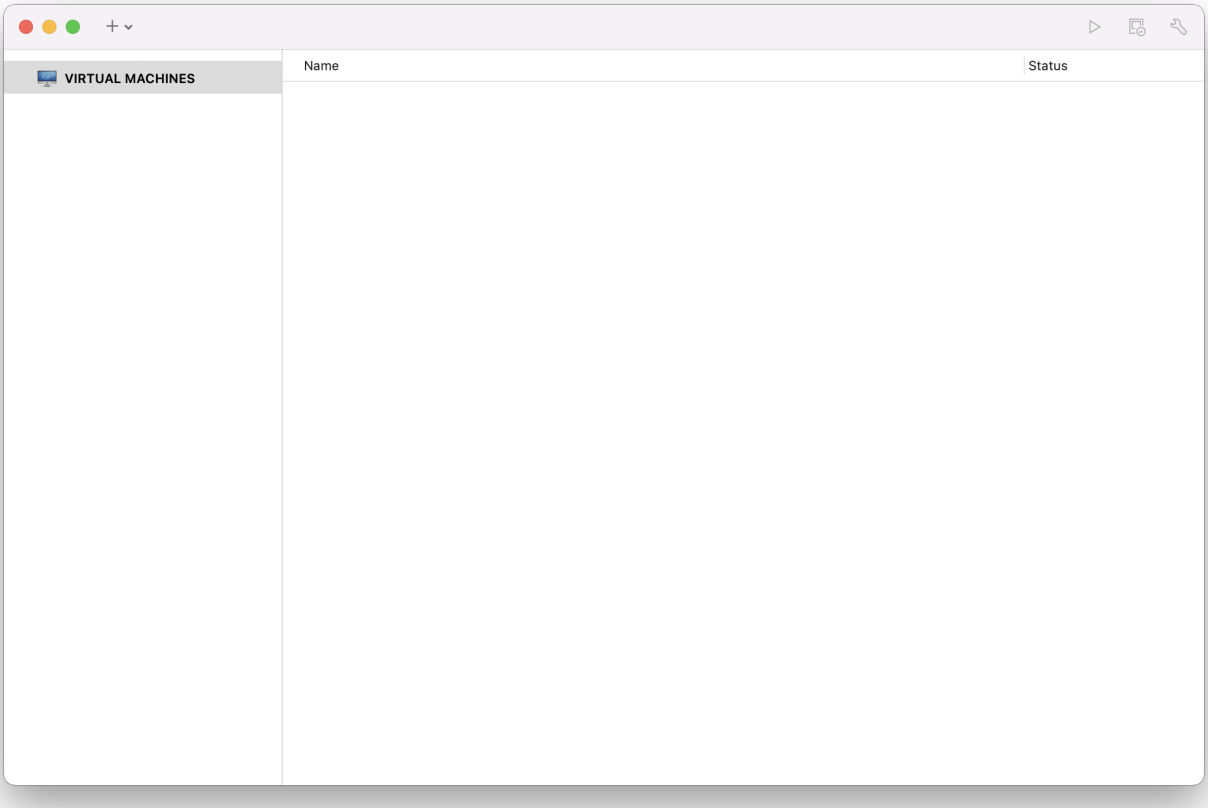
Die einfachste Netzwerkkonfiguration nutzt von DHCP, um die virtuelle Maschine mit dem gleichen lokalen Netzwerk, in dem sich der Host-Microsoft® Windows® befindet, zu verbinden. Dazu fügen Sie die Zeile `ifconfig_de0="DHCP"` in /etc/rc.conf ein. Weitere Informationen zur Konfiguration des Netzwerks unter FreeBSD finden Sie in Netzwerkverbindung konfigurieren.

# 39.4. FreeBSD als Gast-Betriebssystem unter VMware Fusion für Mac OS®

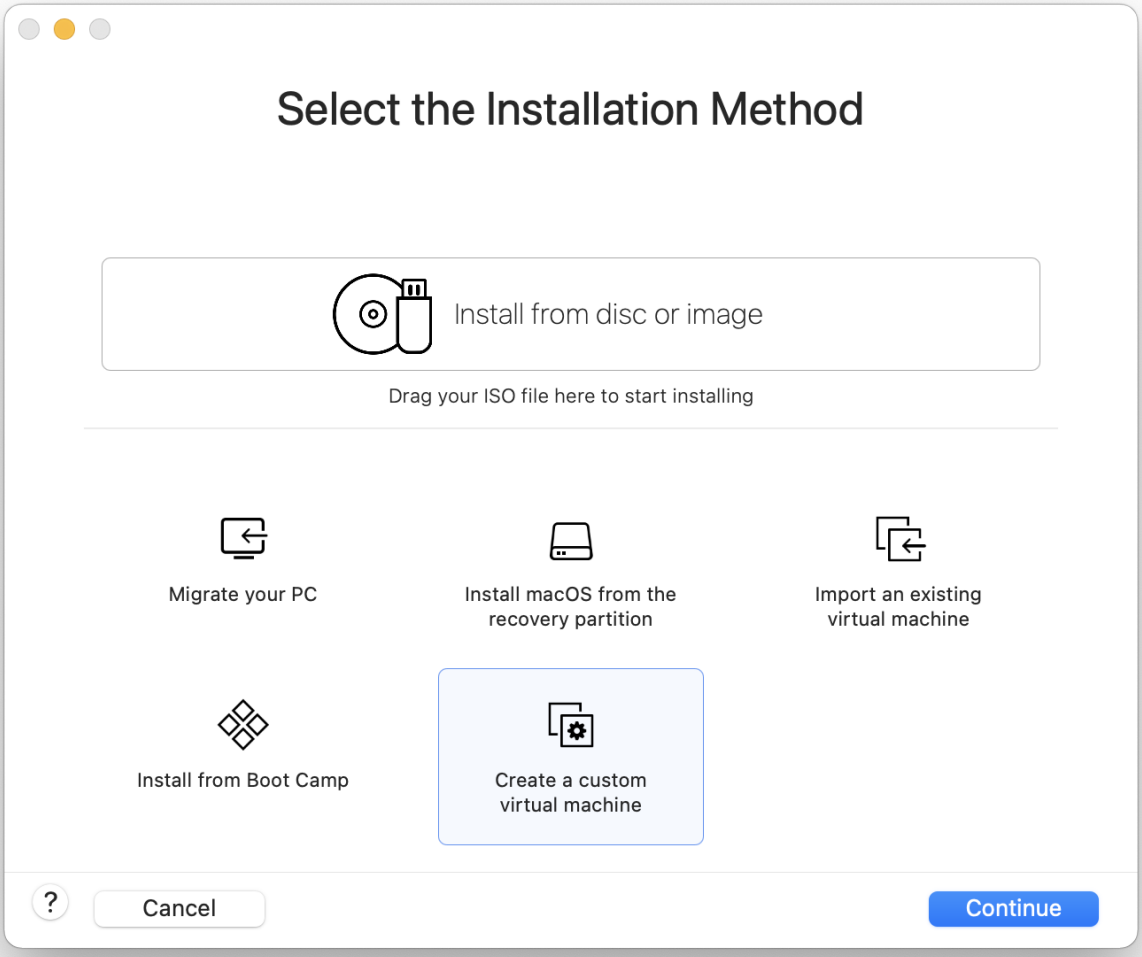
VMware Fusion für Mac® ist ein kommerzielles Programm, das für Intel® basierte Apple® Mac®-Computer mit Mac OS® 10.4.9 oder neuer erhältlich ist. FreeBSD wird von diesem Produkt vollständig als Gast-Betriebssystem unterstützt. Nachdem Sie VMware Fusion unter Mac OS® X installiert haben, können Sie eine virtuelle Maschine konfigurieren und das gewünschte Gastbetriebssystem installieren.

## 39.4.1. FreeBSD in VMware Fusion installieren

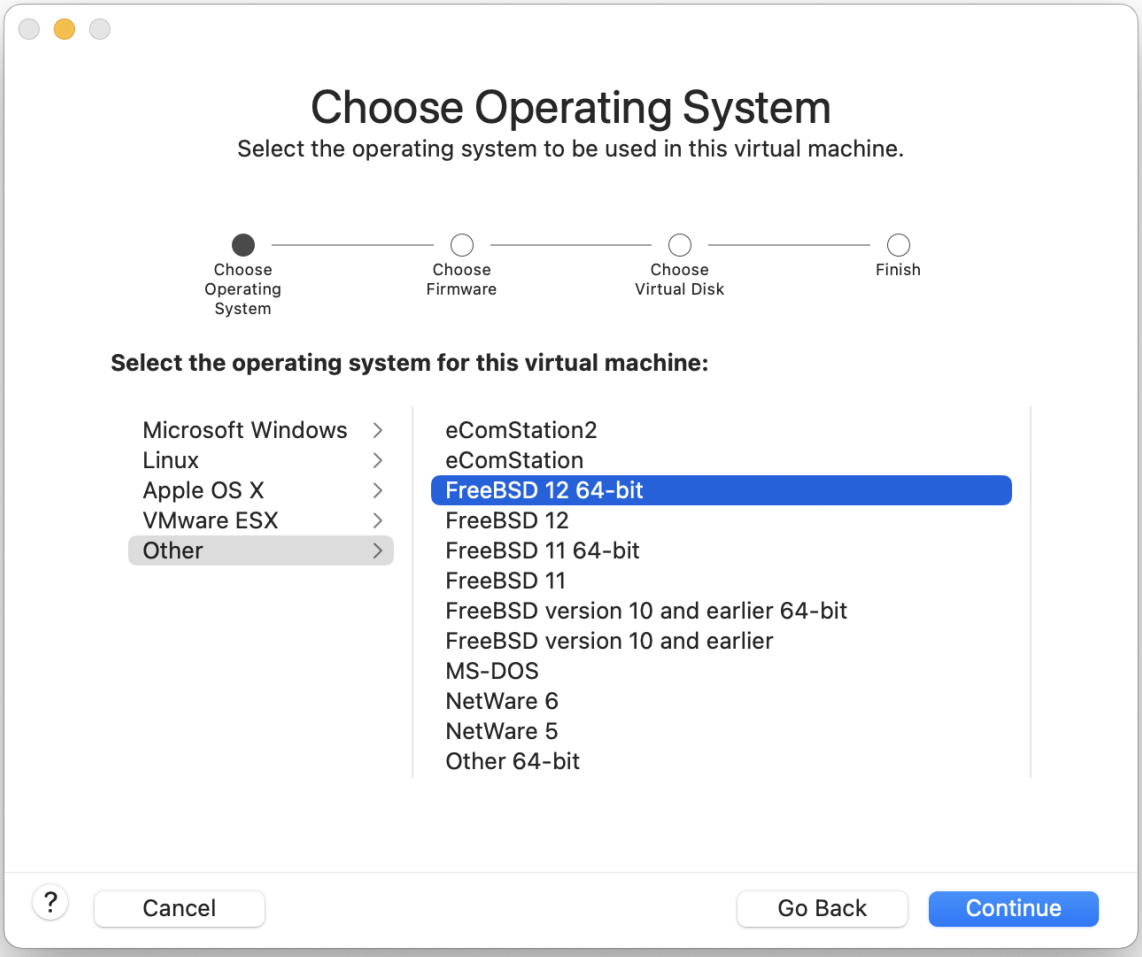
Zuerst müssen Sie VMware Fusion starten, um eine virtuelle Maschine zu erstellen. Dazu wählen Sie die Option New:



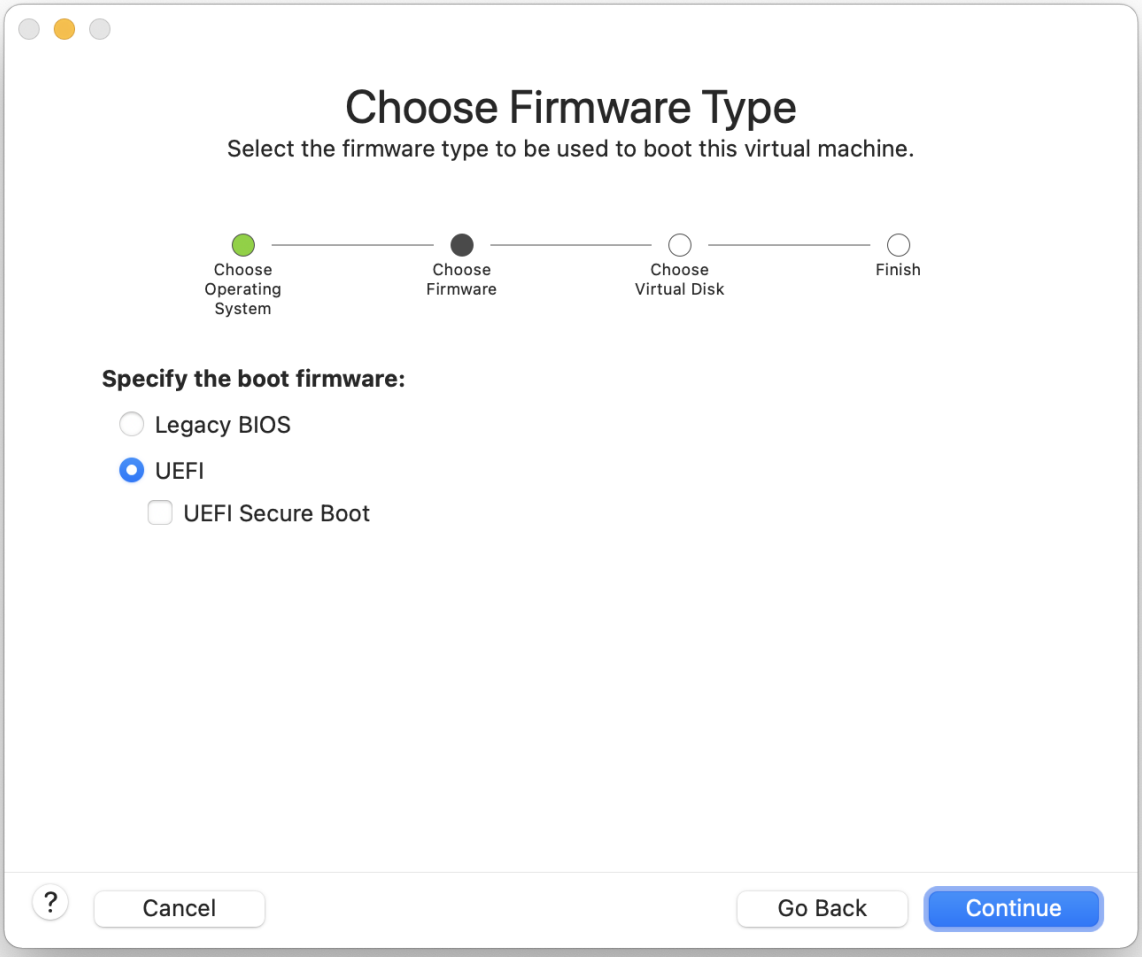

Dadurch wird ein Assistent gestartet, der bei der Erzeugung einer neuen virtuellen Maschine behilflich ist. Klicken Sie auf Continue, um den Prozess zu starten:



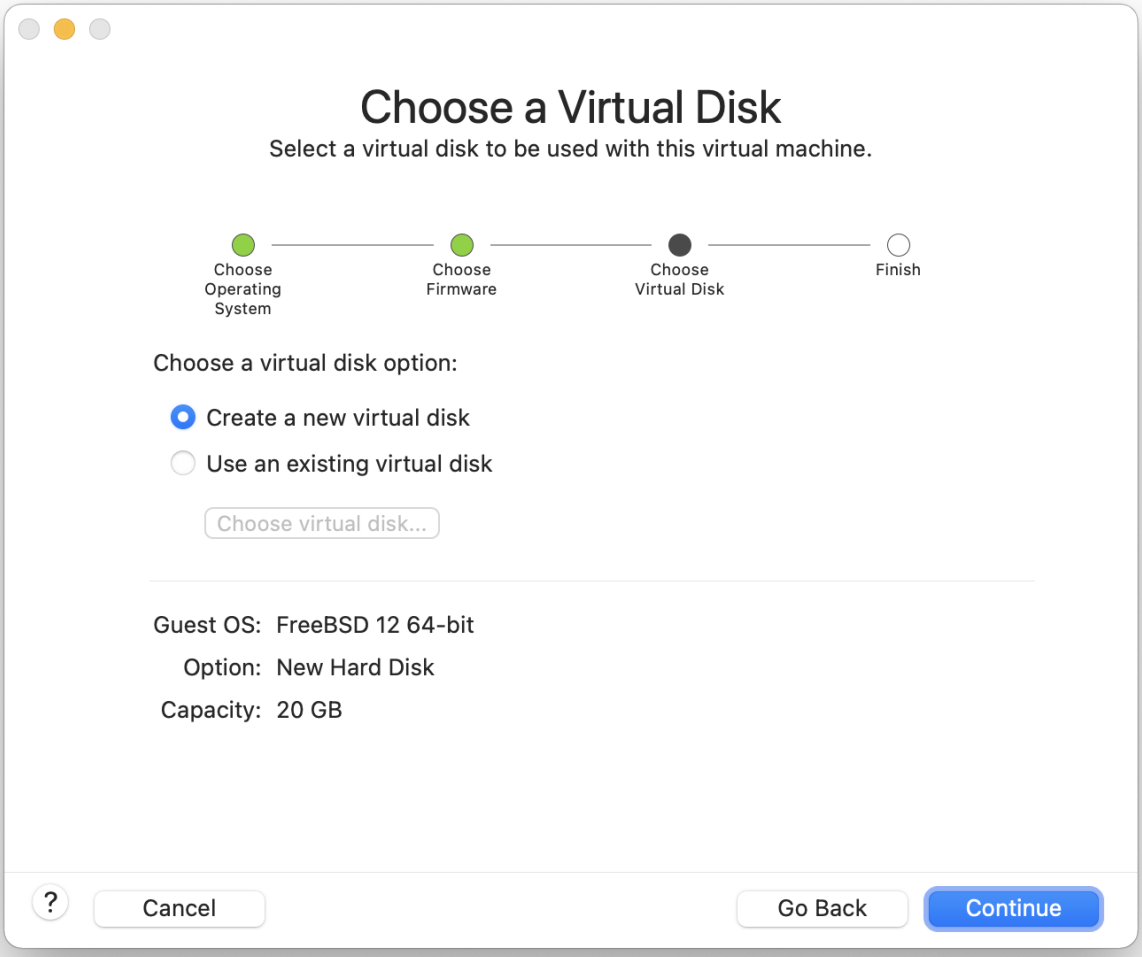

Wählen Sie Other als das Operating System, danach FreeBSD oder FreeBSD 64-bit, je nach dem, welche Version Sie installieren wollen, wenn Sie nach der zu installierenden **Version** gefragt werden:



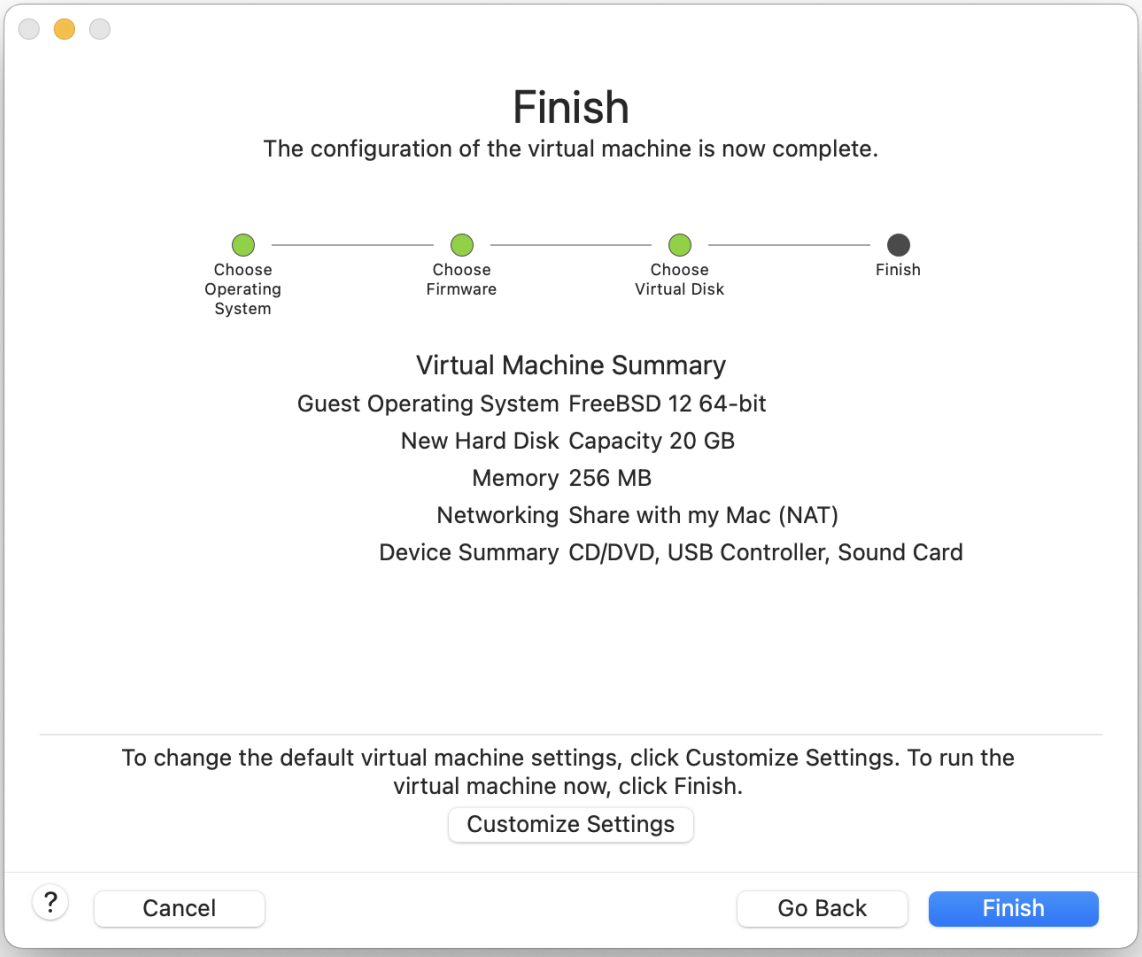

Vergeben Sie einen Namen für die virtuelle Maschine und legen Sie den Speicherort fest:



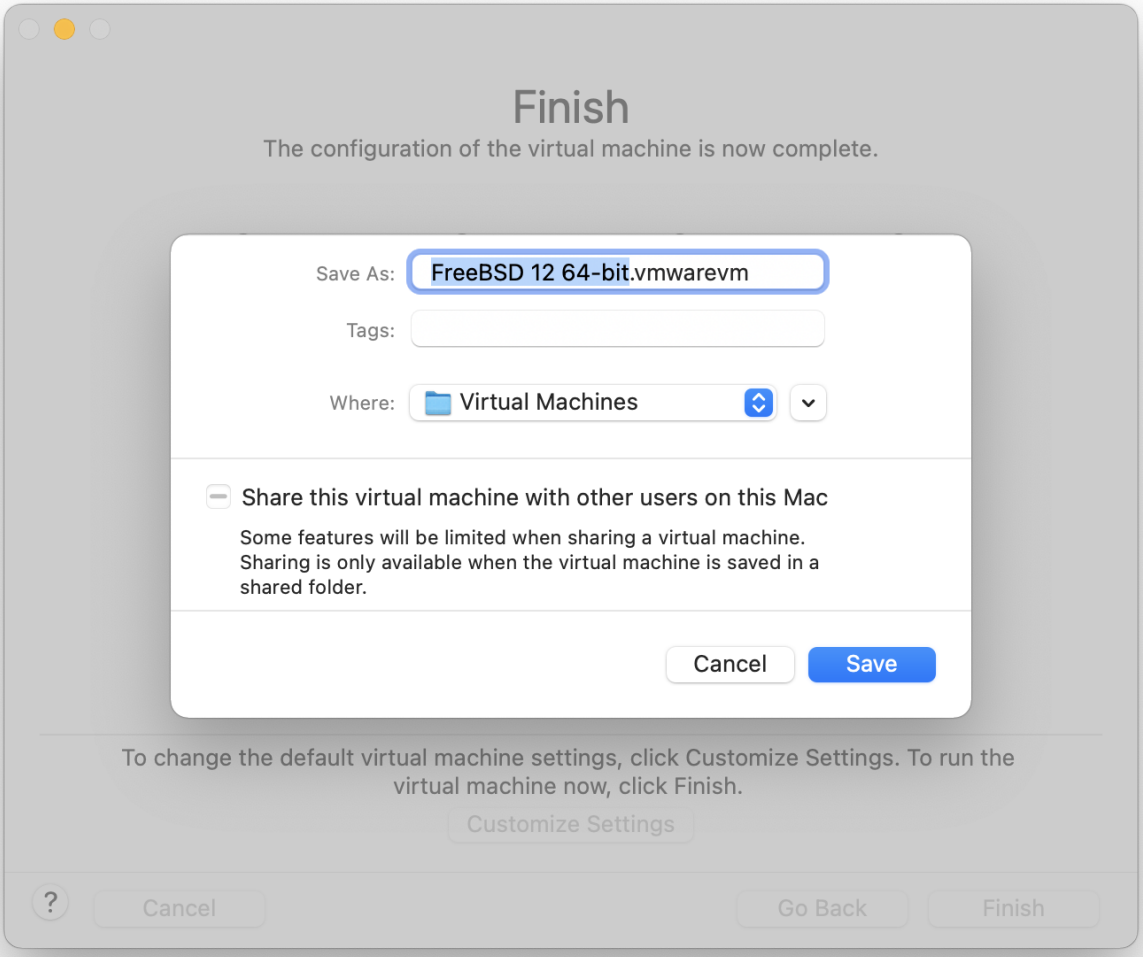

Legen Sie die Größe der virtuellen Festplatte für die virtuelle Maschine fest:




Wählen Sie die Installationsmethode für die virtuelle Maschine. Entweder von einem ISO-Abbild oder von einer CD/DVD:




Nachdem Sie auf Finish geklickt haben, wird die virtuelle Maschine gestartet:



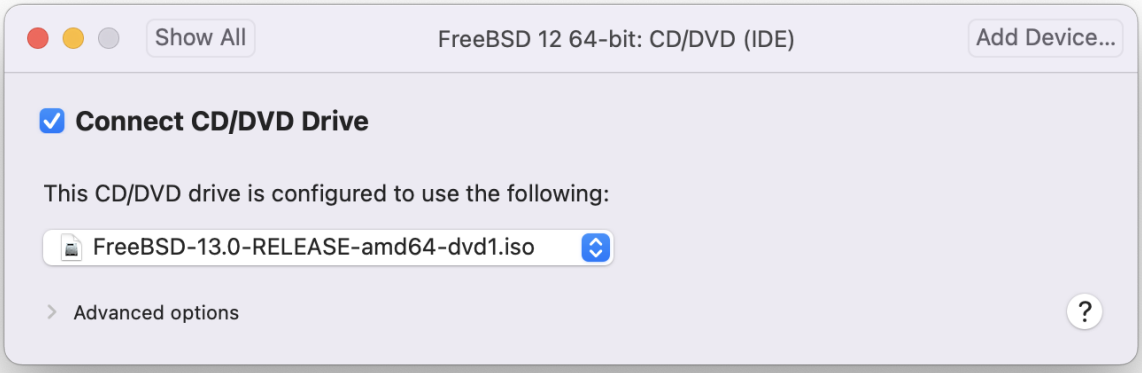

Nun können Sie FreeBSD wie gewohnt installieren:




Nachdem die Installation abgeschlossen ist, können noch verschiedene Parameter der virtuellen Maschine, wie etwa der Speicherverbrauch, konfiguriert werden:



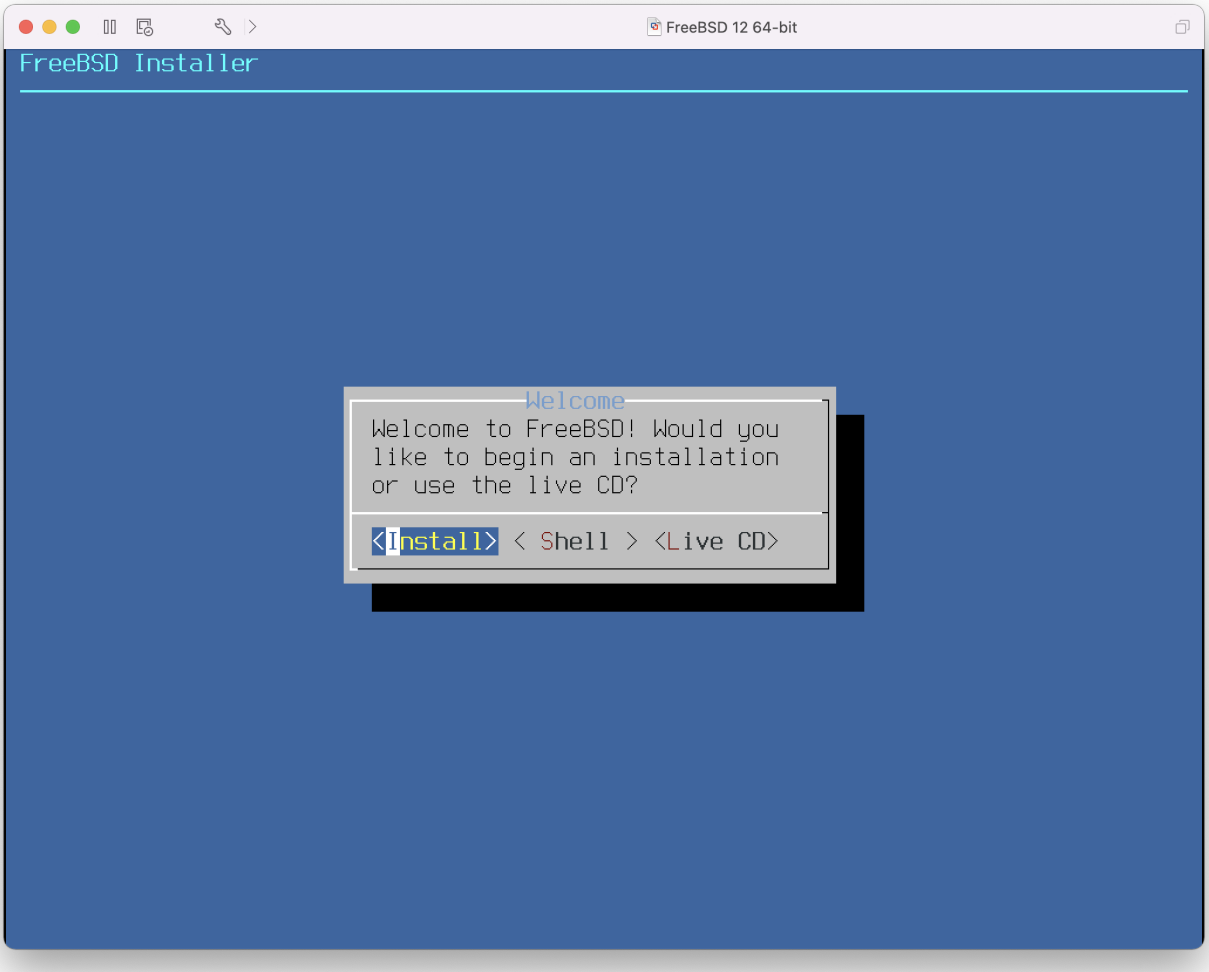
Die Hardware der virtuellen Maschine kann nicht geändert werden, solange die virtuelle Maschine läuft.



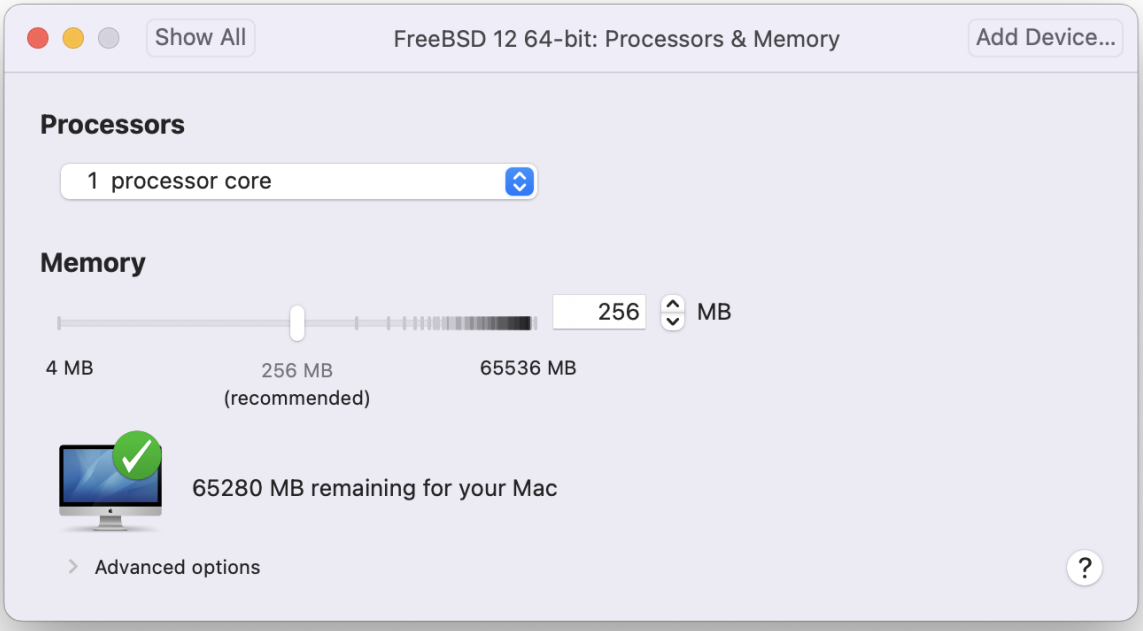

Die Anzahl der CPUs der virtuellen Maschine:




Den Status des CD-Laufwerks. Sie können die CD/DVD/ISO von der virtuellen Maschine lösen, wenn Sie es nicht benötigen.




Zuletzt sollten Sie noch festlegen, wie sich die virtuelle Maschine mit dem Netzwerk verbinden soll. Sollen neben dem Gastsystem auch andere Rechner auf die virtuelle Maschine zugreifen können, muss die Option Connect directly to the physical network (Bridged) gewählt werden. Ist dies nicht der Fall, sollte die Option Share the host's internet connection (NAT) gewählt werden. In dieser Einstellung kann die virtuelle Maschine zwar auf das Internet zugreifen, andere Rechner dürfen aber nicht auf die virtuelle Maschine zugreifen.




Nachdem die Konfiguration abgeschlossen ist, kann FreeBSD gestartet werden.

### 39.4.2. FreeBSD unter VMware Fusion konfigurieren

Nachdem Sie FreeBSD erfolgreich unter VMware Fusion installiert haben, sollten Sie das virtuelle FreeBSD noch anpassen, um eine optimale Funktion zu gewährleisten.

1. Die wichtigste Änderung ist es, die Variable `kern.hz` zu verkleinern, um so die CPU-Auslastung in der VMware Fusion-Umgebung zu verringern.

   ```
   kern.hz=100
   ```

   Ohne diese Einstellung kann ein unbeschäftigtes FreeBSD unter VMware Fusion trotzdem rund 15 Prozent der CPU-Leistung eines Single Prozessor iMac®'s verbrauchen. Nach dieser Änderung reduziert sich dieser Wert auf etwa 5 Prozent.

2. Erstellen einer neuen Kernelkonfigurationsdatei

   Alle FireWire- und USB-Laufwerks-Treiber können aus der Kernelkonfigurationsdatei entfernt werden. VMware Fusion stellt einen virtuellen Netzwerkadapter bereit, der den em(4)-Treiber verwendet. Daher können alle Netzwerkgeräte bis auf em(4) und miibus(4) aus dem Kernel entfernt werden.

3. Netzwerkbetrieb einrichten

   Die einfachste Netzwerkkonfiguration verwendet DHCP, um die virtuelle Maschine mit dem gleichen lokalen Netzwerk, in dem sich der Host-Mac® befindet, zu verbinden. Dazu fügen Sie die Zeile `ifconfig_em0="DHCP"` in /etc/rc.conf ein. Weitere Informationen zur Konfiguration des

Netzwerks unter FreeBSD finden Sie im Netzwerkverbindung konfigurieren.

## 39.5. FreeBSD als Gast mit VirtualBox™

FreeBSD funktioniert einwandfrei als Gast-Betriebssystem unter VirtualBox™. Die Virtualisierungs-Software steht für die meisten Betriebssysteme zur Verfügung, einschließlich FreeBSD.

Die VirtualBox™ Gasterweiterungen bieten Unterstützung für:

- Gemeinsame Zwischenablage.
- Mauszeiger-Integration.
- Zeitsynchronisation mit dem Host.
- Skalierung von Fenstern.
- Nahtloser Modus.



Diese Kommandos werden im FreeBSD Gastsystem ausgeführt.

Installieren Sie das Paket oder den Port emulators/virtualbox-ose-additions in das FreeBSD Gastsystem. Dieses Beispiel installiert den Port:

```
# cd /usr/ports/emulators/virtualbox-ose-additions
# make install clean
```

Fügen Sie folgende Einträge in /etc/rc.conf hinzu:

```
vboxguest_enable="YES"
vboxservice_enable="YES"
```

Wenn ntpd(8) oder ntpdate(8) verwendet wird um die Uhrzeit zu synchronisieren, dann deaktivieren Sie die Synchronisierung mit dem Host:

```
vboxservice_flags="--disable-timesync"
```

Xorg wird den `vboxvideo`-Treiber automatisch erkennen. Alternativ kann auch manuell ein entsprechender Eintrag in /etc/X11/xorg.conf hinzugefügt werden:

```
Section "Device"
    Identifier "Card0"
    Driver "vboxvideo"
    VendorName "InnoTek Systemberatung GmbH"
    BoardName "VirtualBox Graphics Adapter"
EndSection
```

Um den vboxmouse_drv-Treiber zu verwenden, muss /etc/X11/xorg.conf ebenfalls angepasst werden:

```
Section "InputDevice"
    Identifier "Mouse0"
    Driver "vboxmouse"
EndSection
```

Benutzer von HAL sollten die Datei /usr/local/etc/hal/fdi/policy/90-vboxguest.fdi erstellen oder sie aus /usr/local/shared/hal/fdi/policy/10osvendor/90-vboxguest.fdi kopieren:

```
<?xml version="1.0" encoding="utf-8"?>
<!--
# Sun VirtualBox
# Hal driver description for the vboxmouse driver
# $Id: chapter.xml,v 1.33 2012-03-17 04:53:52 eadler Exp $

    Copyright (C) 2008-2009 Sun Microsystems, Inc.

    This file is part of VirtualBox Open Source Edition (OSE, as
    available from http://www.virtualbox.org. This file is free software;
    you can redistribute it and/or modify it under the terms of the GNU
    General Public License (GPL) as published by the Free Software
    Foundation, in version 2 as it comes in the "COPYING" file of the
    VirtualBox OSE distribution. VirtualBox OSE is distributed in the
    hope that it will be useful, but WITHOUT ANY WARRANTY of any kind.

    Please contact Sun Microsystems, Inc., 4150 Network Circle, Santa
    Clara, CA 95054 USA or visit http://www.sun.com if you need
    additional information or have any questions.
-->
<deviceinfo version="0.2">
  <device>
    <match key="info.subsystem" string="pci">
      <match key="info.product" string="VirtualBox guest Service">
    <append key="info.capabilities" type="strlist">input</append>
    <append key="info.capabilities" type="strlist">input.mouse</append>
    <merge key="input.x11_driver" type="string">vboxmouse</merge>
    <merge key="input.device" type="string">/dev/vboxguest</merge>
      </match>
    </match>
  </device>
</deviceinfo>
```

Gemeinsame Ordner für die Dateitransfer zwischen Host und VM sind verfügbar, wenn sie mit mount_vboxvfs eingebunden werden. Ein gemeinsamer Ordner kann auf dem Host über die graphische Oberfläche von VirtualBox oder mit vboxmanage erstellt werden. Um beispielsweise einen freigegebenen Ordner namens *myshare* unter /mnt/bsdboxshare für die VM *BSDBox* zu erstellen, führen Sie folgendes Kommando aus:

```
# vboxmanage sharedfolder add 'BSDBox' --name myshare --hostpath /mnt/bsdboxshare
```

Beachten Sie, dass der Name des gemeinsamen Ordners keine Leerzeichen enthalten darf. Sie können den freigegebenen Ordner innerhalb des Gastsystems wie folgt einbinden:

```
# mount_vboxvfs -w myshare /mnt
```

# 39.6. FreeBSD als Host mit Virtualbox

VirtualBox™ ist ein vollständigesVirtualisierungspaket, das aktiv weiterentwickelt wird und für die meisten Betriebssysteme einschließlich Windows®, Mac OS®, Linux® und FreeBSD zur Verfügung steht. Es kann sowohl Windows® als auch UNIX®-ähnliche Gastsysteme betreiben. Es wird als Open Source Software veröffentlicht, jedoch mit Closed-Source-Komponenten in einem separaten Erweiterungspaket. Zu diesen Komponenten gehört Unterstützung für USB 2.0-Geräte. Weitere Informationen finden Sie auf der Downloads-Seite im VirtualBox™ Wiki. Derzeit sind diese Erweiterungen für FreeBSD nicht verfügbar.

## 39.6.1. VirtualBox™ installieren

VirtualBox™ steht als Paket oder Port in emulators/virtualbox-ose bereit. Der Port kann mit folgendem Kommando installiert werden:

```
# cd /usr/ports/emulators/virtualbox-ose
# make install clean
```

Eine nützliche Option im Konfigurationsdialog ist die `GuestAdditions`-Programmsammlung. Diese stellen eine Reihe von nützlichen Eigenschaften in den Gastbetriebssystemen zur Verfügung, wie beispielsweise Mauszeigerintegration (was es ermöglicht, die Maus zwischen dem Host und dem Gast zu teilen ohne eine spezielle Tastenkombination für diesen Wechsel zu drücken), sowie schnelleres Rendern von Videos, besonders in Windows® Gästen. Diese Gastzusätze sind im **Devices**-Menü zu finden, nachdem die Installation des Gastbetriebssystem abgeschlossen ist.

Ein paar Konfigurationsänderungen sind notwendig, bevor VirtualBox™ das erste Mal gestartet wird. Der Port installiert ein Kernelmodul in /boot/modules, das in den laufenden Kernel geladen werden muss:

```
# kldload vboxdrv
```

Um sicherzustellen, dass das Modul immer nach einem Neustart geladen wird, fügen Sie die folgende Zeile in /boot/loader.conf ein:

```
vboxdrv_load="YES"
```

Um die Kernelmodule für die Unterstützung von Netzwerkbrücken oder Host-Only Netzwerken zu laden, fügen Sie folgendes in /etc/rc.conf ein und starten Sie den Computer neu:

```
vboxnet_enable="YES"
```

Die Gruppe `vboxusers` wird während der Installation von VirtualBox™ angelegt. Alle Benutzer, die Zugriff auf VirtualBox™ haben sollen, müssen in diese Gruppe aufgenommen werden. `pw` kann benutzt werden, um neue Mitglieder hinzuzufügen:

```
# pw groupmod vboxusers -m yourusername
```

Damit Netzwerkbrücken funktionieren, müssen die in der Voreinstellung eingeschränkten Berechtigungen für /dev/vboxnetctl angepasst werden:

```
# chown root:vboxusers /dev/vboxnetctl
# chmod 0600 /dev/vboxnetctl
```

Um diese Berechtigungen dauerhaft zu speichern, fügen Sie folgende Einträge in /etc/devfs.conf hinzu:

```
own     vboxnetctl root:vboxusers
perm    vboxnetctl 0600
```

Um VirtualBox™ zu starten, geben Sie folgenden Befehl in der Xorg-Sitzung ein:

```
% VirtualBox
```

Besuchen Sie die offizielle Webseite von VirtualBox™ unter http://www.virtualbox.org, um weitere Informationen zur Konfiguration und Verwendung zu erhalten. FreeBSD-spezifische Informationen und Anleitungen zur Fehlerbehebung finden Sie auf der entsprechenden Seite im FreeBSD-Wiki unter http://wiki.FreeBSD.org/VirtualBox.

### 39.6.2. USB Unterstützung für VirtualBox™

Sie können VirtualBox™ so konfigurieren, dass USB-Geräte an das Gastsystem weitergeleitet werden. So lange das Erweiterungspaket für USB 2.0 und 3.0 auf FreeBSD nicht verfügbar ist, ist der Host-Controller der OSE-Version auf die Emulation von USB 1.1-Geräten beschränkt.

Damit VirtualBox™ angeschlossene USB-Geräte am Rechner erkennt, muss der Benutzer Mitglied der Gruppe `operator` sein.

```
# pw groupmod operator -m ihrbenutzername
```

Anschließend fügen Sie folgenden Eintrag in /etc/devfs.rules ein. Wenn die Datei nicht existiert, muss sie zuvor erstellt werden:

```
[system=10]
add path 'usb/*' mode 0660 group operator
```

Um diese neuen Regeln zu laden, fügen Sie Folgendes in /etc/rc.conf hinzu:

```
devfs_system_ruleset="system"
```

Danach starten Sie devfs neu:

```
# service devfs restart
```

Sie müssen die Anmeldesitzung und VirtualBox™ neu starten, damit die Änderungen wirksam werden. Danach können Sie nach Bedarf neue USB-Filter erstellen.

### 39.6.3. Host CD/DVD-Zugriff in VirtualBox™

Ein Gastsystem kann auf die DVD/CD-Laufwerke des Hosts zugreifen. Der Zugriff für die virtuellen Maschinen wird in den Einstellungen von VirtualBox™ konfiguriert. Falls erforderlich, erstellen Sie zunächst ein leeres IDEDVD/CD-Gerät und wählen Sie dann ein entsprechendes Medium für dieses Laufwerk aus. Das Kontrollkästchen `Passthrough` besagt, dass die virtuelle Maschine die Hardware direkt verwenden kann. Audio-CDs und Brenner funktionieren nur, wenn diese Option ausgewählt ist.

Damit die CD/DVD-Funktionen von VirtualBox™ funktionieren, muss HAL in /etc/rc.conf aktiviert und anschließend gestartet werden:

```
hald_enable="YES"
```

```
# service hald start
```

Damit die CD/DVD-Funktionen von Benutzern verwendet werden können, benötigen diese Zugriff auf /dev/xpt0, /dev/cdN und /dev/passN. Dies wird in der Regel dadurch erreicht, den Benutzer zum Mitglied der Gruppe `operator` zu machen. Die Berechtigungen für diese Geräte werden mit folgenden Zeilen in /etc/devfs.conf konfiguriert:

```
perm cd* 0660
perm xpt0 0660
perm pass* 0660
```

```
# service devfs restart
```

## 39.7. FreeBSD als Host mit bhyve

Beginnend mit FreeBSD 10.0-RELEASE ist bhyve, ein BSD-lizensierter Hypervisor, Teil des Basissystems. Dieser Hypervisor unterstützt eine Reihe von Gastbetriebssystemen, darunter FreeBSD, OpenBSD und viele Linux® Distributionen. In der Voreinstellung unterstützt bhyve eine serielle Konsole, graphische Konsolen werden nicht emuliert. bhyve verwendet Offload-Funktionen von neueren CPUs, um manuelle Speicherzuordnungen und Anweisungen zu vermeiden.

Das Design von bhyve erfordert einen Prozessor, der Intel® Extended Page Tables (EPT), AMD® Rapid Vitualization Indexing (RVI) oder Nested Page Tables (NPT) unterstützt. FreeBSD- oder Linux®-Gastsysteme mit mehr als einer vCPU benötigen VMX unrestricted mode support (UG). Die meisten neueren Prozessoren, speziell Intel® Core™ i3/i5/i7 und Intel® Xeon™ E3/E5/E7, unterstützen diese Funktionen. Unterstützung für UG wurde mit Intel's Westmere Mikroarchitektur eingeführt. Eine vollständige Liste der Intel®-Prozessoren mit EPT-Unterstützung finden Sie unter https://ark.intel.com/content/www/us/en/ark/search/featurefilter.html?productType=873&0_ExtendedPageTables=True. RVI wird seit der dritten Generation der AMD Opteron™-Prozessoren (Barcelona) unterstützt. Um zu sehen ob der Prozessor bhyve unterstützt, prüfen Sie die Ausgabe von `dmesg` oder /var/run/dmesg.boot. Für AMD®-Prozessoren suchen Sie in der Zeile `Features2` nach `POPCNT`. Für Intel®-Prozessoren suchen Sie in der Zeile `VT-x` nach `EPT` und `UG`.

### 39.7.1. Vorbereitung des Hosts

Der erste Schritt bei der Erstellung einer virtuellen Maschine in bhyve ist die Konfiguration des Host-Systems. Laden Sie zunächst das bhyve Kernelmodul:

```
# kldload vmm
```

Erstellen Sie ein tap-Gerät, um dieses mit der Netzwerk-Schnittstelle der virtuellen Maschine zu verbinden. Damit sich die Schnittstelle mit dem Netzwerk verbinden kann, müssen Sie zusätzlich eine Bridge-Schnittstelle erzeugen, bestehend aus dem tap-Gerät und der physikalischen Schnittstelle. In diesem Beispiel wird die physikalische Schnittstelle igb0 verwendet:

```
# ifconfig tap0 create
# sysctl net.link.tap.up_on_open=1
net.link.tap.up_on_open: 0 -> 1
# ifconfig bridge0 create
# ifconfig bridge0 addm igb0 addm tap0
# ifconfig bridge0 up
```

### 39.7.2. Ein FreeBSD-Gastsystem erstellen

Erzeugen Sie eine Datei, die als virtuelle Festplatte für das Gastsystem verwendet wird. Geben Sie die Größe und den Namen der virtuellen Festplatte an:

```
# truncate -s 16G guest.img
```

Laden Sie ein Installationsabbild von FreeBSD:

```
# fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/10.3/FreeBSD-10.3-
RELEASE-amd64-bootonly.iso
FreeBSD-10.3-RELEASE-amd64-bootonly.iso       100% of  230 MB  570 kBps 06m17s
```

FreeBSD enthält ein Beispielskript um eine virtuelle Maschine in bhyve auszuführen. Das Skript wird die virtuelle Maschine starten und sie in einer Schleife ausführen. Sollte die virtuelle Maschine abstürzen, wird sie vom Skript automatisch neu gestartet. Das Skript akzeptiert einige Optionen, um die Konfiguration der virtuellen Maschine zu kontrollieren: `-c` bestimmt die Anzahl der virtuellen CPUs, `-m` begrenzt den verfügbaren Speicher des Gastsystems, `-t` bestimmt das verwendete tap-Gerät, `-d` gibt das zu benutzende Festplattenabbild an, `-i` sagt bhyve dass es von CD booten soll und `-I` bestimmt das CD-Abbild. Der letzte Parameter ist der Name der virtuellen Maschine. Dieses Beispiel startet die virtuelle Maschine im Installationsmodus:

```
# sh /usr/shared/examples/bhyve/vmrun.sh -c 1 -m 1024M -t tap0 -d guest.img -i -I
FreeBSD-10.3-RELEASE-amd64-bootonly.iso guestname
```

Die virtuelle Maschine wird starten und das Installationsprogramm ausführen. Nachdem das System in der virtuellen Maschine installiert ist, werden Sie gefragt, ob eine Shell gestartet werden soll. Wählen Sie **[ Yes ]**.

Starten Sie die virtuelle Maschine neu. Ein Neustart der virtuellen Maschine wird bhyve beenden, aber da das vmrun.sh-Skript in einer Schleife läuft, wird bhyve automatisch neu gestartet. Wenn dies passiert, wählen Sie die Option `Reboot` im Bootloader-Menü, um die Schleife zu unterbrechen. Anschließend kann das Gastsystem von der virtuellen Festplatte gestartet werden:

```
# sh /usr/shared/examples/bhyve/vmrun.sh -c 4 -m 1024M -t tap0 -d guest.img guestname
```

### 39.7.3. Ein Linux®-Gastsystem erstellen

Um andere Betriebssysteme als FreeBSD zu booten, muss zunächst der Port sysutils/grub2-bhyve installiert werden.

Als nächstes erzeugen Sie eine Datei, die das Gastsystem als virtuelle Festplatte verwenden kann:

```
# truncate -s 16G linux.img
```

Der Start einer virtuellen Maschine mit bhyve ist ein zweistufiger Prozess. Zuerst muss ein Kernel geladen werden, dann kann das Gastsystem gestartet werden. Der Linux®-Kernel wird mit sysutils/grub2-bhyve geladen. Erstellen Sie eine device.map, damit grub die virtuellen Geräte den Dateien auf dem Hostsystem zuordnen kann:

```
(hd0) ./linux.img
(cd0) ./somelinux.iso
```

Benutzen Sie sysutils/grub2-bhyve um den Linux®-Kernel vom ISO-Abbild zu laden:

```
# grub-bhyve -m device.map -r cd0 -M 1024M linuxguest
```

Damit wird grub gestartet. Wenn die Installations-CD eine Datei namens grub.cfg enthält, wird ein Menü angezeigt. Wenn nicht, müssen die Dateien vmlinuz und initrd manuell geladen werden:

```
grub> ls
(hd0) (cd0) (cd0,msdos1) (host)
grub> ls (cd0)/isolinux
boot.cat boot.msg grub.conf initrd.img isolinux.bin isolinux.cfg memtest
splash.jpg TRANS.TBL vesamenu.c32 vmlinuz
grub> linux (cd0)/isolinux/vmlinuz
grub> initrd (cd0)/isolinux/initrd.img
grub> boot
```

Nun, da der Linux®-Kernel geladen ist, kann das Gastsystem gestartet werden:

```
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s 3:0,virtio-
blk,./linux.img \
    -s 4:0,ahci-cd,./somelinux.iso -l com1,stdio -c 4 -m 1024M linuxguest
```

Das System wird booten und das Installtionsprogramm starten. Starten Sie die virtuelle Maschine nach der Installation des Betriebssystems neu. Dies führt auch dazu, dass bhyve beendet wird. Die Instanz der virtuellen Maschine muss zerstört werden, bevor sie erneut in Betrieb genommen werden kann:

```
# bhyvectl --destroy --vm=linuxguest
```

Nun kann das Gastsystem direkt von der virtuellen Festplatte gestartet werden. Laden Sie den Kernel:

```
# grub-bhyve -m device.map -r hd0,msdos1 -M 1024M linuxguest
grub> ls
(hd0) (hd0,msdos2) (hd0,msdos1) (cd0) (cd0,msdos1) (host)
(lvm/VolGroup-lv_swap) (lvm/VolGroup-lv_root)
grub> ls (hd0,msdos1)/
lost+found/ grub/ efi/ System.map-2.6.32-431.el6.x86_64 config-2.6.32-431.el6.x
86_64 symvers-2.6.32-431.el6.x86_64.gz vmlinuz-2.6.32-431.el6.x86_64
initramfs-2.6.32-431.el6.x86_64.img
grub> linux (hd0,msdos1)/vmlinuz-2.6.32-431.el6.x86_64 root=/dev/mapper/VolGroup-
```

```
lv_root
grub> initrd (hd0,msdos1)/initramfs-2.6.32-431.el6.x86_64.img
grub> boot
```

Starten Sie die virtuelle Maschine:

```
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 \$    -s
3:0,virtio-blk,./linux.img -l com1,stdio -c 4 -m 1024M linuxguest
```

Linux® wird jetzt in der virtuellen Maschine gestartet und präsentiert Ihnen vielleicht einen Anmeldeprompt. Sie können sich anmelden und die virtuelle Maschine benutzen. Wenn Sie fertig sind, starten Sie die virtuelle Maschine neu, um bhyve zu verlassen. Anschließend zerstören Sie die Instanz der virtuellen Maschine:

```
# bhyvectl --destroy --vm=linuxguest
```

### 39.7.4. bhyve virtuelle Maschinen mit UEFI Firmware booten

Neben bhyveload und grub-bhyve kann der bhyve Hypervisor virtuelle Maschinen auch über die UEFI-Userspace-Firmware booten. Mit dieser Option werden Gastsysteme unterstützt, die von anderen Bootloadern nicht unterstützt werden.

Um die UEFI-Unterstützung in bhyve nutzen zu können, benötigen Sie zuerst die Abbilder der UEFI-Firmware. Dazu können Sie den Port oder das Paket sysutils/bhyve-firmware installieren.

Mit der Firmware an Ort und Stelle, fügen Sie die Option `-l bootrom,/pfad/zur/firmware` zur bhyve-Befehlszeile hinzu. Der eigentliche bhyve-Befehl könnte wie folgt lauten:

```
# bhyve -AHP -s 0:0,hostbridge -s 1:0,lpc \
-s 2:0,virtio-net,tap1 -s 3:0,virtio-blk,./disk.img \
-s 4:0,ahci-cd,./install.iso -c 4 -m 1024M \
-l bootrom,/usr/local/shared/uefi-firmware/BHYVE_UEFI.fd \
guest
```

sysutils/bhyve-firmware enthält auch eine CSM-fähige Firmware, um Gastsysteme ohne UEFI-Unterstützung im alten BIOS-Modus zu booten:

```
# bhyve -AHP -s 0:0,hostbridge -s 1:0,lpc \
-s 2:0,virtio-net,tap1 -s 3:0,virtio-blk,./disk.img \
-s 4:0,ahci-cd,./install.iso -c 4 -m 1024M \
-l bootrom,/usr/local/shared/uefi-firmware/BHYVE_UEFI_CSM.fd \
guest
```

### 39.7.5. Graphische Framebuffer für bhyve-Gastsysteme

Die Unterstützung von UEFI-Firmware ist bei graphischen Betriebssystemen, wie Microsoft Windows®, besonders nützlich.

Unterstützung für den UEFI-GOP Framebuffer kann auch über die Option `-s 29,fbuf,tcp=0.0.0.0:5900` aktiviert werden. Die Framebuffer-Auflösung kann mit `w=800` und `h=600` konfiguriert werden. Mit der Option `wait` können Sie bhyve anweisen, auf eine VNC-Verbindung zu warten, bevor das Gastsystem gebootet wird. Vom Host oder aus dem Netzwerk kann über das VNC-Protokoll auf den Framebuffer zugegriffen werden. Zusätzlich kann `-s 30,xhci,tablet` hinzugefügt werden, um eine präzise Mauszeigersynchronisation mit dem Host zu gewährleisten.

Der daraus resultierende Befehl würde so aussehen:

```
# bhyve -AHP -s 0:0,hostbridge -s 31:0,lpc \
-s 2:0,virtio-net,tap1 -s 3:0,virtio-blk,./disk.img \
-s 4:0,ahci-cd,./install.iso -c 4 -m 1024M \
-s 29,fbuf,tcp=0.0.0.0:5900,w=800,h=600,wait \
-l bootrom,/usr/local/shared/uefi-firmware/BHYVE_UEFI.fd \
-s 30,xhci,tablet \
guest
```

Beachten Sie, dass der Framebuffer im BIOS-Modus keine Befehle mehr empfängt, sobald die Steuerung von der Firmware an das Gastsystem übergeben wird.

### 39.7.6. Verwendung von ZFS mit bhyve-Gastsystemen

Wenn auf dem Host-Rechner ZFS eingerichtet ist, können Sie ZFS-Volumes anstelle eines Festplattenabbilds verwenden. Dies kann erhebliche Leistungsvorteile für das Gastsystem mit sich bringen. Ein ZFS-Volume kann wie folgt erstellt werden:

```
# zfs create -V16G -o volmode=dev zroot/linuxdisk0
```

Geben Sie das ZFS-Volume beim Start der virtuellen Maschine an:

```
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s3:0,virtio
-blk,/dev/zvol/zroot/linuxdisk0 \
    -l com1,stdio -c 4 -m 1024M linuxguest
```

### 39.7.7. Konsolen in der virtuellen Maschine

Es ist vorteilhaft, die bhyve-Konsole mit einem Werkzeug wie sysutils/tmux oder sysutils/screen zu bedienen. Damit ist es leicht, die Konsole zu verbinden oder zu trennen. Es ist auch möglich, die Konsole als Nullmodem-Gerät zu nutzen, auf das Sie mit `cu` zugreifen können. Laden Sie dazu das nmdm Kernelmodul und ersetzen Sie `-l com1,stdio` mit `-l com1,/dev/nmdm0A`. Die /dev/nmdm-Geräte werden bei Bedarf automatisch erstellt, jeweils paarweise, entsprechend den beiden Enden eines

Nullmodemkabels (/dev/nmdm0A und /dev/nmdm0B). Weitere Informationen finden Sie in nmdm(4).

```
# kldload nmdm
# bhyve -A -H -P -s 0:0,hostbridge -s 1:0,lpc -s 2:0,virtio-net,tap0 -s 3:0,virtio-
blk,./linux.img \
    -l com1,/dev/nmdm0A -c 4 -m 1024M linuxguest
# cu -l /dev/nmdm0B
Connected

Ubuntu 13.10 handbook ttyS0

handbook login:
```

### 39.7.8. Virtuelle Maschinen verwalten

Für jede virtuelle Maschine wird unterhalb von /dev/vmm ein Gerätename erzeugt. Dadurch kann der Administrator einfach feststellen, welche virtuellen Maschinen zur Zeit ausgeführt werden:

```
# ls -al /dev/vmm
total 1
dr-xr-xr-x   2 root  wheel    512 Mar 17 12:19 ./
dr-xr-xr-x  14 root  wheel    512 Mar 17 06:38 ../
crw-------   1 root  wheel  0x1a2 Mar 17 12:20 guestname
crw-------   1 root  wheel  0x19f Mar 17 12:19 linuxguest
crw-------   1 root  wheel  0x1a1 Mar 17 12:19 otherguest
```

Mit Hilfe von `bhyvectl` kann eine virtuelle Maschine zerstört werden:

```
# bhyvectl --destroy --vm=guestname
```

### 39.7.9. Persistente Konfiguration

Um das System so zu konfigurieren, dass bhyve-Gastsysteme beim Booten gestartet werden, müssen die folgenden Konfigurationen in den jeweiligen Dateien vorgenommen werden:

1. /etc/sysctl.conf

   ```
   net.link.tap.up_on_open=1
   ```

2. /etc/rc.conf

   ```
   cloned_interfaces="bridge0 tap0"
   ifconfig_bridge0="addm igb0 addm tap0"
   ```

```
kld_list="vmm nmdm"
```

# 39.8. FreeBSD als Xen™-Host

Xen ist ein GPLv2-lizensierter Typ-1-Hypervisor für Intel® und ARM® Architekturen. Seit FreeBSD 8.0 gibt es Unterstützung für i386™ und AMD® 64-Bit DomU sowie Amazon EC2 unpriviligierte Domänen (virtuelle Maschinen). Dom0 priviligierte Domänen (Host) wird seit FreeBSD 11.0 unterstützt. Aus Performancegründen wurde in FreeBSD 11 die Unterstützung für paravirtualisierte Domänen (PV) zugunsten von Hardware virtualisierten Domänen (HVM) entfernt.

Xen™ ist ein Bare-Metal-Hypervisor, was bedeutet, dass es das erste Programm ist, welches nach dem BIOS geladen wird. Anschließend wird ein spezieller priviligierter Gast namens Domain-0 (kurz `Dom0`) gestartet. Dom0 nutzt seine speziellen Privilegien, um direkt auf die zugrunde liegende Hardware zuzugreifen, was es zu einer sehr leistungsstarken Lösung macht. Es ist in der Lage, direkt auf Festplattencontroller und Netzwerkadapter zuzugreifen. Die Xen™ Werkzeuge zum Verwalten und Steuern des Xen™ Hypervisors werden auch von Dom0 zum Erstellen, Auflisten und Zerstören von VMs verwendet. Dom0 stellt virtuelle Festplatten und Netzwerkfunktionalität für unpriviligierte Domänen bereit, die oft als DomU bezeichnet werden. Dom0 kann mit der Servicekonsole anderer Hypervisor verglichen werden, wohingegen DomU die einzelnen Gast-VMs ausführt.

Xen™ kann VMs zwischen verschiedenen Xen™ Servern migrieren. Wenn beide Xen-Hosts denselben zugrundeliegenden Speicher teilen, kann die Migration durchgeführt werden, ohne dass die VM zuerst heruntergefahren werden muss. Stattdessen wird die Migration live durchgeführt, während die DomU läuft. Sie brauchen daher keinen Neustart oder Ausfallzeit einplanen. Dies ist bei Wartungsarbeiten und Upgrade-Fenstern sinnvoll, um sicherzustellen, dass die von der DomU bereitgestellten Dienste weiterhin zur Verfügung stehen. Viele weitere Funktionen von Xen™ finden Sie im Xen Wiki. Sie sollten jedoch beachten, dass derzeit noch nicht alle Funktionen von FreeBSD unterstützt werden.

## 39.8.1. Hardwareanforderungen für Xen™ Dom0

Um den Xen™ Hypervisor auf einem Host auszuführen, ist eine bestimmte Hardwarefunktionalität erforderlich. Hardware-virtualisierte Domänen benötigen Unterstützung für Extended Page Table ( EPT) und Input/Output Memory Management Unit (IOMMU) im Host-Prozessor.



Um ein FreeBSD Xen™ Dom0 betreiben zu können, muss die Maschine mit Legacy Boot (BIOS) gestartet werden.

## 39.8.2. Xen™ Dom0 Control Domain Konfiguration

Benutzer von FreeBSD 11 sollten die Pakete emulators/xen-kernel47 und sysutils/xen-tools47 installieren. Diese Pakete basieren auf Xen Version 4.7. Mit FreeBSD-12.0 und neueren Versionen können die Pakete emulators/xen-kernel411 und sysutils/xen-tools411 für Xen 4.11 verwendet werden.

Nach der Installation der Xen Pakete müssen die Konfigurationsdateien angepasst werden, um den Host für die Integration von Dom0 vorzubereiten. Ein Eintrag in /etc/sysctl.conf deaktiviert die Begrenzung für Speicherseiten. Andernfalls lassen sich DomU VMs mit höheren Speicheranforderungen nicht ausführen.

```
# echo 'vm.max_wired=-1' >> /etc/sysctl.conf
```

Für eine andere speicherbezogene Einstellung muss in /etc/login.conf die Option `memorylocked` auf `unlimited` gesetzt werden. Ansonsten kann das Erstellen von DomU-Domänen mit der Meldung `Cannot allocate memory` fehlschlagen. Nachdem Sie die Änderung in /etc/login.conf gemacht haben, müssen Sie `cap_mkdb` ausführen um die Datenbank zu aktualisieren. Einschränkung von Ressourcen enthält hierzu ausführliche Informationen.

```
# sed -i '' -e 's/memorylocked=64K/memorylocked=unlimited/' /etc/login.conf
# cap_mkdb /etc/login.conf
```

Fügen Sie einen Eintrag für die Xen™ Konsole in /etc/ttys ein:

```
# echo 'xc0     "usr/libexec/getty Pc"        xterm   onifconsole  secure' >>
/etc/ttys
```

Dom0 wird durch die Auswahl eines Xen™-Kernels in /boot/loader.conf aktiviert. Xen™ benötigt von dem Hostsystem auch Ressourcen wie CPU und Speicher, sowohl für sich selbst als auch für andere DomU Domains. Wie viele Ressourcen benötigt werden, hängt von den individuellen Anforderungen und der eingesetzten Hardware ab. In diesem Beispiel werden der Dom0 8 GB Speicher und 4 virtuelle CPUs zur Verfügung gestellt. Die serielle Konsole und Protokollierung wird ebenfalls aktiviert.

Benutzen Sie die folgenden Kommandos, wenn Sie die Xen 4.7 Pakete verwenden:

```
# sysrc -f /boot/loader.conf hw.pci.mcfg=0
# sysrc -f /boot/loader.conf if_tap_load="YES"
# sysrc -f /boot/loader.conf xen_kernel="/boot/xen"
# sysrc -f /boot/loader.conf xen_cmdline="dom0_mem=8192M dom0_max_vcpus=4 dom0pvh=1
console=com1,vga com1=115200,8n1 guest_loglvl=all loglvl=all"
```

Für Xen Version 4.11 oder höher, benutzen Sie stattdessen diese Kommandos:

```
# sysrc -f /boot/loader.conf if_tap_load="YES"
# sysrc -f /boot/loader.conf xen_kernel="/boot/xen"
# sysrc -f /boot/loader.conf xen_cmdline="dom0_mem=8192M dom0_max_vcpus=4 dom0=pvh
console=com1,vga com1=115200,8n1 guest_loglvl=all loglvl=all"
```



Protokolldateien, die Xen™ für die Dom0- und DomU-VMs erstellt, werden in

> /var/log/xen gespeichert. Sie sollten dieses Verzeichnis überprüfen, falls es zu Problemen kommt.

Aktivieren Sie den xencommons Dienst während des Systemstarts:

```
# sysrc xencommons_enable=yes
```

Diese Einstellungen reichen zwar aus, um ein Dom0-fähiges System zu starten, allerdings fehlt es dann an Netzwerkfunktionalität für die DomU-Rechner. Um dies zu beheben, können Sie eine Netzwerkbrücke über die Netzwerkschnittstelle des Hosts herstellen, die die DomU-VMs für die Verbindung zum Netzwerk benutzen können. Ersetzen Sie *em0* durch den Namen der Netzwerkschnittstelle des Hosts.

```
# sysrc cloned_interfaces="bridge0"
# sysrc ifconfig_bridge0="addm em0 SYNCDHCP"
# sysrc ifconfig_em0="up"
```

Starten Sie den Host neu, um den Xen™-Kernel zu laden und den Dom0 zu starten.

```
# reboot
```

Nach dem erfolgreichen Booten des Xen™-Kernels und der Anmeldung am System wird das Xen™-Werkzeug `xl` verwendet, um Informationen über die Domänen anzuzeigen.

```
# xl list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  8192     4     r-----     962.0
```

Die Ausgabe bestätigt, dass der Dom0 (auch Domain-0 genannt) die ID `0` hat und ausgeführt wird. Der vorher in /boot/loader.conf definierte Speicher und die virtuellen CPUs sind ebenfalls vorhanden. Weitere Informationen finden Sie in der Xen™ Dokumentation. Jetzt können DomU Gast-VMs erstellt werden.

### 39.8.3. Xen™ DomU Gast-VM Konfiguration

Unpriviligierte Domänen bestehen aus einer Konfigurationsdatei und virtuellen oder physikalischen Festplatten. Der virtuelle Plattenspeicher für die DomU kann aus Dateien bestehen, die mit truncate(1) erstellt wurden, oder ZFS Volumes wie in "Volumes erstellen und zerstören" beschrieben. In diesem Beispiel wird ein 20 GB Volume verwendet. Eine VM wird mit dem ZFS Volume erstellt, ein FreeBSD ISO-Abbild, 1 GB RAM und zwei virtuelle CPUs. Das ISO-Abbild mit den Installationsdateien wird mit fetch(1) heruntergeladen und lokal in der Datei freebsd.iso gespeichert.

```
# fetch ftp://ftp.freebsd.org/pub/FreeBSD/releases/ISO-IMAGES/12.0/FreeBSD-12.0-
```

```
RELEASE-amd64-bootonly.iso -o freebsd.iso
```

Ein ZFS Volume von 20 GB namens xendisk0 wird erstellt und dient der VM als Festplatte.

```
# zfs create -V20G -o volmode=dev zroot/xendisk0
```

Die neue DomU Gast-VM wird in einer Datei definiert. Einige spezifische Einstellungen wie Name, Tastaturbelegung und VNC-Verbindungsdetails werden ebenfalls konfiguriert. Für dieses Beispiel enthält die folgende freebsd.cfg eine minimale DomU-Konfiguration:

```
# cat freebsd.cfg
builder = "hvm" ①
name = "freebsd" ②
memory = 1024 ③
vcpus = 2 ④
vif = [ 'mac=00:16:3E:74:34:32,bridge=bridge0' ] ⑤
disk = [
'/dev/zvol/tank/xendisk0,raw,hda,rw', ⑥
'/root/freebsd.iso,raw,hdc:cdrom,r' ⑦
]
vnc = 1 ⑧
vnclisten = "0.0.0.0"
serial = "pty"
usbdevice = "tablet"
```

Erklärung der einzelnen Zeilen:

① Dies definiert, welche Art von Virtualisierung verwendet wird. `hvm` bezieht sich auf hardwaregestützte Virtualisierung oder Hardware Virtual Machine. Gastbetriebssysteme können unverändert auf der CPU mit Virtualisierungserweiterungen laufen und bieten nahezu die gleiche Leistung wie auf physikalischer Hardware. `generic` ist der voreingestellte Wert und erstellt eine PV-Domain.

② Der Name dieser virtuellen Maschine. Er dient zur Unterscheidung von anderen virtuellen Maschinen auf der selben Dom0. Diese Angabe ist zwingend erforderlich.

③ Die Größe an RAM in Megabytes, die der VM zur Verfügung steht. Die Größe wird vom verfügbaren Speicher des Hypervisors subtrahiert, nicht vom Speicher der Dom0.

④ Die Anzahl der virtuellen CPUs, die dem Gast zur Verfügung stehen. Für die beste Leistung sollten Sie dem Gast nicht mehr CPUs zuteilen, als die Anzahl der CPUs auf dem physikalischen Host.

⑤ Der virtuelle Netzwerkadapter. Dies ist die Brücke, die mit der Netzwerkschnittstelle des Hosts verbunden ist. Der Parameter `mac` definiert die MAC-Adresse der virtuellen Schnittstelle. Dieser Parameter ist optional. Falls keine MAC definiert ist, wird Xen™ eine zufällige MAC generieren.

⑥ Der vollständige Pfad zur Festplatte, Datei, oder ZFS Volume für den Plattenspeicher dieser VM. Optionen und Festplattendefinitionen werden durch Kommata getrennt.

⑦ Das Boot-Medium, aus dem das initiale Betriebssystem installiert wird. In diesem Beispiel wird das zuvor heruntergeladene ISO-Abbild benutzt. Andere Geräte und weitere Optionen sind in der Xen™ Dokumentation beschrieben.

⑧ Optionen, die die VNC-Konnektivität der seriellen Konsole der DomU steuern. Dabei handelt es sich um die aktive VNC-Unterstützung, die verwendete IP-Adresse, der Gerätename der seriellen Konsole und die Eingabemethoden für Maus, Tastatur und andere Geräte. `keymap` konfiguriert die Tastaturbelegung, die in der Voreinstellung `english` ist.

Nachdem die Konfigurationsdatei mit allen notwendigen Optionen erstellt wurde, wird die DomU erstellt, indem die Datei als Parameter an `xl` übergeben wird.

```
# xl create freebsd.cfg
```



Jedes mal, wenn die Dom0 neu gestartet wird, muss die Konfigurationsdatei nochmals an `xl` übergeben werden, um die DomU neu zu erstellen. In der Voreinstellung wird nur die Dom0 nach einem Neustart angelegt, nicht die einzelnen VMs. Die VMs können dort fortfahren, wo sie aufgehört haben, weil sie das Betriebssystem auf der virtuellen Festplatte gespeichert haben. Die Konfiguration der virtuellen Maschine kann sich mit der Zeit ändern (bspw. beim Hinzufügen von mehr Arbeitsspeicher). Die Konfigurationsdateien der virtuellen Maschinen müssen ordnungsgemäß gesichert und vorgehalten werden, um die Gast-VM bei Bedarf neu erstellen zu können.

Die Ausgabe von `xl list` bestätigt, dass die DomU erstellt wurde.

```
# xl list
Name                                        ID   Mem VCPUs      State   Time(s)
Domain-0                                     0  8192     4     r-----  1653.4
freebsd                                      1  1024     1     -b----   663.9
```

Um die Installation des Basis-Betriebssystems zu beginnen, starten Sie den VNC-Client und verbinden Sie sich mit Netzwerkadresse des Hosts oder mit der IP-Adresse, die auf der Zeile `vnclisten` in freebsd.cfg konfiguriert wurde. Nachdem das Betriebssystem installiert ist, fahren Sie die DomU herunter und trennen den VNC-Viewer. Bearbeiten Sie dann die freebsd.cfg, entfernen Sie die Zeile mit der `cdrom` Definiton, oder kommentieren Sie die Zeile mit `#` aus. Um diese neue Konfiguration zu laden, ist es notwendig, die alte DomU mit `xl` zu zerstören, indem Sie entweder den Namen oder die ID als Parameter übergeben. Danach kann die DomU mit der angepassten freebsd.cfg neu erstellt werden.

```
# xl destroy freebsd
# xl create freebsd.cfg
```

Auf die Maschine kann jetzt wieder mit dem VNC-Viewer zugegriffen werden. Dieses mal wird sie von einer virtuellen Festplatte booten, auf der das Betriebssystem installiert wurde. Die virtuelle Maschine kann nun verwendet werden.

### 39.8.4. Fehlerbehebung

Dieser Abschnitt enthält grundlegende Informationen, um Probleme zu beheben, die bei der Verwendung von FreeBSD als Host oder Gast von Xen™ auftreten können.

#### Fehlerbehebung beim Booten des Hosts

Bitte beachten Sie, dass die folgenden Tipps zur Fehlerbehebung für Xen™ 4.11 oder neuer gedacht sind. Wenn Sie noch Xen™ 4.7 benutzen und Probleme haben, sollten Sie die Migration auf eine neuere Version in Betracht ziehen.

Um Probleme beim Booten des Hosts zu beheben, benötigen Sie wahrscheinlich ein serielles Kabel oder ein USB-Kabel. Ausführliche Informationen während des Bootens erhalten Sie, wenn Sie die Option `xen_cmdline` in loader.conf hinzufügen. Einige relevante Optionen sind:

- `iommu=debug`: kann benutzt werden, um zusätzliche Informationen über das iommu auszugeben.
- `dom0=verbose`: kann benutzt werden, um zusätzliche Informationen über den dom0 Build Prozess auszugeben.
- `sync_console`: diese Option erzwingt eine synchrone Konsolenausgabe. Dies ist sehr nützlich für die Fehlersuche, um den Verlust von Nachrichten durch die Begrenzung zu vermeiden. Verwenden Sie diese Option niemals in produktiven Umgebungen, da sie es böswilligen Gästen ermöglichen kann, DoS-Angriffe gegen Xen™ über die Konsole durchzuführen.

Um Probleme zu identifizieren, sollte FreeBSD beim Booten ebenfalls detaillierte Informationen anzeigen. Dies können Sie wie folgt aktivieren:

```
# sysrc -f /boot/loader.conf boot_verbose="YES"
```

Wenn keine dieser Optionen zur Lösung des Problems beiträgt, senden Sie bitte das serielle Bootprotokoll zur weiteren Analyse an freebsd-xen@FreeBSD.org und xen-devel@lists.xenproject.org.

#### Fehlerbehebung beim Erstellen von Gastsystemen

Die folgenden Informationen können helfen, Probleme beim Erstellen von Gastsystemen zu diagnostizieren.

Die häufigste Ursache für Fehler beim Erstellen von Gastsystemen ist der `xl` Befehl, der einen Fehler generiert und mit einem Rückgabewert ungleich 0 endet. Wenn der angezeigte Fehler nicht ausreicht, um das Problem zu identifizieren, kann auch eine umfangreichere Ausgabe von `xl` erhalten werden, indem die Option `v` wiederholt verwendet wird.

```
# xl -vvv create freebsd.cfg
Parsing config from freebsd.cfg
libxl: debug: libxl_create.c:1693:do_domain_create: Domain 0:ao 0x800d750a0: create:
how=0x0 callback=0x0 poller=0x800d6f0f0
libxl: debug: libxl_device.c:397:libxl__device_disk_set_backend: Disk vdev=xvda
spec.backend=unknown
```

```
libxl: debug: libxl_device.c:432:libxl__device_disk_set_backend: Disk vdev=xvda, using
backend phy
libxl: debug: libxl_create.c:1018:initiate_domain_create: Domain 1:running bootloader
libxl: debug: libxl_bootloader.c:328:libxl__bootloader_run: Domain 1:not a PV/PVH
domain, skipping bootloader
libxl: debug: libxl_event.c:689:libxl__ev_xswatch_deregister: watch w=0x800d96b98:
deregister unregistered
domainbuilder: detail: xc_dom_allocate: cmdline="", features=""
domainbuilder: detail: xc_dom_kernel_file: filename
="/usr/local/lib/xen/boot/hvmloader"
domainbuilder: detail: xc_dom_malloc_filemap    : 326 kB
libxl: debug: libxl_dom.c:988:libxl__load_hvm_firmware_module: Loading BIOS:
/usr/local/shared/seabios/bios.bin
...
```

Wenn die ausführliche Ausgabe nicht bei der Diagnose des Problems hilft, gibt es auch noch die Protokolle des QEMU und Xen™ Toolstacks in /var/log/xen. Beachten Sie, dass der Name der Domäne an den Protokollnamen angehängt wird. Wenn die Domäne also `freebsd` heißt, sollten Sie wahrscheinlich die Dateien /var/log/xen/xl-freebsd.log und /var/log/xen/qemu-dm.freebsd.log finden. Beide Dateien können nützliche Informationen zur Fehlerbehebung enthalten. Wenn nichts davon zur Lösung des Problems beiträgt, senden Sie bitte die Beschreibung des Problems und so viele Informationen wie möglich an freebsd-xen@FreeBSD.org und xen-devel@lists.xenproject.org, um Hilfe zu erhalten.

# Kapitel 40. Localization - i18n/L10n Usage and Setup

## 40.1. Übersicht

FreeBSD ist ein verteiltes Projekt mit Nutzern und Mitwirkenden auf der ganzen Welt. Als solches unterstützt FreeBSD Lokalisierung für viele Sprachen, so dass Benutzer Daten in anderen Sprachen als Englisch anzeigen, eingeben und verarbeiten können. Sie können zwischen den meisten der verbreitetsten Sprachen der Welt wählen, unter anderem Chinesisch, Japanisch, Koreanisch, Französisch, Russisch, Vietnamesisch und Deutsch.

Der Begriff internationalization (englisch für Internationalisierung) wurde zu I18N abgekürzt, weil sich zwischen dem ersten und letzten Buchstaben des Worts 18 Buchstaben befinden. L10N benutzt die gleiche Namensgebung und ist eine Abkürzung des Wortes localization (englisch für Lokalisierung). Mit I18N/L10N-Methoden, -Protokollen und -Anwendungen können Benutzer eine Sprache ihrer Wahl verwenden.

Dieses Kapitel behandelt die Internationalisierung und Lokalisierung von FreeBSD. Nachdem Sie dieses Kapitel gelesen haben, werden Sie wissen:

- wie der Name einer Locale aufgebaut ist.
- wie die Locale einer Login-Shell gesetzt wird.
- wie die Konsole für nicht-englische Sprachen konfiguriert wird.
- wie Xorg mit verschiedenen Sprachen benutzt wird.
- wie I18N-fähige Anwendungen gefunden werden können.
- Wo Sie weitere Informationen über verschiedene Sprachen finden.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Wissen, wie Sie zusätzliche Anwendungen installieren.

## 40.2. Lokale Anpassungen benutzen

Lokale Anpassungen werden durch die Angabe von drei Werten erreicht: dem Sprachcode, dem Ländercode und der Codierung. Die Zusammenfassung dieser Werte wird "Locale" genannt und sieht wie folgt aus:

```
Sprachcode_Ländercode.Codierung
```

*Sprachcode* und *Ländercode* werden verwendet, um das Land und die spezifische Sprachvariation zu bestimmen. Gebräuchliche Sprach- und Ländercodes enthält dazu einige Beispiele:

*Tabelle 14. Gebräuchliche Sprach- und Ländercodes*

| Sprachcode_Ländercode | Beschreibung |
|---|---|
| en_US | Englisch, Vereinigte Staaten |
| ru_RU | Russisch, Russland |
| zh_TW | Traditionelles Chinesisch, Taiwan |

Eine vollständige Liste der verfügbaren Lokalisierungen erhalten Sie durch die Eingabe von:

```
% locale -a | more
```

Die aktuelle Ländereinstellung erhalten Sie mit:

```
% locale
```

Sprachspezifische Zeichensätze, wie ISO8859-1, ISO8859-15, KOI8-R und CP437 werden in multibyte(3) beschrieben. Eine Liste der Zeichensätze finden Sie in der IANA Registry.

Einige Sprachen, darunter Chinesisch und Japanisch, können nicht mit ASCII-Zeichen dargestellt werden und benötigen eine erweiterte Sprachcodierung mit Wide- oder Multibyte-Zeichen. EUC und Big5 sind Beispiele für Wide- oder Multibyte-Codierungen. Ältere Anwendungen erkennen diese Zeichen nicht und halten sie fälschlicherweise für Steuerzeichen, während neure Anwendungen diese Zeichen in der Regel erkennen. Es hängt allerdings von der Implementierung ab, ob man eine Anwendung neu kompilieren muss, um lokale Zeichensätze zu bekommen, oder ob sie nur richtig konfiguriert werden muss.



FreeBSD verwendet Xorg-kompatible Codierungen.

Der Rest dieses Abschnitts beschreibt die verschiedenen Methoden zur Konfiguration von der Locale auf einem FreeBSD-System. Der folgende Abschnitt beschreibt den Bau von Anwendungen mit I18N-Unterstützung.

### 40.2.1. Einstellen der Locale für die Login-Shell

Die Einstellungen für Locale werden entweder in der ~/.login_conf des Benutzers, oder der Startdatei der Shell (~/.profile, ~/.bashrc oder ~/.cshrc) konfiguriert.

Zwei Umgebungsvariablen sollten konfiguriert werden:

- `LANG`, das die Locale einstellt.
- `MM_CHARSET`, das den MIME Zeichensatz für Anwendungen einstellt.

Neben der Shell-Konfiguration des Benutzers sollten diese Variablen auch für spezifische Anwendungen und Xorg-Konfigurationen eingestellt werden.

Es gibt zwei Methoden, die Locale zu setzen: die erste und empfohlene Methode ist, die Umgebungsvariablen in der Login-Klasse zu setzen, die zweite Methode ist, sie in den Startdateien der Shell zu setzen. In den nächsten Abschnitten werden beide Methoden vorgestellt.

**Lokalisierung in der Login-Klasse**

Die erste Methode wird empfohlen, da sie die Umgebungsvariablen für die Login-Klasse und den MIME Zeichensatz für alle Shells zuweist. Die Lokalisierung kann von einem Benutzer selbst, oder vom Superuser für alle Benutzer eingestellt werden.

.login_conf im Heimatverzeichnis eines Benutzers sollte mindestens die folgenden Einträge enthalten, damit beide Variablen für den Gebrauch der Latin-1 Codierung gesetzt werden:

```
me:\
    :charset=ISO-8859-1:\
    :lang=de_DE.ISO8859-1:
```

Damit traditionelles Chinesisch (BIG-5 Codierung) verwendet werden kann, sind in ~/.login_conf des Benutzers die nachstehenden Ergänzungen vorzunehmen. Einige Programme behandeln die Lokalisierung für Chinesisch, Japanisch und Koreanisch falsch, daher müssen mehr Variablen als üblich gesetzt werden:

```
#Users who do not wish to use monetary units or time formats
#of Taiwan can manually change each variable
me:\
    :lang=zh_TW.Big5:\

:setenv=LC_ALL=zh_TW.Big5,LC_COLLATE=zh_TW.Big5,LC_CTYPE=zh_TW.Big5,LC_MESSAGES=zh_TW.
Big5,LC_MONETARY=zh_TW.Big5,LC_NUMERIC=zh_TW.Big5,LC_TIME=     zh_TW.Big5:\
    :charset=big5:\
    :xmodifiers="@im=gcin": #Set gcin as the XIM Input Server
```

Alternativ kann der Superuser die Lokalisierung für alle Benutzer konfigurieren. Die folgenden Variablen in /etc/login.conf setzen die richtige Login-Klasse und den richtigen MIME Zeichensatz:

```
Sprache|Account-Typ-Beschreibung:\
    :charset=MIME_Zeichensatz:\
    :lang=Locale:\
    :tc=default:
```

Die für Latin-1 erforderlichen Einträge würden wie folgt aussehen:

```
german|German Users Accounts:\
    :charset=ISO-8859-1:\
    :lang=de_DE.ISO8859-1:\
    :tc=default:
```

Weitere Einzelheiten über diese Variablen finden Sie in login.conf(5). Beachten Sie, dass die Datei bereits vordefinierte *russische* Login-Klassen enthält.

Jedes Mal, wenn /etc/login.conf bearbeitet wurde, muss die Datenbank mit dem folgenden Kommando aktualisiert werden:

```
# cap_mkdb /etc/login.conf
```



Der reguläre Benutzer muss den Befehl `cap_mkdb` auf seine ~/.login_conf anwenden, damit die Änderungen wirksam werden.

#### Werkzeuge zum Ändern der Login-Klasse

Neben der manuellen Konfiguration von /etc/login.conf, stehen mehrere Werkzeuge bereit, um die Login-Klasse für neue Benutzer einzustellen.

Wenn Sie neue Accounts mit `vipw` anlegen, setzen Sie im Feld *Sprache* die gewünschte Sprache ein:

```
user:password:1111:11:Sprache:0:0:Benutzername:/home/user:/bin/sh
```

Wenn Sie mit `adduser` neue Benutzer anlegen, können Sie die voreingestellte Sprache für alle Benutzer, oder für einen einzelnen Benutzer einstellen:

Falls alle Benutzer die gleiche Sprache benutzen, setzen Sie `defaultclass=Sprache` in /etc/adduser.conf.

Wenn Sie diese Einstellung beim Anlegen des Benutzers überschreiben wollen, geben Sie entweder die gewünschte Login-Klasse am Prompt ein:

```
Enter login class: default []:
```

oder übergeben Sie die Login-Klasse beim Aufruf von `adduser`:

```
# adduser -class Sprache
```

Wenn Sie neue Benutzer mit `pw` anlegen, geben Sie die Login-Klasse wie folgt an:

```
# pw useradd Benutzername -L Sprache
```

Um die Login-Klasse eines bestehenden Benutzers zu ändern, kann `chpass` verwendet werden. Rufen Sie das Kommando als Superuser auf und geben Sie als Argument den entsprechenden Benutzernamen mit:

```
# chpass Benutzername
```

#### Lokalisierung in den Startdateien der Shells

Diese zweite Methode wird nicht empfohlen, da jede Shell unterschiedlich eingerichtet wird, eine unterschiedliche Konfigurationsdatei und Syntax verwendet. Um beispielsweise die deutsche Sprache für die `sh` zu setzen, fügen Sie für einen Benutzer die folgende Zeilen in ~/.profile ein. Sie können diese Zeilen auch für alle Benutzer der `sh` Shell in /etc/profile oder /usr/shared/skel/dot.profile hinzufügen:

```
LANG=de_DE.ISO8859-1; export LANG
MM_CHARSET=ISO-8859-1; export MM_CHARSET
```

Die `csh` Shell verwendet jedoch eine andere Konfigurationsdatei und eine andere Syntax. Dies sind die entsprechenden Einstellungen für ~/.csh.login, /etc/csh.login oder /usr/shared/skel/dot.login:

```
setenv LANG de_DE.ISO8859-1
setenv MM_CHARSET ISO-8859-1
```

Die Syntax zur Konfiguration von Xorg in ~/.xinitrc hängt ebenfalls von der verwendeten Shell ab. Das erste Beispiel ist für die `sh` Shell, das zweite für die `csh` Shell:

```
LANG=de_DE.ISO8859-1; export LANG
```

```
setenv LANG de_DE.ISO8859-1
```

### 40.2.2. Einrichten der Konsole

Für die Konsole stehen mehrere lokalisierte Sprachen zur Verfügung. Eine Liste der verfügbaren Schriften erhalten Sie mit `ls /usr/shared/syscons/fonts`. Um die Schriftart für die Konsole zu konfigurieren, setzen Sie den gewünschten *Zeichensatz* ohne die Endung .fnt in /etc/rc.conf:

```
font8x16=Zeichensatz
font8x14=Zeichensatz
font8x8=Zeichensatz
```

Die Tasten- und Bildschirmzuordnung (keymap und screenmap) kann in mit den folgenden Einträgen in /etc/rc.conf gesetzt werden:

```
scrnmap=screenmap_name
keymap=keymap_name
keychange="fkey_number sequence"
```

Eine Liste der verfügbaren Bildschirmzuordnungen erhalten Sie mit `ls /usr/shared/syscons/scrnmaps`. Spezifizieren Sie *screenmap_name* ohne die Endung .scm. Eine

Bildschirmzuordnung und der zugehörige Zeichensatz verbreitert die Zeichenmatrix von VGA Karten von 8 Bit auf 9 Bit. Sie wird benötigt, wenn der Zeichensatz des Bildschirms 8 Bit verwendet.

Eine Liste der verfügbaren Tastenzuordnungen erhalten Sie mit `ls /usr/shared/syscons/keymaps`. Spezifizieren Sie *keymap_name* ohne die Endung .kbd. Eine Tastenzuordnung können Sie ohne einen Neustart mit kbdmap(1) ausprobieren.

Der Eintrag `keychange` programmiert die Funktionstasten so, dass sie zu dem ausgesuchten Terminal passen. Die Sequenzen der Funktionstasten können nicht in Tastenzuordnungen definiert werden.

Setzen Sie als nächstes für alle Terminals den richtigen Terminaltyp in /etc/ttys. Terminaltypen für Zeichensätze enthält eine Zusammenfassung der verfügbaren Terminaltypen.

*Tabelle 15. Terminaltypen für Zeichensätze*

| Zeichensatz | Terminaltyp |
|---|---|
| ISO8859-1 oder ISO8859-15 | `cons25l1` |
| ISO8859-2 | `cons25l2` |
| ISO8859-7 | `cons25l7` |
| KOI8-R | `cons25r` |
| KOI8-U | `cons25u` |
| CP437 (VGA default) | `cons25` |
| US-ASCII | `cons25w` |

Mit Wide- oder Multibyte-Zeichensätzen müssen Sie die entsprechende Konsole aus der FreeBSD Ports-Sammlung installieren. Die verfügbaren Ports sind in Konsolen aus der Ports-Sammlung zusammengefasst. Nachdem Sie einen Port installiert haben, finden Sie in der Manualpage oder der pkg-message des Ports Anweisungen zur Konfiguration und Benutzung der Konsole.

*Tabelle 16. Konsolen aus der Ports-Sammlung*

| Sprache | Port |
|---|---|
| traditionelles Chinesisch (BIG-5) | chinese/big5con |
| Chinesisch/Japanisch/Koreanisch | chinese/cce |
| Chinesisch/Japanisch/Koreanisch | chinese/zhcon |
| Japanisch | chinese/kon2 |
| Japanisch | japanese/kon2-14dot |
| Japanisch | japanese/kon2-16dot |

Wenn Sie moused in /etc/rc.conf aktiviert haben, ist vielleicht noch weitere Konfiguration nötig. Der Mauszeiger des syscons(4) Treibers belegt in der Voreinstellung den Bereich von `0xd0` bis `0xd3` des Zeichensatzes. Wenn dieser Bereich ebenfalls von der eingestellten Sprache benötigt wird, müssen Sie den Mauszeiger verschieben. Fügen Sie dazu die folgende Zeile in /etc/rc.conf ein:

```
mousechar_start=3
```

### 40.2.3. Einrichtung von Xorg

Das X-Window-System beschreibt die Installation und Konfiguration von Xorg. Wenn Xorg für die Lokalisierung eingerichtet wird, stehen zusätzliche Zeichensätze und Eingabemethoden in der FreeBSD Ports-Sammlung zur Verfügung. Anwendungsspezifische I18N-Einstellungen, wie etwa Zeichensätze und Menüs, können in ~/.Xresources angepasst werden, damit in den graphischen Anwendungen des Benutzers die gewählte Sprache angezeigt wird.

Das X Input Method (XIM) Protokoll ist ein Xorg-Standard für die Eingabe von nicht-englischen Zeichen. Verfügbare Eingabemethoden fasst die aus der FreeBSD Ports-Sammlung verfügbaren Anwendungen für die Eingabemethoden zusammen. Zusätzliche Fcitx- und Uim-Anwendungen sind ebenfalls verfügbar.

*Tabelle 17. Verfügbare Eingabemethoden*

| Sprache | Eingabemethode |
|---|---|
| Chinesisch | chinese/gcin |
| Chinesisch | chinese/ibus-chewing |
| Chinesisch | chinese/ibus-pinyin |
| Chinesisch | chinese/oxim |
| Chinesisch | chinese/scim-fcitx |
| Chinesisch | chinese/scim-pinyin |
| Chinesisch | chinese/scim-tables |
| Japanisch | japanese/ibus-anthy |
| Japanisch | japanese/ibus-mozc |
| Japanisch | japanese/ibus-skk |
| Japanisch | japanese/im-ja |
| Japanisch | japanese/kinput2 |
| Japanisch | japanese/scim-anthy |
| Japanisch | japanese/scim-canna |
| Japanisch | japanese/scim-honoka |
| Japanisch | japanese/scim-honoka-plugin-romkan |
| Japanisch | japanese/scim-honoka-plugin-wnn |
| Japanisch | japanese/scim-prime |
| Japanisch | japanese/scim-skk |
| Japanisch | japanese/scim-tables |
| Japanisch | japanese/scim-tomoe |

| Sprache | Eingabemethode |
|---|---|
| Japanisch | japanese/scim-uim |
| Japanisch | japanese/skkinput |
| Japanisch | japanese/skkinput3 |
| Japanisch | japanese/uim-anthy |
| Koreanisch | korean/ibus-hangul |
| Koreanisch | korean/imhangul |
| Koreanisch | korean/nabi |
| Koreanisch | korean/scim-hangul |
| Koreanisch | korean/scim-tables |
| Vietnamesisch | vietnamese/xvnkb |
| Vietnamesisch | vietnamese/x-unikey |

## 40.3. I18N-Programme

I18N-Anwendungen werden mit Hilfe von I18N-Bibliotheken programmiert. Diese erlauben es Entwicklern, eine einfache Sprachdatei zu schreiben und Menüs und Texte an jede Sprache anzupassen.

Die FreeBSD Ports-Sammlung enthält Programme mit Unterstützung für Wide- und Mulitbyte-Zeichensätze für verschiedene Sprachen. Konsultieren Sie die I18N-Dokumentation des entsprechenden Ports für Informationen, wie das Programm zu konfigurieren ist und welche Optionen beim Übersetzen anzugeben sind.

Viele Anwendungen aus der FreeBSD Ports-Sammlung bieten I18N-Unterstützung. Diese enthalten, zur einfachen Identifikation, `-i18n` im Namen. Es werden jedoch nicht alle Sprachen unterstützt.

Einige Anwendungen können mit einem bestimmten Zeichensatz konfiguriert werden. Dies erfolgt entweder im Makefile, oder über spezielle Parameter, die an configure übergeben werden. Lesen Sie die I18N-Dokumentation des entsprechenden Ports für Informationen, wie das Programm zu konfigurieren ist und welche Optionen beim Übersetzen anzugeben sind.

## 40.4. Lokalisierung für einzelne Sprachen

Dieser Abschnitt beschreibt die Lokalisierung eines FreeBSD-Systems für die russische Sprache. Außerdem werden einige zusätzliche Ressourcen für die Lokalisierung in anderen Sprachen zur Verfügung gestellt.

### 40.4.1. Russisch (KOI8-R Codierung)

Um diese Locale für die Login-Shell zu setzen, fügen Sie die folgenden Zeilen in die ~/.login_conf des Benutzers ein:

```
me:My Account:\
    :charset=KOI8-R:\
    :lang=ru_RU.KOI8-R:
```

Fügen Sie folgende Zeile für die Konsole in /etc/rc.conf ein:

```
keymap="ru.utf-8"
scrnmap="utf-82cp866"
font8x16="cp866b-8x16"
font8x14="cp866-8x14"
font8x8="cp866-8x8"
mousechar_start=3
```

Benutzen Sie `cons25r` als Terminaltyp für jeden `ttyv` Eintrag in /etc/ttys.

Damit der Druck funktioniert, wird ein spezieller Filter zur Übersetzung von KOI8-R nach CP866 benötigt, da die meisten Drucker mit russischen Zeichen die Codetabelle CP866 verwenden. FreeBSD enthält im Basissystem einen Filter zu diesem Zweck. Um diesen Filter zu benutzen, fügen Sie folgenden Eintrag in /etc/printcap ein:

```
lp|Russian local line printer:\
    :sh:of=/usr/libexec/lpr/ru/koi2alt:\
    :lp=/dev/lpt0:sd=/var/spool/output/lpd:lf=/var/log/lpd-errs:
```

printcap(5) enthält eine ausführlichere Erklärung.

Russische Dateinamen auf MS-DOS® Dateisystemen werden durch `-L` und dem Namen der Locale in /etc/fstab erkannt:

```
/dev/ad0s2      /dos/c  msdos   rw,-Lru_RU.KOI8-R 0 0
```

Weitere Informationen finden Sie in mount_msdosfs(8).

Wenn Sie Xorg verwenden, installieren Sie das Paket x11-fonts/xorg-fonts-cyrillic. Im Abschnitt `"Files"` von /etc/X11/xorg.conf fügen Sie dann den folgenden Eintrag *vor* allen anderen `FontPath` Einträgen ein:

```
FontPath   "/usr/local/lib/X11/fonts/cyrillic"
```

Zusätzliche kyrillische Schriftarten finden Sie in der Ports-Sammlung.

Die Unterstützung für eine russische Tastatur aktivieren Sie im Abschnitt `"Keyboard"` von xorg.conf:

```
Option "XkbLayout"   "us,ru"
```

```
Option "XkbOptions"  "grp:toggle"
```

Stellen Sie zudem sicher, dass `XkbDisable` auskommentiert ist.

Beim Einsatz von `grp:toggle` können Sie mit `Right Alt` (Alt Gr) zwischen dem RUS- und LAT-Modus wechseln, verwenden Sie hingegen `grp:ctrl_shift_toggle`, so erfolgt der Wechsel mit `Ctrl` + `Shift`. Für `grp:caps_toggle` ist zum Wechseln des RUS/LAT-Modus `CapsLock` zuständig. Die alte Funktion von `CapsLock` steht nur im LAT-Modus mit der Tastenkombination `Shift` + `CapsLock` zur Verfügung. `grp:caps_toggle` funktioniert aus unbekannten Gründen unter Xorg nicht.

Wenn die Tastatur Windows®-Tasten besitzt und nicht-alphanumerische Tasten nicht funktionieren, fügen Sie die folgende Zeile in xorg.conf ein:

```
Option "XkbVariant" ",winkeys"
```



Die russische XKB-Tastatur funktioniert vielleicht nicht mit nicht-lokalisierten Anwendungen. Lokalisierte Anwendungen sollten mindestens die Funktion `XtSetLanguageProc (NULL, NULL, NULL);` frühzeitig aufrufen.

Weitere Informationen über die Lokalisierung von Xorg-Anwendungen erhalten Sie auf der Webseite http://koi8.pp.ru/xwin.html. Allgemeine Informatinen über die KOI8-R Codierung finden Sie auf http://koi8.pp.ru.

### 40.4.2. Weitere sprachspezifische Ressourcen

Dieser Abschnitt enthält einige zusätzliche Ressourcen für die Konfiguration anderer Lokalisierungen.

**Traditionelles Chinesisch für Taiwan**

Das taiwanesische FreeBSD Projekt stellt ein Tutorium unter http://netlab.cse.yzu.edu.tw/~statue/freebsd/zh-tut/ zur Verfügung.

**Griechische Lokalisierung**

Ein Artikel über die Unterstützung für Griechisch steht unter http://www.freebsd.org/doc/el/articles/greek-language-support/. Bitte beachten Sie, dass dies *nur* für Griechisch gilt.

**Japanische und koreanische Lokalisierung**

Informationen über die japanische Lokalisierung entnehmen Sie bitte http://www.jp.FreeBSD.org/, Informationen über die koreanische Lokalisierung erhalten Sie unter http://www.kr.FreeBSD.org/.

**Nicht-englische FreeBSD-Dokumentation**

Teile der FreeBSD Dokumentation wurden von Beitragenden in andere Sprachen übersetzt. Folgen Sie den Links auf der FreeBSD-Webseite oder schauen Sie in /usr/shared/doc nach.

# Kapitel 41. FreeBSD aktualisieren

## 41.1. Übersicht

FreeBSD wird zwischen einzelnen Releases ständig weiter entwickelt. Manche Leute bevorzugen die offiziellen Release-Versionen, während andere wiederum lieber auf dem aktuellen Stand der Entwicklung bleiben möchten. Wie dem auch sei, sogar offizielle Release-Versionen werden oft mit Sicherheitsaktualisierungen und anderen kritischen Fehlerbereinigungen versorgt. Unabhängig von der eingesetzten Version bringt FreeBSD alle nötigen Werkzeuge mit, um das System aktuell zu halten und es innerhalb verschiedener Versionen zu aktualisieren. Dieses Kapitel beschreibt, wie man einem Entwicklungssystem folgen kann, sowie die grundlegenden Werkzeuge um FreeBSD zu aktualisieren.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- wissen, wie das System mit freebsd-update oder Subversion aktualisiert wird.
- wissen, wie man das aktuell installierte System mit einer ursprünglichen Version vergleicht.
- wissen, wie die installierte Dokumentation mit Subversion oder Dokumentations-Ports aktualisiert wird.
- den Unterschied zwischen den beiden Entwicklungszweigen FreeBSD-STABLE und FreeBSD-CURRENT kennen.
- wissen, wie das komplette Basissystem neu gebaut und installiert wird.

Bevor Sie dieses Kapitel lesen, sollten Sie

- das Netzwerk richtig konfiguriert haben (Weiterführende Netzwerkthemen).
- wissen, wie Software Dritter installiert wird (Installieren von Anwendungen: Pakete und Ports).



In diesem Kapitel wird `svnlite` benutzt, um die FreeBSD Quellen zu beziehen und zu aktualisieren. Alternativ kann auch der Port oder das Paket devel/subversion installiert werden.

## 41.2. FreeBSD-Update

Das zeitnahe Einspielen von Sicherheitsaktualisierungen und die Aktualisierung des Betriebssystems sind wichtige Aspekte der Systemadministration. FreeBSD enthält das Werkzeug `freebsd-update`, mit dem Sie diese beiden Aufgaben erfüllen können.

Dieses Werkzeug ermöglicht die Anwendung von Sicherheitsaktualisierungen im Binärformat auf das FreeBSD Basissystem, ohne dieses neu zu übersetzen und zu installieren. Die Aktualisierungen im Binärformat sind für alle Architekturen und Versionen verfügbar, welche vom FreeBSD Sicherheitsteam unterstützt werden. Eine Liste der unterstützten Versionen und der End-of-Life-Daten ist unter https://www.FreeBSD.org/security/ aufgeführt.

`freebsd-update` unterstützt auch die Aktualisierung des Betriebssystems auf eine neuere

Unterversion sowie eine Aktualisierung auf einen anderen Release-Zweig. Bevor Sie auf eine neue Version aktualisieren, sollten Sie die aktuellen Ankündigungen zu dem Release gelesen haben, da diese wichtige Informationen zu dem entsprechenden Release enthalten. Ankündigungen finden Sie unter https://www.FreeBSD.org/releases/.



Wenn eine `crontab` existiert, welche die Eigenschaften von freebsd-update(8) verwendet, muss diese deaktiviert werden, bevor das Betriebssystem aktualisiert wird.

Dieser Abschnitt beschreibt die Verwendung der Konfigurationsdatei von `freebsd-update`. Es wird gezeigt wie Sie Sicherheitsaktualisierungen einspielen, oder wie Sie das Betriebssystem auf neuere Haupt- und Unterversionen aktualisieren können.

### 41.2.1. Die Konfigurationsdatei

In der Regel muss die Konfigurationsdatei von `freebsd-update` nicht bearbeitet werden. Manche Benutzer möchten die Standard-Konfigurationsdatei /etc/freebsd-update.conf trotzdem anpassen, um bessere Kontrolle über den gesamten Prozess zu besitzen. Die Kommentare in dieser Datei beschreiben die verfügbaren Optionen, jedoch benötigen die folgenden ein paar zusätzliche Erklärungen:

```
# Components of the base system which should be kept updated.
Components world kernel
```

Dieser Parameter kontrolliert, welche Teile von FreeBSD auf dem aktuellen Stand gehalten werden sollen. In der Voreinstellung wird das gesamte Basissystem sowie der Kernel aktualisiert. Es können auch einzelne Komponenten, wie `src/base` oder `src/sys`, angegeben werden. Die beste Einstellung ist, diese Option so zu belassen, da eine Änderung es bedingt, dass man als Benutzer jede Komponente auflisten muss, die aktualisiert werden soll. Dies könnte katastrophale Folgen nach sich ziehen, da der Quellcode und die Binärdateien dadurch nicht mehr synchron wären.

```
# Paths which start with anything matching an entry in an IgnorePaths
# statement will be ignored.
IgnorePaths /boot/kernel/linker.hints
```

Fügen Sie Pfade wie /bin oder /sbin hinzu, um diese speziellen Verzeichnisse während des Aktualisierungsprozesses unberührt zu lassen. Diese Option kann verwendet werden, um zu verhindern, dass `freebsd-update` lokale Änderungen überschreibt.

```
# Paths which start with anything matching an entry in an UpdateIfUnmodified
# statement will only be updated if the contents of the file have not been
# modified by the user (unless changes are merged; see below).
UpdateIfUnmodified /etc/ /var/ /root/ /.cshrc /.profile
```

Diese Option aktualisiert nur unmodifizierte Konfigurationsdateien in den angegebenen Verzeichnissen. Jede Änderung, die der Benutzer daran vorgenommen hat, wird die automatische

Aktualisierung dieser Dateien verhindern. Es gibt eine weitere Option `KeepModifiedMetadata`, die `freebsd-update` instruiert, die Änderungen während der Zusammenführung zu speichern.

```
# When upgrading to a new FreeBSD release, files which match MergeChanges
# will have any local changes merged into the version from the new release.
MergeChanges /etc/ /var/named/etc/ /boot/device.hints
```

Eine Liste von Verzeichnissen mit Konfigurationsdateien, in denen `freebsd-update` Zusammenführungen versuchen soll. Dieser Verschmelzungsprozess von Dateien ist eine Serie von diff(1)-Korrekturen, ähnlich wie mergemaster(8), aber mit weniger Optionen. Die Änderungen werden entweder akzeptiert, oder öffnen einen Editor, oder `freebsd-update` bricht ab. Im Zweifelsfall sichern Sie /etc und akzeptieren einfach die Änderungen. Lesen Sie mergemaster(8), um Informationen über `mergemaster` zu erhalten.

```
# Directory in which to store downloaded updates and temporary
# files used by FreeBSD Update.
# WorkDir /var/db/freebsd-update
```

In diesem Verzeichnis werden alle Korrekturen und temporären Dateien abgelegt. Im Falle einer Versionsaktualisierung sollte diesem Verzeichnis mindestens ein Gigabyte Festplattenspeicher zur Verfügung stehen.

```
# When upgrading between releases, should the list of Components be
# read strictly (StrictComponents yes) or merely as a list of components
# which *might* be installed of which FreeBSD Update should figure out
# which actually are installed and upgrade those (StrictComponents no)?
# StrictComponents no
```

Wenn diese Option auf `yes` gesetzt ist, wird `freebsd-update` annehmen, dass die `Components`-Liste vollständig ist und nicht versuchen, Änderungen ausserhalb dieser Liste zu tätigen. Tatsächlich wird `freebsd-update` versuchen, jede Datei zu aktualisieren, die zu der `Components`-Liste gehört.

### 41.2.2. Sicherheitskorrekturen anwenden

Das Einspielen von FreeBSD Sicherheitskorrekturen wurde dahingehend vereinfacht, dass der Administrator nun das gesamte System mit `freebsd-update` auf dem aktuellen Stand halten kann. Weitere Informationen zu FreeBSD Sicherheitshinweisen finden Sie in FreeBSD Sicherheitshinweise.

Sicherheitskorrekturen für FreeBSD können wie folgt heruntergeladen und installiert werden. Das erste Kommando prüft, ob noch ausstehende Korrekturen verfügbar sind, und wenn dass der Fall ist, zeigt es welche Dateien davon betroffen wären. Das zweite Kommando wird die Korrekturen auf das System anwenden.

```
# freebsd-update fetch
```

```
# freebsd-update install
```

Wenn während der Aktualisierung Korrekturen am Kernel angewendet werden, muss das System neu gestartet werden, damit der korrigierte Kernel gebootet wird. Wenn die Korrekturen auf laufende Binärdateien angewendet werden, sollten die betroffenen Anwendungen neu gestartet werden, damit die korrigierte Version der Binärdatei geladen wird.



Im Regelfall muss der Benutzer darauf vorbereitet sein, das System neu zu starten. Um herauszufinden, ob ein Neustart durch eine Aktualisierung des Kernels erforderlich ist, führen Sie die Befehle `freebsd-version -k` und `uname -r` aus. Ist die Ausgabe dieser Befehle unterschiedlich, ist ein Neustart des Systems erforderlich.

Mit dem folgenden Eintrag in /etc/crontab wird das System einmal täglich nach Aktualisierungen suchen:

```
@daily                                  root    freebsd-update cron
```

Wenn Korrekturen existieren, werden diese automatisch heruntergeladen, aber nicht eingespielt. Der `root`-Benutzer bekommt eine Nachricht, damit die Korrekturen überprüft und mit `freebsd-update install` manuell installiert werden können.

Wenn etwas schief geht, kann `freebsd-update` den letzten Satz von Änderungen mit folgendem Befehl rückgängig machen:

```
# freebsd-update rollback
Uninstalling updates... done.
```

Wie bereits erwähnt, sollte das System neu gestartet werden, wenn der Kernel oder ein Kernelmodul verändert wurde. Betroffene Anwendungen sollten neu gestartet werden, wenn Binärdateien verändert wurden.

Das `freebsd-update`-Werkzeug kann nur den GENERIC-Kernel automatisch aktualisieren. Wenn ein angepasster Kernel verwendet wird, muss dieser neu erstellt und installiert werden, nachdem `freebsd-update` die Aktualisierungen durchgeführt hat. Der voreingestellte Kernel ist *GENERIC*. Benutzen Sie das Kommando uname(1) um dies zu überprüfen.



Behalten Sie immer eine Kopie des GENERIC-Kernels in /boot/GENERIC. Das wird bei der Diagnose von verschiedenen Problemen sowie bei der Durchführung von Versionsaktualisierungen eine große Hilfe sein. Im Angepasste Kernel unter FreeBSD 9.X und später wird beschrieben, wie Sie eine Kopie des GENERIC-Kernels bekommen.

Solange die Standardkonfiguration in /etc/freebsd-update.conf nicht geändert wurde, wird `freebsd-update` die aktualisierten Quellcodedateien des Kernels zusammen mit dem Rest der Neuerungen installieren. Die erneute Übersetzung und Installation eines neuen, angepassten Kernels kann dann auf die übliche Art und Weise durchgeführt werden.

Die Aktualisierungen, die über `freebsd-update` verteilt werden, betreffen nicht immer den Kernel. Es ist nicht notwendig, den angepassten Kernel neu zu erstellen, wenn die Kernelquellen nicht durch `freebsd-update install` geändert wurden. Allerdings wird `freebsd-update` immer /usr/src/sys/conf/newvers.sh aktualisieren. Der aktuelle Patch-Level, der mit der `-p`-Nummer bei `uname -r` ausgegeben wird, wird aus dieser Datei ausgelesen. Die Neuinstallation des angepassten Kernels, selbst wenn sich daran nichts geändert hat, erlaubt es `uname`, den aktuellen Patch-Level des Systems korrekt wiederzugeben. Dies ist besonders hilfreich, wenn mehrere Systeme gewartet werden, da es eine schnelle Einschätzung der installierten Aktualisierungen in jedem einzelnen System ermöglicht.

### 41.2.3. Aktualisierungen an Haupt- und Unterversionen

Aktualisierungen einer Unterversion von FreeBSD zur nächsten Version ist beispielsweise die Aktualisierung von FreeBSD 9.0 auf FreeBSD 9.1. Die Aktualisierung einer Hauptversion ist beispielsweise von FreeBSD 9.X auf FreeBSD 10.X. Beide Arten der Aktualisierungen können durchgeführt werden, indem man `freebsd-update` eine Release-Version als Ziel übergibt.



Wenn auf dem System ein angepasster Kernel eingesetzt wird, stellen Sie sicher, dass eine Kopie des GENERIC-Kernels in /boot/GENERIC existiert. Im Angepasste Kernel unter FreeBSD 9.X und später wird beschrieben, wie Sie eine Kopie des GENERIC-Kernels bekommen.

Wenn Sie das folgende Kommando auf einem System mit FreeBSD 9.0 ausführen, wird das System auf FreeBSD 9.1 aktualisiert:

```
# freebsd-update -r 9.1-RELEASE upgrade
```

Nach der Eingabe des Kommandos überprüft `freebsd-update` die Konfigurationsdatei und das aktuelle System, um die nötigen Informationen für die Systemaktualisierung zu sammeln. Eine Bildschirmausgabe wird anzeigen, welche Komponenten erkannt und welche nicht erkannt wurden. Zum Beispiel:

```
Looking up update.FreeBSD.org mirrors... 1 mirrors found.
Fetching metadata signature for 9.0-RELEASE from update1.FreeBSD.org... done.
Fetching metadata index... done.
Inspecting system... done.

The following components of FreeBSD seem to be installed:
kernel/smp src/base src/bin src/contrib src/crypto src/etc src/games
src/gnu src/include src/krb5 src/lib src/libexec src/release src/rescue
src/sbin src/secure src/share src/sys src/tools src/ubin src/usbin
world/base world/info world/lib32 world/manpages

The following components of FreeBSD do not seem to be installed:
kernel/generic world/catpages world/dict world/doc world/games
world/proflibs
```

```
Does this look reasonable (y/n)? y
```

An diesem Punkt wird `freebsd-update` versuchen, alle notwendigen Dateien für die Aktualisierung herunter zu laden. In manchen Fällen wird der Benutzer mit Fragen konfrontiert, um festzustellen, was installiert werden soll oder auf welche Art und Weise fortgesetzt werden soll.

Wenn ein angepasster Kernel benutzt wird, produziert der vorherige Schritt eine Warnung ähnlich zu der folgenden:

```
WARNING: This system is running a "
MYKERNEL" kernel, which is not a
kernel configuration distributed as part of FreeBSD 9.0-RELEASE.
This kernel will not be updated: you MUST update the kernel manually
before running "/usr/sbin/freebsd-update install"
```

Diese Warnung kann an dieser Stelle problemlos ignoriert werden. Der aktualisierte GENERIC -Kernel wird als ein Zwischenschritt im Aktualisierungsprozess verwendet.

Nachdem alle Korrekturen auf das lokale System heruntergeladen wurden, werden diese eingespielt. Dieser Prozess kann eine gewisse Zeit in Anspruch nehmen, abhängig von der Geschwindigkeit und Auslastung der Maschine. Konfigurationsdateien werden ebenfalls zusammengefügt. Dieser Teil der Prozedur verlangt einige Benutzereingaben, da eine Datei möglicherweise von Hand zusammengefasst werden muss oder ein Editor erscheint auf dem Bildschirm zum manuellen bearbeiten. Die Ergebnisse von jeder erfolgreichen Zusammenfassung werden dem Benutzer angezeigt, während der Prozess weiter läuft. Eine fehlgeschlagene oder ignorierte Zusammenfassung wird den Prozess sofort beenden. Benutzer sollten eine Sicherung von /etc anlegen und wichtige Dateien später manuell vereinen, beispielsweise master.passwd oder group.



Das System ist zu diesem Zeitpunkt noch nicht verändert worden, da alle Korrekturen und Vereinigungen in einem anderen Verzeichnis vorgenommen wurden. Wenn alle Korrekturen erfolgreich eingespielt, alle Konfigurationsdateien zusammengefügt wurden und es den Anschein hat, dass der Prozess problemlos verlaufen wird, müssen die Änderungen vom Anwender noch angewendet und auf die Platte geschrieben werden:

```
# freebsd-update install
```

Der Kernel und die Module werden zuerst aktualisiert. Wenn das System einen angepassten Kernel verwendet, benutzen Sie nextboot(8), um den Kernel für den nächsten Neustart auf /boot/GENERIC zu setzen:

```
# nextboot -k GENERIC
```



Bevor das System mit dem GENERIC-Kernel neu gestartet wird, vergewissern Sie

> sich, dass für den Neustart alle benötigten Treiber enthalten sind. Falls auf die Maschine aus der Ferne zugegriffen wird, stellen Sie sicher, dass das System ordnungsgemäß an das Netzwerk angeschlossen ist. Achten Sie besonders darauf, dass wenn der angepasste Kernel Funktionalität beinhaltet, die normalerweise von Kernelmodulen zur Verfügung gestellt werden, dass diese temporär über /boot/loader.conf in den GENERIC-Kernel übernommen werden. Zudem wird empfohlen, nicht benötigte Dienste, eingehängte Platten und verbundene Netzlaufwerke zu deaktivieren, bis der Aktualisierungsprozess abgeschlossen ist.

Die Maschine sollte nun mit dem aktualisierten Kernel neu gestartet werden:

```
# shutdown -r now
```

Sobald das System wieder hochgefahren ist, muss `freebsd-update` erneut gestartet werden. Da der Zustand des Prozesses zuvor gesichert wurde, wird `freebsd-update` nicht von vorne beginnen, sondern mit der nächsten Phase fortfahren und alle alten Bibliotheken und Objektdateien löschen.

```
# freebsd-update install
```



> Abhängig davon, ob irgendwelche Bibliotheksversionen erhöht wurden, kann es sein, dass nur zwei Installationsphasen anstatt drei durchlaufen werden.

Die Aktualisierung ist nun abgeschlossen. Wenn es sich hierbei um eine Aktualisierung auf eine neue Hauptversion handelt, müssen alle Ports und Pakete neu installiert werden. Dieser Vorgang wird in Aktualisierung der Pakete nach einem Upgrade auf eine Hauptversion beschrieben.

**Angepasste Kernel unter FreeBSD 9.X und später**

Stellen Sie vor der ersten Benutzung von `freebsd-update` sicher, dass eine Kopie des GENERIC -Kernels in /boot/GENERIC existiert. Wenn ein angepasster Kernel erstmalig gebaut wurde, ist der Kernel in /boot/kernel.old der GENERIC-Kernel. Benennen Sie dieses Verzeichnis einfach in /boot/GENERIC um.

Wenn bereits mehrfach ein angepasster Kernel gebaut wurde, oder nicht bekannt ist wie oft ein angepasster Kernel gebaut wurde, behalten Sie besser eine Kopie des GENERIC-Kernels, welcher mit der aktuellen Version des Betriebssystems übereinstimmt. Wenn ein direkter Zugriff auf die Maschine möglich ist, kann eine Kopie des GENERIC-Kernels von den Installationsmedien installiert werden:

```
# mount /cdrom
# cd /cdrom/usr/freebsd-dist
# tar -C/ -xvf kernel.txz boot/kernel/kernel
```

Alternativ kann der GENERIC-Kernel aus den Quellen neu gebaut und installiert werden:

```
# cd /usr/src
# make kernel __MAKE_CONF=/dev/null SRCCONF=/dev/null
```

Damit dieser Kernel als GENERIC-Kernel von `freebsd-update` erkannt wird, darf die GENERIC -Konfigurationsdatei in keiner Weise geändert worden sein. Es wird ebenfalls empfohlen, dass dieser ohne irgendwelche speziellen Optionen erstellt wird.

Der Neustart in den GENERIC-Kernel ist nicht notwendig, da `freebsd-update` lediglich /boot/GENERIC benötigt.

**Aktualisierung der Pakete nach einem Upgrade auf eine Hauptversion**

In der Regel funktionieren nach einer Aktualisierung einer Unterversion die installierten Anwendungen weiterhin problemlos. Neue Hauptversionen verwenden jedoch andere Binärschnittstellen (ABIs), was dazu führt, dass die meisten Anwendungen von Drittherstellern nicht mehr funktionieren. Nach der Aktualisierung auf eine Hauptversion, müssen alle installierten Ports und Pakete aktualisiert werden. Benutzen Sie `pkg upgrade` um Pakte zu aktualisieren. Installierte Ports können Sie mit einem Werkzeug wie ports-mgmt/portmaster aktualisiert werden.

Bei einer erzwungenen Aktualisierung aller installierten Pakete, werden diese durch eine neue Version aus dem Repository ersetzt, sogar dann, wenn sich die Versionsnummer nicht erhöht hat. Dieser Schritt ist erforderlich, da sich die ABI bei einer Aktualisierung der Hauptversion von FreeBSD verändert hat. Eine erzwungene Aktualisierung aller installierten Pakete geschieht wie folgt:

```
# pkg-static upgrade -f
```

Ein Neubau der installierten Ports führen Sie mit diesem Kommando durch:

```
# portmaster -af
```

Dieser Befehl wird die Konfigurationen für jede Anwendung anzeigen, und der Benutzer hat die Möglichkeit, die Optionen anzupassen. Wenn Sie ausschließlich die voreingestellten Optionen verwenden möchten, verwenden Sie mit dem obigen Befehl den Parameter `-G`.

Sobald dies abgeschlossen ist, beenden Sie den Aktualisierungsprozess mit einem letzten Aufruf von `freebsd-update`. Geben Sie den folgenden Befehl ein, um alle losen Enden des Aktualisierungsprozesses miteinander zu verknüpfen:

```
# freebsd-update install
```

Wenn der GENERIC-Kernel temporär Verwendung fand, ist dies der richtige Zeitpunkt, einen neuen, angepassten Kernel nach den Anweisungen in Konfiguration des FreeBSD-Kernels zu bauen und zu installieren.

Booten Sie anschließend die Maschine in die neue FreeBSD-Version. Der Aktualisierungsprozess ist damit abgeschlossen.

### 41.2.4. Vergleich des Systemzustands

`freebsd-update IDS` kann verwendet werden, um den Zustand der installierten FreeBSD-Version gegenüber einer bekannten und funktionierenden Kopie zu vergleichen. Dieses Kommando vergleicht die aktuelle Version von Systemwerkzeugen, Bibliotheken sowie Konfigurationsdateien und kann als integriertes Intrusion Detection System (IDS) benutzt werden.



Dieses Programm ist kein Ersatz für ein echtes IDS-System wie security/snort. Da `freebsd-update` Daten auf der Festplatte speichert, ist die Möglichkeit von Verfälschungen offensichtlich. Obwohl diese Möglichkeit durch die Verwendung von `kern.securelevel` oder die Speicherung von Daten auf einem Nur-Lese Dateisystem eingedämmt werden kann, besteht eine bessere Lösung darin, das System gegen ein gesichertes Medium, wie eine DVD oder einen externen, separat aufbewahrten USB-Plattenspeicher, zu vergleichen. Eine alternative Methode zur Bereitstellung von IDS-Funktionaliäten wird in Überprüfung von Binärdateien beschrieben.

Beginnen Sie den Vergleich, indem Sie das Programm starten und eine Ausgabedatei festlegen:

```
# freebsd-update IDS >> outfile.ids
```

Das System wird nun überprüft. Dabei wird eine lange Liste von Dateien zusammen mit den SHA256-Hashwerten der Release-Version und den Werten des aktuell installierten Systems, in die angegebene Ausgabedatei geschrieben.

Die Zeilen in der Ausgabe sind extrem lang, aber das Ausgabeformat kann einfach verarbeitet werden. Um beispielsweise eine Liste von allen Dateien zu erhalten, die sich vom aktuellen Release unterscheiden, geben Sie das folgende Kommando ein:

```
# cat outfile.ids | awk '{ print $1 }' | more
/etc/master.passwd
/etc/motd
/etc/passwd
/etc/pf.conf
```

Diese Beispielausgabe wurde abgeschnitten, da noch viele weitere Dateien vorhanden sind. Einige Dateien wurden auf natürliche Art verändert. /etc/passwd wurde beispielsweise geändert, wenn Benutzer zum System hinzugefügt wurden. Kernelmodule können sich unterscheiden, wenn `freebsd-update` diese aktualisiert hat. Um bestimmte Dateien oder Verzeichnisse auszuschließen, fügen Sie diese an die `IDSIgnorePaths`-Option in /etc/freebsd-update.conf an.

## 41.3. Aktualisieren der Dokumentationssammlung

Dokumentation ein wichtiger Bestandteil des FreeBSD Betriebssystems. Obwohl eine aktuelle Version der FreeBSD Dokumentation jederzeit auf der FreeBSD Webseite ( https://www.freebsd.org/doc/) verfügbar ist, kann es nützlich sein, eine lokale Kopie der FreeBSD Webseite, Handbücher, FAQ und Artikel zu haben.

Dieser Abschnitt beschreibt, wie Sie die FreeBSD Dokumentation über die Quellen oder die FreeBSD Ports-Sammlung aktuell halten.

Informationen zum Bearbeiten und Einreichen von Korrekturen finden Sie in der Fibel für neue Mitarbeiter des FreeBSD-Dokumentationsprojekts.

### 41.3.1. Die FreeBSD-Dokumentation aus den Quellen installieren

Der Bau der FreeBSD Dokumentation aus den Quellen erfordert einige Werkzeuge, die nicht Teil des Basissystems sind. Die erforderlichen Werkzeuge können über den Port oder das Paket textproc/docproj installiert werden.

Benutzen Sie nach der Installation svnlite, um eine saubere Kopie der Dokumentationsquellen zu holen:

```
# svnlite checkout https://svn.FreeBSD.org/doc/head /usr/doc
```

Es dauert eine Weile, bis die Quellen das allererste Mal heruntergeladen werden. Lassen Sie den Vorgang laufen, bis es fertig ist.

Zukünftige Aktualisierungen der Dokumentationsquellen können wie folgt durchgeführt werden:

```
# svnlite update /usr/doc
```

Sobald ein aktueller Schnappschuss der Dokumentationsquellen nach /usr/doc heruntergeladen wurde, ist alles bereit für eine Aktualisierung der bestehenden Dokumentation.

Eine komplette Aktualisierung aller Sprachen kann durch folgende Eingabe erreicht werden:

```
# cd /usr/doc
# make install clean
```

Wenn nur eine Aktualisierung einer bestimmten Sprache gewünscht wird, kann `make` in einem sprachspezifischen Unterverzeichnis von /usr/doc aufgerufen werden:

```
# cd /usr/doc/en_US.ISO8859-1
# make install clean
```

Alternativ kann der folgende Befehl in /usr/doc oder einem sprachspezifischen Unterverzeichnis abgesetzt werden, um die Dokumentation zu aktualisieren:

```
# make update
```

Die zu installierenden Ausgabeformate können durch das Setzen von FORMATS angegeben werden:

```
# cd /usr/doc
# make FORMATS='html html-split' install clean
```

Es existieren ein paar Optionen, welche den Prozess der Aktualisierung von Teilen der Dokumentation oder einer bestimmten Übersetzung erleichtern. Diese Optionen können entweder systemweit in /etc/make.conf gesetzt, oder als Kommandozeilenoptionen an make übergeben werden.

Zu den Optionen gehören:

DOC_LANG
: Eine Liste von Sprachen und Kodierungen, die gebaut und installiert werden sollen, z.B. en_US.ISO8859-1, um nur die englische Dokumentation zu erhalten.

FORMATS
: Ein einzelnes Format oder eine Liste von Ausgabeformaten, das gebaut werden soll. Momentan werden html, html-split, txt, ps und pdf unterstützt.

DOCDIR
: Wohin die Dokumentation installiert werden soll. Der Standardpfad ist /usr/shared/doc.

Für weitere make-Variablen, die als systemweite Optionen in FreeBSD unterstützt werden, lesen Sie make.conf(5).

### 41.3.2. Die Dokumentation aus den Ports aktualisieren

Im vorherigen Abschnitt wurde eine Methode gezeigt, wie die FreeBSD-Dokumentation aus den Quellen gebaut werden kann. Dieser Abschnitt beschreibt eine alternative Methode, in der die Ports-Sammlung verwendet wird und die es ermöglicht:

- vorgefertigte Schnappschüsse der Dokumentation zu installieren, ohne vorher die Werkzeugsammlung der Dokumentation installieren zu müssen.
- die Dokumentationsquellen durch das Ports-System erstellen zu lassen, was die Schritte zum Auschecken und Erstellen etwas erleichtert.

Diese Methoden der Aktualisierung der FreeBSD-Dokumentation werden durch eine Menge von Dokumentations-Ports und Paketen unterstützt, die von Documentation Engineering Team <doceng@FreeBSD.org> monatlich aktualisiert wird. Diese sind in der FreeBSD Ports-Sammlung unter der Kategorie "docs" gelistet ( http://www.freshports.org/docs/).

Die Dokumentations-Ports sind wie folgt organisiert:

- Das Paket oder der Port misc/freebsd-doc-en installiert die englische Dokumentation.
- Das Paket oder der Port misc/freebsd-doc-all installiert die komplette Dokumentation in allen verfügbaren Sprachen.
- Es gibt noch ein Paket oder einen Port für jede Übersetzung, beispielsweise misc/freebsd-doc-hu für die ungarische Dokumentation.

Wenn Sie Pakete benutzen, wird die FreeBSD-Dokumentation in allen verfügbaren Formaten der jeweiligen Sprache installiert. Das folgende Beispiel wird das aktuelle Paket der ungarischen Dokumentation installieren:

```
# pkg install hu-freebsd-doc
```



Pakete verwenden ein Format, welches sich von dem Namen des dazugehörigen Ports unterscheidet: `lang-freebsd-doc`. *lang* entspricht hier der Kurzform des Sprachcodes, z.B. `hu` für Ungarisch, oder `zh_cn` für vereinfachtes Chinesisch.

Um das Format der Dokumentation zu bestimmen, muss anstelle des Pakets der Port gebaut werden. Das folgende Beispiel baut und installiert die englische Dokumentation:

```
# cd /usr/ports/misc/freebsd-doc-en
# make install clean
```

Der Port enthält ein Konfigurationsmenü, in dem das Format ausgewählt werden kann. In der Voreinstellung sind `html-split` und `pdf` ausgewählt.

Alternativ können bei der Erstellung eines Dokumentations-Ports verschiedene `make`-Optionen angegeben werden. Dazu gehören:

`WITH_HTML`
: Erstellt das HTML-Format mit einer einzigen HTML-Datei pro Dokument. Die formatierte Dokumentation wird als Datei mit dem Namen article.html oder book.html gespeichert.

`WITH_PDF`
: Die formatierte Dokumentation wird als Datei mit dem Namen article.pdf oder book.pdf gespeichert.

`DOCBASE`
: Legt den Pfad fest, wohin die Dokumentation installiert werden soll. Die Voreinstellung ist /usr/local/shared/doc/freebsd.

Dieses Beispiel verwendet Variablen, um die ungarische Dokumentation als PDF in ein bestimmtes Verzeichnis zu installieren:

```
# cd /usr/ports/misc/freebsd-doc-hu
```

```
# make -DWITH_PDF DOCBASE=share/doc/freebsd/hu install clean
```

Dokumentations-Ports oder -Pakete können nach den Anweisungen in Installieren von Anwendungen: Pakete und Ports aktualisiert werden. Beispielsweise aktualisiert das folgende Kommando die installierte ungarische Dokumentation mittels ports-mgmt/portmaster unter Verwendung von Paketen:

```
# portmaster -PP hu-freebsd-doc
```

# 41.4. Einem Entwicklungszweig folgen

FreeBSD besitzt zwei Entwicklungszweige: FreeBSD-CURRENT und FreeBSD-STABLE.

Dieser Abschnitt beschreibt beide Zweige sowie deren Interessengruppen und erläutert, wie ein System auf dem aktuellen Stand eines jeweiligen Zweiges gehalten wird.

## 41.4.1. FreeBSD-CURRENT

FreeBSD-CURRENT ist die allerneueste Entwicklung von FreeBSD. Benutzer von FreeBSD-CURRENT sollten über sehr gute technische Fähigkeiten verfügen. Benutzer mit weniger technischen Fähigkeiten sollten stattdessen FreeBSD-STABLE benutzen, wenn sie einem Entwicklungszweig folgen möchten.

FreeBSD-CURRENT besteht aus den neuesten Quellen des FreeBSD-Systems und enthält Sachen, an denen gerade gearbeitet wird, experimentelle Änderungen und Übergangsmechanismen, die im nächsten offiziellen Release enthalten sein können oder nicht. Obwohl FreeBSD-CURRENT täglich von vielen Entwicklern gebaut wird, gibt es Zeiträume, in denen sich das System vielleicht nicht bauen lässt. Diese Probleme werden so schnell wie möglich behoben, aber ob Sie mit FreeBSD-CURRENT eine Katastrophe erleben oder neue Funktionen erhalten, kann von dem Zeitpunkt abhängen, an dem der Quelltext synchronisiert wurde.

FreeBSD-CURRENT wird hauptsächlich für drei Interessengruppen zur Verfügung gestellt:

1. Mitglieder der FreeBSD Gemeinschaft, die aktiv an einem Teil des Quellbaums arbeiten.
2. Mitglieder der FreeBSD Gemeinschaft, die aktive Tester sind. Diese Personen sind bereit, Zeit in das Lösen von Problemen zu investieren, Vorschläge zu Änderungen oder der generellen Entwicklung von FreeBSD zu machen und Fehlerkorrekturen einzureichen.
3. Benutzer, die die Entwicklung im Auge behalten, oder die Quellen zu Referenzzwecken benutzen wollen. Diese Gruppe macht auch Vorschläge oder steuert Quellcode bei.

FreeBSD-CURRENT ist *nicht* der schnellste Weg, neue Funktionen vor dem offiziellen Release auszuprobieren. Bedenken Sie, dass neue Funktionen noch nicht im vollen Umfang getestet wurden und daher höchstwahrscheinlich Fehler enthalten. Es ist auch nicht der schnellste Weg, um an Fehlerbehebungen (engl. bug fixes) zu kommen. Jede Fehlerbehebung führt mit gleicher Wahrscheinlichkeit neue Fehler ein, mit der sie alte behebt. FreeBSD-CURRENT wird in keiner Weise "offiziell unterstützt".

Um FreeBSD-CURRENT zu folgen:

1. Lesen Sie die Mailinglisten FreeBSD-CURRENT und SVN commit messages for the src tree for head/-current. Dies ist *notwendig*, um die Kommentare über den akutellen Status des Systems und wichtige Mitteilungen zum aktuellen Zustand von FreeBSD-CURRENT zu erfahren.

   Die SVN commit messages for the src tree for head/-current Mailingliste erfasst die Commit-Logs für jede Änderung und enthält alle relevanten Informationen zu möglichen Seiteneffekten.

   Um diese Listen zu abonnieren, besuchen Sie https://lists.freebsd.org, klicken Sie auf die gewünschte Liste und folgen Sie den Anweisungen. Wenn Sie die Änderungen am gesamten Quellbaum verfolgen möchten, abonnieren Sie die SVN commit messages for the entire src tree (except for "user" and "projects") Liste.

2. Synchronisieren Sie die Quellen für FreeBSD-CURRENT. In der Regel wird svnlite benutzt, um die Quellen für -CURRENT aus dem Zweig `head` zu laden. Verwenden Sie dazu einen Subversion Spiegel aus “Subversion Mirror Sites”.
3. Aufgrund der Größe des Repositories ist es empfehlenswert, nur die gewünschten Teilbäume auszuchecken. Wenn Sie die Quellen einsetzen und nicht nur darin lesen wollen, laden Sie sich die *kompletten* Quellen von FreeBSD-CURRENT und nicht nur ausgesuchte Teile.

   Lesen Sie /usr/src/Makefile sehr aufmerksam und folgen Sie den Anweisungen in FreeBSD aus den Quellen aktualisieren. Lesen Sie die Mailingliste FreeBSD-CURRENT und /usr/src/UPDATING, um über Änderungen im Installationsverfahren, die manchmal vor der Einführung eines neuen Releases notwendig sind, informiert zu sein.

4. Seien Sie aktiv! Benutzer von FreeBSD-CURRENT werden aufgefordert ihre Verbesserungsvorschläge oder Fehlerbehebungen einzureichen. Verbesserungsvorschläge, die Code enthalten, sind jederzeit herzlich willkommen.

### 41.4.2. FreeBSD-STABLE

FreeBSD-STABLE ist der Entwicklungszweig, auf dem Releases erstellt werden. Dieser Zweig ändert sich langsamer als FreeBSD-CURRENT und alle Änderungen sollten zuvor in FreeBSD-CURRENT ausgetestet sein. Beachten Sie, dass dies *immer noch* ein Entwicklungszweig ist und daher zu jedem Zeitpunkt die Quellen von FreeBSD-STABLE verwendbar sein können oder eben auch nicht. FreeBSD-STABLE ist Teil des Entwicklungsprozesses und nicht für Endanwender gedacht. Benutzer, die nicht über die notwendigen Ressourcen zum Testen verfügen, sollten stattdessen ein aktuelles Release von FreeBSD benutzen.

Wer daran interessiert ist den Entwicklungsprozess von FreeBSD zu verfolgen oder dazu beizutragen, insbesondere im Hinblick auf das nächste Release, der sollte es in Erwägung ziehen FreeBSD-STABLE zu benutzen.

Obwohl wir versuchen sicherzustellen, dass sich FreeBSD-STABLE jederzeit übersetzen lässt und lauffähig ist, können wir dafür keine Garantie übernehmen. Auch wenn Neuentwicklungen in FreeBSD-CURRENT stattfinden, ist es jedoch so, dass mehr Leute FreeBSD-STABLE anstelle von FreeBSD-CURRENT benutzen und es daher unvermeidlich ist, dass Fehler und Grenzfälle erst in FreeBSD-STABLE auffallen. Aus diesen Gründen empfehlen wir, FreeBSD-STABLE *nicht* blindlings

zu benutzen.

Um FreeBSD-STABLE zu folgen:

1. Lesen Sie die Mailingliste FreeBSD-STABLE;, damit Sie über Abhängigkeiten beim Bau von FreeBSD-STABLE und Dinge, die besondere Aufmerksamkeit erfordern, informiert sind. Umstrittene Fehlerbehebungen oder Änderungen werden von den Entwicklern auf dieser Liste bekannt gegeben. Dies erlaubt es den Benutzern, Einwände gegen die vorgeschlagenen Änderungen vorzubringen.

   Abonnieren Sie die passende svn-Liste für den jeweiligen Zweig, den Sie verfolgen. Wenn Sie beispielsweise den Zweig 9-STABLE verfolgen, lesen Sie SVN commit messages for only the 9-stable src tree. Diese Liste enthält zu jeder Änderung das Commit-Log, das Informationen zu möglichen Seiteneffekten enthält.

   Um diese Listen zu abonnieren, besuchen Sie die Seite https://lists.freebsd.org. Klicken Sie auf die gewünschte Liste und folgen Sie den Anweisungen. Wenn Sie daran interessiert sind, Änderungen am gesamten Quellbaum zu verfolgen, abonnieren Sie SVN commit messages for the entire src tree (except for "user" and "projects").

2. Wenn Sie ein neues System installieren und dazu einen der monatlich aus FreeBSD-STABLE erzeugten Snapshots verwenden wollen, sollten Sie zuerst www.freebsd.org/snapshots" auf aktuelle Informationen überprüfen. Alternativ können Sie auch das neueste FreeBSD-STABLE-Release von den FreeBSD Spiegeln beziehen.

   Um ein bestehendes FreeBSD-System auf FreeBSD-STABLE zu aktualisieren, benutzen Sie svn um den gewünschten Entwicklungs- oder Release-Zweig auszuchecken. Die Zweige, wie beispielsweise `stable/9`, sind unter www.freebsd.org/releng aufgeführt.

3. Lesen Sie /usr/src/Makefile sehr aufmerksam bevor Sie FreeBSD-STABLE aktualisieren und folgen Sie den Anweisungen in FreeBSD aus den Quellen aktualisieren. Lesen Sie die Mailingliste FreeBSD-STABLE; und /usr/src/UPDATING, um über Änderungen im Installationsablauf, die manchmal vor der Einführung eines neuen Releases notwendig sind, informiert zu sein.

## 41.5. FreeBSD aus den Quellen aktualisieren

Das Aktualisieren von FreeBSD aus den Quellen bietet im Vergleich zu binären Updates mehrere Vorteile. Der Quellcode kann mit Optionen gebaut werden, um die Vorteile von spezifischer Hardware zu nutzen. Teile des Basissystems können mit veränderten Einstellungen gebaut, oder falls nicht gewünscht, auch ganz ausgelassen werden. Dieser Prozess dauert zwar länger als die Aktualisierung mit binären Updates, ermöglicht aber eine vollständige Anpassung, um eine individuelle Version von FreeBSD zu erstellen.

### 41.5.1. Schnellstartanleitung

Diese kurze Referenz zeigt die typischen Schritte um FreeBSD aus den Quellen zu aktualisieren. Spätere Abschnitte beschreiben die Prozedur im Detail.

- Aktualisierung und Bauprozess*

```
# svnlite update /usr/src  ①
check /usr/src/UPDATING  ②
# cd /usr/src            ③
# make -j4 buildworld    ④
# make -j4 kernel        ⑤
# shutdown -r now        ⑥
# cd /usr/src            ⑦
# make installworld      ⑧
# mergemaster -Ui        ⑨
# shutdown -r now        ⑩
```

① Holt die neueste Version der Quellen. Den Quellcode aktualisieren enthält weitere Informationen zum Aktualisieren und Bauen der Quellen.

② /usr/src/UPDATING enthält Anweisungen für alle manuellen Schritte, die vor oder nach dem Bau der Quellen erforderlich sind.

③ Wechsel in das Bauverzeichnis.

④ Bau des Basissystems, mit Ausnahme des Kernels.

⑤ Bau und Installation des Kernels. Dieser Schritt ist gleichbedeutend mit `make buildkernel installkernel`.

⑥ Installation des Basissystems.

⑦ Aktualisierung und Zusammenführung der Konfigurationsdateien in /etc.

⑧ Neustart des Systems mit dem neu erstellten Basissystem und Kernel.

### 41.5.2. Vorbereitungen zum Aktualisieren aus den Quellen

Lesen Sie /usr/src/UPDATING. Jeder manuelle Schritt, welcher vor oder nach der Aktualisierung erforderlich ist, wird in dieser Datei beschrieben.

### 41.5.3. Den Quellcode aktualisieren

Der Quellcode von FreeBSD befindet sich in /usr/src/. Die bevorzugte Methode zur Aktualisierung dieser Quellen ist über das Versionskontrollsystem Subversion. Sie sollten sicherstellen, dass der Quellcode unter Versionskontrolle steht:

```
# svnlite info /usr/src
Path: /usr/src
Working Copy Root Path: /usr/src
...
```

Dies ist ein Hinweis darauf, dass /usr/src/ unter Versionskontrolle steht und mit svnlite(1) aktualisiert werden kann.

```
# svnlite update /usr/src
```

Dieser Vorgang kann einige Zeit in Anspruch nehmen, falls das Verzeichnis nicht zuletzt aktualisiert wurde. Nach Beendigung ist der Quellcode aktuell und der im nächsten Abschnitt beschriebene Bauprozess kann beginnen.



**Synchronisation der Quellen**

Meldet die Ausgabe `'/usr/src' is not a working copy`, dann fehlen entweder Dateien, oder das Verzeichnis wurde mit einer anderen Methode aktualisiert. Ein erneuter Checkout der Quellen ist jetzt erforderlich.

*Tabelle 18. FreeBSD Versionen und Repository-Pfade*

| Ausgabe von uname -r | Repository-Pfad | Beschreibung |
|---|---|---|
| `X.Y-RELEASE` | `base/releng/X.Y` | Die Release-Version inklusive kritischer Sicherheits- und Bugfix-Patches. Dieser Zweig wird für die meisten Benutzer empfohlen. |
| `X.Y-STABLE` | `base/stable/X` | Die Release-Version und alle weitere Versionen auf diesem Zweig. *STABLE* bezieht sich darauf, dass die Binärschnittstelle (ABI) sich nicht ändert, sodass Anwendungen welche auf älteren Versionen erstellt wurden weiterhin lauffähig sind. Eine Anwendung, welche für FreeBSD 10.1 übersetzt wurde, läuft auch auf FreeBSD 10-STABLE.<br><br>STABLE-Zweige haben gelegentlich Fehler und Inkompatibilitäten, welche den Benutzer beeinträchtigen könnten. In der Regel werden diese Fehler aber zügig behoben. |
| `X-CURRENT` | `base/head/` | Die neueste unveröffentlichte Version von FreeBSD. Der CURRENT-Zweig kann viele Fehler und Inkompatibilitäten enthalten und wird daher nur für fortgeschrittene Benutzer empfohlen. |

Ermitteln Sie mit uname(1) die verwendete FreeBSD-Version:

```
# uname -r
10.3-RELEASE
```

Basierend auf FreeBSD Versionen und Repository-Pfade ist `base/releng/10.3` der Repository-Pfad zur Aktualisierung von `10.3-RELEASE`. Dieser Pfad wird beim Auschecken der Quellen benutzt:

```
# mv /usr/src /usr/src.bak  ①
```

```
# svnlite checkout https://svn.freebsd.org/base/releng/10.3 /usr/src
②
```

① Verschiebt das alte Verzeichnis. Wenn es keine lokalen Änderungen in diesem Verzeichnis gibt, kann es gelöscht werden.

② Der Pfad aus FreeBSD Versionen und Repository-Pfade wird der Repository-URL hinzugefügt. Der dritte Parameter ist das lokale Zielverzeichnis für den Quellcode.

### 41.5.4. Den Quellcode bauen

Die Welt, also das gesamte Basissystem mit Ausnahme des Kernels, wird zuerst übersetzt, um aktuelle Werkzeuge zum Erstellen des Kernels bereitzustellen. Anschließend wird der Kernel gebaut:

```
# cd /usr/src
# make buildworld
# make buildkernel
```

Das Ergebnis wird in /usr/obj abgelegt.

Dies sind die grundlegenden Schritte. Weitere Optionen zur Kontrolle des Bauprozesses sind nachfolgend beschrieben.

#### Umgebung für den Bauprozess säubern

Einige Versionen von FreeBSD hinterlassen bereits übersetzten Code im temporären Objektverzeichnis /usr/obj. Dies kann nachfolgende Bauprozesse beschleunigen, da Code, der nicht verändert wurde, nicht neu übersetzt werden muss. Um eine saubere Umgebung für den Bauprozess zu schaffen, benutzen Sie `cleanworld` bevor Sie mit dem Bau beginnen.

```
# make cleanworld
```

#### Anzahl der Prozesse einstellen

Eine höhere Anzahl an Prozessen kann die Geschwindigkeit auf Mehrprozessor-Systemen verbessern. Die Anzahl der Kerne lässt sich mit `sysctl hw.cpu` bestimmen. Prozessoren variieren ebenso, wie die verschiedenen Build-Systeme von FreeBSD. Sie müssen daher mehrere Versuche starten um zu sehen, wie die Anzahl der Prozesse die Geschwindigkeit beeinflusst. Als Ausgangspunkt können Sie die halbe bis doppelte Anzahl der Kerne als Wert probieren. Die Anzahl der Prozesse wird mit `-j` angegeben.

*Beispiel 43. Die Anzahl der Prozesse erhöhen*

Das Basissystem und den Kernel mit vier Prozessen bauen:

```
# make -j4 buildworld buildkernel
```

#### Nur den Kernel erstellen

Wenn sich der Quellcode verändert hat, muss ein `buildworld` ausgeführt werden. Danach kann der Kernel mit `buildkernel` übersetzt werden. Um lediglich den Kernel zu übersetzen:

```
# cd /usr/src
# make buildkernel
```

#### Einen angepassten Kernel erstellen

Der FreeBSD Standard-Kernel basiert auf einer *Konfigurationsdatei* namens GENERIC. Der GENERIC -Kernel enthält die gängigsten Gerätetreiber und Optionen. Manchmal ist es aber sinnvoll oder gar notwendig, einen angepassten Kernel zu erstellen, um Gerätetreiber oder Optionen hinzuzufügen oder zu entfernen, um bestimmte Anforderungen zu erfüllen.

Zum Beispiel könnte jemand, der einen kleinen eingebetteten Rechner mit eingeschränktem RAM entwickelt, nicht benötigte Gerätetreiber oder Optionen entfernen, um den Kernel etwas kleiner zu machen.

Die Kernelkonfigurationsdateien befinden sich in /usr/src/sys/arch/conf/, wobei *arch* die Ausgabe von `uname -m` ist. Auf den meisten Rechnern ist dies `amd64`, demnach befinden sich die Konfigurationsdateien in /usr/src/sys/amd64/conf/.



/usr/src kann aus Versehen gelöscht oder neu erstellt werden. Daher ist es vorzuziehen, angepasste Kernelkonfigurationsdateien in einen separaten Verzeichnis, wie bspw. /root zu speichern und diese in das conf-Verzeichnis zu verlinken. Wenn dieses Verzeichnis gelöscht oder überschrieben wird, kann die Kernelkonfigurationsdatei einfach neu verknüpft werden.

Eine benutzerdefinierte Konfigurationsdatei kann durch Kopieren der GENERIC -Konfigurationsdatei erstellt werden. In diesem Beispiel ist der neue Kernel für einen Speicherserver, heißt also STORAGESERVER:

```
# cp /usr/src/sys/amd64/conf/GENERIC /root/STORAGESERVER
# cd /usr/src/sys/amd64/conf
# ln -s /root/STORAGESERVER .
```

Jetzt kann /root/STORAGESERVER bearbeitet werden. Die Manualpage config(5) zeigt, wie Treiber und Optionen hinzugefügt oder entfernt werden.

Der angepasste Kernel wird mit der Variablen `KERNCONF`, die auf die Kernelkonfigurationsdatei verweist, übersetzt:

```
# make buildkernel KERNCONF=STORAGESERVER
```

### 41.5.5. Installation des Codes

Nachdem die Schritte `buildworld` und `buildkernel` abgeschlossen sind, wird der neue Kernel und die Welt installiert:

```
# cd /usr/src
# make installkernel
# shutdown -r now
# cd /usr/src
# make installworld
# shutdown -r now
```

Wenn ein angepasster Kernel erstellt wurde, muss zusätzlich die Variable `KERNCONF` gesetzt werden:

```
# cd /usr/src
# make installkernel KERNCONF=STORAGESERVER
# shutdown -r now
# cd /usr/src
# make installworld
# shutdown -r now
```

### 41.5.6. Die Aktualisierung abschließen

Ein paar abschließende Aufgaben beenden die Aktualisierung. Alle Konfigurationsdateien werden mit den neuen Versionen zusammengeführt, veraltete Bibliotheken werden entfernt, dann wird das System neu gestartet.

#### Konfigurationsdateien mit mergemaster(8) zusammenführen

mergemaster(8) bietet einen einfachen Weg, um die Konfigurationsdateien des Systems mit den neuen Versionen dieser Dateien zusammenzuführen.

Mit der Option `-Ui` aktualisiert mergemaster(8) automatisch Dateien, welche nicht vom Benutzer verändert wurden und installiert neue Dateien, die noch nicht vorhanden sind:

```
# mergemaster -Ui
```

Wenn eine Datei manuell zusammengeführt werden muss, erlaubt eine interaktive Anzeige, zu wählen, welche Teile der Dateien beibehalten werden. Die Manualpage mergemaster(8) enthält weitere Informationen.

**Veraltete Dateien und Bibliotheken entfernen**

Nach einer Aktualisierung können sich immer noch veraltete Dateien und Verzeichnisse im System befinden. Diese lassen sich mit folgendem Kommando auflisten:

```
# make check-old
```

und löschen:

```
# make delete-old
```

Einige veraltete Bibliotheken können ebenfalls noch vorhanden sein. Diese werden mit folgenden Kommando aufgelistet:

```
# make check-old-libs
```

und wie folgt gelöscht:

```
# make delete-old-libs
```

Programme, die diese alten Bibliotheken noch verwenden, werden nicht mehr funktionieren, wenn die Bibliothek gelöscht wurde. Diese Programme müssen nach dem Löschen der alten Bibliotheken neu gebaut oder ersetzt werden.



Wenn Sie sich sicher sind, dass alle Dateien und Verzeichnisse gelöscht werden können, dann setzen Sie `BATCH_DELETE_OLD_FILES`, um nicht jede einzelne Datei mit `y` und `Enter` bestätigen zu müssen. Zum Beispiel:

```
# make BATCH_DELETE_OLD_FILES=yes delete-old-libs
```

**Neustart des Systems**

Zum Abschluss der Aktualisierung muss das System neu gestartet werden, damit alle Änderungen wirksam werden:

```
# shutdown -r now
```

## 41.6. Installation mehrerer Maschinen

Wenn Sie mehrere Maschinen auf dem gleichen Stand halten wollen, ist es eine Verschwendung von Ressourcen, die Quellen auf jeder Maschine vorzuhalten und zu übersetzen. Die Lösung dazu ist, eine Maschine den Großteil der Arbeit durchführen zu lassen und den anderen Maschinen das

Ergebnis mit NFS zur Verfügung zu stellen. Dieser Abschnitt zeigt eine Methode dies zu tun. Weitere Informationen zu NFS finden Sie in Network File System (NFS).

Stellen Sie zuerst eine Liste der Maschinen zusammen, die auf demselben Stand sein sollen. Wir nennen diese Maschinen die *Baugruppe*. Jede dieser Maschinen kann mit einem eigenen Kernel laufen, doch sind die Programme des Userlands auf allen Maschinen gleich. Wählen Sie aus der Baugruppe eine Maschine aus, auf der der Bau durchgeführt wird, den *Bau-Master*. Dies sollte eine Maschine sein, die über die nötigen CPU-Ressourcen für `make buildworld` und `make installworld` verfügt.

Sie brauchen auch eine *Testmaschine*, auf der Sie die Updates testen, bevor Sie sie in Produktion installieren. Dies *muss* eine Maschine sein, die über einen längeren Zeitraum nicht zur Verfügung stehen kann.

Alle Maschinen der Baugruppe müssen /usr/obj und /usr/src über NFS vom Bau-Master an gleichem Ort einhängen. Wenn Sie mehrere Baugruppen haben, sollte sich /usr/src auf einem Bau-Master befinden und über NFS für den Rest der Maschinen zur Verfügung gestellt werden.

Stellen Sie sicher, dass /etc/make.conf und /etc/src.conf auf allen Maschinen einer Baugruppe mit der Datei des Bau-Masters übereinstimmt. Der Bau-Master muss jeden Teil des Systems bauen, den irgendeine Maschine der Baugruppe benötigt. Auf dem Bau-Master müssen in /etc/make.conf alle zu bauenden Kernel mit der Variablen `KERNCONF` bekannt gegeben werden. Geben Sie dabei den Kernel des Bau-Masters zuerst an. Für jeden zu bauenden Kernel muss auf dem Bau-Master die entsprechende Konfigurationsdatei unter /usr/src/sys/arch/conf abgelegt werden.

Bauen Sie auf dem Bau-Master, wie in FreeBSD aus den Quellen aktualisieren beschrieben, den Kernel und die Welt, installieren Sie aber nichts. Wechseln Sie auf die Testmaschine und installieren Sie den gerade gebauten Kernel. Hängen Sie auf der Testmaschine /usr/src und /usr/obj über NFS ein. Geben Sie dann `shutdown now` ein, um in den Single-User-Modus zu gelangen, von wo aus Sie den neuen Kernel und das System installieren. Lassen Sie anschließend `mergemaster` laufen. Wenn Sie fertig sind, booten Sie die Maschine wieder in den Mehrbenutzermodus.

Nachdem Sie sichergestellt haben, dass die Testmaschine einwandfrei funktioniert, wiederholen Sie diese Prozedur für jede Maschine in der Baugruppe.

Dasselbe Verfahren können Sie auch für die Ports-Sammlung anwenden. Zuerst müssen alle Maschinen einer Baugruppe /usr/ports über NFS zur Verfügung gestellt bekommen. Setzen Sie ein Verzeichnis für die Quellen auf, das sich alle Maschinen teilen. Dieses Verzeichnis können Sie in /etc/make.conf mit der Variablen `DISTDIR` angeben. Das Verzeichnis sollte für den Benutzer beschreibbar sein, auf den der Benutzer `root` vom NFS Subsystem abgebildet wird. Jede Maschine sollte noch `WRKDIRPREFIX` auf ein lokales Bauverzeichnis setzen. Wenn Sie vorhaben, Pakete zu bauen und zu verteilen, sollten Sie `PACKAGES` auf ein Verzeichnis mit den gleichen Eigenschaften wie `DISTDIR` setzen.

# Kapitel 42. DTrace

## 42.1. Überblick

DTrace, auch bekannt als Dynamic Tracing, wurde von Sun™ als ein Werkzeug zur Analyse von Performance-Problemen in Produktiv- und Entwicklungssystemen entwickelt. Zusätzlich zur Diagnose von Performance-Problemen kann DTrace auch verwendet werden, um bei der Untersuchung und Behebung von unerwartetem Verhalten im FreeBSD-Kernel und den Anwenderprogrammen zu helfen.

DTrace ist ein bemerkenswertes Werkzeug zur Profilerstellung, mit einer beeindruckenden Palette von Eigenschaften zur Diagnose von Systemereignissen. Es kann auch dazu verwendet werden, bestehende Skripte ablaufen zu lassen, um einen Nutzen aus deren Möglichkeiten zu ziehen. Nutzer können mittels der Programmiersprache D von DTrace ihre eigenen Hilfsmittel schreiben, was es ermöglicht, die eigenen Profile nach Ihren Bedürfnissen anzupassen.

Die DTrace-Implementierung in FreeBSD bietet experimentelle Unterstützung für DTrace im Userland. Userland DTrace erlaubt es Anwendern, function boundary tracing für Anwendungsprogramme über den `pid`-Provider hinweg vorzunehmen und um statische Sonden in Anwendungsprogramme für die spätere Aufzeichnung einzufügen. Manche Ports, wie beispielsweise databases/postgresql12-server und lang/php74 besitzen eine DTrace-Option, um statische Sonden zu aktivieren.

Eine offizielle Anleitung für DTrace wird vom Illumos Projekt im DTrace Guide bereitgestellt.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie Folgendes wissen:

- Was DTrace ist und welche Funktionen es zur Verfügung stellt.
- Unterschiede zwischen der Solaris™ DTrace Implementierung und derjenigen, die FreeBSD bereitstellt.
- Wie man DTrace auf FreeBSD aktiviert und verwendet.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- UNIX® und FreeBSD Grundlagen verstehen (Grundlagen des FreeBSD Betriebssystems).
- Vertraut sein mit Sicherheitsaspekten und wie diese FreeBSD betreffen (Sicherheit).



Diese Funktion ist als experimentell anzusehen. Manche Einstellungen enthalten möglicherweise nicht alle Funktionalitäten, andere Teile könnten gar nicht laufen. Mit der Zeit, wenn diese Funktion als für den Produktivbetrieb geeignet erscheint, wird auch diese Dokumentation geändert, um diesem Umstand gerecht zu werden.

## 42.2. Unterschiede in der Implementierung

Obwohl DTrace in FreeBSD sehr ähnlich zu dem in Solaris™ ist, existieren doch Unterschiede. Der Hauptunterschied besteht darin, dass in FreeBSD DTrace als eine Menge von Kernelmodulen

implementiert ist und DTrace nicht verwendet werden kann, bis diese Module geladen wurden. Um alle nötigen Module zu laden, geben Sie ein:

```
# kldload dtraceall
```

Beginnend mit FreeBSD 10.0-RELEASE werden die Module automatisch geladen, sobald `dtrace` aufgerufen wird.

FreeBSD verwendet die Kerneloption `DDB_CTF`, um die Unterstützung im Kernel für das Laden von CTF-Daten aus Kernelmodulen und dem Kernel selbst zu ermöglichen. CTF ist das Compact C Type Format von Solaris™, welches eine reduzierte Form von Debug-Informationen kapselt, ähnlich zu DWARF und den antiken Stabs. Diese CTF-Daten werden dem Binärcode von den `ctfconvert` und `ctfmerge` Befehlen den Werkzeugen zum Bauen des Systems hinzugefügt. Das `ctfconvert` -Dienstprogramm parst die vom Compiler erstellten DWARFELF Debug-Abschnitte und `ctfmerge` vereint CTFELF-Abschnitte aus Objekten, entweder in ausführbare Dateien oder Shared-Libraries.

Einige Provider in FreeBSD unterscheiden sich von der Solaris™-Implementierung. Am deutlichsten wird das beim `dtmalloc`-Provider, welcher das Aufzeichnen von `malloc()` nach Typen im FreeBSD-Kernel ermöglicht. Manche der Provider in Solaris™ wie `cpc` und `mib` sind in FreeBSD nicht vorhanden. Diese können in zukünftigen FreeBSD-Versionen auftauchen. Weiterhin sind manche der Provider in beiden Betriebssystemen nicht zueinander kompatibel, in dem Sinne daß deren Sonden unterschiedliche Argumenttypen aufweisen. Dadurch können D-Skripte, die unter Solaris™ geschrieben wurden, evtl. unter FreeBSD funktionieren oder auch nicht, umgekehrt ist das genauso.

In FreeBSD darf DTrace wegen unterschiedlicher Sicherheitskonzepte nur von `root` verwendet werden. Solaris™ besitzt ein paar Audit-Funktionen auf den unteren Ebenen, die noch nicht in FreeBSD implementiert sind. Deshalb kann nur `root` auf /dev/dtrace/dtrace zugreifen.

Zum Schluss muss noch erwähnt werden, dass die DTrace-Software unter die CDDL Lizenz fällt. Die `Common Development and Distribution License` wird von FreeBSD mitgeliefert, sehen Sie sich dazu /usr/src/cddl/contrib/opensolaris/OPENSOLARIS.LICENSE an, oder lesen Sie die Online-Version unter http://opensource.org/licenses/CDDL-1.0. Während der FreeBSD-Kernel mit den DTrace-Optionen immer noch BSD-lizenziert ist, tritt die CDDL in Kraft, wenn Module in Binärform vertrieben werden oder die Binärdateien geladen werden.

## 42.3. Die DTrace Unterstützung aktivieren

In FreeBSD 9.2 und 10.0 ist die Unterstützung von DTrace im GENERIC-Kernel bereits eingebaut. Nutzer von früheren Versionen sollten die folgenden Zeilen in eine eigene Kernelkonfigurationsdatei einfügen und den Kernel mittels der Anleitung in Konfiguration des FreeBSD-Kernels neu übersetzen:

```
options         KDTRACE_HOOKS
options         DDB_CTF
makeoptions         DEBUG=-g
makeoptions         WITH_CTF=1
```

Besitzer der AMD64-Architektur werden wahrscheinlich noch die folgende Zeile zur Kernelkonfigurationsdatei hinzufügen:

```
options         KDTRACE_FRAME
```

Diese Option liefert die Unterstützung für die FBT-Eigenschaft. DTrace wird auch ohne diese Option funktionieren; jedoch wird dann Function Boundary Tracing nur eingeschränkt unterstützt.

Sobald FreeBSD in den neuen Kernel gebootet oder die DTrace-Kernelmodule mittels `kldload dtraceall` geladen wurden, benötigt das System Unterstützung für die Korn-Shell, da DTrace mehrere Dienstprogramme enthält, die in `ksh` implementiert sind. Vergewissern Sie sich, dass das Paket oder der Port shells/ksh93 installiert ist. Es ist auch möglich, diese Werkzeuge unter shells/pdksh oder shells/mksh laufen zu lassen.

Zum Schluss sollten Sie noch den aktuellen DTrace-Werkzeugsatz beschaffen. Die DTrace-Werkzeugsammlung enthält gebrauchsfertige Skripte, um Systeminformationen zu sammeln. Es gibt Skripte zum Überprüfen von offenen Dateien, Speicher- und CPU-Gebrauch und noch viel mehr. FreeBSD 10 installiert ein paar dieser Skripte in /usr/shared/dtrace. Für andere FreeBSD-Versionen oder um die volle DTrace-Werkzeugsammlung zu installieren, verwenden Sie den sysutils/dtrace-toolkit Port oder das Paket.



Die Skripte in /usr/shared/dtrace wurden speziell für FreeBSD portiert. Nicht alle Skripte in der DTrace-Werkzeugsammlung werden in FreeBSD unverändert funktionieren und manche Skript benötigen einigen Aufwand, damit diese auf FreeBSD funktionieren.

Der DTrace-Werkzeugsatz beinhaltet viele Skripte in der speziellen Sprache von DTrace. Diese Sprache wird die D-Sprache genannt und ist sehr ähnlich zu C++. Eine detaillierte Beschreibung dieser Sprache würde den Rahmen dieses Dokuments sprengen. Im Illumos Dynamic Tracing Guide wird diese Sprache ausführlich beschrieben.

## 42.4. DTrace verwenden

DTrace-Skripte bestehen aus einer Liste von einer oder mehreren *Sonden* oder Instrumentationspunkten, an denen jede Sonde mit einer Aktion verknüpft ist. Jedesmal, wenn die Bedingung für eine Sonde zutrifft, wird die verknüpfte Aktion ausgeführt. Beispielsweise könnte eine Aktion ausgeführt werden, wenn eine Datei geöffnet, ein Prozess gestartet oder eine Codezeile ausgeführt wird. Die Aktion könnte die Protokollierung von Informationen sein oder die Änderung von Kontextvariablen. Das Lesen und Schreiben von Kontextvariablen erlaubt es den Sonden, Informationen auszutauschen und kooperativ die Korrelation bestimmter Ereignisse zu analysieren.

Um alle Sonden anzuzeigen, kann der Administrator nun den folgenden Befehl eingeben:

```
# dtrace -l | more
```

Jede Sonde besitzt eine ID, einen PROVIDER (dtrace oder fbt), ein MODULE und einen FUNCTION NAME. Lesen Sie dtrace(1) für weitere Informationen zu diesem Kommando.

Die Beispiele in diesem Abschnitt geben einen Überblick, wie man zwei dieser voll funktionsfähigen Skripte aus der DTrace-Werkzeugsammlung verwendet: die Skripte hotkernel und procsystime.

Das hotkernel Skript wurde entworfen, um zu identifizieren, welche Funktion die meiste Kernelzeit beansprucht. Es wird es Ausgaben ähnlich der Folgenden produzieren:

```
# cd /usr/local/share/dtrace-toolkit
# ./hotkernel
Sampling... Hit Ctrl-C to end.
```

Verwenden Sie wie angegeben die Tastenkombination `Ctrl` + `C` drücken, um den Prozess zu stoppen. Nach dem Abbruch wird das Skript eine Liste von Kernelfunktionen und Zeitmessungen ausgeben, aufsteigend sortiert nach den Zeiten:

```
kernel`_thread_lock_flags                               2   0.0%
0xc1097063                                              2   0.0%
kernel`sched_userret                                    2   0.0%
kernel`kern_select                                      2   0.0%
kernel`generic_copyin                                   3   0.0%
kernel`_mtx_assert                                      3   0.0%
kernel`vm_fault                                         3   0.0%
kernel`sopoll_generic                                   3   0.0%
kernel`fixup_filename                                   4   0.0%
kernel`_isitmyx                                         4   0.0%
kernel`find_instance                                    4   0.0%
kernel`_mtx_unlock_flags                                5   0.0%
kernel`syscall                                          5   0.0%
kernel`DELAY                                            5   0.0%
0xc108a253                                              6   0.0%
kernel`witness_lock                                     7   0.0%
kernel`read_aux_data_no_wait                            7   0.0%
kernel`Xint0x80_syscall                                 7   0.0%
kernel`witness_checkorder                               7   0.0%
kernel`sse2_pagezero                                    8   0.0%
kernel`strncmp                                          9   0.0%
kernel`spinlock_exit                                   10   0.0%
kernel`_mtx_lock_flags                                 11   0.0%
kernel`witness_unlock                                  15   0.0%
kernel`sched_idletd                                   137   0.3%
0xc10981a5                                          42139  99.3%
```

Dieses Skript funktioniert auch mit Kernelmodulen. Um diese Eigenschaft zu verwenden, starten Sie das Skript mit -m:

```
# ./hotkernel -m
Sampling... Hit Ctrl-C to end.
^C
MODULE                                                  COUNT   PCNT
0xc107882e                                                  1   0.0%
0xc10e6aa4                                                  1   0.0%
0xc1076983                                                  1   0.0%
0xc109708a                                                  1   0.0%
0xc1075a5d                                                  1   0.0%
0xc1077325                                                  1   0.0%
0xc108a245                                                  1   0.0%
0xc107730d                                                  1   0.0%
0xc1097063                                                  2   0.0%
0xc108a253                                                 73   0.0%
kernel                                                    874   0.4%
0xc10981a5                                             213781  99.6%
```

Das procsystime Skript fängt die Systemaufruf-Zeiten für eine gegebene Prozess-ID (PID) oder einen Prozessnamen ab und gibt diese aus. Im folgenden Beispiel wurde eine neue Instanz von /bin/csh erzeugt. Dann wurde procsystime ausgeführt und verbleibt so, während ein paar Befehle in die andere Instanz von csh eingegeben werden. Dies sind die Ergebnisse dieses Versuchs:

```
# ./procsystime -n csh
Tracing... Hit Ctrl-C to end...
^C

Elapsed Times for processes csh,

         SYSCALL          TIME (ns)
          getpid               6131
       sigreturn               8121
           close              19127
           fcntl              19959
             dup              26955
         setpgid              28070
            stat              31899
       setitimer              40938
           wait4              62717
       sigaction              67372
     sigprocmask             119091
    gettimeofday             183710
           write             263242
          execve             492547
           ioctl             770073
           vfork            3258923
      sigsuspend            6985124
            read         3988049784
```

Wie aus der Ausgabe ersichtlich ist, verbraucht der `read()`-Systemaufruf die meiste Zeit in Nanosekunden, während der Systemaufruf `getpid()` hingegen am schnellsten läuft.

# Kapitel 43. USB Gerätemodus

## 43.1. Übersicht

Dieses Kapitel behandelt die Verwendung des USB Gerätemodus und USB On-The-Go (USB OTG) unter FreeBSD. Dazu gehören virtuelle serielle Konsolen, virtuelle Netzwerkkarten und virtuelle USB-Laufwerke.

Wenn die eingesetzte Hardware den USB-Gerätemodus oder USB OTG unterstützt, kann FreeBSDs USB-Stack im Gerätemodus ausgeführt werden. Solche Hardware wird häufig in eingebetteten Systeme verbaut. Der Gerätemodus ermöglicht es dem Rechner verschiedene Arten von USB-Geräteklassen darzustellen, einschließlich serieller Schnittstellen, Netzwerkkarten und Massenspeicher oder Kombinationen davon. Ein USB-Host, beispielsweise ein Notebook oder ein Desktop-Rechner, kann wie auf ein physisches USB-Gerät darauf zugreifen.

Es gibt zwei grundlegende Möglichkeiten, wie die Hardware den Gerätemodus bereitstellen kann: mit einem separaten "Client Modus", der nur den Gerätemodus unterstützt, und mit einem USB-OTG-Port, der sowohl den Geräte- als auch den Hostmodus bereitstellen kann. Bei USB-OTG-Ports wechselt der USB-Stack automatisch zwischen host- und geräteseitig, je nachdem, was an dem Port angeschlossen ist. Wenn Sie ein USB-Gerät wie einen Speicherstick an den Port anschließen, wechselt FreeBSD in den Hostmodus. Wenn Sie einen USB-Host wie einen Computer anschließen, wechselt FreeBSD in den Gerätemodus. "Client Ports" arbeiten immer im Gerätemodus.

Was FreeBSD dem USB-Host präsentiert, hängt von der sysctl-Variable `hw.usb.template` ab. Einige Vorlagen bieten ein einzelnes Gerät, beispielsweise ein serielles Terminal, andere bieten mehrere, die alle gleichzeitig verwendet werden können. Ein Beispiel ist die Vorlage 10, die ein Massenspeichergerät, eine serielle Konsole und eine Netzwerkkarte bereitstellt. usb_template(4) enthält eine Liste der verfügbaren Werte.

Beachten Sie, dass in einigen Fällen, abhängig von der Hardware und dem Betriebssystem des Hosts, die Änderung an der Konfiguration nur dann bemerkt werden kann, wenn der Host entweder physisch getrennt und wieder verbunden oder gezwungen wird, den USB-Bus auf eine systemspezifische Weise neu zu scannen. Wenn FreeBSD auf dem Host läuft, kann usbconfig(8) `reset` verwendet werden. Dies muss auch nach dem Laden von usb_template.ko geschehen, wenn der USB-Host bereits an der USB OTG-Buchse angeschlossen war.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie wissen:

- wie man den USB Gerätemodus unter FreeBSD einrichtet.
- wie man die virtuelle serielle Schnittstelle unter FreeBSD konfiguriert.
- wie man sich mit der virtuellen seriellen Schnittstelle von verschiedenen Betriebssystemen aus verbindet.

## 43.2. Virtuelle serielle USB-Ports

### 43.2.1. Konfiguration des USB-Gerätemodus für serielle Ports

Die virtuellen seriellen Ports werden durch die Vorlagen 3, 8 und 10 unterstützt. Beachten Sie, dass Vorlage 3 mit Microsoft Windows 10 ohne spezielle Treiber und INF-Dateien funktioinert. Andere Host-Betriebssysteme arbeiten mit allen drei Vorlagen. Die beiden Kernelmodule usb_template(4) und umodem(4) müssen geladen werden.

Um die seriellen Ports im USB-Gerätemodus zu aktivieren, fügen Sie folgenden Zeilen in /etc/ttys hinzu:

```
ttyU0   "/usr/libexec/getty 3wire"  vt100   onifconsole secure
ttyU1   "/usr/libexec/getty 3wire"    vt100   onifconsole secure
```

Danach fügen Sie folgende Zeilen in /etc/devd.conf hinzu:

```
notify 100 {
    match "system"      "DEVFS";
    match "subsystem"   "CDEV";
    match "type"        "CREATE";
    match "cdev"        "ttyU[0-9]+";
    action "/sbin/init q";
};
```

Laden Sie die Konfiguration neu, falls devd(8) bereits läuft:

```
# service devd restart
```

Stellen Sie sicher, dass die notwendigen Module geladen sind und die richtige Vorlage beim Booten gesetzt ist. Fügen Sie dazu folgende Zeilen in /boot/loader.conf ein:

```
umodem_load="YES"
hw.usb.template=3
```

Um das Modul zu laden und die Vorlage ohne Neustart zu aktivieren, verwenden Sie:

```
# kldload umodem
# sysctl hw.usb.template=3
```

### 43.2.2. FreeBSD mit der seriellen Schnittstelle im USB-Gerätemodus verbinden

Um eine Verbindung zu einer Karte herzustellen, die so konfiguriert ist, dass sie serielle Ports im USB-Gerätemodus bereitstellt, schließen Sie den USB-Host, beispielsweise einen Laptop, an den USB OTG- oder USB-Client-Port der Karte an. Verwenden Sie `pstat -t` auf dem Host, um die

Terminalzeilen aufzulisten. Am Ende der Liste sollten Sie einen seriellen USB-Anschluss sehen, zum Beispiel "ttyU0". Um die Verbindung zu öffnen, benutzen Sie:

```
# cu -l /dev/ttyU0
```

Nach mehrmaligem Drücken der `Enter`-Taste erscheint ein Anmeldeprompt.

### 43.2.3. macOS mit der seriellen Schnittstelle im USB-Gerätemodus verbinden

Um eine Verbindung zu einer Karte herzustellen, die so konfiguriert ist, dass sie serielle Ports im USB-Gerätemodus bereitstellt, schließen Sie den USB-Host, beispielsweise einen Laptop, an den USB OTG- oder USB-Client-Port der Karte an. Um die Verbindung zu öffnen, benutzen Sie:

```
# cu -l /dev/cu.usbmodemFreeBSD1
```

### 43.2.4. Linux mit der seriellen Schnittstelle im USB-Gerätemodus verbinden

Um eine Verbindung zu einer Karte herzustellen, die so konfiguriert ist, dass sie serielle Ports im USB-Gerätemodus bereitstellt, schließen Sie den USB-Host, beispielsweise einen Laptop, an den USB OTG- oder USB-Client-Port der Karte an. Um die Verbindung zu öffnen, benutzen Sie:

```
# minicom -D /dev/ttyACM0
```

### 43.2.5. Windows 10 mit der seriellen Schnittstelle im USB-Gerätemodus verbinden

Um eine Verbindung zu einer Karte herzustellen, die so konfiguriert ist, dass sie serielle Ports im USB-Gerätemodus bereitstellt, schließen Sie den USB-Host, beispielsweise einen Laptop, an den USB OTG- oder USB-Client-Port der Karte an. Um die Verbindung zu öffnen, benötigen Sie ein Terminalprogramm mit Unterstützung für serielle Schnittstellen, zum Beispiel PuTTY. Um den von Windows® verwendeten COM-Port zu ermitteln, starten Sie den Geräte-Manager und erweitern Sie "Ports (COM & LPT)". Dort sehen Sie einen Namen wie "USB Serial Sevice (COM4)". Starten Sie das Terminalprogramm Ihrer Wahl, zum Beispiel PuTTY. Im Dialog von PuTTY setzen Sie den "Connection type" auf "Serial" und notieren im Feld "Serial line" den ermittelten COM-Namen. Danach klicken Sie auf "Open".

## 43.3. Netzwerkkarten im USB-Gerätemodus

Virtuelle Netzwerkkarten werden durch die Vorlagen 1, 8 und 10 unterstützt. Beachten Sie, dass keine dieser Vorlagen mit Windows® funktioniert. Andere Host-Betriebssysteme arbeiten mit allen drei Vorlagen. Die Kernelmodule usb_template(4) und if_cdce(4) müssen geladen sein.

Stellen Sie sicher, dass die notwendigen Module geladen sind und die richtige Vorlage beim Booten gesetzt ist. Fügen Sie dazu folgende Zeilen in /boot/loader.conf ein:

```
if_cdce_load="YES"
hw.usb.template=1
```

Um das Modul zu laden und die Vorlage ohne Neustart zu aktivieren, verwenden Sie:

```
# kldload if_cdce
# sysctl hw.usb.template=1
```

## 43.4. Virtuelle USB-Speichergeräte



cfumass(4) ist ein USB-Gerätetreiber, der seit FreeBSD 12.0 verfügbar ist.

Virtuelle Speichergeräte werden durch die Vorlagen 0 und 10 unterstützt. Die Kernelmodule usb_template(4) und cfumass(4) müssen geladen sein. cfumass(4) ist die Schnittstelle zum CTL-Subsystem, das auch für iSCSI- oder Fibre-Channel-Targets benutzt wird. Auf dem Host können Initiatioren von USB-Massenspeichern nur auf eine einzige LUN, LUN 0 zugreifen.

### 43.4.1. Konfiguration von USB-Massenspeicher Targets mit dem cfumass-Startskript

Der einfachste Weg, ein schreibgeschütztes USBSpeicherziel einzurichten, ist die Verwendung des cfumass rc-Skripts. Kopieren Sie einfach die Dateien, die dem USB-Host präsentiert werden sollen, in das Verzeichnis /var/cfumass und fügen Sie diese Zeile in /etc/rc.conf ein:

```
cfumass_enable="YES"
```

Um das Ziel ohne Neustart zu konfigurieren, führen Sie diesen Befehl aus:

```
# service cfumass start
```

Im Gegensatz zur seriellen und Netzwerkfunktionalität sollte die Vorlage in /boot/loader.conf nicht auf 0 oder 10 gesetzt werden, da die LUN vor dem Setzen der Vorlage konfiguriert werden muss. Das cfumass rc-Skript setzt beim Start automatisch die richtige Vorlage.

### 43.4.2. USB-Massenspeicher mit anderen Werkzeugen konfigurieren

Der Rest dieses Kapitels enthält eine detaillierte Beschreibung der Konfiguration ohne die Verwendung des cfumass rc-Skripts. Dies ist beispielsweise notwendig, wenn man eine beschreibbare LUN zur Verfügung stellen will.

Im Gegensatz zu iSCSI ist es bei USB-Massenspeichern nicht zwingend erforderlich, dass der ctld(8) Daemon läuft. Es gibt zwei Möglichkeiten, das Target zu konfigurieren: ctladm(8) oder ctld(8). Beide erfordern, dass das Kernelmodul cfumass.ko geladen ist. Das Modul kann manuell geladen werden:

```
# kldload cfumass
```

Wenn cfumass nicht im Kernel integriert ist, kann /boot/loader.conf angepasst werden, damit das Modul beim Booten geladen wird:

```
cfumass_load="YES"
```

Eine LUN kann ohne den ctld(8) Daemon erstellt werden:

```
# ctladm create -b block -o file=/data/target0
```

Dies stellt den Inhalt des Abbilds von /data/target0 als LUN auf dem USB-Host dar. Die Datei muss vor der Ausführung des Befehls vorhanden sein. Um die LUN beim Systemstart zu konfigurieren, muss das Kommando in /etc/rc.local eingetragen werden.

ctld(8) kann auch benutzt werden, um LUNs zu verwalten. Dazu erstellen Sie eine /etc/ctl.conf und fügen eine Zeile in /etc/rc.conf hinzu, um sicherzustellen, dass ctld(8) beim Booten automatisch gestartet wird. Danach kann der Daemon gestartet werden.

Es folgt ein Beispiel einer einfachen Konfiguration für /etc/ctl.conf. Eine ausführliche Beschreibung der Optionen finden Sie in ctl.conf(5).

```
target naa.50015178f369f092 {
    lun 0 {
        path /data/target0
        size 4G
    }
}
```

Dieses Beispiel erstellt ein einzelnes Target mit einer einzigen LUN. `naa.50015178f369f092` ist eine Gerätekennung, die aus 32 zufälligen Hexadezimalziffern besteht. `path` definiert den absoluten Pfad zu einer Datei oder eines zvol, welches die LUN als Speicher nutzen kann. Diese Datei muss vor dem Start von ctld(8) existieren. Die zweite Zeile ist optional und definiert die Größe der LUN.

Damit der ctld(8) Daemon beim Booten gestartet wird, muss diese Zeile in /etc/rc.conf hinzugefügt werden:

```
ctld_enable="YES"
```

Sie können ctld(8) mit diesem Befehl direkt starten:

```
# service ctld start
```

Der ctld(8) Daemon liest beim Start /etc/ctl.conf. Wenn diese Datei nach dem Start des Daemons bearbeitet wird, müssen die Änderungen neu geladen werden, damit sie sofort wirksam werden:

```
# service ctld reload
```

params: path: "/books/handbook/partiv/" :leveloffset: +1

# Teil IV: Serielle Datenübertragung

# Kapitel 44. Übersicht

UNIX® Systeme unterstützten schon immer die serielle Datenübertragung. Tatsächlich wurden Ein- und Ausgaben auf den ersten UNIX® Maschinen über serielle Leitungen durchgeführt. Seit der Zeit, in der ein durchschnittlicher Terminal aus einem seriellen Drucker mit 10 Zeichen/Sekunde und einer Tastatur bestand, hat sich viel verändert. Dieses Kapitel behandelt einige Möglichkeiten, serielle Datenübertragung unter FreeBSD zu verwenden.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie Folgendes wissen:

- Wie Sie Terminals an ein FreeBSD-System anschließen.
- Wie Sie sich mit einem Modem auf entfernte Rechner einwählen.
- Wie Sie entfernten Benutzern erlauben, sich mit einem Modem in ein FreeBSD-System einzuwählen.
- Wie Sie ein FreeBSD-System über eine serielle Konsole booten.

Bevor Sie dieses Kapitel lesen, sollten Sie

- einen angepassten Kernel konfigurieren und installieren können.
- Berechtigungen und Prozesse unter FreeBSD verstehen.
- Zugriff auf die Handbücher der seriellen Komponenten haben, die mit FreeBSD verwendet werden sollen.

# Kapitel 45. Begriffe und Hardware

Die folgenden Begriffe werden oft verwendet, wenn es um serielle Kommunikation geht:

**bps**

Bits pro Sekunde (bps) ist die Einheit für die Übertragungsgeschwindigkeit.

**DEE (DTE)**

Eine Datenendeinrichtung (Data Terminal Equipment) ist einer der beiden Endpunkte bei der seriellen Kommunikation. Zum Beispiel ein Computer.

**DÜE (DCE)**

Datenübertragungseinrichtung (Data Communications Equipment) ist der andere Endpunkt bei der seriellen Kommunikation. Typischerweise ein Modem.

**RS-232**

Der originale Standard, der serielle Datenübertragung definiert. Er wird heutzutage als TIA-232 bezeichnet.

In diesem Abschnitt wird der Begriff "Baud" nicht für Übertragungsgeschwindigkeiten gebraucht. Baud bezeichnet elektrische Zustandswechsel pro Zeiteinheit, die Taktfrequenz, während "bps" der *richtige* Begriff für die Übertragungsgeschwindigkeit ist.

Um ein Modem oder einen Terminal an ein FreeBSD-System anzuschließen, muss der Computer über eine serielle Schnittstelle verfügen. Zusätzlich wird das passende Kabel benötigt, um das Gerät mit der Schnittstelle zu verbinden. Benutzer, die mit seriellen Geräten und den nötigen Kabeln schon vertraut sind, können diesen Abschnitt überspringen.

## 45.1. Kabel und Schnittstellen

Es gibt verschiedene serielle Kabel. Die zwei häufigsten sind Nullmodemkabel und Standard-RS-232-Kabel. Die Dokumentation der Hardware sollte beschreiben, welcher Kabeltyp benötigt wird.

Ein Nullmodemkabel verbindet einige Signale, wie die Betriebserde, eins zu eins, andere Signale werden getauscht: Die Sende- und Empfangsleitungen werden zum Beispiel gekreuzt.

Nullmodemkabel für die Anbindung eines Terminals können auch selbst hergestellt werden. Die folgende Tabelle enthält die Signalnamen von RS-232C sowie die Pinbelegung für einen Stecker vom Typ DB-25. Obwohl der Standard eine direkte Verbindung von Pin 1 zu Pin 1 (*Protective Ground*) vorschreibt, ist diese in vielen Fällen nicht vorhanden. Einige Terminals benötigen nur die Pins 2, 3 und 7 für eine korrekte Funktion, während andere eine unterschiedliche Konfiguration als die in den folgenden Beispielen gezeigte benötigen.

*Tabelle 19. Nullmodemkabel vom Typ DB-25-zu-DB-25*

| Signal | Pin # | | Pin # | Signal |
|---|---|---|---|---|
| SG | 7 | verbunden mit | 7 | SG |
| TD | 2 | verbunden mit | 3 | RD |

| Signal | Pin # | | Pin # | Signal |
|---|---|---|---|---|
| RD | 3 | verbunden mit | 2 | TD |
| RTS | 4 | verbunden mit | 5 | CTS |
| CTS | 5 | verbunden mit | 4 | RTS |
| DTR | 20 | verbunden mit | 6 | DSR |
| DTR | 20 | verbunden mit | 8 | DCD |
| DSR | 6 | verbunden mit | 20 | DTR |
| DCD | 8 | verbunden mit | 20 | DTR |

Die folgenden zwei Schemata werden heutzutage ebenfalls häufig eingesetzt:

*Tabelle 20. Nullmodemkabel vom Typ DB-9-zu-DB-9*

| Signal | Pin # | | Pin # | Signal |
|---|---|---|---|---|
| RD | 2 | verbunden mit | 3 | TD |
| TD | 3 | verbunden mit | 2 | RD |
| DTR | 4 | verbunden mit | 6 | DSR |
| DTR | 4 | verbunden mit | 1 | DCD |
| SG | 5 | verbunden mit | 5 | SG |
| DSR | 6 | verbunden mit | 4 | DTR |
| DCD | 1 | verbunden mit | 4 | DTR |
| RTS | 7 | verbunden mit | 8 | CTS |
| CTS | 8 | verbunden mit | 7 | RTS |

*Tabelle 21. Nullmodemkabel vom Typ DB-9-zu-DB-25*

| Signal | Pin # | | Pin # | Signal |
|---|---|---|---|---|
| RD | 2 | verbunden mit | 2 | TD |
| TD | 3 | verbunden mit | 3 | RD |
| DTR | 4 | verbunden mit | 6 | DSR |
| DTR | 4 | verbunden mit | 8 | DCD |
| SG | 5 | verbunden mit | 7 | SG |
| DSR | 6 | verbunden mit | 20 | DTR |
| DCD | 1 | verbunden mit | 20 | DTR |
| RTS | 7 | verbunden mit | 5 | CTS |
| CTS | 8 | verbunden mit | 4 | RTS |



Wird ein Pin eines Kabels mit zwei Pins des anderen Kabels verbunden, werden dazu in der Regel zuerst die beiden Pins mit einem kurzem Draht verbunden.

> Danach wird dieser Draht mit dem Pin des anderen Endes verbunden.

Die eben besprochenen Schemata scheinen die beliebtesten zu sein. Weitere Varianten verbinden SG mit SG, TD mit RD, RTS und CTS mit DCD, DTR mit DSR, und umgekehrt.

Ein Standard-RS-232C-Kabel verbindet alle Signale direkt. Das Signal "Transmitted Data" wird mit dem Signal "Transmitted Data" der Gegenstelle verbunden. Dieses Kabel wird benötigt, um ein Modem mit einem FreeBSD-System zu verbinden. Manche Terminals benötigen dieses Kabel ebenfalls.

Über serielle Schnittstellen werden Daten zwischen dem FreeBSD-System und dem Terminal übertragen. Dieser Abschnitt beschreibt die verschiedenen Schnittstellen und wie sie unter FreeBSD angesprochen werden.

Da es verschiedene Schnittstellen gibt, sollte vor dem Kauf oder Selbstbau eines Kabels sichergestellt werden, dass dieses zu den Schnittstellen des Terminals und des FreeBSD-Systems passt.

Die meisten Terminals besitzen DB-25-Stecker. Personal Computer haben DB-25- oder DB-9-Stecker. Eine serielle Multiportkarte hat vielleicht RJ-12- oder RJ-45-Anschlüsse.

Die Dokumentation der Geräte sollte Aufschluss über den Typ der benötigten Anschlüsse geben. Oft hilft es, wenn Sie sich den Anschluss einfach ansehen.

Unter FreeBSD wird jede serielle Schnittstelle (Port) über einen Eintrag in /dev angesprochen. Es gibt dort zwei verschiedene Einträge:

- Schnittstellen für eingehende Verbindungen werden /dev/ttyuN genannt. Dabei ist *N* die Nummer der Schnittstelle, deren Zählung bei Null beginnt. Allgemein wird diese Schnittstelle für Terminals benutzt. Diese Schnittstelle funktioniert nur, wenn ein "Data Carrier Detect" Signal (DCD) vorliegt.
- Für ausgehende Verbindungen wird in FreeBSD 8.X und neueren Versionen /dev/cuauN verwendet. FreeBSD 7.X und ältere Versionen verwenden /dev/cuadN. Dieser Port wird normalerweise nur von Modems genutzt. Er kann allerdings auch für Terminals benutzt werden, die das "Data Carrier Detect" Signal nicht unterstützen.

Wenn ein Terminal an die erste serielle Schnittstelle (COM1) angeschlossen ist, wird er über /dev/ttyu0 angesprochen. Wenn er an der zweiten seriellen Schnittstelle (COM2) angeschlossen ist, verwenden Sie /dev/ttyu1, usw.

## 45.2. Kernelkonfiguration

In der Voreinstellung benutzt FreeBSD vier serielle Schnittstellen, die unter MS-DOS® als COM1, COM2, COM3 und COM4 bekannt sind. Momentan unterstützt FreeBSD einfache Multiportkarten, wie bspw. die BocaBoard 1008 und 2016 und bessere wie die von Digiboard und Stallion Technologies. In der Voreinstellung sucht der Kernel allerdings nur nach den Standardanschlüssen.

Um zu überprüfen, ob der Kernel die seriellen Schnittstellen erkennt, achten Sie auf die Meldungen beim Booten, oder schauen sich diese später mit `/sbin/dmesg` an. Achten Sie auf Meldungen die mit

uart beginnen:

```
# /sbin/dmesg | grep 'uart'
```

Wenn der Kernel nicht alle seriellen Schnittstellen erkennt, müssen Sie /boot/device.hints konfigurieren. Wenn Sie diese Datei editieren, können Sie die Einträge für Geräte, die auf dem System nicht vorhanden sind, auskommentieren oder komplett entfernen.



port IO_COM1 ist ein Ersatz für port 0x3f8, IO_COM2 bedeutet port 0x2f8, IO_COM3 bedeutet port 0x3e8 und IO_COM4 steht für port 0x2e8. Die angegebenen IO-Adressen sind genau wie die Interrupts 4, 3, 5 und 9 üblich für serielle Schnittstellen. Beachten Sie, dass sich normale serielle Schnittstellen auf ISA-Bussen *keine* Interrupts teilen können. Multiportkarten besitzen zusätzliche Schaltkreise, die es allen 16550As auf der Karte erlauben, sich einen oder zwei Interrupts zu teilen.

## 45.3. Gerätedateien

Die meisten Geräte im Kernel werden durch Gerätedateien in /dev angesprochen. Die sio Geräte werden durch /dev/ttyuN für eingehende Verbindungen und durch /dev/cuauN für ausgehende Verbindungen angesprochen. Zum Initialisieren der Geräte stellt FreeBSD die Dateien /dev/ttyuN.init und /dev/cuauN.init zur Verfügung. Zusätzlich existieren Dateien für das Sperren von Gerätedateien (Locking). Dabei handelt es sich um die Dateien /dev/ttyuN.lock und /dev/cuauN.lock. Diese Dateien werden benutzt, um Kommunikationsparameter beim Öffnen eines Ports vorzugeben. Für Modems, die zur Flusskontrolle RTS/CTS benutzen, kann damit crtscts gesetzt werden. Die Geräte /dev/ttyldN und /dev/cualaN (locking devices) werden genutzt, um bestimmte Parameter festzuschreiben und vor Veränderungen zu schützen. Weitere Informationen zu Terminals finden Sie in termios(4), sio(4) erklärt die Dateien zum Initialisieren und Sperren der Geräte, stty(1) beschreibt schließlich Terminal-Einstellungen.

## 45.4. Konfiguration der seriellen Schnittstelle

Anwendungen benutzen normalerweise die Geräte ttyuN oder cuauN. Das Gerät besitzt einige Voreinstellungen für Terminal-I/O, wenn es von einem Prozess geöffnet wird. Mit dem folgenden Kommando können Sie sich diese Einstellungen ansehen:

```
# stty -a -f /dev/ttyu1
```

Wenn diese Einstellungen verändert werden, bleiben sie nur solange wirksam, bis das Gerät geschlossen wird. Wenn das Gerät danach wieder geöffnet wird, sind die Voreinstellungen wieder wirksam. Um die Voreinstellungen dauerhaft zu ändern, öffnen Sie das Gerät, das zum Initialisieren dient und verändern dessen Einstellungen. Um beispielsweise für ttyu5 den CLOCAL Modus, 8-Bit Kommunikation und XON/XOFF Flusssteuerung einzuschalten, setzen Sie das folgende Kommando ab:

```
# stty -f /dev/ttyu5.init clocal cs8 ixon ixoff
```

In /etc/rc.d/rc.serial werden die systemweiten Voreinstellungen für serielle Geräte vorgenommen.

Um zu verhindern, dass Einstellungen von Anwendungen verändert werden, können Sie die Geräte zum Festschreiben von Einstellungen ("locking devices") benutzen. Wenn sie beispielsweise die Geschwindigkeit von ttyu5 auf 57600 bps festlegen wollen, benutzen Sie das folgende Kommando:

```
# stty -f /dev/ttyld5 57600
```

Eine Anwendung, die ttyu5 öffnet, kann nun nicht mehr die Geschwindigkeit ändern und muss 57600 bps benutzen.

Die Geräte zum Initialisieren und Festschreiben von Einstellungen sollten selbstverständlich nur von `root` beschreibbar sein.

# Kapitel 46. Terminals

Wenn Sie sich nicht an der Konsole oder über ein Netzwerk an ein FreeBSD-System anmelden können, sind Terminals ein bequemer und kostengünstiger Weg, um auf ein System zuzugreifen. Dieser Abschnitt beschreibt wie Sie Terminals mit FreeBSD benutzen.

Das ursprüngliche UNIX® System besaß keine Konsolen. Zum Anmelden und Starten von Programmen wurden stattdessen Terminals benutzt, die an den seriellen Schnittstellen des Rechners angeschlossen waren.

Die Möglichkeit, über eine serielle Schnittstelle eine Anmeldesitzung herzustellen, existiert heute noch in fast jedem UNIX®-artigen Betriebssystem, einschließlich FreeBSD. Der Einsatz eines Terminals, das an einem freien seriellen Port angeschlossen ist, ermöglicht es dem Benutzer sich anzumelden und dort jedes Textprogramm zu starten, das normalerweise an der Konsole oder in einem `xterm` Fenster ausgeführt wird.

Viele Terminals können an einem FreeBSD-System angeschlossen werden. Ein alter Computer kann als Terminal an ein leistungsfähiges FreeBSD-System angeschlossen werden. Damit kann ein Einzelarbeitsplatz in ein leistungsfähiges Mehrbenutzersystem verwandelt werden.

FreeBSD unterstützt drei Arten von Anschlüssen:

**Dumb-Terminals**

Dumb-Terminals (unintelligente Datenstationen) sind Geräte, die über die serielle Schnittstelle mit einem Rechner verbunden werden. Sie werden "unintelligent" genannt, weil sie nur Text senden und empfangen und keine Programme laufen lassen können. Alle benötigten Programme befinden sich auf dem Rechner, der mit dem Terminal verbunden ist.

Es gibt viele Dumb-Terminals, die von verschiedenen Herstellern produziert werden, und so gut wie jeder der verschiedenen Terminals sollte mit FreeBSD zusammenarbeiten. Manche High-End Geräte verfügen sogar über Grafikfähigkeiten, die allerdings nur von spezieller Software genutzt werden kann.

Dumb-Terminals sind in Umgebungen beliebt, in denen keine Grafikanwendungen benötigt werden.

**Computer, die als Terminal fungieren**

Jeder Computer kann die Funktion eines Dumb-Terminals, der ja nur Text senden und empfangen kann, übernehmen. Dazu wird lediglich das richtige Kabel benötigt und eine *Terminalemulation*, die auf dem Computer läuft.

Diese Konfiguration ist sehr nützlich. Wenn ein Benutzer zum Beispiel gerade an der FreeBSD-Konsole arbeitet, kann ein anderer Benutzer einen weniger leistungsstarken Computer, der als Terminal mit dem FreeBSD-System verbunden ist, benutzen, um dort gleichzeitig im Textmodus zu arbeiten.

Bereits im Basissystem sind mindestens zwei Werkzeuge vorhanden, die Sie zur Arbeit über eine serielle Konsole einsetzen können: cu(1) sowie tip(1).

Um sich von einem FreeBSD-System aus über eine serielle Verbindung mit einem anderen System zu verbinden, geben Sie folgenden Befehl ein:

```
# cu -l /dev/cuauN
```

Die Ports sind von Null beginnend nummeriert. Das bedeutet, dass COM1 dem Gerät /dev/cuau0 entspricht.

In der Ports-Sammlung finden sich weitere Programme, wie beispielsweise comms/minicom, mit denen eine Verbindung über eine serielle Schnittstelle hergestellt werden kann.

**X-Terminals**

X-Terminals sind die ausgereiftesten der verfügbaren Terminals. Sie werden nicht mit der seriellen Schnittstelle sondern mit einem Netzwerk, wie dem Ethernet, verbunden. Diese Terminals sind auch nicht auf den Textmodus beschränkt, sondern können jede Xorg-Anwendung darstellen.

Die Einrichtung und Verwendung von X-Terminals wird in diesem Abschnitt nicht beschrieben.

# 46.1. Konfiguration

Dieser Abschnitt beschreibt, wie Sie ein FreeBSD-System konfigurieren müssen, um sich an einem Terminal anzumelden. Dabei wird vorausgesetzt, dass der Kernel bereits die serielle Schnittstelle, die mit dem Terminal verbunden ist, unterstützt. Weiterhin sollte der Terminal schon angeschlossen sein.

Der `init` Prozess ist für das Initialisieren des Systems und den Start von Prozessen zum Zeitpunkt des Systemstarts verantwortlich. Unter anderem liest `init`/etc/ttys ein und startet für jeden verfügbaren Terminal einen `getty` Prozess. `getty` wiederum fragt beim Anmelden den Benutzernamen ab und startet `login`.

Um Terminals auf einem FreeBSD-System einzurichten, führen Sie folgenden Schritte als `root` durch:

1. Fügen Sie einen Eintrag in /etc/ttys für die serielle Schnittstelle aus /dev ein, falls dieser nicht bereits vorhanden ist.
2. Geben Sie `/usr/libexec/getty` als auszuführendes Programm an. Als Parameter für *getty* geben Sie den passenden Verbindungstyp aus /etc/gettytab an.
3. Geben Sie den Terminaltyp an.
4. Aktivieren Sie den Anschluss.
5. Geben Sie die Sicherheit des Anschlusses an.
6. Veranlassen Sie `init`/etc/ttys erneut zu lesen.

Optional können Sie in /etc/gettytab auch einen auf Ihre Zwecke angepassten Terminaltyp erstellen. gettytab(5) und getty(8) enthalten dazu weitere Informationen.

### 46.1.1. Hinzufügen eines Eintrags in /etc/ttys

In /etc/ttys werden alle Terminals aufgeführt, an denen eine Anmeldung auf dem FreeBSD-System möglich ist. Hier findet sich zum Beispiel ein Eintrag für die erste virtuelle Konsole /dev/ttyv0, der es Benutzern ermöglicht, sich dort anzumelden. Die Datei enthält weitere Einträge für andere virtuelle Konsolen, serielle Schnittstellen und Pseudoterminals. Um einen Terminal zu konfigurieren, fügen Sie einen Eintrag für den Namen des Gerätes aus /dev ohne das Präfix /dev hinzu. Zum Beispiel wird /dev/ttyv0 als `ttyv0` aufgeführt.

In der Voreinstellung enthält /etc/ttys Einträge für die ersten vier seriellen Schnittstellen: ttyu0 bis ttyu3. Wird an eine von diesen Schnittstellen ein Terminal angeschlossen, braucht in dieser Datei kein weiter Eintrag hinzugefügt werden.

*Beispiel 44. Einträge in /etc/ttys hinzufügen*

Dieses Beispiel konfiguriert zwei Terminals: Einen Wyse-50 und einen alten 286 IBM PC, der mit Procomm einen VT-100 Terminal emuliert. Der Wyse-Terminal ist mit der zweiten seriellen Schnittstelle verbunden und der 286 mit der sechsten seriellen Schnittstelle, einem Anschluss auf einer Multiportkarte. Die entsprechenden Einträge in /etc/ttys würden dann wie folgt aussehen:

```
ttyu1  "/usr/libexec/getty std.38400"  wy50  on  insecure
ttyu5   "/usr/libexec/getty std.19200"  vt100  on  insecure
```

Das erste Feld gibt normalerweise den Namen der Gerätedatei aus /dev an.

Im zweiten Feld wird das auszuführende Kommando, normal ist das getty(8), angegeben. `getty` initialisiert und öffnet die Verbindung, setzt die Geschwindigkeit und fragt den Benutzernamen ab. Danach führt es login(1) aus.

`getty` akzeptiert einen optionalen Parameter auf der Kommandozeile, den Verbindungstyp, der die Eigenschaften der Verbindung, wie die Geschwindigkeit und Parität, festlegt. Die Typen und die damit verbundenen Eigenschaften liest `getty` aus /etc/gettytab.

/etc/gettytab enthält viele Einträge sowohl für neue wie auch alte Terminalverbindungen. Die meisten Einträge, die mit `std` beginnen, sollten mit einem festverdrahteten Terminal funktionieren. Für jede Geschwindigkeit zwischen 110 bps und 115200 bps gibt es einen `std` Eintrag. Weitere Informationen dazu finden Sie in gettytab(5).

Wenn Sie den Verbindungstyp in /etc/ttys eintragen, stellen Sie sicher, dass die Kommunikationseinstellungen auch mit denen des Terminals übereinstimmen.

In diesem Beispiel verwendet der Wyse-50 keine Parität und 38400 bps, der 286 PC benutzt ebenfalls keine Parität und arbeitet mit 19200 bps.

Das dritte Feld gibt den Terminaltyp an, der normalerweise mit diesem Anschluss verbunden ist. Für Einwählverbindungen wird oft `unknown` oder `dialup` benutzt, da sich die Benutzer praktisch mit beliebigen Terminals oder Emulatoren anmelden können. Bei festverdrahteten Terminals ändert sich der Typ nicht, so dass in diesem Feld ein richtiger Typ aus der

termcap(5) Datenbank angegeben werden kann.In diesem Beispiel benutzt der Wyse-50 den entsprechenden Typ aus termcap(5), der 286 PC wird als VT-100, den er ja emuliert, angegeben.

Das vierte Feld gibt an, ob der Anschluss aktiviert werden soll. Ist das Feld auf `on` gesetzt, startet `init` das Programm, das im zweiten Feld angegeben ist. Normalerweise ist dies `getty`. Wenn das Feld auf `off` gesetzt wird, wird `getty` nicht ausgeführt und folglich kann sich niemand an dem betreffenden Terminal anmelden.

Das letzte Feld gibt die Sicherheit des Anschlusses an. Wenn hier `secure` angegeben wird, darf sich `root`, oder jeder Account mit der UID `0` über diese Verbindung anmelden. Wenn `insecure` angegeben wird, dürfen sich nur unprivilegierte Benutzer anmelden. Diese können später mit su(1) oder einem ähnlichen Mechanismus zu `root` wechseln.Es wird dringend empfohlen `insecure` zu verwenden, sogar für Terminals hinter verschlossenen Türen. Es ist ganz einfach sich mit `su` anzumelden, wenn Superuser-Rechte benötigt werden.

### 46.1.2. `init` zwingen, /etc/ttys erneut zu lesen

Nachdem Änderungen in /etc/ttys vorgenommen wurden, schicken Sie `init` ein SIGHUP-Signal (hangup), um es zu veranlassen, seine Konfigurationsdatei neu zu lesen:

```
# kill -HUP 1
```



Da `init` immer der erste Prozess auf einem System ist, besitzt es immer die Prozess-ID `1`.

Wenn alles richtig eingerichtet ist, alle Kabel angeschlossen und die Terminals eingeschaltet sind, sollte für jeden Terminal ein `getty` Prozess laufen und auf jedem Terminal sollte eine Anmeldeaufforderung zu sehen sein.

## 46.2. Fehlersuche

Selbst wenn Sie den Anweisungen akribisch gefolgt sind, kann es immer noch zu Fehlern beim Einrichten eines Terminals kommen. Hier eine Liste der häufigsten Symptome, sowie einige mögliche Lösungen:

Wenn kein Anmeldeprompt erscheint, stellen Sie sicher, dass der Terminal verbunden und eingeschaltet ist. Wenn ein PC als Terminal fungiert, überprüfen Sie, dass die Terminalemulation auf den richtigen Schnittstellen läuft.

Stellen Sie sicher, dass Sie das richtige Kabel verwenden und dass das Kabel fest mit dem Terminal und dem FreeBSD-Rechner verbunden ist.

Stellen Sie sicher, dass die Einstellungen für die Geschwindigkeit (bps) und Parität auf dem FreeBSD-System und dem Terminal gleich sind. Wenn der Terminal einen Bildschirm besitzt, überprüfen Sie die richtige Einstellung von Helligkeit und Kontrast. Wenn der Terminal druckt, stellen Sie die ausreichende Versorgung mit Papier und Tinte sicher.

Überprüfen Sie mit `ps`, dass der `getty` Prozess für den Terminal läuft:

```
# ps -axww|grep getty
```

Für jeden Terminal sollte ein Eintrag vorhanden sein. Aus dem folgenden Beispiel ist zu erkennen, dass `getty` auf der zweiten seriellen Schnittstelle tyyd1 läuft und den Verbindungstyp `std.38400` aus /etc/gettytab benutzt:

```
22189  d1  Is+    0:00.03 /usr/libexec/getty std.38400 ttyu1
```

Wenn `getty` nicht läuft, überprüfen Sie, ob der Anschluss in /etc/ttys aktiviert ist. Denken Sie daran `kill -HUP 1` auszuführen, nachdem /etc/ttys geändert wurde.

Wenn `getty` läuft, aber der Terminal immer noch kein Anmeldeprompt ausgibt, oder am Anmeldeprompt nichts eingegeben werden kann, kann es sein, dass der Terminal oder Kabel keinen Hardware-Handshake unterstützt. Ändern Sie dann den Eintrag `std.38400` in /etc/ttys zu `3wire.38400`. Nachdem Sie /etc/ttys geändert haben, setzen Sie `kill -HUP 1` ab. Der Eintrag `3wire` besitzt ähnliche Eigenschaften wie der Eintrag `std`, ignoriert aber den Hardware-Handshake. Wenn Sie den Eintrag `3wire` verwenden, muss vielleicht die Geschwindigkeit verkleinert oder die Software-Flusssteuerung aktiviert werden, um Pufferüberläufe zu vermeiden.

Wenn nur unverständliche Zeichen erscheinen, stellen Sie sicher, dass die Einstellungen für die Geschwindigkeit (bps) und Parität auf dem FreeBSD-System und dem Terminal gleich sind. Kontrollieren Sie den `getty` Prozess und stellen Sie sicher, dass der richtige Verbindungstyp aus /etc/gettytab benutzt wird. Wenn das nicht der Fall ist, editieren Sie /etc/ttys und setzen das Kommando `kill-HUP 1` ab.

Wenn Zeichen doppelt und eingegebene Passwörter im Klartext erscheinen, stellen Sie den Terminal oder die Terminalemulation von "half duplex" oder "local echo" auf "full duplex" um.

# Kapitel 47. Einwählverbindungen

Das Einrichten von Einwählverbindungen auf FreeBSD-Systemen ähnelt dem Anschließen von Terminals, nur dass anstelle eines Terminals ein Modem verwendet wird. FreeBSD unterstützt sowohl externe als auch interne Modems.

Externe Modems sind für Einwählverbindungen besser geeignet, da sie die Konfiguration in nicht flüchtigem RAM speichern können. Zudem verfügen Sie über Leuchtanzeigen, die den Status wichtiger RS-232 Signale anzeigen.

Interne Modems verfügen normalerweise nicht über nicht flüchtiges RAM und lassen sich meist nur über DIP-Schalter konfigurieren. Selbst wenn ein internes Modem Leuchtanzeigen besitzt, sind diese meist schwer einzusehen, wenn das Modem eingebaut ist.

Mit einem externen Modem muss das passende Kabel verwendet werden. Ein Standard RS-232C Kabel, bei dem die folgenden Signale miteinander verbunden sind, sollte ausreichen:

*Tabelle 22. Signalnamen*

| Abkürzung | Bedeutung |
|---|---|
| RD | Received Data |
| TD | Transmitted Data |
| DTR | Data Terminal Ready |
| DSR | Data Set Ready |
| DCD | Data Carrier Detect (dadurch erkennt RS-232 das Signal *Received Line*) |
| SG | Signal Ground |
| RTS | Request to Send |
| CTS | Clear to Send |

Ab Geschwindigkeiten von 2400 bps benötigt FreeBSD die Signale RTS und CTS für die Flusssteuerung. Das Signal CD zeigt an, ob ein Träger vorliegt, das heißt ob die Verbindung aufgebaut ist oder beendet wurde. DTR zeigt an, dass das Gerät betriebsbereit ist. Es gibt einige Kabel, bei denen nicht alle nötigen Signale verbunden sind. Wenn Probleme dieser Art auftreten, dass zum Beispiel die Sitzung nicht beendet wird, obwohl die Verbindung beendet wurde, kann das an einem solchen Kabel liegen.

Wie andere UNIX® Betriebssysteme auch, benutzt FreeBSD Hardwaresignale, um festzustellen, ob ein Anruf beantwortet wurde, eine Verbindung beendet wurde, oder um die Verbindung zu schließen und das Modem zurückzusetzen. FreeBSD vermeidet es, dem Modem Kommandos zu senden, oder den Statusreport des Modems abzufragen.

## 47.1. Schnittstellenbausteine

FreeBSD unterstützt EIA RS-232C (CCITT V.24) serielle Schnittstellen, die auf den NS8250, NS16450,

NS16550 oder NS16550A Bausteinen basieren. Die Bausteine der Serie 16550 verfügen über einen 16 Byte großen Puffer, der als FIFO angelegt ist. Wegen Fehler in der FIFO-Logik kann der Puffer in einem 16550 Baustein allerdings nicht genutzt werden, das heißt der Baustein muss als 16450 betrieben werden. Bei allen Bausteinen ohne Puffer und dem 16550 Baustein muss jedes Byte einzeln von dem Betriebssystem verarbeitet werden, was Fehler bei hohen Geschwindigkeiten oder großer Systemlast erzeugt. Es sollten daher nach Möglichkeit serielle Schnittstellen, die auf 16550A Bausteinen basieren, eingesetzt werden.

## 47.2. Überblick

Wie bei Terminals auch, startet `init` für jede serielle Schnittstelle, die eine Einwählverbindung zur Verfügung stellt, einen `getty` Prozess. Wenn das Modem beispielsweise an /dev/ttyu0 angeschlossen ist, sollte in der Ausgabe von `ps ax` eine Zeile wie die folgende erscheinen:

```
4850 ??  I      0:00.09 /usr/libexec/getty V19200 ttyu0
```

Wenn sich ein Benutzer einwählt und die Verbindung aufgebaut ist, zeigt das Modem dies durch das CD Signal (Carrier Detect) an. Der Kernel merkt, dass ein Signal anliegt und weist `getty` an, die Schnittstelle zu öffnen. Dann sendet `getty` das Anmeldeprompt mit der ersten für die Verbindung vereinbarten Geschwindigkeit und wartet auf eine Antwort. Wenn die Antwort unverständlich ist, weil zum Beispiel die Geschwindigkeit des Modems von `getty`s Geschwindigkeit abweicht, versucht `getty` die Geschwindigkeit solange anzupassen, bis es eine verständliche Antwort erhält.

Nachdem der Benutzer seinen Benutzernamen eingegeben hat, führt `getty`/usr/bin/login aus, welches das Passwort abfragt und danach die Shell des Benutzers startet.

## 47.3. Konfigurationsdateien

Drei Konfigurationsdateien in /etc steuern, ob eine Einwahl in das FreeBSD-System möglich ist. /etc/gettytab, konfiguriert den /usr/libexec/getty Dæmon. In /etc/ttys wird festgelegt, auf welchen Schnittstellen /sbin/init einen `getty` Prozess startet. Schließlich bietet /etc/rc.d/serial die Möglichkeit, Schnittstellen zu initialisieren.

Es gibt zwei Ansichten darüber, wie Modems für Einwählverbindungen unter UNIX® zu konfigurieren sind. Zum einen kann die Geschwindigkeit zwischen dem Modem und dem Computer fest eingestellt werden. Sie ist damit unabhängig von der Geschwindigkeit, mit der sich der entfernte Benutzer einwählt. Dies hat den Vorteil, dass der entfernte Benutzer das Anmeldeprompt sofort bekommt. Der Nachteil bei diesem Verfahren ist, dass das System die tatsächliche Geschwindigkeit der Verbindung nicht kennt. Damit können bildschirmorientierte Programme wie Emacs ihren Bildschirmaufbau nicht an langsame Verbindungen anpassen, um die Antwortzeiten zu verbessern.

Die andere Möglichkeit besteht darin, die Geschwindigkeit der RS-232 Schnittstelle des lokalen Modems an die Geschwindigkeit des entfernten Modems anzupassen. Bei einer V.32bis (14400 bps) Verbindung kann das lokale Modem die RS-232 Schnittstelle mit 19200 bps betreiben, während bei einer Verbindung mit 2400 bps die RS-232 Schnittstelle mit 2400 bps betrieben wird. Da `getty` die Verbindungsgeschwindigkeit des Modems nicht kennt, startet es den Anmeldevorgang mit der

Ausgabe von `login:` und wartet auf eine Antwort. Wenn der Benutzer der Gegenstelle nun nur unverständliche Zeichen erhält, muss er solange `Enter` drücken, bis das Anmeldeprompt erscheint. Solange die Geschwindigkeiten nicht übereinstimmen, sind die Antworten der Gegenstelle für `getty` ebenfalls unverständlich. In diesem Fall wechselt `getty` zur nächsten Geschwindigkeit und gibt wieder `login:` aus. In aller Regel erhält der Benutzer der Gegenstelle nach ein bis zwei Tastendrücken eine erkennbare Anmeldeaufforderung. Diese Anmeldeprozedur sieht nicht so sauber wie die Methode mit einer festen Geschwindigkeit aus, bietet dem Benutzer einer langsamen Verbindung allerdings den Vorteil, dass sich bildschirmorientierte Programme an die Geschwindigkeit anpassen können.

Im Folgenden wird die Konfiguration für beide Methoden besprochen, doch die Methode der angepassten Geschwindigkeit wird bei der Diskussion bevorzugt.

### 47.3.1. /etc/gettytab

Mit /etc/gettytab wird getty(8) im Stil von termcap(5) konfiguriert. Das Format dieser Datei und die Bedeutung der Einträge wird in gettytab(5) beschrieben.

Wenn die Modemgeschwindigkeit vorgeben wird, sollten Anpassungen in /etc/gettytab nicht erforderlich sein.

Wenn jedoch die Geschwindigkeit angepasst werden soll, erstellen Sie einen Eintrag in /etc/gettytab, um `getty` die Geschwindigkeit für das Modem mitzuteilen. Für ein 2400 bps Modem kann der vorhandene `D2400` Eintrag benutzt werden.

```
#
# Fast dialup terminals, 2400/1200/300 rotary (can start either way)
#
D2400|d2400|Fast-Dial-2400:\
        :nx=D1200:tc=2400-baud:
3|D1200|Fast-Dial-1200:\
        :nx=D300:tc=1200-baud:
5|D300|Fast-Dial-300:\
        :nx=D2400:tc=300-baud:
```

Wird ein Modem mit einer höheren Geschwindigkeit eingesetzt, müssen weitere Einträge in /etc/gettytab erstellt werden. Dieses Beispiel zeigt einen Eintrag für ein 14400 bps Modem mit einer Geschwindigkeit bis zu 19200 bps:

```
#
# Additions for a V.32bis Modem
#
um|V300|High Speed Modem at 300,8-bit:\
        :nx=V19200:tc=std.300:
un|V1200|High Speed Modem at 1200,8-bit:\
        :nx=V300:tc=std.1200:
uo|V2400|High Speed Modem at 2400,8-bit:\
        :nx=V1200:tc=std.2400:
```

```
up|V9600|High Speed Modem at 9600,8-bit:\
        :nx=V2400:tc=std.9600:
uq|V19200|High Speed Modem at 19200,8-bit:\
        :nx=V9600:tc=std.19200:
```

Die damit erzeugten Verbindungen verwenden 8 Bit und keine Parität.

Im obigen Beispiel startet die Geschwindigkeit bei 19200 bps (eine V.32bis Verbindung) und geht dann über 9600 bps (V.32), 400 bps, 1200 bps und 300 bps wieder zurück zu 19200 bps. Das Schlüsselwort `nx=` (next table) sorgt für das zyklische Durchlaufen der Geschwindigkeiten. Jede Zeile zieht zudem noch mit `tc=` (table continuation) die Vorgabewerte für die jeweilige Geschwindigkeit an.

Wenn Sie ein 28800 bps Modem besitzen und/oder Kompression mit einem 14400 bps Modem benutzen wollen, brauchen Sie höhere Geschwindigkeiten als 19200 bps. Das folgende Beispiel startet mit 57600 bps:

```
#
# Additions for a V.32bis or V.34 Modem
# Starting at 57600 bps
#
vm|VH300|Very High Speed Modem at 300,8-bit:\
        :nx=VH57600:tc=std.300:
vn|VH1200|Very High Speed Modem at 1200,8-bit:\
        :nx=VH300:tc=std.1200:
vo|VH2400|Very High Speed Modem at 2400,8-bit:\
        :nx=VH1200:tc=std.2400:
vp|VH9600|Very High Speed Modem at 9600,8-bit:\
        :nx=VH2400:tc=std.9600:
vq|VH57600|Very High Speed Modem at 57600,8-bit:\
        :nx=VH9600:tc=std.57600:
```

Wenn Sie eine langsame CPU oder ein stark ausgelastetes System besitzen und sich kein 16550A im System befindet, erhalten Sie bei 57600 bps vielleicht `sio` Fehlermeldungen der Form "silo overflow".

### 47.3.2. /etc/ttys

/etc/ttys wurde bereits in Einträge in /etc/ttys hinzufügen besprochen. Die Konfiguration für Modems ist ähnlich, allerdings braucht `getty` ein anderes Argument und es muss ein anderer Terminaltyp angegeben werden. Der Eintrag für beide Methoden (feste und angepasste Geschwindigkeit) hat die folgende Form:

```
ttyu0   "/usr/libexec/getty xxx"   dialup on
```

Das erste Feld der obigen Zeile gibt die Gerätedatei für diesen Eintrag an. ttyu0 bedeutet, dass `getty` mit /dev/ttyu0 arbeitet. Das zweite Feld `"/usr/libexec/getty xxx"` gibt das Kommando an, das `init`

für dieses Gerät startet (*xxx* wird durch einen passenden Eintrag aus /etc/gettytab ersetzt). Die Vorgabe für den Terminaltyp, hier `dialup`, wird im dritten Feld angegeben. Das vierte Feld, `on`, zeigt `init` an, dass die Schnittstelle aktiviert ist. Im fünften Feld könnte noch `secure` angegeben werden, um Anmeldungen von `root` zu erlauben, doch sollte das wirklich nur für physikalisch sichere Terminals, wie die Systemkonsole, aktiviert werden.

Die Vorgabe für den Terminaltyp, `dialup` im obigen Beispiel, hängt von lokalen Gegebenheiten ab. Traditionell wird `dialup` für Einwählverbindungen verwendet, so dass die Benutzer in ihren Anmeldeskripten den Terminaltyp auf ihren Terminal abstimmen können, wenn der Typ auf `dialup` gesetzt ist. Wenn Sie nur VT102 Terminals oder Emulatoren einsetzen, können Sie den Terminaltyp hier auch fest auf `vt102` setzen.

Nachdem /etc/ttys geändert wurde, muss `init` ein HUP Signal schicken, damit es die Datei wieder einliest:

```
# kill -HUP 1
```

Stellen Sie sicher, dass das Modem richtig konfiguriert und angeschlossen ist, bevor Sie das Signal an `init` schicken.

Das Argument von `getty` muss in diesem Fall eine feste Geschwindigkeit vorgeben. Der Eintrag für ein Modem, das fest auf 19200 bps eingestellt ist, könnte wie folgt aussehen:

```
ttyu0   "/usr/libexec/getty std.19200"   dialup on
```

Wenn das Modem auf eine andere Geschwindigkeit eingestellt ist, setzen Sie anstelle von `std.19200` einen passenden Eintrag der Form `std.speed` ein. Stellen Sie sicher, dass dies auch ein gültiger Verbindungstyp aus /etc/gettytab ist.

Das Argument von `getty` muss hier auf einen der Einträge aus /etc/gettytab zeigen, der zu einer Kette von Einträgen gehört, die die zu probierenden Geschwindigkeiten beschreiben. Wenn Sie dem obigen Beispiel gefolgt sind und zusätzliche Einträge in /etc/gettytab erzeugt haben, können Sie die folgende Zeile verwenden:

```
ttyu0   "/usr/libexec/getty V19200"   dialup on
```

### 47.3.3. /etc/rc.d/serial

Modems, die höhere Geschwindigkeiten unterstützen, zum Beispiel V.32, V.32bis und V.34 Modems, benutzen Hardware-Flusssteuerung (`RTS/CTS`). Für die entsprechenden Schnittstellen können Sie die Flusssteuerung mit `stty` in /etc/rc.d/serial einstellen.

Um beispielsweise die Hardware-Flusssteuerung für die Geräte zur Ein- und Auswahl der zweiten seriellen Schnittstelle (COM2) zu aktivieren, benutzen Sie die Dateien zur Initialisierung der entsprechenden Geräte und fügen die folgenden Zeilen in /etc/rc.d/serial hinzu:

```
# Serial port initial configuration
stty -f /dev/ttyu1.init crtscts
stty -f /dev/cuad1.init crtscts
```

## 47.4. Modemkonfiguration

Für ein Modem, das seine Konfiguration in nicht flüchtigem RAM speichert, wird ein Terminalprogramm wie Telix unter MS-DOS® oder `tip` unter FreeBSD benötigt, um die Parameter einzustellen. Verbinden Sie sich mit derselben Geschwindigkeit, die `getty` zuerst benutzen würde, mit dem Modem und treffen Sie folgende Einstellungen:

- DCD ist eingeschaltet, wenn das Trägersignal des entfernten Modems erkannt wird.
- Im Betrieb liegt DTR an. Bei einem Verlust von DTR legt das Modem auf und setzt sich zurück.
- CTS Flusssteuerung ist für ausgehende Daten aktiviert.
- XON/XOFF Flusssteuerung ist ausgeschaltet.
- RTS Flusssteuerung ist für eingehende Daten aktiviert.
- Keine Rückmeldungen ausgeben.
- Die Echo-Funktion ist deaktiviert.

Lesen Sie die Dokumentation für das Modem, um herauszufinden welche Befehle und/oder DIP-Schalterstellungen benötigt werden.

Für ein externes 14400 gelten zum Beispiel die folgenden Befehle:

```
ATZ
ATC1D2H1I0R2W
```

Bei dieser Gelegenheit können Sie auch gleich andere Einstellungen, zum Beispiel ob Sie V42.bis und/oder MNP5 Kompression benutzen wollen, an Ihrem Modem vornehmen.

Bei einem externen 14400 müssen Sie auch noch einige DIP-Schalter einstellen. Die folgenden Einstellungen können verwendet werden:

- Schalter 1: OBEN - DTR normal
- Schalter 2: N/A (Rückmeldungen als Text/numerische Rückmeldungen)
- Schalter 3: OBEN - Keine Rückmeldungen ausgeben
- Schalter 4: UNTEN - Echo-Funktion aus
- Schalter 5: OBEN - Rufannahme aktiviert
- Schalter 6: OBEN - Carrier Detect normal
- Schalter 7: OBEN - Einstellungen aus dem NVRAM laden
- Schalter 8: N/A (Smart Mode/Dumb Mode)

Für Einwählverbindungen sollten die Rückmeldungen deaktiviert sein, da sonst `getty` dem Modem das Anmeldeprompt `login:` schickt und das Modem im Kommandomodus das Prompt wieder ausgibt (Echo-Funktion) oder eine Rückmeldung gibt. Das führt dann zu einer länglichen und fruchtlosen Kommunikation zwischen dem Modem und `getty`.

Die Geschwindigkeit zwischen Modem und Computer muss auf einen festen Wert eingestellt werden. Mit einem externen 14400 Modem setzen die folgenden Kommandos die Geschwindigkeit auf den Wert der Datenendeinrichtung fest:

```
ATZ
ATB1W
```

In diesem Fall muss die Geschwindigkeit der seriellen Schnittstelle des Modems der eingehenden Geschwindigkeit angepasst werden. Für ein externes 14400 Modem erlauben die folgenden Befehle eine Anpassung der Geschwindigkeit der seriellen Schnittstelle für Verbindungen, die keine Fehlerkorrektur verwenden:

```
ATZ
ATB2W
```

Verbindungen mit Fehlerkorrektur (V.42, MNP) verwenden die Geschwindigkeit der Datenendeinrichtung.

### 47.4.1. Überprüfen der Modemkonfiguration

Die meisten Modems verfügen über Kommandos, die die Konfiguration des Modems in lesbarer Form ausgeben. Auf einem externen 14400 zeigt `ATI5` die Einstellungen im nicht flüchtigen RAM an. Um die wirklichen Einstellungen unter Berücksichtigung der DIP-Schalter zu sehen, benutzen Sie `ATZ` gefolgt von `ATI4`.

Wenn Sie ein anderes Modem benutzen, schauen Sie bitte in der Dokumentation des Modems nach, wie Sie die Konfiguration des Modems überprüfen können.

## 47.5. Fehlersuche

Bei Problemen können Sie die Einwählverbindung anhand der folgenden Punkte überprüfen:

Schließen Sie das Modem an das FreeBSD-System an und booten Sie das System. Wenn das Modem über Statusindikatoren verfügt, überprüfen Sie, ob der DTR Indikator leuchtet, wenn das Anmeldeprompt erscheint. Dies zeigt an, dass das FreeBSD-System einen `getty` Prozess auf der entsprechenden Schnittstelle gestartet hat und das Modem auf einkommende Verbindungen wartet.

Wenn der DTR-Indikator nicht leuchtet, melden Sie sich an dem FreeBSD-System an und überprüfen mit `ps ax`, ob FreeBSD einen `getty`-Prozess auf der entsprechenden Schnittstelle gestartet hat:

```
114 ??  I      0:00.10 /usr/libexec/getty V19200 ttyu0
115 ??  I      0:00.10 /usr/libexec/getty V19200 ttyu1
```

Wenn das Modem noch keinen Anruf entgegengenommen hat und Sie stattdessen die folgende Zeile sehen

```
114 d0  I      0:00.10 /usr/libexec/getty V19200 ttyu0
```

bedeutet dies, dass `getty` die Schnittstelle schon geöffnet hat und zeigt Kabelprobleme oder eine falsche Modemkonfiguration an, da `getty` die Schnittstelle erst dann öffnen kann, wenn das CD Signal (Carrier Detect) vom Modem anliegt.

Wenn Sie keine `getty`-Prozesse auf den gewünschten ttyuN Ports finden, untersuchen Sie /etc/ttys auf Fehler. Suchen Sie auch in /var/log/messages nach Meldungen von `init` oder `getty`. Wenn Sie dort Meldungen finden, sollten Sie noch einmal die beiden Konfigurationsdateien /etc/ttys und /etc/gettytab nach Fehlern durchsehen. Überprüfen Sie auch, ob die Gerätedateien /dev/ttyuN vorhanden sind.

Versuchen Sie als nächstes, sich in das System einzuwählen. Auf dem entfernten System stellen Sie bitte die folgenden Kommunikationsparameter ein: 8 Bit, keine Parität, ein Stop-Bit. Wenn kein Anmeldeprompt erscheint oder nur unleserliche Zeichen, drücken Sie mehrmals, in Abständen von ungefähr einer Sekunde, `Enter`. Wenn Sie immer noch nicht die `login:` Meldung sehen, schicken Sie ein `BREAK` Kommando. Wenn Sie zur Einwahl ein Highspeed-Modem benutzen, verwenden Sie eine feste Geschwindigkeit auf der seriellen Schnittstelle des Modems.

Wenn jetzt immer noch kein Anmeldeprompt erscheint, überprüfen Sie nochmals /etc/gettytab und stellen sicher, dass:

- der Verbindungstyp in /etc/ttys zu einem gültigen Eintrag in /etc/gettytab gehört.
- jeder der `nx=` Einträge in gettytab gültig ist und
- jeder `tc=` Eintrag auf einen gültigen Eintrag in gettytab verweist.

Wenn das Modem am FreeBSD-System auf einen eingehenden Anruf nicht antwortet, stellen Sie sicher, dass das Modem so konfiguriert ist, dass es einen Anruf beantwortet, wenn DTR anliegt. Wenn das Modem Statusindikatoren besitzt, können Sie das Anliegen von DTR anhand der Leuchten überprüfen.

Wenn Sie alles schon mehrfach überprüft haben und es immer noch noch nicht funktioniert, versuchen Sie es zu einem späteren Zeitpunkt erneut. Wenn es immer noch nicht funktioniert, können Sie eine Mail an die Mailingliste schicken, in der Sie Ihr Modem und Ihr Problem beschreiben.

# Kapitel 48. Verbindungen nach Außen

Die folgenden Ratschläge beschreiben, wie Sie mit einem Modem eine Verbindung zu einem anderen Computer herstellen. Dies können Sie nutzen, um sich auf einem entfernten Computer anzumelden.

Weiterhin ist diese Art von Verbindungen nützlich, wenn PPP mal nicht funktioniert. Wenn Sie zum Beispiel eine Datei mit FTP übertragen wollen und das über PPP gerade nicht möglich ist, melden Sie sich auf dem entfernten Rechner an und führen dort die FTP-Sitzung durch. Die Dateien können danach mit zmodem auf den lokalen Rechner übertragen werden.

## 48.1. Ein Hayes Modem benutzen

Es gibt einen eingebauten, allgemeinen Hayes Wähler in `tip`. Verwenden Sie `at=hayes` in /etc/remote.

Der Hayes-Treiber ist nicht schlau genug, um ein paar der erweiterten Funktionen von neueren Modems, bspw. `BUSY`, `NO DIALTONE` oder `CONNECT 115200` zu nutzen. Schalten Sie diese Nachrichten mit Hilfe von `ATX0W` ab, wenn Sie `tip` benutzen.

Der Anwahl-Timeout von `tip` beträgt 60 Sekunden. Das Modem sollte weniger verwenden, oder `tip` denkt, dass ein Kommunikationsfehler vorliegt. Versuchen Sie es mit `ATS7=45W`.

## 48.2. AT-Befehle benutzen

Erstellen Sie einen `direct` Eintrag in /etc/remote. Wenn das Modem zum Beispiel an der ersten seriellen Schnittstelle, /dev/cuad0, angeschlossen ist, dann fügen Sie die folgende Zeile hinzu:

```
cuad0:dv=/dev/cuad0:br#19200:pa=none
```

Verwenden Sie die höchste bps-Rate, die das Modem in der `br` Fähigkeit unterstützt. Geben Sie dann `tip cuad0` ein und Sie sind mit dem Modem verbunden.

Oder benutzen Sie `cu` als `root` mit dem folgenden Befehl:

```
# cu -lline -sspeed
```

*line* steht für die serielle Schnittstelle (/dev/cuad0) und *speed* für die Geschwindigkeit (`57600`). Wenn Sie mit dem Eingeben der AT Befehle fertig sind, beenden Sie mit `~.`.

## 48.3. Das @ Zeichen funktioniert nicht

Das `@` Zeichen in der Telefonnummerfähigkeit sagt `tip`, dass es in /etc/phones nach einer Nummer suchen soll. Aber `@` ist auch ein spezielles Zeichen in den Dateien, in denen Fähigkeiten beschrieben werden, wie /etc/remote. Schreiben Sie es mit einem Backslash:

```
pn=\@
```

## 48.4. Wie kann ich von der Kommandozeile eine Telefonnummer wählen?

Setzen Sie einen allgemeinen Eintrag in /etc/remote. Zum Beispiel:

```
tip115200|Dial any phone number at 115200 bps:\
        :dv=/dev/cuad0:br#115200:at=hayes:pa=none:du:
tip57600|Dial any phone number at 57600 bps:\
        :dv=/dev/cuad0:br#57600:at=hayes:pa=none:du:
```

Folgendes sollte jetzt funktionieren:

```
# tip -115200 5551234
```

Benutzer, die `cu` gegenüber `tip` bevorzugen, können einen allgemeinen `cu`-Eintrag verwenden:

```
cu115200|Use cu to dial any number at 115200bps:\
        :dv=/dev/cuad1:br#57600:at=hayes:pa=none:du:
```

und benutzen zum Wählen das Kommando:

```
# cu 5551234 -s 115200
```

## 48.5. Die bps-Rate angeben

Schreiben Sie einen `tip1200`- oder einen `cu1200`-Eintrag, aber geben Sie auch die bps-Rate an, die das Modem wirklich unterstützt. Leider denkt tip(1), dass 1200 bps ein guter Standardwert ist und deswegen sucht es nach einem `tip1200`-Eintrag. Natürlich müssen Sie nicht 1200 bps benutzen.

## 48.6. Über einen Terminal-Server auf verschiedene Rechner zugreifen

Sie müssen nicht warten bis Sie verbunden sind, und jedes Mal `CONNECT Rechner` eingeben, benutzen Sie `tip`s `cm`-Fähigkeit. Sie können diese Einträge in /etc/remote verwenden. Mit den Befehlen `tip pain` oder `tip muffin` können Sie eine Verbindungen zu den Rechnern `pain` oder `muffin` herstellen; mit `tip deep13` verbinden Sie sich mit dem Terminalserver.

```
pain|pain.deep13.com|Forrester's machine:\
        :cm=CONNECT pain\n:tc=deep13:
```

```
muffin|muffin.deep13.com|Frank's machine:\
        :cm=CONNECT muffin\n:tc=deep13:
deep13:Gizmonics Institute terminal server:\
        :dv=/dev/cuad2:br#38400:at=hayes:du:pa=none:pn=5551234:
```

## 48.7. Mehr als eine Verbindung mit `tip` benutzen

Das ist oft ein Problem, wenn eine Universität mehrere Telefonleitungen hat und viele tausend Studenten diese benutzen wollen.

Erstellen Sie einen Eintrag in /etc/remote und benutzen Sie `@` für die `pn`-Fähigkeit:

```
big-university:\
        :pn=\@:tc=dialout
dialout:\
        :dv=/dev/cuad3:br#9600:at=courier:du:pa=none:
```

Listen Sie dann die Telefonnummern in /etc/phones auf:

```
big-university 5551111
big-university 5551112
big-university 5551113
big-university 5551114
```

`tip` probiert jede der Nummern in der aufgelisteten Reihenfolge und gibt dann auf. Möchten Sie, dass `tip` beim Versuchen eine Verbindung herzustellen nicht aufgibt, lassen Sie es in einer `while` -Schleife laufen.

## 48.8. Eine Übertragung erzwingen

`Ctrl` + `P` ist das voreingestellte Zeichen, mit dem eine Übertragung erzwungen werden kann und wird benutzt, um `tip` zu sagen, dass das nächste Zeichen direkt gesendet werden soll und nicht als Fluchtzeichen interpretiert werden soll. Mit Hilfe der Fluchtsequenz `~s`, mit der man Variablen setzen kann, können Sie jedes andere Zeichen als "force"-Zeichen definieren.

Geben Sie `~sforce=Zeichen` gefolgt von `Enter` ein. Für *Zeichen* können Sie ein beliebiges einzelnes Zeichen einsetzen. Wenn Sie *Zeichen* weglassen, ist das "force"-Zeichen "nul", das Sie mit `Ctrl` + `2` oder `Ctrl` + `Leertaste` eingeben können. Ein guter Wert für *Zeichen* ist `Shift` + `Ctrl` + `6`, welches nur auf wenigen Terminal Servern benutzt wird.

Sie können das "force"-Zeichen auch bestimmen, indem Sie in $HOME/.tiprc das Folgende einstellen:

```
force=single-char
```

## 48.9. Großbuchstaben

Dies passiert, wenn `Ctrl` + `A` eingegeben wurde, das "raise"-Zeichen von `tip`, das speziell für Leute mit defekten caps-lock Tasten eingerichtet wurde. Benutzen Sie `~s` wie oben und setzen Sie die Variable `raisechar` auf etwas, das Ihnen angemessen erscheint. Tatsächlich kann die Variable auf das gleiche Zeichen wie das "force"-Zeichen gesetzt werden, wenn diese Fähigkeiten niemals benutzt werden sollen.

Hier ist ein Muster der .tiprc Datei für Emacs Benutzer, die `Ctrl` + `2` und `Ctrl` + `A` tippen müssen:

```
force=^^
raisechar=^^
```

Geben Sie für `^^` `Shift` + `Ctrl` + `6` ein.

## 48.10. Dateien mit `tip` übertragen

Wenn Sie mit einem anderen UNIX® System kommunizieren, können Sie mit `~p` (put) und `~t` (take) Dateien senden und empfangen. Diese Befehle lassen `cat` und `echo` auf dem entfernten System laufen, um Dateien zu empfangen und zu senden. Die Syntax ist:

`~p` local-file [ remote-file ]

`~t` remote-file [ local-file ]

Es gibt keine Fehlerkontrolle, deshalb sollte besser ein anderes Protokoll, wie zmodem, benutzt werden.

## 48.11. zmodem mit `tip` benutzen

Um Dateien zu empfangen, starten Sie das Programm zum Senden auf dem entfernten Computer. Geben Sie dann `~C rz` ein, um die Dateien lokal zu empfangen.

Um Dateien zu senden, starten Sie das Programm zum Empfangen auf dem entfernten Computer. Geben Sie dann `~C sz Dateien` ein, um Dateien auf das entfernte System zu senden.

# Kapitel 49. Einrichten der seriellen Konsole

FreeBSD kann ein System mit einem Dumb-Terminal (unintelligente Datenstation) an einer seriellen Schnittstelle als Konsole booten. Diese Konfiguration ist besonders nützlich für Systemadministratoren, die FreeBSD auf Systemen ohne Tastatur oder Monitor installieren wollen, und Entwickler, die den Kernel oder Gerätetreiber debuggen.

Wie in FreeBSDs Bootvorgang beschrieben, besitzt FreeBSD drei Bootphasen. Der Code für die ersten beiden Bootphasen befindet sich im Bootsektor am Anfang der FreeBSD-Slice der Bootplatte. Dieser Bootblock lädt den Bootloader in Phase drei.

Um eine serielle Konsole einzurichten, muss der Bootblock, der Bootloader und der Kernel konfiguriert werden.

## 49.1. Schnelle Konfiguration der seriellen Konsole

Dieser Abschnitt bietet einen schnellen Überblick über die Einrichtung einer seriellen Konsolen. Es wird vorausgesetzt, dass die Voreinstellungen verwendet werden.

1. Verbinden Sie die serielle Konsole mit COM1 sowie dem Kontrollterminal.
2. Um die Startmeldungen der seriellen Konsole zu sehen, geben Sie als `root` folgendes ein:

   ```
   ```

3. Ändern Sie in /etc/ttys den Eintrag für ttyu0 von `off` auf `on`. Zusätzlich sollten Sie den Wert `dialup` auf `vt100` ändern. Nur so wird auf der seriellen Konsole eine Eingabeaufforderung mit einer Passwortabfrage aktiviert.
4. Starten Sie nun das System neu, damit die serielle Konsole aktiviert wird.

Wenn Sie eine unterschiedliche Konfiguration benötigen, lesen Sie den nächsten Abschnitt für eine tiefer gehende Erklärung.

## 49.2. Konfiguration der seriellen Konsole

1. Bereiten Sie ein serielles Kabel vor.

   Sie benötigen entweder ein Nullmodemkabel oder ein serielles Standard Kabel mit einem Nullmodemkabel-Adapter. In Kabel und Schnittstellen werden serielle Kabel beschrieben.

2. Trennen Sie die Tastatur vom Computer.

   Viele PC Systeme suchen beim Power On Self Test (POST) nach einer Tastatur und geben eine Fehlermeldung aus, wenn sie keine finden. Einige Maschinen werden sich sogar weigern, ohne Tastatur zu booten.

   Wenn der Rechner trotz einer Fehlermeldung normal weiterbootet, brauchen Sie weiter nichts zu tun.

Wenn das System ohne Tastatur nicht booten will, müssen Sie das BIOS so konfigurieren, dass es diesen Fehler ignoriert (wenn das möglich ist). Das Handbuch zum Motherboard sollte beschreiben, wie das zu bewerkstelligen ist.



Selbst wenn Sie im BIOS "Not installed" für die Tastatur einstellen, können Sie eine Tastatur angeschlossen haben und diese auch weiterhin benutzen, da sie mit dieser Anweisung das BIOS lediglich anweisen, nach dem Einschalten des Rechners nicht nach einer Tastatur zu suchen und den Rechner ohne entsprechende Fehlermeldung zu starten. Wenn die oben beschriebene Option nicht im BIOS vorhanden ist, halten Sie stattdessen Ausschau nach einer "Halt on Error" Option. Sie können den gleichen Effekt wie oben erzielen, wenn Sie diese Option auf "All but Keyboard" oder sogar "No Errors" setzen.



Wenn das System über eine PS/2® Maus verfügt, müssen Sie diese wahrscheinlich auch abziehen. Da sich die PS/2® Maus und die Tastatur einige Hardwarekomponenten teilen, kann das dazu führen, dass die Hardwareerkennung fälschlicherweise eine Tastatur findet, wenn eine PS/2® Maus angeschlossen ist.

3. Schließen Sie einen Dumb-Terminal an COM1 (sio0) an.

   Wenn Sie keinen Dumb-Terminal besitzen, können Sie einen alten Computer mit einem Terminalemulator oder die serielle Schnittstelle eines anderen UNIX® Rechners benutzen. Sie benötigen auf jeden Fall eine freie erste serielle Schnittstelle (COM1). Zurzeit ist es nicht möglich, in den Bootblöcken eine andere Schnittstelle zu konfigurieren, ohne diese neu zu kompilieren. Wenn Sie COM1 bereits für ein anderes Gerät benutzen, müssen Sie dieses Gerät temporär entfernen und einen neuen Bootblock sowie Kernel installieren, wenn FreeBSD erst einmal installiert ist.

4. Stellen Sie sicher, dass die Kernelkonfiguration die richtigen Optionen für COM1 (sio0) enthält.

   Relevante Optionen sind:

   `0x10`

   Aktiviert die Konsolenunterstützung für dieses Gerät. Zurzeit kann nur ein Gerät die Konsolenunterstützung aktiviert haben. Das erste, in der Konfigurationsdatei aufgeführte Gerät, mit dieser Option, verfügt über eine aktivierte Konsolenunterstützung. Beachten Sie, dass diese Option alleine nicht ausreicht, um die serielle Konsole zu aktivieren. Setzen Sie entweder noch die nachfolgend diskutierte Option oder verwenden Sie beim Booten, wie unten beschrieben, den Schalter `-h`.

   `0x20`

   Das erste Gerät in der Kernelkonfigurationsdatei mit dieser Option wird, unabhängig von dem unten diskutierten Schalter `-h`, zur Konsole. Die Option `0x20` muss zusammen mit `0x10` verwendet werden.

`0x40`

Reserviert dieses Gerät und sperrt es für normale Zugriffe. Sie sollten diese Option nicht auf dem Gerät setzen, das Sie als serielle Konsole verwenden wollen. Der Zweck dieser Option ist es, dieses Gerät für das Remote-Debuggen zu reservieren. Das FreeBSD Developers' Handbook enthält dazu weitere Informationen.

Beispiel:

```
device sio0 at isa? port IO_COM1 tty flags 0x10 irq 4
```

Weitere Einzelheiten finden Sie in sio(4).

Wenn diese Optionen nicht gesetzt sind, müssen Sie auf einer anderen Konsole beim Booten UserConfig starten oder den Kernel neu kompilieren.

5. Erstellen Sie boot.config im Rootverzeichnis der `a`-Partition des Bootlaufwerks.

   Der Code des Bootblocks entnimmt dieser Datei, wie Sie Ihr System booten möchten. Um die serielle Konsole zu aktivieren, müssen Sie hier eine oder mehrere Optionen (alle in derselben Zeile) angeben. Die folgenden Optionen stehen zur Auswahl der Konsole zur Verfügung:

   `-h`

   Schaltet zwischen der internen und der seriellen Konsole um. Wenn Sie beispielsweise von der internen Konsole (Bildschirm) booten, weist `-h` den Bootloader und den Kernel an, die serielle Schnittstelle als Konsole zu nehmen. Wenn die Konsole normal auf der seriellen Schnittstelle liegt, wählen Sie mit `-h` den Bildschirm aus.

   `-D`

   Schaltet zwischen Einzelkonsole und Dual-Konsole um. Die Einzelkonsole ist entweder die interne Konsole (der Bildschirm) oder die serielle Schnittstelle, je nach dem Stand von `-h`. Im Dual-Konsolen Betrieb ist die Konsole, unabhängig von `-h`, gleichzeitig der Bildschirm und die serielle Schnittstelle. Dies trifft aber nur zu, wenn der Bootblock ausgeführt wird. Sobald der Bootloader ausgeführt wird, wird die durch `-h` gegebene Konsole die alleinige Konsole.

   `-P`

   Veranlasst den Bootblock nach einer Tastatur zu suchen. Wenn keine Tastatur gefunden wird, werden `-D` und `-h` automatisch gesetzt.

   Wegen Platzbeschränkungen in den Bootblöcken kann `-P` nur erweiterte Tastaturen erkennen. Tastaturen mit weniger als 101 Tasten und ohne F11 und F12 Tasten werden wahrscheinlich, wie vielleicht auch die Tastaturen einiger Laptops, nicht erkannt. Wenn das der Fall ist, können Sie `-P` nicht verwenden, da es leider keine Abhilfe für dieses Problem gibt.

   Benutzen Sie also entweder `-P`, um die Konsole automatisch zu setzen, oder `-h`, um die

serielle Konsole zu verwenden.

Weitere Optionen werden in boot(8) beschrieben.

Mit Ausnahme von -P werden die Optionen an den Bootloader weitergegeben. Der Bootloader untersucht dann einzig -h um festzustellen, welches Gerät die Konsole wird. Wenn Sie also nur -D angegeben haben, können Sie die serielle Schnittstelle nur als Konsole verwenden während der Bootblock ausgeführt wird. Danach wird der Bootloader, da ja -h fehlt, den Bildschirm zur Konsole machen.

6. Booten Sie die Maschine.

   Wenn Sie das FreeBSD-System starten, werden die Bootblöcke den Inhalt von /boot.config auf der Konsole ausgeben:

   ```
   /boot.config: -P
   Keyboard: no
   ```

   Die zweite Zeile sehen Sie nur, wenn Sie in /boot.config-P angegeben haben. Sie zeigt an, ob eine Tastatur angeschlossen ist oder nicht. Die Meldungen gehen je nach den Einstellungen in /boot.config auf die interne Konsole, die serielle Konsole, oder beide Konsolen.

| **Optionen** | **Meldungen erscheinen auf** |
|---|---|
| keine | der internen Konsole |
| -h | der seriellen Konsole |
| -D | der seriellen und der internen Konsole |
| -Dh | der seriellen und der internen Konsole |
| -P, mit Tastatur | der internen Konsole |
| -P, ohne Tastatur | der seriellen Konsole |

   Nach den oben gezeigten Meldungen gibt es eine kleine Verzögerung bevor die Bootblöcke den Bootloader laden und weitere Meldungen auf der Konsole erscheinen. Sie können die Ausführung der Bootblöcke unterbrechen, um zu überprüfen, ob auch alles richtig aufgesetzt ist, brauchen das aber unter normalen Umständen nicht zu tun.

   Drücken Sie eine Taste außer `Enter` um den Bootvorgang zu unterbrechen. Sie erhalten dann ein Prompt, an dem Sie weitere Eingaben tätigen können:

   ```
   FreeBSD/i386 BOOT
   Default: 0:ad(0,a)/boot/loader
   boot:
   ```

   Je nach Inhalt von /boot.config erscheint das Prompt auf der seriellen Konsole, der internen Konsole oder beiden Konsolen. Wenn die Meldung auf der richtigen Konsole

erscheint, drücken Sie `Enter` um fortzufahren.

Wenn kein Prompt auf der seriellen Konsole erscheint, liegt ein Fehler in den Einstellungen vor. Als Abhilfe geben Sie an der momentanen Konsole `-h` ein, um den Bootblock und den Bootloader auf die serielle Konsole umzustellen. Führen Sie dann den Bootvorgang mit `Enter` weiter und wenn das System gebootet hat, können Sie die fehlerhaften Einstellungen korrigieren.

Während der dritten Bootphase können Sie immer noch zwischen der internen und der seriellen Konsole auswählen. Setzen Sie dazu, wie in Die Konsole im Bootloader ändern beschrieben, die entsprechenden Variablen des Bootloaders.

## 49.3. Zusammenfassung

Die folgende Tabelle bietet eine Zusammenfassung der verschiedenen Einstellungen, die in diesem Abschnitt diskutiert wurden:

*Tabelle 23. Fall 1: Option 0x10 für sio0*

| Optionen in /boot.config | Konsole in den Bootblöcken | Konsole im Bootloader | Konsole im Kernel |
|---|---|---|---|
| keine | interne | interne | interne |
| `-h` | serielle | serielle | serielle |
| `-D` | serielle und interne | interne | interne |
| `-Dh` | serielle und interne | serielle | serielle |
| `-P`, mit Tastatur | interne | interne | interne |
| `-P`, ohne Tastatur | serielle und interne | serielle | serielle |

*Tabelle 24. Fall 2: Option 0x30 für sio0*

| Optionen in /boot.config | Konsole in den Bootblöcken | Konsole im Bootloader | Konsole im Kernel |
|---|---|---|---|
| keine | interne | interne | serielle |
| `-h` | serielle | serielle | serielle |
| `-D` | serielle und interne | interne | serielle |
| `-Dh` | serielle und interne | serielle | serielle |
| `-P`, mit Tastatur | interne | interne | serielle |
| `-P`, ohne Tastatur | serielle und interne | serielle | serielle |

## 49.4. Hinweise zur seriellen Konsole

### 49.4.1. Verwenden einer höheren Geschwindigkeit

Die Vorgabewerte für die Kommunikationsparameter der seriellen Schnittstelle sind: 9600 baud, 8 Bit, keine Parität und ein Stopp-Bit. Um die Standardgeschwindigkeit zu ändern, stehen folgende Möglichkeiten zur Verfügung:

- Geben Sie die neue Konsolengeschwindigkeit mit `BOOT_COMCONSOLE_SPEED` an und kompilieren Sie die Bootblöcke neu. Ausführliche Informationen zum Bau und zur Installation von neuen Bootblöcken finden Sie im Eine andere Schnittstelle als sio0 benutzen des Handbuchs.

  Wenn die serielle Konsole nicht mit der Option `-h` gestartet wird, oder wenn die verwendete serielle Konsole sich von der von den Bootblöcken verwendeten unterscheidet, müsssen Sie zusätzlich die folgende Option in die Kernelkonfigurationsdatei aufnehmen und den Kernel neu bauen:

  ```
  options CONSPEED=19200
  ```

- Verwenden Sie die Option `-S`, um den Kernel zu booten. Eine Beschreibung dieses Vorgangs sowie eine Auflistung der von /boot.config unterstützten Optionen finden Sie in boot(8).
- Aktivieren Sie die Option `comconsole_speed` in /boot/loader.conf.

  Diese Option setzt voraus, dass auch die Optionen `console`, `boot_serial`, sowie `boot_multicons` in /boot/loader.conf gesetzt sind. Im Folgenden finden Sie ein Beispiel, in dem `comconsole_speed` verwendet wird, um die Geschwindigkeit der seriellen Konsole zu ändern:

  ```
  boot_multicons="YES"
  boot_serial="YES"
  comconsole_speed="115200"
  console="comconsole,vidconsole"
  ```

### 49.4.2. Eine andere Schnittstelle als sio0 benutzen

Wenn Sie, warum auch immer, ein anderes Gerät als sio0 für die serielle Konsole einsetzen wollen, kompilieren Sie bitte die Bootblöcke, den Bootloader und den Kernel nach dem folgenden Verfahren neu.

1. Installieren Sie die Kernelquellen wie im FreeBSD aktualisieren beschrieben.
2. Setzen Sie in /etc/make.conf`BOOT_COMCONSOLE_PORT` auf die Adresse der Schnittstelle (0x3F8, 0x2F8, 0x3E8 oder 0x2E8), die Sie benutzen möchten. Sie können nur sio0 bis sio3 (COM1 bis COM4) benutzen, Multiportkarten können Sie nicht als Konsole benutzen. Interrupts müssen Sie hier nicht angeben.
3. Erstellen Sie eine angepasste Kernelkonfiguration und geben Sie dort die richtigen Optionen für die Schnittstelle, die Sie benutzen möchten, an. Wenn Sie zum Beispiel sio1 (COM2) zur Konsole machen wollen, geben Sie dort entweder

```
device sio1 at isa? port IO_COM2 tty flags 0x10 irq 3
```

oder

```
device sio1 at isa? port IO_COM2 tty flags 0x30 irq 3
```

an. Keine andere serielle Schnittstelle sollte als Konsole definiert werden.

4. Übersetzen und installieren Sie die Bootblöcke und den Bootloader:

```
# cd /sys/boot
# make clean
# make
# make install
```

5. Bauen und installieren Sie einen neuen Kernel.
6. Schreiben Sie die Bootblöcke mit bsdlabel(8) auf die Bootplatte und booten Sie den neuen Kernel.

### 49.4.3. DDB Debugger über die serielle Schnittstelle

Wenn Sie den Kerneldebugger über eine serielle Verbindung bedienen möchten, übersetzen Sie einen angepassten Kernel mit den folgenden Optionen. Das ist nützlich, kann aber gefährlich sein, wenn auf der Leitung falsche BREAK-Signale generiert werden.

```
options BREAK_TO_DEBUGGER
options DDB
```

### 49.4.4. Benutzung der seriellen Konsole zum Anmelden

Da Sie schon die Bootmeldungen auf der Konsole verfolgen können und den Kerneldebugger über die Konsole bedienen können, wollen Sie sich vielleicht auch an der Konsole anmelden.

Öffnen Sie /etc/ttys in einem Editor und suchen Sie nach den folgenden Zeilen:

```
ttyu0 "/usr/libexec/getty std.9600" unknown off secure
ttyu1 "/usr/libexec/getty std.9600" unknown off secure
ttyu2 "/usr/libexec/getty std.9600" unknown off secure
ttyu3 "/usr/libexec/getty std.9600" unknown off secure
```

ttyu0 bis ttyu3 entsprechen COM1 bis COM4. Ändern Sie für die entsprechende Schnittstelle `off` zu `on`. Wenn Sie auch die Geschwindigkeit der seriellen Schnittstelle geändert haben, müssen Sie `std.9600` auf die momentane Geschwindigkeit anpassen.

Auch kann den Terminaltyp von `unknown` auf den tatsächlich verwendeten Terminal gesetzt werden.

Damit die Änderungen wirksam werden, müssen Sie noch `kill -HUP 1` absetzen.

# 49.5. Die Konsole im Bootloader ändern

In den vorigen Abschnitten wurde beschrieben, wie Sie die serielle Konsole durch Änderungen im Bootblock aktivieren. Dieser Abschnitt zeigt, wie Sie mit Kommandos und Umgebungsvariablen die Konsole im Bootloader definieren. Da der Bootloader die dritte Phase im Bootvorgang ist und nach den Bootblöcken ausgeführt wird, überschreiben seine Einstellungen die des Bootblocks.

## 49.5.1. Festlegen der Konsole

Mit einer einzigen Zeile in /boot/loader.conf können Sie den Bootloader und den Kernel anweisen, die serielle Schnittstelle zur Konsole zu machen:

```
console="comconsole"
```

Unabhängig von den Einstellungen im Bootblock legt dies die Konsole fest.

Die obige Zeile sollte die erste Zeile in /boot/loader.conf sein, so dass die Bootmeldungen so früh wie möglich auf der Konsole zu sehen sind.

Analog können Sie die interne Konsole verwenden:

```
console="vidconsole"
```

Wenn die Umgebungsvariable `console` nicht gesetzt ist, bestimmt der Bootloader und damit auch der Kernel, die Konsole über die `-h` Option des Bootblocks.

Die Bootkonsole kann in /boot/loader.conf.local oder /boot/loader.conf angegeben werden.

Weitere Informationen erhalten Sie in loader.conf(5).



Momentan gibt es im Bootloader nichts vergleichbares zu `-P` im Bootblock. Damit kann die Konsole nicht automatisch über das Vorhandensein einer Tastatur festgelegt werden.

## 49.5.2. Eine andere Schnittstelle als sio0 benutzen

Der Bootloader muss neu kompiliert werden, wenn eine andere Schnittstelle als sio0 benutzt werden soll. Folgen Sie der Anleitung aus Eine andere Schnittstelle als sio0 benutzen.

# 49.6. Vorbehalte

Obwohl es die meisten Systeme erlauben, ohne Tastatur zu booten, gibt es nur wenige Systeme, die

ohne eine Grafikkarte booten. Maschinen mit einem AMI BIOS können ohne Grafik booten, indem Sie den Grafikadapter im CMOS-Setup auf `Not installed` setzen.

Viele Maschinen unterstützen diese Option allerdings nicht. Damit diese Maschinen booten, müssen sie über eine Grafikkarte, auch wenn es nur eine alte Monochromkarte ist, verfügen. Allerdings brauchen Sie keinen Monitor an die Karte anzuschließen. Sie können natürlich auch versuchen, auf diesen Maschinen ein AMI BIOS zu installieren.

# Kapitel 50. PPP

## 50.1. Übersicht

FreeBSD unterstützt das Point-to-Point (PPP) Protokoll, mit dem über ein Modem eine Verbindung mit einem Netzwerk oder dem Internet hergestellt werden kann. Dieses Kapitel beschreibt die Konfiguration von Modem-basierten Kommunikationsdiensten unter FreeBSD.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie wissen:

- Wie Sie PPP einrichten, benutzen, sowie Fehler beheben.
- Was zu tun ist, um PPP over Ethernet (PPPoE) einzurichten.
- Wie Sie PPP over ATM (PPPoA) einrichten.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Mit den grundlegenden Begriffen der Netzwerktechnik vertraut sein.
- Die Grundlagen und den Zweck einer Einwahlverbindung sowie PPP kennen.

## 50.2. PPP konfigurieren

FreeBSD enthält ppp(8), um Einwählverbindungen über PPP zu verwalten. Der FreeBSD-Kernel enthält Unterstützung für die tun-Schnittstelle, die benutzt wird um mit einem Modem zu interagieren. Für die Konfiguration muss mindestens eine Datei bearbeitet werden. Beispiele sind in den Konfigurationsdateien ebenfalls enthalten. Schlussendlich wird `ppp` benutzt, um die Verbindungen zu starten und zu verwalten.

Für eine PPP-Verbindung sind folgende Dinge erforderlich:

- Ein Account bei einem Internet Service Provider (ISP).
- Ein Modem.
- Die Einwahlnummer(n) des ISPs.
- Den Login-Namen und das Passwort, welches vom ISP zugewiesen wurde.
- Die IP-Adresse von einem oder mehreren DNSServern. Üblicherweise werden diese Daten vom ISP zur Verfügung gestellt. Falls dies nicht der Fall ist, können Sie FreeBSD so konfigurieren, das es die DNS-Daten automatisch aushandeln kann.

Sollte eine dieser Informationen fehlen, kontaktieren Sie den ISP!

Die folgenden Informationen werden möglicherweise durch den ISP zur Verfügung gestellt, sie sind aber nicht zwingend erforderlich:

- Die IP-Adresse des Standard-Gateways. Steht diese Information nicht zur Verfügung, wird der PPP-Server des ISPs beim Verbindungsaufbau eine gültige Adresse übermitteln. Diese Adresse wird in der Konfiguration von PPP unter FreeBSD als `HISADDR` bezeichnet.

- Die Netzmaske. Falls der ISP keine Netzmaske vorgegeben hat, können Sie in der Konfigurationsdatei von ppp(8) `255.255.255.255` verwenden. *

  Wenn der ISP eine statische IP-Adresse und einen Rechnernamen zugewiesen hat, sollten diese Informationen in die Konfigurationsdatei eingetragen werden. Andernfalls werden diese Informationen automatisch beim Verbindungsaufbau zur Verfügung gestellt.

Der Rest dieses Abschnitts beschreibt, wie FreeBSD für gebräuchliche PPP-Verbindungsszenarien konfiguriert wird. Die erforderliche Konfigurationsdatei ist /etc/ppp/ppp.conf. Zusätzliche Dateien und Beispiele sind in /usr/shared/examples/ppp/ verfügbar.



Die Beispieldateien, die in diesem Kapitel dargestellt werden, enthalten Zeilennummern. Die Nummerierung dient lediglich einer leichteren Orientierung und sollte nicht in die Dateien übernommen werden.

Achten Sie auf die richtige Einrückung, wenn Sie eine Konfigurationsdatei bearbeiten. Zeilen die mit einem `:` enden, beginnen in der ersten Spalte (am Beginn der Zeile). Alle anderen Zeilen sollten wie dargestellt durch Leerzeichen oder Tabulatoren eingerückt werden.

### 50.2.1. Grundlegende Konfiguration

Um eine PPP-Verbindung zu konfigurieren, tragen Sie zuerst die Zugangsdaten des ISPs in /etc/ppp/ppp.conf ein. Diese Datei wird wie folgt beschrieben:

```
1     default:
2       set log Phase Chat LCP IPCP CCP tun command
3       ident user-ppp VERSION
4       set device /dev/cuau0
5       set speed 115200
6       set dial "ABORT BUSY ABORT NO\\sCARRIER TIMEOUT 5 \
7                 \"\" AT OK-AT-OK ATE1Q0 OK \\dATDT\\T TIMEOUT 40 CONNECT"
8       set timeout 180
9       enable dns
10
11    provider:
12      set phone "(123) 456 7890"
13      set authname foo
14      set authkey bar
15      set timeout 300
16      set ifaddr x.x.x.x/0 y.y.y.y/0 255.255.255.255 0.0.0.0
17      add default HISADDR
```

**Zeile 1**

Gibt den Standardeintrag an. Befehle dieses Eintrags (Zeile 2 bis 9) werden automatisch ausgeführt, wenn `ppp` läuft.

**Zeile 2**

Schaltet die ausführliche Protokollierung ein. Sobald die Verbindung zufriedenstellend funktioniert, können Sie diese Zeile verkürzen:

```
set log phase tun
```

Dies verhindert ein übermäßiges Anwachsen der Logdateien.

**Zeile 3**

Übermittelt die Version von ppp(8) an die PPP-Software der Gegenstelle.

**Zeile 4**

Gibt das Device an, an dem das Modem angeschlossen ist. COM1 entspricht /dev/cuad0 und COM2 entspricht /dev/cuad1.

**Zeile 5**

Legt die Verbindungsgeschwindigkeit fest. Falls ein Wert von `115200` bei älteren Modems nicht funktioniert, versuchen Sie es stattdessen mit `38400`.

**Zeile 6 & 7**

Die Zeichenfolge für die Einwahl in einer expect-send Syntax. Weitere Informationen finden Sie in chat(8).

Beachten Sie, dass dieser Befehl aufgrund der besseren Lesbarkeit auf der nächsten Zeile weitergeht. Das kann für jeden Befehl in ppp.conf gelten, wenn `\` das letzte Zeichen in einer Zeile ist.

**Zeile 8**

Legt den Zeitrahmen in Sekunden fest, innerhalb dessen eine Reaktion erfolgen muss.

**Zeile 9**

Weist die Gegenstelle an, die DNS-Einstellungen zu bestätigen. Wenn es im lokalen Netzwerk einen DNS-Server gibt, sollte diese Zeile auskommentiert oder gelöscht werden.

**Zeile 10**

Eine leere Zeile zur besseren Lesbarkeit. Leere Zeilen werden von ppp(8) ignoriert.

**Zeile 11**

Bestimmt einen Provider, namens `provider`. Wenn Sie hier den Namen des ISP einsetzen, können Sie später die Verbindung mit `load ISP` aufbauen.

**Zeile 12**

Gibt die Telefonnummer des Providers an. Mehrere Telefonnummern können angegeben werden, indem Doppelpunkte (`:`) oder Pipe-Zeichen (`|`) als Trennzeichen verwendet werden. Wenn Sie die verschiedenen Nummern abwechselnd verwenden möchten, sollten Sie die Nummern durch einen Doppelpunkt trennen. Wenn Sie immer die erste Nummer verwenden möchten und die anderen nur zum Einsatz kommen sollen, wenn eine Einwahl mit der ersten

Telefonnummer nicht möglich ist, sollten Sie das Pipe-Zeichen zur Trennung verwenden. Sie sollten immer die gesamte Reihe der Telefonnummern in Anführungszeichen (") setzen, um Wählfehler zu vermeiden.

**Zeile 13 & 14**

Gibt den Benutzernamen und das Passwort für den ISP an.

**Zeile 15**

Setzt einen Zeitrahmen in Sekunden, innerhalb dessen eine Reaktion erfolgen muss. In diesem Fall, wird die Verbindung nach 300 Sekunden automatisch geschlossen, wenn keine Aktivität zu verzeichnen ist. Wenn Sie keinen Zeitrahmen festlegen wollen, nach dessen Überschreiten die Verbindung geschlossen wird, können Sie diesen Wert auf `0` setzen.

**Zeile 16**

Legt die Adresse für die Schnittstelle fest. Die verwendeten Werte hängen davon ab, ob Sie vom ISP eine statische IP-Adresse zugeteilt bekommen haben, oder ob beim Verbindungsaufbau eine dynamische Adresse ausgehandelt wird.

Wenn Ihnen der ISP keine statische IP-Adresse zugeteilt hat, ändern Sie diese Zeile auf den folgenden Wert. Dadurch weiß ppp(8), dass es das IP Configuration Protocol (IPCP) benutzen soll um die dynamische IP-Adresse auszuhandeln.

```
set ifaddr 10.0.0.1/0 10.0.0.2/0 255.255.255.255 0.0.0.0
```

**Zeile 17**

Fügt eine Defaultroute für das Gateway hinzu. Belassen Sie die Zeile so wie sie ist. `HISADDR` wird dabei durch die in Zeile 16 angegebene Gateway-Adresse ersetzt. Wichtig ist, dass diese Zeile nach Zeile 16 erscheint.

Je nachdem, ob ppp(8) manuell oder automatisch gestartet wird, muss vielleicht auch /etc/ppp/ppp.linkup mit dem folgenden Inhalt erstellt werden. Diese Datei ist erforderlich, falls `ppp` im `-auto`-Modus ausgefürht wird. Die Datei wird verwendet, nachdem die Verbindung hergestellt wurde. An diesem Punkt wird die IP-Adresse zugewiesen und es sollte nun möglich sein, Einträge in die Routingtabelle hinzuzufügen. Stellen Sie bei der Bearbeitung der Datei sicher, dass der Eintrag für *provider* mit dem Wert aus Zeile 11 in ppp.conf übereinstimmt.

```
provider:
      add default HISADDR
```

Diese Datei wird ebenfalls benötigt, wenn bei einer Konfiguration mit statischer IP-Adresse die Adresse des Standard-Gateways "erraten" wird. In solchen Fällen entfernen Sie Zeile 17 aus ppp.conf und erstellen Sie /etc/ppp/ppp.linkup mit den oben genannten Zeilen. Weitere Beispiele für diese Datei finden Sie in /usr/shared/examples/ppp/.

In der Voreinstellung muss `ppp` als `root` ausgeführt werden. Um diesen Standard zu ändern, muss das Konto eines Benutzers, der `ppp` ausführen soll, zur Gruppe `network` in /etc/group hinzugefügt werden.

Danach geben Sie dem Benutzer ebenfalls Zugriff auf einen oder mehrere Abschnitte der Konfigurationsdatei /etc/ppp/ppp.conf geben müssen, indem Sie den `allow` Befehl verwenden. Um beispielsweise den Benutzern `fred` und `mary` die Berechtigung für den Eintrag `provider:` zu geben, fügen Sie in der Sektion `provider` folgende Zeile ein:

```
allow users fred mary
```

Wenn dieser Befehl stattdessen in der Sektion `default` verwendet wird, erhalten die angegebenen Benutzer vollständigen Zugriff.

### 50.2.2. Fortgeschrittene Konfiguration

Es ist möglich PPP so zu konfigurieren, dass bei Bedarf DNS und NetBIOS Nameserveradressen bereitgestellt werden.

Um diese Erweiterungen für die PPP Version 1.x zu aktivieren, sollte der entsprechende Abschnitt der Datei /etc/ppp/ppp.conf um folgende Zeilen ergänzt werden:

```
enable msext
set ns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Für PPP Version 2 und höher:

```
accept dns
set dns 203.14.100.1 203.14.100.2
set nbns 203.14.100.5
```

Damit werden den Clients die primären und sekundären Nameserveradressen sowie ein NetBIOS Nameserver-Host mitgeteilt.

In Version 2 und höher verwendet PPP die Werte, die in /etc/resolv.conf zu finden sind, wenn die Zeile `set dns` weggelassen wird.

#### Authentifizierung durch PAP und CHAP

Einige ISPs haben ihr System so eingerichtet, dass der Authentifizierungsteil eines Verbindungsaufbaus mit Hilfe von PAP oder CHAP-Mechanismen durchgeführt wird. Wenn das der Fall sein sollte, wird der ISP bei der Verbindung keinen `login:`-Prompt präsentieren, sondern sofort mit der Aushandlung der PPP-Verbindung beginnen.

PAP ist nicht so sicher wie CHAP, doch die Sicherheit ist hierbei normalerweise kein Problem, da Passwörter, obgleich von PAP im Klartext versandt, lediglich über die serielle Verbindung verschickt werden. Es gibt für Angreifer wenig Möglichkeiten zu "lauschen".

Die folgenden Veränderungen müssen vorgenommen werden:

```
13      set authname MyUserName
14      set authkey MyPassword
15      set login
```

**Zeile 13**

Diese Zeile legt den PAP/CHAP Benutzernamen fest. Sie müssen den richtigen Wert für *MyUserName* eingeben.

**Zeile 14**

Diese Zeile legt das PAP/CHAP Passwort fest. Sie müssen den richtigen Wert für *MyPassword* eingeben. Sie können eine zusätzliche Zeile, wie etwa:

```
16      accept PAP
```

oder

```
16      accept CHAP
```

verwenden, um deutlich zu machen, dass dies beabsichtigt ist, aber sowohl PAP wie auch CHAP als standardmäßig akzeptiert werden.

**Zeile 15**

Der ISP wird normalerweise keine Anmeldung am Server verlangen, wenn PAP oder CHAP verwendet wird. Sie müssen deshalb den String "set login" deaktivieren.

### PPP NAT benutzen

PPP kann Network Address Translation (NAT) ohne Hilfe des Kernels durchführen. Wenn Sie diese Funktion benutzen wollen, fügen Sie die folgende Zeile in /etc/ppp/ppp.conf ein:

```
nat enable yes
```

NAT kann mit der Option `-nat` auf der Kommandozeile aktiviert werden. Weiterhin kann NAT in /etc/rc.conf mit der Variablen `ppp_nat` aktiviert werden. Dies ist auch die Voreinstellung.

Die nachstehende /etc/ppp/ppp.conf benutzt NAT für bestimmte eingehende Verbindungen:

```
nat port tcp 10.0.0.2:ftp ftp
nat port tcp 10.0.0.2:http http
```

Wenn Sie Verbindungen von außen überhaupt nicht trauen, benutzen Sie die folgende Zeile:

```
nat deny_incoming yes
```

### 50.2.3. Abschließende Systemkonfiguration

Obwohl `ppp` nun konfiguriert ist, müssen noch einige Änderungen in /etc/rc.conf vorgenommen werden.

Gehen Sie diese Datei von oben nach unten durch, und stellen Sie als Erstes sicher, dass die Zeile `hostname=` vorhanden ist:

```
hostname="foo.example.com"
```

Wenn der ISP eine statische IP-Adresse und einen Namen zugewiesen hat, verwenden Sie diesen Namen als Hostnamen.

Schauen Sie nach der Variable `network_interfaces`. Wenn Sie das System so konfigurieren möchten, dass es bei Bedarf eine Verbindung zum ISP aufbaut, sollten Sie das Gerät tun0 zu der Liste hinzufügen oder es andernfalls entfernen.

```
network_interfaces="lo0 tun0"
ifconfig_tun0=
```



Die Variable `ifconfig_tun0` sollte leer sein und eine Datei namens /etc/start_if.tun0 sollte erstellt werden. Diese Datei sollte die nachfolgende Zeile enthalten:

```
ppp -auto mysystem
```

Dieses Skript startet den ppp-Daemon im Automatik-Modus. Es wird bei der Netzwerkkonfiguration ausgeführt. Wenn der Rechner als Gateway für ein LAN fungiert, möchten Sie vielleicht auch die Option `-alias` verwenden. In der Manualpage sind weitere Einzelheiten zu finden.

Stellen Sie sicher, dass der Start eines Routerprogramms in /etc/rc.conf wie folgt deaktiviert ist:

```
router_enable="NO"
```

Es ist wichtig, dass der `routed`-Daemon nicht gestartet wird da `routed` dazu tendiert, die von `ppp` erstellten Einträge der Standardroute zu überschreiben.

Es ist außerdem sinnvoll, darauf zu achten, dass die Zeile `sendmail_flags` nicht die Option `-q` enthält, da `sendmail` sonst ab und zu die Netzwerkverbindung prüfen wird, was möglicherweise dazu führt, dass sich der Rechner einwählt. Sie können hier Folgendes angeben:

```
sendmail_flags="-bd"
```

Der Nachteil dieser Lösung ist, dass Sie `sendmail` nach jedem Aufbau einer ppp-Verbindung

auffordern müssen, die Mailwarteschlange zu überprüfen. Verwenden Sie den Befehl `!bg` in ppp.linkup, um dies zu automatisieren:

```
1     provider:
2       delete ALL
3       add 0 0 HISADDR
4       !bg sendmail -bd -q30m
```

Alternativ ist es möglich, einen "dfilter" einzusetzen, um SMTP-Verkehr zu blockieren. Weitere Einzelheiten hierzu finden Sie in den Beispieldateien.

### 50.2.4. `ppp` benutzen

Das Einzige, was nun noch zu tun bleibt, ist den Rechner neu zu starten. Nach dem Neustart können Sie entweder:

```
# ppp
```

und danach `dial provider` eingeben, um eine PPP-Sitzung zu starten, oder Sie geben:

```
# ppp -auto provider
```

ein, um `ppp` bei Datenverkehr aus dem Netzwerk heraus, automatisch eine Verbindung herstellen zu lassen (vorausgesetzt Sie haben kein start_if.tun0 Skript erstellt).

Es ist möglich, dem Programm `ppp` Befehle zu erteilen, während es im Hintergrund läuft. Dazu ist jedoch die Einrichtung eines passenden Diagnose-Ports erforderlich. Ergänzen Sie hierzu die Konfigurationsdatei um folgende Zeile:

```
set server /var/run/ppp-tun%d DiagnosticPassword 0177
```

Damit wird PPP angewiesen, auf den angegebenen UNIX®-Domainsocket zu hören und Clients nach dem angegebenen Passwort zu fragen, bevor der Zugang gewährt wird. Das `%d` wird durch die Nummer des benutzten tun-Devices ersetzt.

Wenn ein Socket eingerichtet ist, kann das Programm pppctl(8) in Skripten verwendet werden, mit denen in das laufende Programm eingegriffen wird.

### 50.2.5. Einwählverbindungen konfigurieren

“Einwählverbindungen” bietet eine gute Beschreibung, wie Einwählverbindungen unter Verwendung von getty(8) genutzt werden können.

Eine Alternative zu `getty` ist comms/mgetty+sendfax, eine raffiniertere Version von `getty`, die mit Blick auf Einwählverbindungen entworfen wurde.

Der Vorteil von mgetty ist, dass es auf aktive Weise mit Modems *spricht*, das heißt wenn ein Port in /etc/ttys ausgeschaltet ist, wird das Modem nicht auf Anrufe reagieren.

Spätere Versionen von mgetty (von 0.99beta aufwärts) unterstützen auch die automatische Erkennung von PPP-Streams, was Clients den skriptlosen Zugang zum Server erlaubt.

http://mgetty.greenie.net/doc/mgetty_toc.html enthält weitere Informationen zu mgetty.

In der Voreinstellung wird comms/mgetty+sendfax mit der Option AUTO_PPP konfiguriert und kompiliert. Dadurch kann mgetty die LCP Phase von PPP-Verbindungen erkennen und automatisch eine ppp-Shell starten. Da hierbei jedoch die Login/Passwort-Sequenz nicht durchlaufen wird, ist es notwendig, Benutzer durch PAP oder CHAP zu authentifizieren.

In diesem Abschnitt wird davon ausgegangen, dass der Benutzer den Port comms/mgetty+sendfax auf seinem System kompiliert und installiert hat.

Stellen Sie sicher, dass /usr/local/etc/mgetty+sendfax/login.config Folgendes enthält:

```
/AutoPPP/ -     - /etc/ppp/ppp-pap-dialup
```

Hierdurch wird mgetty angewiesen, ppp-pap-dialup für die erkannten PPP-Verbindungen auszuführen.

Erstellen Sie eine ausführbare Datei namens /etc/ppp/ppp-pap-dialup mit folgendem Inhalt:

```
#!/bin/sh
exec /usr/sbin/ppp -direct pap$IDENT
```

Erstellen Sie bitte für jede Einwählverbindung, die Sie in /etc/ttys ermöglicht haben, einen korrespondierenden Eintrag in der Datei /etc/ppp/ppp.conf. Diese Einträge können problemlos, mit den Definitionen die weiter oben gemacht wurden, koexistieren.

```
pap:
  enable pap
  set ifaddr 203.14.100.1 203.14.100.20-203.14.100.40
  enable proxy
```

Jeder Benutzer, der sich auf diese Weise anmeldet, benötigt einen Benutzernamen und ein Passwort in der Datei /etc/ppp/ppp.secret. Sie haben auch die Möglichkeit, Benutzer mit Hilfe von PAP zu authentifizieren, indem Sie in /etc/passwd folgende Option hinzufügen:

```
enable passwdauth
```

Um bestimmten Benutzern eine statische IP-Adresse zuzuweisen, können Sie die Adresse als drittes Argument in /etc/ppp/ppp.secret angeben. Beispiele finden Sie in /usr/shared/examples/ppp/ppp.secret.sample.

## 50.3. Probleme bei PPP-Verbindungen

Dieser Abschnitt behandelt Probleme, die auftauchen können, wenn PPP über ein Modem verwendet wird. Einige ISPs verwenden `ssword`, andere verwenden `password`. Wenn das Einwahlskript falsch ist, scheitert die Anmeldung. Üblicherweise suchen Sie nach Fehlern der PPP-Verbindung indem Sie sich manuell verbinden.

### 50.3.1. Gerätedateien überprüfen

Wenn Sie einen eigenen Kernel verwenden, stellen Sie sicher, dass die folgende Zeile in der Kernelkonfigurationsdatei vorhanden ist:

```
device   uart
```

Das uart-Gerät ist bereits im `GENERIC`-Kernel vorhanden, deshalb sind in diesem Fall keine zusätzlichen Schritte vonnöten. Kontrollieren Sie die Ausgabe von `dmesg`:

```
# dmesg | grep uart
```

In der Ausgabe sollten die entsprechenden uart-Geräte, beispielsweise uart1 (COM2), angezeigt werden. Wird ein passendes Gerät angezeigt, braucht der Kernel nicht neu erstellt werden. Wenn das Modem an uart1 angeschlossen ist, ist /dev/cuau1 die dazugehörende Gerätedatei.

### 50.3.2. Manuelle Verbindungen

Ein Verbindungsaufbau zum Internet durch manuelle Steuerung von `ppp` geht schnell, ist einfach und stellt einen guten Weg dar, eine Verbindung auf Fehler hin zu überprüfen oder einfach Informationen darüber zu sammeln, wie der ISP Verbindungen handhabt. Lassen Sie uns PPP von der Kommandozeile aus starten. Beachten Sie, dass in allen Beispielen *example* der Hostname der Maschine ist, auf der PPP läuft. `ppp` starten Sie wie folgt:

```
# ppp
```

```
ppp ON example> set device /dev/cuau1
```

Mit dem zweiten Befehl wird das Gerät cuau1 festgelegt.

```
ppp ON example> set speed 115200
```

Dieser Befehlt setzt die Verbindungsgeschwindigkeit auf 115200 kbps.

```
ppp ON example> enable dns
```

Dieser Befehl weist `ppp` an, den Resolver zu konfigurieren und in /etc/resolv.conf Einträge für den Nameserver hinzuzufügen. Falls `ppp` nicht in der Lage ist den Hostnamen selbst zu bestimmen, kann dieser auch später manuell eingetragen werden.

```
ppp ON example> term
```

Wechselt in den "Terminal"-Modus, um das Modem manuell kontrollieren zu können.

```
deflink: Entering terminal mode on /dev/cuau1
type '~h' for help
```

```
at
OK
atdt123456789
```

Sie verwenden `at` zur Initialisierung des Modems und dann `atdt` sowie die Nummer des ISPs, um den Einwählprozess zu starten.

```
CONNECT
```

Dies ist die Bestätigung, dass eine Verbindung aufgebaut wurde. Falls wir Verbindungsprobleme bekommen, die nicht mit der Hardware zusammenhängen, werden wir an dieser Stelle ansetzen müssen, um eine Lösung zu finden.

```
ISP Login:myusername
```

Hier werden Sie nach einem Benutzernamen gefragt. Geben Sie am Prompt den Namen ein, den Ihnen der ISP zur Verfügung gestellt hat.

```
ISP Pass:mypassword
```

An dieser Stelle müssen Sie das Passwort angeben, das Ihnen vom ISP vorgegeben wurde. Das Passwort wird, analog dem normalen Anmeldevorgang, nicht angezeigt.

```
Shell or PPP:ppp
```

Abhängig vom ISP, kann es sein, dass dieser Prompt nicht erscheint. Wir werden hier gefragt, ob wir eine Shell beim Provider verwenden oder `ppp` starten wollen. Weil wir eine Internetverbindung aufbauen wollen, haben wir uns in diesem Beispiel für `ppp` entschieden.

```
Ppp ON example>
```

Beachten Sie, dass sich in diesem Beispiel das erste p in einen Großbuchstaben verwandelt hat. Dies zeigt, dass wir erfolgreich eine Verbindung zum ISP hergestellt haben.

```
PPp ON example>
```

An dieser Stelle haben wir uns erfolgreich beim ISP authentifiziert und warten darauf, dass uns eine IP-Adresse zugewiesen wird.

```
PPP ON example>
```

Wir haben uns mit der Gegenstelle auf eine IP-Adresse geeinigt und den Verbindungsaufbau erfolgreich abgeschlossen.

```
PPP ON example> add default HISADDR
```

Hier geben wir unsere Standardroute an. Weil zu diesem Zeitpunkt unsere einzige Verbindung zu unserer Gegenstelle besteht, müssen wir dies tun, bevor wir Kontakt zur Außenwelt aufnehmen können. Falls dies aufgrund bestehender Routen nicht funktionieren sollte, können Sie ein Ausrufungszeichen ! vor add setzen. Sie können diese Standardroute aber auch vor dem eigentlichen Verbindungsaufbau angeben und PPP wird entsprechend eine neue Route aushandeln.

Wenn alles gut ging, sollten wir nun eine aktive Internetverbindung haben, die wir mit Ctrl + z in den Hintergrund schicken können. Wenn Sie feststellen, dass PPP wieder zu ppp wird, ist die Verbindung abgebrochen. Es ist gut dies zu wissen, weil dadurch der Verbindungsstatus angezeigt wird. Große Ps zeigen an, dass eine Verbindung zum ISP besteht und kleine ps zeigen an, dass keine Verbindung besteht.

### 50.3.3. Fehlersuche

Wenn keine Verbindung aufgebaut werden kann, schalten Sie die Hardware-Flusssteuerung CTS/RTS aus, indem Sie die Option set ctsrts off verwenden. Dies ist zumeist dann der Fall, wenn Sie mit einem PPP-fähigen Terminalserver verbunden sind. Hier bleibt PPP bei dem Versuch hängen, Daten über die Nachrichtenverbindung zu schicken, weil auf einCTS-Signal (Clear-to-Send) gewartet wird, das vielleicht nie kommt. Wenn Sie diese Option jedoch gebrauchen, sollten Sie auch die Option set accmap verwenden, die erforderlich sein kann, um bestimmte Hardware zu kontrollieren, die auf die Übertragung bestimmter Zeichen zwischen den Kommunikations-Endpunkten (zumeist XON/XOFF) angewiesen ist. Die Manualpage ppp(8) bietet mehr Informationen zu dieser Option und ihrer Verwendung.

Für ein älteres Modem benötigen Sie vielleicht die Option set parity even. Standardmäßig wird keine Parität vorausgesetzt, sie ist aber für die Fehlerprüfung bei älteren Modems und bei bestimmten ISPs erforderlich.

PPP kehrt möglicherweise nicht in den Befehlsmodus zurück, was normalerweise auf einen Fehler bei der Aushandlung hinweist, wobei der ISP wartet, dass der Aushandlungsprozess beginnt. Die Option ~p erzwingt in diesem Fall den Beginn des Aushandlungsprozesses.

Wenn der Login-Prompt nie erscheint, wird wahrscheinlich PAP oder CHAP für die Authentifizierung benötigt. Um PAP oder CHAP zu verwenden, ergänzen Sie PPP um folgende Optionen, bevor Sie in den Terminalmodus wechseln:

```
ppp ON example> set authname myusername
```

Hierbei sollte *myusername* durch den Benutzernamen ersetzt werden, den Sie vom ISP bekommen haben.

```
ppp ON example> set authkey mypassword
```

*mypassword* sollten Sie durch das Passwort ersetzen, das Ihnen der ISP zugewiesen hat.

Wenn die Verbindung aufgebaut wird, Sie aber keine Rechner unter dem Domänen-Namen erreichen können, versuchen Sie, einen Rechner mit ping(8) und seiner IP-Adresse zu erreichen. Wenn 100% der Pakete verloren gehen, ist es sehr wahrscheinlich, dass keine Standardroute zugewiesen wurde. Überprüfen Sie, ob während des Verbindungsaufbaus die Option `add default HISADDR` gesetzt war. Wenn Sie zu einer entfernten IP-Adresse eine Verbindung aufbauen können, ist es möglich, dass die Adresse eines Nameservers nicht in /etc/resolv.conf eingetragen wurde. Diese Datei sollte folgendermaßen aussehen:

```
domain example.com
nameserver x.x.x.x
nameserver y.y.y.y
```

```
Dabei sollten _x.x.x.x_ und _y.y.y.y_ durch die IP-Adressen der DNS-Server des ISPs
ersetzt werden.
```

Mit syslog(3) kann die PPP-Verbindung protokolliert werden. Fügen Sie einfach die folgende Zeile in /etc/syslog.conf ein:

```
!ppp
*.*     /var/log/ppp.log
```

## 50.4. PPP over Ethernet (PPPoE)

Dieser Abschnitt beschreibt, wie Sie PPP over Ethernet (PPPoE) einrichten.

Dies ist ein Beispiel einer funktionierenden ppp.conf:

```
default:
  set log Phase tun command # you can add more detailed logging if you wish
  set ifaddr 10.0.0.1/0 10.0.0.2/0
```

```
name_of_service_provider:
  set device PPPoE:xl1 # replace xl1 with your Ethernet device
  set authname YOURLOGINNAME
  set authkey YOURPASSWORD
  set dial
  set login
  add default HISADDR
```

Als `root`, geben Sie ein:

```
# ppp -ddial name_of_service_provider
```

Fügen Sie folgende Zeilen in /etc/rc.conf ein:

```
ppp_enable="YES"
ppp_mode="ddial"
ppp_nat="YES"   # if you want to enable nat for your local network, otherwise NO
ppp_profile="name_of_service_provider"
```

### 50.4.1. Verwendung einer PPPoE-Dienstbezeichnung (service tag)

Manchmal kann es notwendig sein, eine Dienstbezeichnung (service tag) zu verwenden, um eine Verbindung aufzubauen. Dienstbezeichnungen werden eingesetzt, um zwischen verschiedenen PPPoE-Servern unterscheiden zu können, die einem bestehenden Netzwerk zugeteilt sind.

Die erforderlichen Dienstbezeichnungen sollten in der Dokumentation, zu finden sein, die der ISP zur Verfügung gestellt hat.

Als letzte Möglichkeit könnten Sie versuchen, net/rr-pppoe zu installieren. Bedenken Sie aber, dass dadurch Daten Ihres Modems gelöscht werden können, so dass es nicht mehr benutzt werden kann. Überlegen Sie also genau, ob Sie dies machen wollen. Installieren Sie einfach das Programm, das Ihnen der Provider zusammen mit dem Modem geliefert hat. Gehen Sie dann in das Menü **System** dieses Programms. Der Name des Profils, sollte in der Liste aufgeführt sein. Normalerweise ist dies *ISP*.

Der Name des Profils (service tag) wird im Eintrag für die PPPoE-Konfiguration in der Datei ppp.conf verwendet, als der Teil des Befehls `set device` (die Manualpage ppp(8) enthält Einzelheiten hierzu), der den Provider angibt. Dieser Eintrag sollte folgendermaßen aussehen:

```
set device PPPoE:xl1:ISP
```

Vergessen Sie nicht, statt *xl1* das richtige Gerät für die Netzwerkkarte anzugeben.

Denken Sie auch daran, *ISP* durch das Profil zu ersetzen.

Weitere Informationen finden Sie unter Cheaper Broadband with FreeBSD on DSL von Renaud Waldura.

### 50.4.2. PPPoE mit einem 3Com® HomeConnect™ ADSL Modem Dual Link

Dieses Modem folgt nicht den in RFC 2516 festgelegten Spezifikationen.

Um FreeBSD in die Lage zu versetzen, mit diesem Gerät zu kommunizieren, muss ein sysctl Befehl angegeben werden. Dies kann beim Systemstart automatisch geschehen, indem die Datei /etc/sysctl.conf angepasst wird:

```
net.graph.nonstandard_pppoe=1
```

oder, wenn der Befehl unmittelbar wirksam werden soll, durch:

```
# sysctl net.graph.nonstandard_pppoe=1
```

Da hiermit eine systemweit gültige Einstellung vorgenommen wird, ist es nicht möglich, gleichzeitig mit einem normalen PPPoE-Client oder Server und einem 3Com® HomeConnect™ ADSL Modem zu kommunizieren.

## 50.5. PPP over ATM (PPPoA)

Nachfolgend wird beschrieben, wie PPP over ATM (PPPoA) eingerichtet wird. PPPoA ist vor allem unter europäischen DSL-Providern populär.

### 50.5.1. Die Verwendung von mpd

Sie können mpd verwenden, um zu einer Reihe von Diensten, insbesondere PPTP-Diensten eine Verbindung herzustellen. Das Programm kann aus den Ports oder als Paket net/mpd5 installiert werden. Viele ADSL Modems sind auf einen PPTP-Tunnel zwischen dem Modem und dem Rechner angewiesen.

Sobald das Programm installiert ist, müssen Sie es nach den Vorgaben des Providers konfigurieren. Der Port installiert auch einige gut dokumentierte Beispielkonfigurationsdateien in /usr/local/etc/mpd/. Ein kompletter Leitfaden zur Konfiguration von mpd ist unter /usr/local/shared/doc/mpd/ zu finden. Hier ist eine Beispielkonfiguration, um mit mpd eine Verbindung zu einem ADSL-Dienst aufzubauen. Die Konfiguration ist auf zwei Dateien verteilt. Zunächst die Datei mpd.conf:



Dieses Beispiel für mpd.conf funktioniert nur mit mpd 4.x.

```
default:
    load adsl

adsl:
```

```
    new -i ng0 adsl adsl
    set bundle authname username ①
    set bundle password password ②
    set bundle disable multilink

    set link no pap acfcomp protocomp
    set link disable chap
    set link accept chap
    set link keep-alive 30 10

    set ipcp no vjcomp
    set ipcp ranges 0.0.0.0/0 0.0.0.0/0

    set iface route default
    set iface disable on-demand
    set iface enable proxy-arp
    set iface idle 0

    open
```

① Der Benutzername, den Sie zur Authentifizierung bei Ihrem ISP verwenden.

② Das Passwort, das Sie zur Authentifizierung bei Ihrem ISP verwenden.

Die Datei mpd.links enthält Informationen über die Verbindung(en), die Sie aufbauen möchten. Eine Beispieldatei mpd.links, die das vorige Beispiel ergänzt, wird unten angegeben:

```
adsl:
    set link type pptp
    set pptp mode active
    set pptp enable originate outcall
    set pptp self 10.0.0.1 ①
    set pptp peer 10.0.0.138 ②
```

① Die IP-Adresse des FreeBSD-Rechners von dem aus Sie mpd verwenden.

② Die IP-Adresse des ADSL-Modems. Das Alcatel SpeedTouch™ Home hat die Adresse `10.0.0.138` voreingestellt.

Ein Verbindungsaufbau kann einfach durch Eingabe des folgenden Befehls als `root` gestartet werden:

```
# mpd -b adsl
```

Sie können sich den Status der Verbindung durch folgenden Befehl anzeigen lassen:

```
% ifconfig ng0
ng0: flags=88d1<UP,POINTOPOINT,RUNNING,NOARP,SIMPLEX,MULTICAST> mtu 1500
     inet 216.136.204.117 --> 204.152.186.171 netmask 0xffffffff
```

Die Verwendung von mpd ist der empfehlenswerteste Weg, um mit FreeBSD eine Verbindung zu einem ADSL-Dienst aufzubauen.

### 50.5.2. Die Verwendung von pptpclient

Es ist außerdem möglich, mit FreeBSD eine Verbindung zu anderen PPPoA-Diensten aufzubauen. Dazu wird net/pptpclient verwendet.

Um mit net/pptpclient eine Verbindung zu einem DSL-Dienst aufbauen zu können, müssen Sie den entsprechenden Port bzw. das Paket installieren und /etc/ppp/ppp.conf bearbeiten. Eine Beispieldatei für ppp.conf ist weiter unten angegeben. Weitere Informationen zu den Optionen von ppp.conf finden Sie in ppp(8).

```
adsl:
 set log phase chat lcp ipcp ccp tun command
 set timeout 0
 enable dns
 set authname username ①
 set authkey password ②
 set ifaddr 0 0
 add default HISADDR
```

① Der Benutzername für den Zugang beim DSL-Provider.

② Das Passwort für Ihren Account.



Weil das Passwort in ppp.conf im Klartext hinzugefügt wird, sollten Sie sicherstellen, dass niemand den Inhalt dieser Datei lesen kann:

```
# chown root:wheel /etc/ppp/ppp.conf
# chmod 600 /etc/ppp/ppp.conf
```

Dies wird einen Tunnel für eine PPP-Session zum DSL-Router öffnen. Ethernet-DSL-Modems haben eine vorkonfigurierte LAN-IP-Adresse, mit der Sie eine Verbindung aufbauen. Im Falle des Alcatel SpeedTouch™ Home handelt es sich dabei um die Adresse 10.0.0.138. In der Dokumentation des Routers sollte angegeben sein, welche Adresse das Gerät verwendet. Um den Tunnel zu öffnen und eine PPP-Session zu starten, führen Sie folgenden Befehl aus:

```
# pptp address adsl
```



Wenn Sie ein kaufmännisches Und ("&") an das Ende dieses Kommandos anfügen, wird pptp den Prompt zurückgeben.

Ein virtuelles Tunnel-Device tun wird für das Zusammenspiel der Prozesse pptp und ppp geschaffen. Wenn Sie den Prompt zurückerhalten haben oder der pptp-Prozess das Vorliegen einer Verbindung bestätigt, können Sie den Tunnel folgendermaßen überprüfen:

```
% ifconfig tun0
tun0: flags=8051<UP,POINTOPOINT,RUNNING,MULTICAST> mtu 1500
        inet 216.136.204.21 --> 204.152.186.171 netmask 0xffffff00
        Opened by PID 918
```

Wenn die Verbindung fehlschlägt, überprüfen Sie die Konfiguration des Routers, den Sie normalerweise mit einem Web-Browser erreichen können. Prüfen Sie auch die Ausgabe des Befehls `pptp` und die Logdatei /var/log/ppp.log.

# Kapitel 51. Elektronische Post (E-Mail)

## 51.1. Terminologie

Das Akronym *MTA* steht für *Mail Transfer Agent* was übersetzt "Mailübertragungs-Agent" bedeutet.

Während die Bezeichnung *Server-Dämon* die Komponente eines MTA benennt, die für eingehende Verbindungen zuständig ist, wird mit dem Begriff *Mailer* öfters die Komponente des MTA bezeichnet, die E-Mails versendet.

## 51.2. Übersicht

"Elektronische Post", besser bekannt als E-Mail, ist eine der am weit verbreitetsten Formen der Kommunikation heutzutage. Dieses Kapitel bietet eine grundlegende Einführung in das Betreiben eines E-Mail-Servers unter FreeBSD. Ebenfalls wird der Versand und Empfang von E-Mails unter FreeBSD behandelt. Eine umfassende Betrachtung zu diesem Thema finden Sie in den Büchern, die in Bibliografie aufgelistet sind.

Dieses Kapitel behandelt die folgenden Punkte:

- Welche Software-Komponenten beim Senden und Empfangen von elektronischer Post involviert sind.
- Wo sich grundlegende Sendmail Konfigurationsdateien in FreeBSD befinden.
- Den Unterschied zwischen entfernten und lokalen Postfächern.
- Wie man Versender von Spam daran hindern kann, E-Mail-Server illegalerweise als Weiterleitung zu verwenden.
- Wie man einen alternativen MTA installiert und konfiguriert, um Sendmail zu ersetzen.
- Wie man oft auftretende E-Mail-Server Probleme behebt.
- Wie E-Mails über einen Relay verschickt werden.
- Wie E-Mails über eine Einwahlverbindung gehandhabt werden.
- Wie SMTP-Authentifizierung einrichtet wird.
- Den Empfang und den Versand von E-Mails mithilfe von Programmen wie mutt.
- Wie E-Mails von einem entfernten Server mit POP oder IMAP abgeholt werden.
- Wie eingehende E-Mail automatisch gefiltert wird.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Die Netzwerk-Verbindung richtig einrichten. (Weiterführende Netzwerkthemen).
- Die DNS-Information für einen E-Mail-Server einstellen (Netzwerkserver).
- Wissen, wie man zusätzliche Dritthersteller-Software installiert (Installieren von Anwendungen: Pakete und Ports).

# 51.3. E-Mail Komponenten

Es gibt fünf größere Komponenten die am Austausch von E-Mails beteiligt sind: der Mail User Agent (MUA), der Mail Transfer Agent (MTA), der Mail Host, ein entferntes oder lokales Postfach, sowie DNS. Dieser Abschnitt enthält eine Übersicht über diese Komponenten.

**Mail User Agent (MUA)**

Der Mail User Agent (MUA) ist das Benutzerprogramm zum Verfassen, Senden und Empfangen von E-Mails. Diese Anwendung kann ein Kommandozeilenprogramm sein, wie das in FreeBSD enthaltene Programm `mail`, oder ein Programm aus der Ports-Sammlung wie beispielsweise mutt, alpine oder elm. In der Ports-Sammlung sind auch dutzende von grafischen Programmen verfügbar, darunter ClawsMail, Evolution und Thunderbird. Einige Unternehmen bieten auch ein Web-Mail-Programm an, das über einen Webbrowser verwaltet werden kann. Weitere Informationen zur Installation und Verwendung von MUAs unter FreeBSD finden Sie im E-Mail-Programme.

**Mail Transfer Agent (MTA)**

Der Mail Transfer Agent (MTA) ist ein E-Mail-Server Daemon, welcher für dem Empfang von eingehenden E-Mails und für den Versand von ausgehenden E-Mails verantwortlich ist. FreeBSD wird mit Sendmail als Standard-MTA ausgeliefert, aber es unterstützt auch weitere E-Mail-Server, darunter Exim, Postfix und qmail. Die Konfiguration von Sendmail wird im Sendmail-Konfigurationsdateien beschrieben. Wenn Sie einen anderen MTA aus der Ports-Sammlung installieren, lesen Sie die Nachrichten die nach der Installation der Anwendung ausgegeben werden, wenn Sie FreeBSD spezifische Informationen benötigen. Allgemeine Informationen zur Konfiguration finden Sie in der Regel auf der Webseite des Herstellers.

**Mail Host und Postfächer**

Der Mail Host ist für die Zustellung und das Empfangen von E-Mails für den Rechner oder eines Netzwerks zuständig. Der Mail Host empfängt alle E-Mails für eine Domäne und speichert diese entweder im voreingestellten mbox-Format, oder im Maildir-Format. Diese E-Mails können lokal mit einem Benutzerprogramm MUA gelesen werden. Mithilfe von Protokollen wie POP oder IMAP können die E-Mails auch von entfernten Rechnern gelesen werden. Wenn die E-Mails direkt auf dem Mail Host gelesen werden, wird kein POP- oder IMAP-Server benötigt.

Um auf entfernte Postfächer zuzugreifen, wird ein Zugang zu einem POP- oder IMAP-Server benötigt. Beide Protokolle ermöglichen es Benutzern, auf ein entferntes Postfach zuzugreifen. IMAP bietet gegenüber POP einige Vorteiler. Dazu zählt die Fähigkeit eine Kopie aller Nachrichten auf einem entfernten Server zu speichern, sowie gleichzeitig ablaufende Aktualisierungen. IMAP kann auch über langsame Verbindungen nützlich sein, da nicht gleich die komplette Nachricht heruntergeladen wird. Weiterhin können E-Mails auf dem Server durchsucht werden, was den Datenverkehr zwischen Clients und dem Server minimiert.

Die Ports-Sammlung enthält einige POP- und IMAP-Server, darunter mail/qpopper, mail/imap-uw, mail/courier-imap und mail/dovecot2.



Beachten Sie, dass sowohl POP als auch IMAP Daten, wie den Benutzernamen und das Passwort, im Klartext übertragen. Um die Übermittlung von Daten über diese Protokolle zu schützen, können Sie Sitzungen über ssh(1) (SSH-

Tunnel) tunneln oder SSL (OpenSSL) verwenden.

**Domain Name System (DNS)**

Das Domain Name System (DNS) und sein Daemon `named` spielen eine große Rolle bei der Auslieferung von E-Mails. Um E-Mails auszuliefern, fragt der MTA im DNS den Rechner ab, der E-Mails für das Zielsystem entgegennimmt. Der gleiche Vorgang läuft ab, wenn eine E-Mail von einem entfernten Server zum MTA zugestellt wird.

Im DNS werden Rechnernamen auf IP-Adressen abgebildet. Daneben werden spezielle Informationen für das Mail-System gespeichert, die *MX-Einträge* (MX record) genannt werden. Der MX-Eintrag (von Mail eXchanger) gibt an, welche Rechner E-Mails für eine Domäne annehmen.

Mit host(1) können die MX-Einträge für eine Domäne abgefragt werden:

```
# host -t mx FreeBSD.org
FreeBSD.org mail is handled by 10 mx1.FreeBSD.org
```

Weitere Informationen zu DNS und dessen Konfiguration finden Sie im Domain Name System (DNS).

# 51.4. Sendmail-Konfigurationsdateien

Sendmail ist der standardmäßig in FreeBSD installierte MTA. Es nimmt E-Mails von E-Mail-Benutzerprogrammen (MUA) entgegen und liefert diese zu den entsprechenden Mail Hosts, die in der Konfigurationsdatei definiert sind. Sendmail kann auch Netzwerkverbindungen annehmen und E-Mails an lokale *Mailboxen*, oder an andere Programme ausliefern.

Die Konfigurationsdateien von Sendmail befinden sich in /etc/mail. In diesem Abschnitt werden diese Dateien im Detail beschrieben.

**/etc/mail/access**

Diese Datenbank bestimmt, welche Rechner oder IP-Adressen Zugriff auf den lokalen Mail-Server haben und welche Art von Zugriff ihnen gestattet wird. Rechner die als `OK` aufgelistet sind, was der Standard ist, sind berechtigt E-Mails zu diesem Rechner zu schicken, solange die endgültige Zieladresse der lokale Rechner ist. Rechner die als `REJECT` aufgelistet sind, werden abgelehnt. Rechner die als `RELAY` aufgelistet sind, wird es erlaubt Post für jede Zieladresse durch diesen Mail-Server zu senden. Rechner die als `ERROR` aufgelistet sind, bekommen ihre E-Mail mit einem speziellen Fehler zurück. Wenn ein Rechner als `SKIP` aufgelistet ist, wird Sendmail die aktuelle Suche abbrechen, ohne die E-Mail zu akzeptieren oder abzulehnen. E-Mails von Rechnern die als `QUARANTAINE` aufgelistet sind, werden vorerst zurückgehalten. Dem sendenden Rechner wird ein festgelegter Text als Grund für die Quarantäne zurückgeschickt.

Beispiele für die Verwendung dieser Optionen für IPv4- und IPv6-Adressen finden Sie in der Beispielkonfiguration /etc/mail/access.sample:

```
# $FreeBSD$
```

```
#
# Mail relay access control list.  Default is to reject mail unless the
# destination is local, or listed in /etc/mail/local-host-names
#
## Examples (commented out for safety)
#From:cyberspammer.com          ERROR:"550 We don't accept mail from spammers"
#From:okay.cyberspammer.com     OK
#Connect:sendmail.org           RELAY
#To:sendmail.org                RELAY
#Connect:128.32                 RELAY
#Connect:128.32.2               SKIP
#Connect:IPv6:1:2:3:4:5:6:7     RELAY
#Connect:suspicious.example.com QUARANTINE:Mail from suspicious host
#Connect:[127.0.0.3]            OK
#Connect:[IPv6:1:2:3:4:5:6:7:8] OK
```

Um die Datenbank zu konfigurieren, verwenden Sie das im Beispiel gezeigte Format, um Einträge in /etc/mail/access hinzuzufügen, aber setzen Sie kein Kommentarsymbol (`#`) vor die Einträge. Erstellen Sie einen Eintrag für jeden Rechner, dessen Zugriff konfiguriert werden soll. E-Mail-Versender, die mit der linken Spalte der Tabelle übereinstimmen, sind betroffen von der Aktion in der rechten Spalte.

Immer wenn diese Datei verändert wurde, muss die Datenbank aktualisiert und Sendmail neu gestartet werden:

```
# makemap hash /etc/mail/access < /etc/mail/access
# service sendmail restart
```

**/etc/mail/aliases**

Diese Datenbank enthält eine Liste der virtuellen Mailboxen, die in andere Benutzer, Dateien, Programme oder andere Aliase expandiert werden. Hier sind ein paar Beispiele, die das Dateiformat verdeutlichen:

```
root: localuser
ftp-bugs: joe,eric,paul
bit.bucket:  /dev/null
procmail: "|/usr/local/bin/procmail"
```

Der Name der Mailbox auf der linken Seite des Doppelpunkts wird mit den Zielen auf der rechten Seite ersetzt. Der erste Eintrag ersetzt die Mailbox `root` mit der Mailbox `localuser`, die dann in der Datenbank /etc/mail/aliases gesucht wird. Wird kein passender Eintrag gefunden, wird die Nachricht zum `localuser` geliefert. Der zweite Eintrag zeigt eine E-Mail-Verteilerliste. E-Mails an `ftp-bugs` werden zu den drei lokalen Mailboxen `joe`, `eric` und `paul` gesendet. Eine entfernte Mailbox kann auch als *user@example.com* angegeben werden. Der dritte Eintrag zeigt wie E-Mails in eine Datei geschrieben werden, in diesem Fall /dev/null. Der letzte Eintrag verdeutlicht das Senden von E-Mails an ein Programm. Hier wird die Nachricht über eine UNIX® Pipe an /usr/local/bin/procmail gesendet. Weitere Informationen zu dem Format dieser

Datei finden Sie in aliases(5).

Wenn diese Datei geändert wird, muss `newaliases` ausgeführt werden, um die Datenbank zu aktualisieren.

**/etc/mail/sendmail.cf**

Dies ist die Hauptkonfigurations-Datei von Sendmail. Sie kontrolliert das allgemeine Verhalten von Sendmail, einschließlich allem vom Umschreiben von E-Mail Adressen bis hin zum Übertragen von Ablehnungsnachrichten an entfernte E-Mail-Server. Dementsprechend ist die Konfigurationsdatei ziemlich komplex. Glücklicherweise muss diese Datei selten für Standard E-Mail-Server geändert werden.

Die Sendmail Hauptkonfigurationsdatei kann mit m4(1) Makros erstellt werden, die Eigenschaften und Verhalten von Sendmail definieren. Einige der Details finden Sie in /usr/src/contrib/sendmail/cf/README.

Wenn Änderungen an dieser Datei vorgenommen werden, muss Sendmail neu gestartet werden, damit die Änderungen Wirkung zeigen.

**/etc/mail/virtusertable**

Diese Datenbank ordnet Adressen für virtuelle Domänen und Benutzern reellen Mailboxen zu. Diese Mailboxen können lokal, auf entfernten Systemen, Aliase in /etc/mail/aliases oder eine Datei sein. Dadurch können mehrere virtuelle Domains auf einem Rechner gehostet werden.

FreeBSD enthält eine Beispielkonfiguration in /etc/mail/virtusertable.sample, die das Format genauer beschreibt. Das folgende Beispiel zeigt, wie benutzerdefinierte Einträge in diesem Format erstellt werden:

```
root@example.com                root
postmaster@example.com          postmaster@noc.example.net
@example.com                    joe
```

Diese Datei wird nach dem ersten übereinstimmenden Eintrag durchsucht. Wenn eine E-Mail-Adresse mit der Adresse auf der linken Seite übereinstimmt, wird sie dem Eintrag auf der rechten Seite zugeordnet. Der erste Eintrag in diesem Beispiel ordnet eine bestimmte E-Mail-Adresse einer lokalen Mailbox zu, während der zweite Eintrag eine bestimmte E-Mail-Adresse einer entfernten Mailbox zuordnet. Zuletzt wird jede E-Mail-Adresse von `example.com`, welche nicht mit einem der vorherigen Einträge übereinstimmt, mit dem letzten Eintrag übereinstimmen und der lokalen Mailbox `joe` zugeordnet. Benutzen Sie dieses Format, wenn Sie neue Einträge in /etc/mail/virtusertable hinzufügen. Jedes Mal, wenn diese Datei bearbeitet wurde, muss die Datenbank aktualisiert und Sendmail neu gestartet werden:

```
# makemap hash /etc/mail/virtusertable < /etc/mail/virusertable
# service sendmail restart
```

**/etc/mail/relay-domains**

In der standardmäßigen FreeBSD-Installation wird Sendmail nur dazu konfiguriert, E-Mails von

dem Rechner, auf dem es läuft, zu senden. Wenn zum Beispiel ein POP-Server installiert ist, können Benutzer ihre E-Mails von entfernten Standorten überprüfen. Sie werden jedoch keine E-Mails von außen verschicken können. Typischerweise wird ein paar Sekunden nach dem Versuch eine E-Mail von MAILER-DAEMON mit einer `5.7 Relaying Denied` Fehlermeldung versendet werden.

Die einfachste Lösung ist, wie im folgenden Beispiel gezeigt, den FQDN des Internet-Dienstanbieters und gegebenenfalls weitere Adressen in /etc/mail/relay-domains einzutragen:

```
your.isq.example.com
other.isp.example.net
users.isp.example.org
www.example.org
```

Nachdem diese Datei erstellt oder editiert wurde, muss Sendmail mittels `service sendmail restart` neu gestartet werden.

Ab jetzt wird jede E-Mail, die von einem in der Liste eingetragenen Rechner durch das System geschickt wird, ihr Ziel erreichen, vorausgesetzt der Benutzer hat einen Account auf dem System. Dies erlaubt es Benutzern aus der Ferne, E-Mails über das System zu versenden, ohne dem Massenversand (SPAM) die Tür zu öffnen.

# 51.5. Wechseln des Mailübertragungs-Agenten

FreeBSD enthält mit Sendmail bereits einen MTA, der für die ein- und ausgehenden E-Mails verantwortlich ist. Der Systemadministrator kann aber den MTA des Systems wechseln. Eine große Auswahl an alternativen MTAs ist in der Kategorie `mail` der FreeBSD Ports-Sammlung verfügbar.

Sobald ein neuer MTA installiert ist, können Sie die neue Software konfigurieren und testen, bevor Sie Sendmail ersetzen. Informationen über die Konfiguration des neu gewählten MTA finden Sie in der dazugehörigen Dokumentation.

Sobald der neue MTA wie gewünscht funktioniert, benutzen Sie die Anweisungen in diesem Abschnitt, um Sendmail zu deaktivieren und stattdessen den neuen MTA zu verwenden.

## 51.5.1. Sendmail deaktivieren



Wenn der ausgehende Mail-Dienst von Sendmail deaktiviert ist, muss für den E-Mail-Versand ein alternatives System installiert werden. Andernfalls sind Systemfunktionen wie periodic(8) nicht mehr in der Lage, ihre Resulate und Meldungen als E-Mail zu versenden. Aber auch viele andere Teile des Systems erwarten einen funktionalen MTA. Sind Programme auf die deaktivierten Sendmail-Binärdateien angewiesen, landen deren E-Mails ansonsten in einer inaktiven Sendmail-Warteschlange und können nicht ausgeliefert werden.

Um Sendmail komplett zu deaktivieren, müssen folgende Zeilen in /etc/rc.conf hinzugefügt oder editiert werden:

```
sendmail_enable="NO"
sendmail_submit_enable="NO"
sendmail_outbound_enable="NO"
sendmail_msp_queue_enable="NO"
```

Um lediglich die Funktion zum Empfang von E-Mails durch Sendmail zu deaktivieren, muss folgender Eintrag in /etc/rc.conf gesetzt werden:

```
sendmail_enable="NO"
```

Weitere Informationen zu den Startoptionen von Sendmail finden Sie in der Manualpage rc.sendmail(8).

### 51.5.2. Den voreingestellten MTA ersetzen

Wenn ein neuer MTA über die Ports-Sammlung installiert wird, werden auch die Startskripten installiert. Die Anweisungen zum starten dieser Skripte werden in den Paketnachrichten erwähnt. Bevor Sie den neuen MTA in Betrieb nehmen, stoppen Sie alle laufenden Sendmail-Prozesse. In diesem Beispiel werden alle notwendigen Dienste gestoppt und danach der Postfix Dienst gestartet:

```
# service sendmail stop
# service postfix start
```

Damit der angegebene MTA automatisch beim Hochfahren des Systems gestartet wird, fügen Sie dessen Konfigurationszeile in /etc/rc.conf hinzu. Dieser Eintrag startet den PostfixMTA:

```
postfix_enable="YES"
```

Da Sendmail allgegenwärtig ist und manche Anwendungen einfach davon ausgehen es bereits installiert und konfiguriert, wird einige zusätzliche Konfiguration benötigt. Überprüfen Sie /etc/periodic.conf und stellen Sie sicher, dass diese Werte auf NO gesetzt werden. Wenn die Datei nicht existiert, erstellen Sie sie mit folgenden Einträgen:

```
daily_clean_hoststat_enable="NO"
daily_status_mail_enable="NO"
daily_status_include_submit_mailq="NO"
daily_submit_queuerun="NO"
```

Viele alternative MTAs stellen ihre eigenen kompatiblen Implementierungen der Sendmail Kommandozeilen-Schnittstelle zur Verfügung, was die Verwendung als "drop-in" Ersatz für Sendmail vereinfacht. Allerdings versuchen einige MUAs Sendmails Standard-Dateien auszuführen, anstelle der Dateien des neuen MTAs. FreeBSD verwendet /etc/mail/mailer.conf um die erwarteten Sendmail Dateien auf die neuen Dateien abzubilden. Weitere Informationen über diese Zuordnungen können in mailwrapper(8) gefunden werden.

In der Voreinstellung sieht /etc/mail/mailer.conf wie folgt aus:

```
# $FreeBSD$
#
# Execute the "real" sendmail program, named /usr/libexec/sendmail/sendmail
#
sendmail        /usr/libexec/sendmail/sendmail
send-mail       /usr/libexec/sendmail/sendmail
mailq           /usr/libexec/sendmail/sendmail
newaliases      /usr/libexec/sendmail/sendmail
hoststat        /usr/libexec/sendmail/sendmail
purgestat       /usr/libexec/sendmail/sendmail
```

Wenn eines der Kommandos auf der linken Seite ausgeführt werden soll, führt das System tatsächlich den damit verbundenen Befehl auf der rechten Seite aus. Mit diesem System lassen sich Programme, die für die Sendmail-Funktionen gestartet werden, leicht ändern.

Einige MTAs aus der Ports-Sammlung können diese Datei aktualisieren. Zum Beispiel würde Postfix die Datei wie folgt aktualisieren:

```
#
# Execute the Postfix sendmail program, named /usr/local/sbin/sendmail
#
sendmail        /usr/local/sbin/sendmail
send-mail       /usr/local/sbin/sendmail
mailq           /usr/local/sbin/sendmail
newaliases      /usr/local/sbin/sendmail
```

Falls die Installation des MTA nicht automatisch /etc/mail/mailer.conf aktualisiert, bearbeiten Sie diese Datei in einem Texteditor, so dass auf die neuen Dateien verwiesen wird. Dieses Beispiel zeigt auf die Dateien, die von mail/ssmtp installiert wurden:

```
sendmail        /usr/local/sbin/ssmtp
send-mail       /usr/local/sbin/ssmtp
mailq           /usr/local/sbin/ssmtp
newaliases      /usr/local/sbin/ssmtp
hoststat        /usr/bin/true
purgestat       /usr/bin/true
```

Sobald alles konfiguriert ist, wird empfohlen, das System neu zu starten. Ein Neustart bietet auch die Möglichkeit sicherzustellen, dass das System korrekt konfiguriert wurde, um den neuen MTA automatisch beim Hochfahren zu starten.

## 51.6. Fehlerbehebung

Hier finden sich ein paar häufig gestellte Fragen und ihre Antworten, die von der FAQ

übernommen wurden.

### 51.6.1. Warum muss ich einen FQDN (fully-qualified domain name / voll ausgeschriebenen Domänennamen) für meine Rechner verwenden?

Vielleicht befindet sich der Rechner in einer anderen Domäne. Um beispielsweise von einem Rechner in `foo.bar.edu` einen Rechner namens `mumble` in der Domäne `foo.bar.edu` zu erreichen, geben Sie seinen voll ausgeschriebenen Domänennamen (FQDN) `mumble.bar.edu`, anstelle von `mumble` an.

Das liegt daran, dass die aktuelle Version von BIND, die mit FreeBSD ausgeliefert wird, keine Standardabkürzungen für nicht komplett angegebene Domänennamen außerhalb der lokalen Domäne unterstützt. Daher muss ein nicht-qualifizierter Rechner, wie `mumble`, entweder als `mumble.foo.bar.edu` gefunden werden, oder er wird in der root Domäne gesucht.

In älteren Versionen von BIND lief die Suche über `mumble.bar.edu` und `mumble.edu`. RFC 1535 erklärt, warum dieses Verhalten als schlechte Praxis oder sogar als Sicherheitsloch angesehen wird.

Um das zu umgehen, setzen Sie die Zeile:

```
search foo.bar.edu bar.edu
```

anstatt der vorherigen

```
domain foo.bar.edu
```

in /etc/resolv.conf ein. Stellen Sie jedoch sicher, dass die Suchordnung nicht die Begrenzung von "lokaler und öffentlicher Administration", wie RFC 1535 sie nennt, überschreitet.

### 51.6.2. Wie kann ich einen E-Mail-Server auf einem Anwahl-PPP Rechner betreiben?

Sie wollen sich mit einem FreeBSD E-Mail Gateway im LAN verbinden. Die PPP-Verbindung ist keine Standleitung.

Ein Weg dies zu tun ist, von einem immer mit dem Internet verbundenen Server einen sekundären MX-Dienst für die Domäne zur Verfügung gestellt zu bekommen. In diesem Beispiel heißt die Domäne `example.com`, und der Internet-Dienstanbieter hat `example.net` so eingestellt, dass er für die Domäne einen sekundären MX-Dienst zur Verfügung stellt:

```
example.com.            MX      10      bigco.com.
                    MX      20      example.net.
```

Nur ein Rechner sollte als Endempfänger angegeben sein. Sendmail fügen Sie `Cw example.com` zu /etc/sendmail.cf auf `example.com` hinzu.

Wenn der MTA des Versenders versucht die E-Mail zuzustellen, wird es versuchen das System `example.com` über die PPP-Verbindung zu erreichen. Es kommt zu einer Zeitüberschreitung, wenn das Zielsystem offline ist. Der MTA wird die E-Mail automatisch der sekundären MX-Seite des Internet-Providers `example.net` zustellen. Die sekundäre MX-Seite wird periodisch versuchen, eine Verbindung zur primären MX-Seite `example.com` aufzubauen.

Verwenden Sie etwas wie dies als Login-Skript:

```
#!/bin/sh
# Put me in /usr/local/bin/pppmyisp
( sleep 60 ; /usr/sbin/sendmail -q ) &
/usr/sbin/ppp -direct pppmyisp
```

Wenn Sie ein separates Login-Skript für einen Benutzer erstellen, benutzen Sie stattdessen `sendmail -qRexample.com` in dem oben gezeigten Skript. Das erzwingt die sofortige Verarbeitung der E-Mails in der Warteschlange für `example.com`

Eine weitere Verfeinerung der Situation kann an diesem Beispiel von FreeBSD Internet service providers entnommen werden:

```
> wir stellen einem Kunden den sekundären MX zur Verfügung.
> Der Kunde verbindet sich mit unseren Diensten mehrmals am Tag
> automatisch um die E-Mails zu seinem primären MX zu holen
> (wir wählen uns nicht bei ihm ein, wenn E-Mails für seine
> Domäne eintreffen).  Unser sendmail sendet den Inhalt der
> E-Mail-Warteschlange alle 30 Minuten.  Momentan muss er 30 Minuten
> eingewählt bleiben um sicher zu sein, dass alle seine E-Mails
> beim primären MX eingetroffen sind.
>
> Gibt es einen Befehl, der sendmail dazu bringt, alle E-Mails sofort
> zu senden? Der Benutzer hat natürlich keine root-Rechte auf
> unserer Maschine.

In der privacy flags Sektion von sendmail.cf befindet sich die
Definition Opgoaway,restrictqrun

Entferne restrictqrun um nicht-root Benutzern zu erlauben, die Verarbeitung
der Nachrichten-Warteschlangen zu starten.  Möglicherweise willst du
auch die MX neu sortieren.  Wir sind der primäre MX für unsere
Kunden mit diesen Wünschen und haben definiert:

# Wenn wir der beste MX für einen Rechner sind, versuche es direkt
# anstatt einen lokalen Konfigurationsfehler zu generieren.
OwTrue

Auf diesem Weg liefern Gegenstellen direkt zu dir, ohne die Kundenverbindung
zu versuchen.  Dann sendest du zu deinem Kunden.  Das funktioniert nur
für Rechner, du musst also deinen Kunden dazu bringen,
ihre E-Mail Maschine customer.com zu nennen, sowie
```

```
hostname.customer.com im DNS.  Setze einfach einen A-Eintrag
in den DNS für customer.com.
```

# 51.7. Weiterführende Themen

Dieser Abschnitt behandelt kompliziertere Themen wie E-Mail-Konfiguration und Einrichtung von E-Mail für eine ganze Domäne.

## 51.7.1. Grundlegende Konfiguration

Mit der Software im Auslieferungszustand sollte es möglich sein, E-Mails an externe Rechner zu senden, vorausgesetzt /etc/resolv.conf ist konfiguriert, oder das Netzwerk hat Zugriff auf einen konfigurierten DNS-Server. Um E-Mails an den MTA auf dem Rechner auszuliefern, stehen zwei Möglichkeiten zur Auswahl:

- Betreiben Sie einen DNS-Server für die Domäne.
- Lassen Sie die E-Mails direkt über den FQDN des Rechners ausliefern.

Um E-Mails direkt zu einem Rechner geliefert zu bekommen, wird eine permanente statische IP-Adresse (keine dynamische IP-Adresse) benötigt. Befindet sich das System hinter einer Firewall, muss diese den SMTP-Verkehr weiterleiten. Um E-Mails direkt am Rechner zu empfangen, muss eines der folgenden Dinge konfiguriert werden:

- Vergewissern Sie sich, dass der MX-Eintrag mit der kleinsten Nummer im DNS auf die statische IP-Adresse des Rechners zeigt.
- Stellen Sie sicher, dass für den Rechner kein MX-Eintrag im DNS existiert.

Jede der erwähnten Konfigurationsmöglichkeiten erlaubt es, E-Mails direkt auf dem Rechner zu empfangen.

Versuchen Sie das:

```
# hostname
example.FreeBSD.org

# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
```

In diesem Beispiel sollte es funktionieren, E-Mails direkt an yourlogin@example.FreeBSD.org zu senden, vorausgesetzt dass Sendmail auf `example.FreeBSD.org` korrekt läuft.

In diesem Beispiel:

```
# host example.FreeBSD.org
example.FreeBSD.org has address 204.216.27.XX
example.FreeBSD.org mail is handled (pri=10) by devnull.FreeBSD.org
```

Hier wird jede an den Rechner example.FreeBSD.org gesandte E-Mail auf hub unter dem gleichen Benutzernamen gesammelt, anstatt diese direkt zu Ihrem Rechner zu senden.

Die obige Information wird von einem DNS-Server verwaltet. Der DNS-Eintrag, der die Information zum E-Mail-Routing enthält, ist der MX-Eintrag. Existiert kein MX-Eintrag, werden E-Mails direkt über die IP-Adresse an den Rechner geliefert.

Der MX-Eintrag für freefall.FreeBSD.org sah einmal so aus:

```
freefall        MX  30  mail.crl.net
freefall        MX  40  agora.rdrop.com
freefall        MX  10  freefall.FreeBSD.org
freefall        MX  20  who.cdrom.com
```

freefall hatte viele MX-Einträge. Die kleinste MX-Nummer definiert de Rechner, der die E-Mails direkt empfängt, wobei die anderen Rechner temporär E-Mails in Warteschlangen einreihen, falls freefall beschäftigt oder unerreichbar ist.

Es ist sehr sinnvoll, dass stellvertretende MX-Seiten separate Internet-Verbindungen verwenden. Ihr ISP kann diesen Dienst zur Verfügung stellen.

### 51.7.2. E-Mails für eine Domäne

Wird ein MTA für ein Netzwerk konfiguriert, dann sollte jede E-Mail die an einen Rechner in dieser Domäne geschickt wird, an den MTA umgeleitet werden, damit die Benutzer ihre E-Mails vom zentralen Mail-Server empfangen können.

Am einfachsten ist es, wenn Accounts mit gleichen *Benutzernamen* sowohl auf dem MTA, als auch auf dem System mit dem MUA existieren. Verwenden Sie adduser(8), um Benutzerkonten anzulegen.

Der MTA muss auf jeder Workstation im Netzwerk als der zuständige Rechner für den E-Mail-Austausch gekennzeichnet werden. Dies wird in der DNS-Konfiguration über den MX-Eintrag gesteuert:

```
example.FreeBSD.org A   204.216.27.XX       ; Workstation
            MX  10 devnull.FreeBSD.org  ; Mailhost
```

Diese Einstellung wird E-Mails für die Workstations zum MTA weiterleiten, egal wo der A-Eintrag hinzeigt. Die E-Mails werden zum MX-Rechner gesendet.

Diese Einstellung muss auf dem DNS-Server konfiguriert werden. Besitzt das Netzwerk keinen eigenen DNS-Server, kontaktieren Sie Ihren ISP oder DNS-Verwalter.

Im Folgenden ist ein Beispiel für virtuelles E-Mail-Hosting. Nehmen wir an, dass für einen Kunden mit der Domäne customer1.org, alle E-Mails für customer1.org an mail.myhost.com gesendet werden sollen. Der entsprechende DNS-Eintrag sollte wie folgt aussehen:

```
customer1.org       MX  10  mail.myhost.com
```

Wenn für die Domäne nur E-Mails verarbeitet werden sollen, wird für `customer1.org` *kein* `A`-Eintrag benötigt. Allerdings wird ein `ping` gegen `customer1.org` nur dann funktionieren, wenn ein `A`-Eintrag existiert.

Teilen Sie dem MTA mit, für welche Domänen bzw. Hostnamen Post entgegengenommen werden soll. Die beiden folgenden Methoden funktionieren für Sendmail:

- Fügen Sie die Rechnernamen in /etc/mail/local-host-names hinzu, wenn `FEATURE(use_cw_file)` verwendet wird.
- Fügen Sie eine Zeile `Cwyour.host.com` in /etc/sendmail.cf hinzu.

## 51.8. Ausgehende E-Mail über einen Relay versenden

In vielen Fällen möchte man E-Mail nur über einen Relay verschicken. Zum Beispiel:

- Der Rechner ist ein Arbeitsplatzrechner und benutzt Programme wie mail(1) über ein Relay des ISP.
- Ein Server, der E-Mails nicht selbst verarbeitet, soll alle E-Mails zu einem Relay schicken.

Obwohl jeder MTA diese Aufgabe erfüllen kann, ist es oft schwierig einen vollwertigen MTA so zu konfigurieren, dass er lediglich ausgehende E-Mails weiterleitet. Es ist übertrieben, Programme wie Sendmail und Postfix nur für diesen Zweck einzusetzen.

Weiterhin kann es sein, dass die Bestimmungen des Internetzugangs es verbieten, einen eigenen Mail-Server zu betreiben.

Um die hier beschriebenen Anforderungen zu erfüllen, installieren Sie einfach den Port mail/ssmtp:

```
# cd /usr/ports/mail/ssmtp
# make install replace clean
```

Nach der Installation kann mail/ssmtp über /usr/local/etc/ssmtp/ssmtp.conf konfiguriert werden:

```
root=yourrealemail@example.com
mailhub=mail.example.com
rewriteDomain=example.com
hostname=_HOSTNAME_
```

Verwenden Sie eine gültige E-Mail-Adresse für `root`. Geben Sie für `mail.example.com` den Mail-Relay des ISPs an. Einige ISPs nennen den Relay "Postausgangsserver" oder "SMTP-Server".

Deaktivieren Sie Sendmail, einschließlich des Services für den Postausgang. Details finden Sie in Sendmail deaktivieren.

mail/ssmtp verfügt über weitere Optionen. Die Beispiele in /usr/local/etc/ssmtp oder die Manualpage von ssmtp enthalten weitere Informationen.

Wird ssmtp wie hier beschrieben eingerichtet, können Anwendungen E-Mails von dem lokalen Rechner verschicken. Man verstößt damit auch nicht gegen Bestimmungen des ISPs und läuft nicht Gefahr, dass der Rechner zum Versenden von Spam missbraucht wird.

## 51.9. E-Mail über Einwahl-Verbindungen

Wird eine feste IP-Adresse verwendet, müssen die Standardeinstellungen wahrscheinlich gar nicht geändert werden. Stellen Sie den Hostnamen auf den entsprechend zugeordneten Internetnamen ein und Sendmail übernimmt das Übrige.

Bei der Verwendung einer dynamisch zugewiesenen IP-Adresse und einer PPP-Wählverbindung mit dem Internet, hat man in der Regel ein Postfach auf dem Mailserver des ISP. In diesem Beispiel ist die Domäne des ISP `example.net`, der Benutzername ist `user`, der Rechnername ist `bsd.home` und der ISP erlaubt es, `relay.example.net` als Mail-Relayhost zu benutzen.

Um Mails aus der Mailbox des ISPs abzuholen, muss ein gesondertes Programm aus der Ports-Sammlung installiert werden. mail/fetchmail ist eine gute Wahl, weil es viele verschiedene Protokolle unterstützt. Für gewöhnlich stellt der ISPPOP zur Verfügung. Falls User-PPP verwendet wird, können durch folgenden Eintrag in /etc/ppp/ppp.linkup E-Mails automatisch abgerufen werden, sobald eine Verbindung zum Netz aufgebaut wird:

```
MYADDR:
!bg su user -c fetchmail
```

Wird Sendmail benutzt, um E-Mails an nicht-lokale Benutzer zu versenden, konfigurieren Sie es so, dass die Warteschlange abgearbeitet wird, sobald eine Verbindung mit dem Internet besteht. Um dies zu erreichen, müssen folgende Zeilen nach dem `fetchmail`-Eintrag in /etc/ppp/ppp.linkup hinzugefügt werden.

```
!bg su user -c "sendmail -q"
```

In diesem Beispiel existiert auf `bsd.home` ein Benutzer `user`. Erstellen Sie auf `bsd.home` im Heimatverzeichnis von `user` die Datei .fetchmailrc mit folgender Zeile:

```
poll example.net protocol pop3 fetchall pass MySecret;
```

Diese Datei sollte für niemandem außer `user` lesbar sein, weil sie das Passwort `MySecret` enthält.

Um Mails mit dem richtigen `from:`-Header zu versenden, müssen Sie Sendmail so konfigurieren, dass es user@example.net und nicht user@bsd.home benutzen soll und das alle Mails über `relay.example.net` versendet werden, um eine schnellere Übertragung von Mails zu gewährleisten.

Die folgende .mc sollte ausreichen:

```
VERSIONID(`bsd.home.mc version 1.0')
OSTYPE(bsd4.4)dnl
FEATURE(nouucp)dnl
MAILER(local)dnl
MAILER(smtp)dnl
Cwlocalhost
Cwbsd.home
MASQUERADE_AS(`example.net')dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(nocanonify)dnl
FEATURE(nodns)dnl
define(`SMART_HOST', `relay.example.net')
Dmbsd.home
define(`confDOMAIN_NAME',`bsd.home')dnl
define(`confDELIVERY_MODE',`deferred')dnl
```

Im vorherigen Abschnitt finden Sie Details dazu, wie Sie diese Datei in das Format sendmail.cf konvertieren können. Vergessen Sie nicht, Sendmail neu zu starten, nachdem sendmail.cf verändert wurde.

## 51.10. SMTP-Authentifizierung

Die Konfiguration von SMTP-Authentifizierung auf dem MTA bietet einige Vorteile. Die erforderliche Authentifizierung erhöht die Sicherheit von Sendmail und mobilen Benutzern, die auf entfernten Rechnern arbeiten. Diese Benutzer können denselben MTA verwenden, ohne jedes Mal das Benutzerprogramm neu konfigurieren zu müssen.

1. Installieren Sie security/cyrus-sasl2 aus der Ports-Sammlung. Dieser Port verfügt über einige Optionen, die während der Übersetzung festgelegt werden. Für die in diesem Abschnitt beschriebene Methode zur SMTP-Authentifizierung muss die Option `LOGIN` aktiviert werden.
2. Nach der Installation von security/cyrus-sasl2 editieren Sie /usr/local/lib/sasl2/Sendmail.conf. Erstellen Sie die Datei, wenn sie nicht existiert und fügen Sie die folgende Zeile hinzu:

   ```
   pwcheck_method: saslauthd
   ```

3. Als nächstes installieren Sie security/cyrus-sasl2-saslauthd, und fügen die folgende Zeile in /etc/rc.conf ein:

   ```
   saslauthd_enable="YES"
   ```

   Abschließend starten Sie den saslauthd-Dämon:

   ```
   # service saslauthd start
   ```

Dieser Dämon agiert als Broker zwischen Sendmail und der FreeBSD-passwd-Datenbank. Dadurch müssen zum Versenden von E-Mails keine zusätzlichen Accounts und Passwörter angelegt werden. Die Benutzer verwenden dasselbe Passwort zum Anmelden wie zum Verschicken von E-Mails.

4. Fügen Sie danach in /etc/make.conf die folgenden Zeilen hinzu:

```
SENDMAIL_CFLAGS=-I/usr/local/include/sasl -DSASL
SENDMAIL_LDADD=/usr/local/lib/libsasl2.so
```

   Beim Übersetzen von Sendmail werden damit die cyrus-sasl2-Bibliotheken benutzt. Stellen Sie daher vor dem Übersetzen von Sendmail sicher, dass cyrus-sasl2 installiert ist.

5. Übersetzen Sie Sendmail mit den nachstehenden Kommandos:

```
# cd /usr/src/lib/libsmutil
# make cleandir && make obj && make
# cd /usr/src/lib/libsm
# make cleandir && make obj && make
# cd /usr/src/usr.sbin/sendmail
# make cleandir && make obj && make && make install
```

   Die Übersetzung sollte keine Probleme bereiten, wenn /usr/src nicht umfangreich verändert wurde und die benötigten Bibliotheken installiert sind.

6. Nachdem Sendmail übersetzt und installiert wurde, editieren Sie /etc/mail/freebsd.mc beziehungsweise die lokale .mc-Datei. Viele Administratoren verwenden die Ausgabe von hostname(1), um der .mc einen eindeutigen Namen zu geben. Fügen Sie die folgenden Zeilen hinzu:

```
dnl set SASL options
TRUST_AUTH_MECH(`GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
define(`confAUTH_MECHANISMS', `GSSAPI DIGEST-MD5 CRAM-MD5 LOGIN')dnl
```

   Diese Anweisungen konfigurieren die Methoden, die Sendmail zur Authentifizierung von Benutzern verwendet. Lesen Sie die Sendmail Dokumentation, wenn eine andere Methode als `pwcheck` verwendet werden soll.

7. Abschließend rufen Sie make(1) in /etc/mail auf. Damit wird aus der .mc-Datei eine neue .cf -Datei erzeugt. Der Name ist entweder freebsd.cf oder der Name der lokalen .mc-Datei. `make install restart` installiert die Datei nach /etc/mail/sendmail.cf und startet Sendmail neu. Weitere Informationen zu diesem Vorgang entnehmen Sie bitte /etc/mail/Makefile.

Um die Konfiguration zu testen, verwenden Sie einen MUA, um eine Testnachricht zu senden. Mail-Benutzerprogramm das Passwort für die Authentifizierung ein und versenden Sie zum Testen eine E-Mail. Zur Fehlersuche, setzen Sie den `LogLevel` von Sendmail auf `13` und untersuchen die Fehlermeldungen in /var/log/maillog.

Weitere Information finden Sie unter SMTP-Authentifizierung.

## 51.11. E-Mail-Programme

Anwendungen, die E-Mails versenden und empfangen, werden als E-Mail-Programme oder Mail-User-Agents (MUA) bezeichnet. Mit der Entwicklung und Ausbreitung von E-Mail wachsen auch die E-Mail-Programme und bieten Benutzern mehr Funktionen und höhere Flexibilität. Die Kategorie `mail` der FreeBSD Ports-Sammlung enthält zahlreiche E-Mail-Programme. Dazu gehören grafische Programme, wie beispielsweise Evolution oder Balsa und Konsolen basierte Programme wie mutt oder alpine.

### 51.11.1. `mail`

Das standardmäßig unter FreeBSD installierte E-Mail-Programm ist mail(1). Das Programm ist konsolenorientiert und enthält alle Funktionen, die zum Versand und Empfang textbasierter E-Mails erforderlich sind. Es bietet eine begrenzte Unterstützung für Anhänge und kann auf lokale Postfächer zugreifen.

`mail` kann nicht direkt auf POP- oder IMAP-Server zugreifen. Entfernte Postfächer können aber mit einer Anwendung wie fetchmail in eine lokale mbox geladen werden.

Um E-Mails zu versenden oder zu empfangen, starten Sie einfach `mail` wie im nachstehenden Beispiel:

```
% mail
```

`mail` liest automatisch den Inhalt des Benutzer-Postfachs im Verzeichnis /var/mail. Sollte das Postfach leer sein, beendet sich `mail` mit der Nachricht, dass keine E-Mails vorhanden sind. Wenn E-Mails vorhanden sind, wird die Benutzeroberfläche gestartet und eine Liste der E-Mails angezeigt. Die E-Mails werden automatisch nummeriert wie im folgenden Beispiel gezeigt:

```
Mail version 8.1 6/6/93.  Type ? for help.
"/var/mail/marcs": 3 messages 3 new
>N  1 root@localhost        Mon Mar  8 14:05  14/510   "test"
 N  2 root@localhost        Mon Mar  8 14:05  14/509   "user account"
 N  3 root@localhost        Mon Mar  8 14:05  14/509   "sample"
```

Einzelne Nachrichten können nun durch Eingabe von `t` gefolgt von der Nummer der Nachricht gelesen werden. Im nachstehenden Beispiel wird die erste E-Mail gelesen:

```
&
 t 1
Message 1:
From root@localhost  Mon Mar  8 14:05:52 2004
X-Original-To: marcs@localhost
Delivered-To: marcs@localhost
```

```
To: marcs@localhost
Subject: test
Date: Mon,  8 Mar 2004 14:05:52 +0200 (SAST)
From: root@localhost (Charlie Root)

Das ist eine Test-Nachricht.  Antworte bitte!
```

Wie in diesem Beispiel zu sehen ist, wird die Nachricht zusammen mit dem vollständigen Nachrichtenkopf angezeigt. Um die Liste der E-Mails erneut zu sehen, drücken Sie wieder die Taste `h`.

Um auf eine E-Mail zu antworten, benutzen Sie entweder `R` oder `r`. `R` weist `mail` an, dem Versender der Nachricht zu antworten, während mit `r` allen Empfängern der Nachricht geantwortet wird. Den Kommandos kann die Zahl der E-Mail, auf die geantwortet werden soll, mitgegeben werden. Nachdem die Antwort E-Mail verfasst worden ist, sollte die Eingabe mit einem einzelnen Punkt (`.`) auf einer neuen Zeile abgeschlossen werden. Hierzu ein Beispiel:

```
&
 R 1
To: root@localhost
Subject: Re: test
 Danke, ich habe deine E-Mail erhalten.
.
EOT
```

Neue E-Mails können mit `m`, gefolgt von der E-Mail-Adresse des Empfängers verschickt werden. Mehrere Empfänger werden durch Kommata (`,`) getrennt, angegeben. Der Betreff (subject) der Nachricht kann dann, gefolgt vom Inhalt der Nachricht eingegeben werden. Die Nachricht wird dann mit einem einzelnen Punkt (`.`) auf einer neuen Zeile abgeschlossen.

```
&
 mail root@localhost
Subject:
 Ich habe die E-Mails im Griff!

Jetzt kann ich E-Mails versenden und empfangen ... :)
.
EOT
```

Die Taste `?` zeigt zu jeder Zeit einen Hilfetext an. Lesen Sie mail(1), wenn Sie weitere Hilfe zur Benutzung von `mail` erhalten möchten.



mail(1) wurde nicht für den Umgang mit Anhängen entworfen und kann daher sehr schlecht mit Anhängen umgehen. Neuere MUAs gehen wesentlich besser mit Anhängen um. Benutzer, die `mail` bevorzugen, werden vielleicht den Port converters/mpack zu schätzen wissen.

### 51.11.2. mutt

mutt ist ein leistungsfähiges E-Mail-Programm mit vielen Funktionen, darunter:

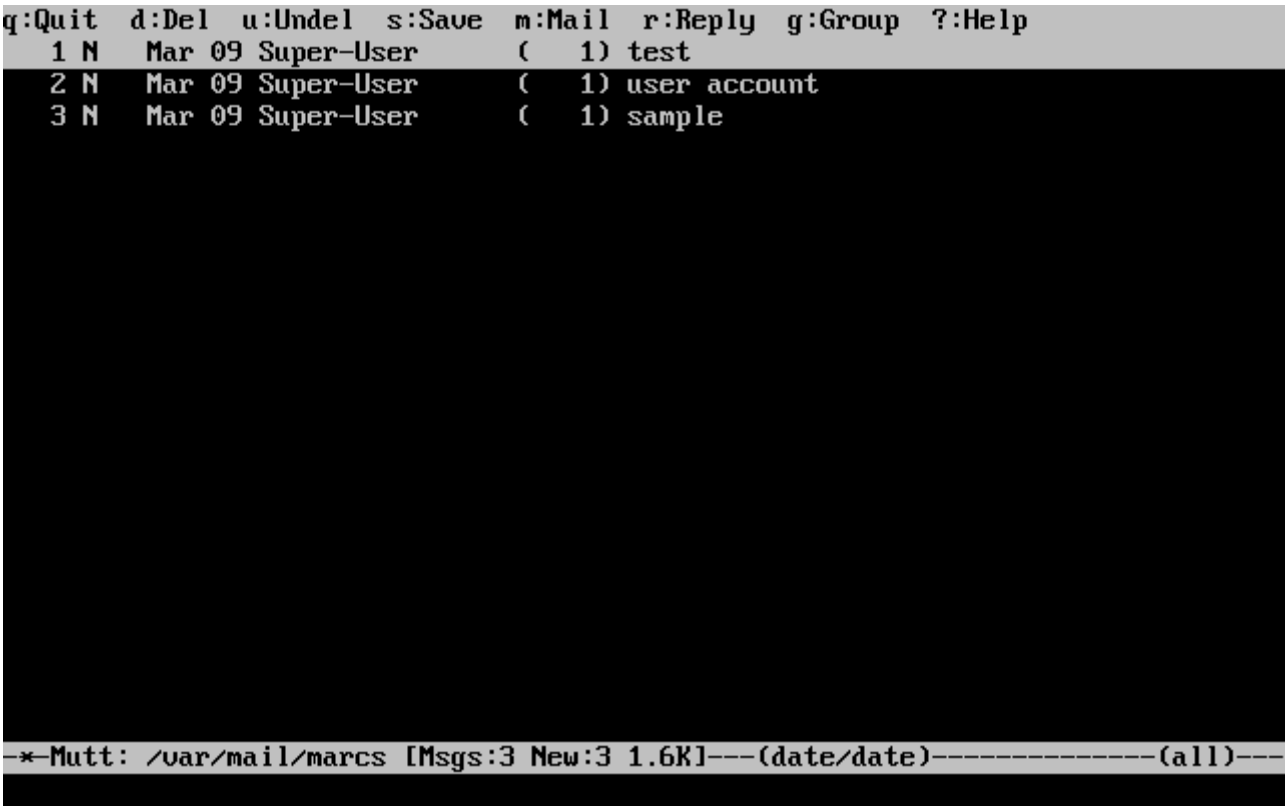
- mutt kann den Verlauf einer Diskussion (threading) darstellen.
- Unterstützung von PGP für das digitale signieren und verschlüsseln von E-Mail.
- MIME-Unterstützung.
- Maildir-Unterstützung.
- mutt lässt sich im höchsten Maße an lokale Bedürfnisse anpassen.

Mehr über mutt erfahren Sie auf der Seite http://www.mutt.org.

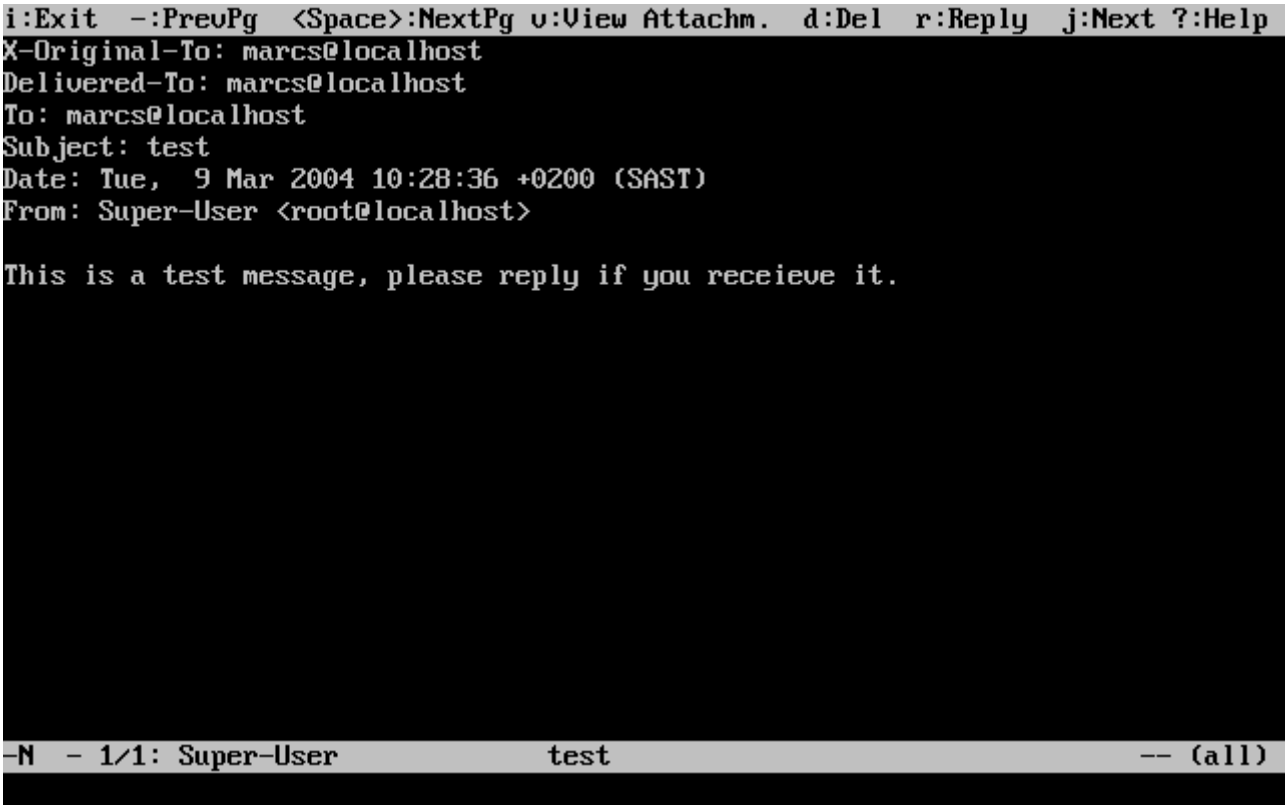
mutt kann über den Port mail/mutt installiert werden. Nachdem der Port installiert ist, kann mutt mit dem folgenden Befehl gestartet werden:

```
% mutt
```

mutt liest automatisch den Inhalt des Benutzer-Postfachs im Verzeichnis /var/mail. Sind keine E-Mails vorhanden, wartet mutt auf Benutzereingaben. Das folgende Beispiel zeigt, wie mutt eine Nachrichten-Liste darstellt:




Um eine E-Mail zu lesen, wählen Sie die Nachricht einfach mit den Pfeiltasten aus und drücken `Enter`. mutt zeigt E-Mails wie folgt an:

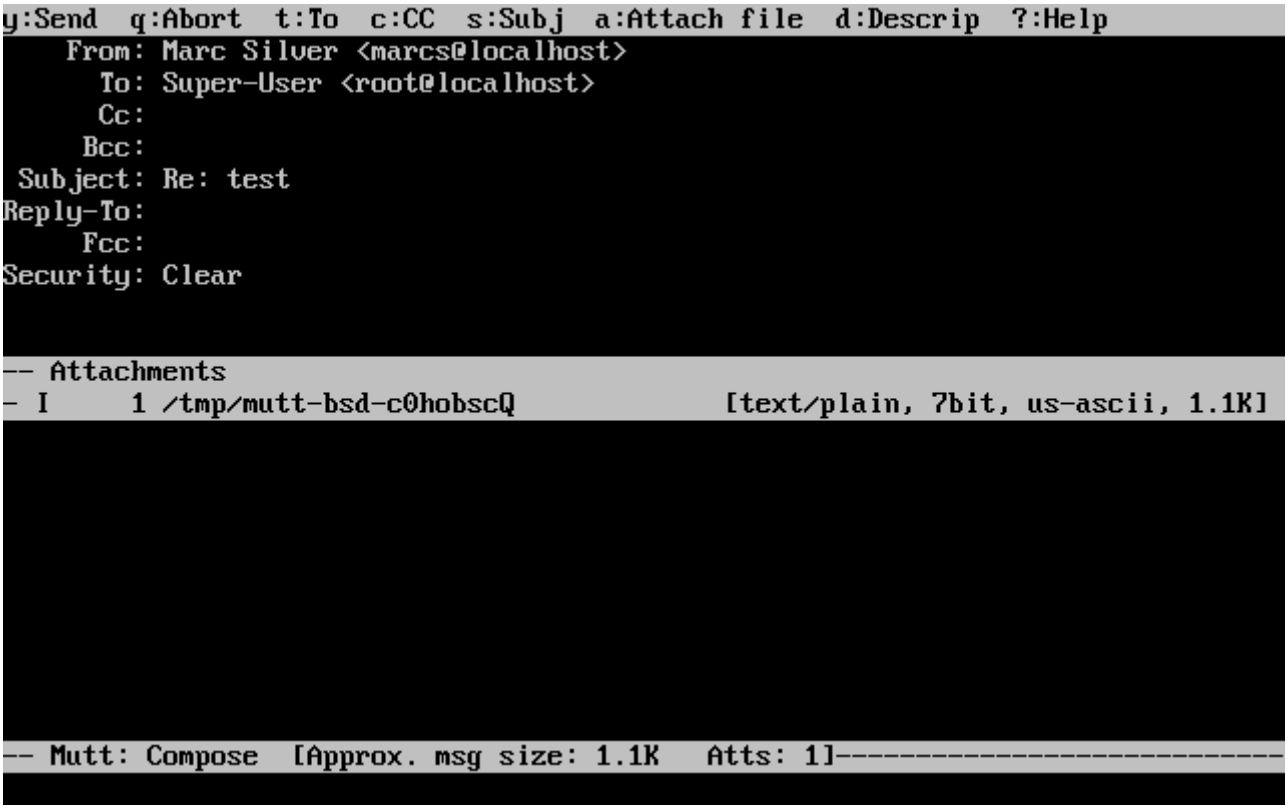



Änlich wie mail(1), kann auch mutt verwendet werden, um nur dem Absender, oder auch allen anderen Empfängern zu antworten. Um nur dem Absender der E-Mail zu antworten, drücken Sie `r`. Um sowohl dem Absender, als auch allen anderen Empfängern zu antworten, drücken Sie `g`.



Zum Erstellen oder zum Beantworten von E-Mails ruft mutt den Editor vi(1) auf. Jeder Benutzer kann diese Einstellung anpassen, indem die Variable `editor` in .muttrc im Heimatverzeichnis gesetzt wird, oder die Umgebungsvariable `EDITOR` entsprechend angepasst wird. Weitere Informationen zur Konfiguration von mutt finden Sie unter http://www.mutt.org/.

Drücken Sie `m`, um eine neue Nachricht zu verfassen. Nachdem der Betreff (subject) eingegeben wurde, startet mutt den vi(1) und die Nachricht kann verfasst werden. Wenn Sie fertig sind, speichern Sie die Nachricht und verlassen den vi(1). mutt wird dann wieder aktiv und zeigt eine Zusammenfassung der zu sendenden Nachricht an. Drücken Sie `y`, um die E-Mail zu versenden. Der nachstehende Bildschirmabzug zeigt die Zusammenfassung der E-Mail:




mutt verfügt über eine umfangreiche Hilfestellung. Aus fast jedem Menü können Hilfeseiten mit `?` aufgerufen werden. In der oberen Statuszeile werden zudem die verfügbaren Tastenkombinationen angezeigt.

### 51.11.3. alpine

alpine wendet sich an Anfänger bietet aber ebenfalls einige Funktionen für Profis.

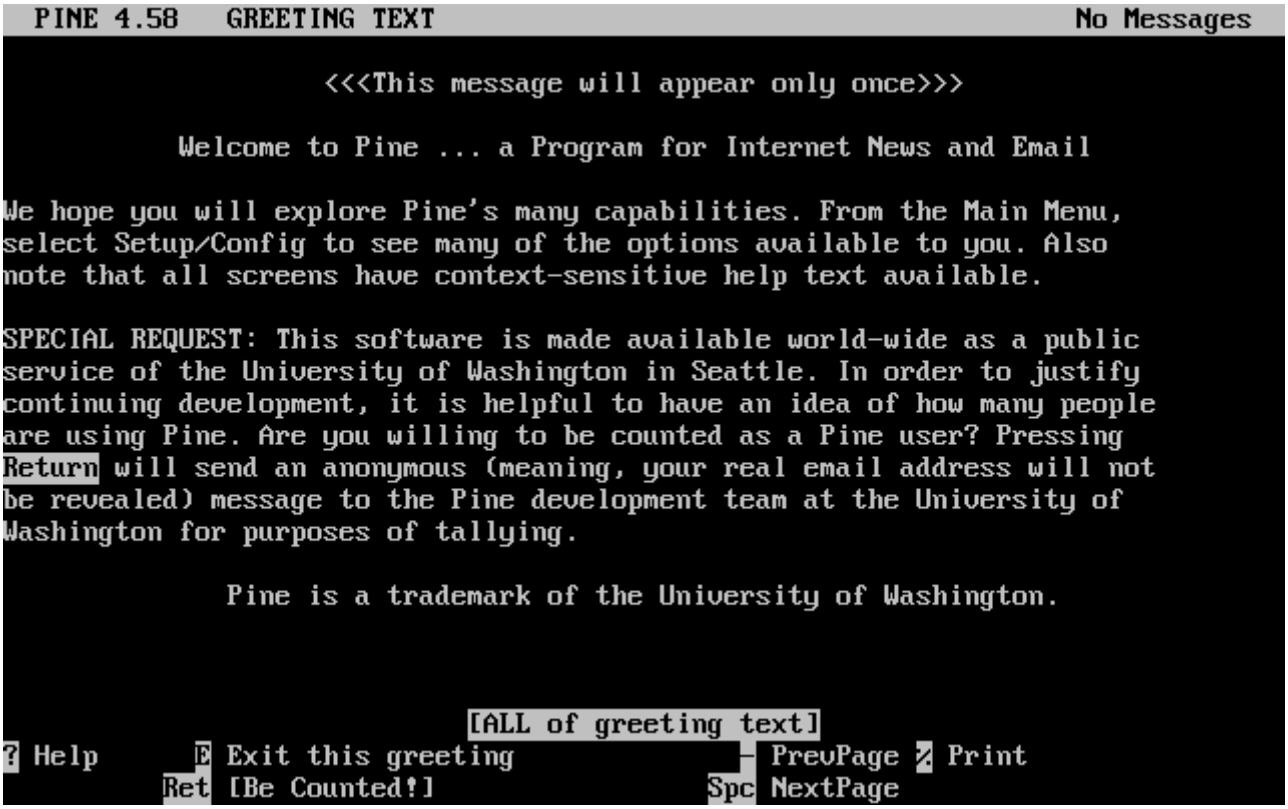


In der Vergangenheit wurden in alpine mehrere Schwachstellen gefunden. Die Schwachstellen gestatteten entfernten Benutzern, durch das Versenden einer besonders verfassten E-Mail, Programme auf dem lokalen System laufen zu lassen. Alle *bekannten* Schwachstellen sind beseitigt worden, doch wird im Quellcode von alpine ein sehr riskanter Programmierstil verwendet, sodass der FreeBSD-Security-Officer von weiteren unbekannten Schwachstellen ausgeht. Benutzer installieren alpine auf eigene Verantwortung!

Der Port mail/alpine enthält die aktuelle Version von alpine. Nach der Installation können Sie alpine mit dem nachstehenden Kommando starten:

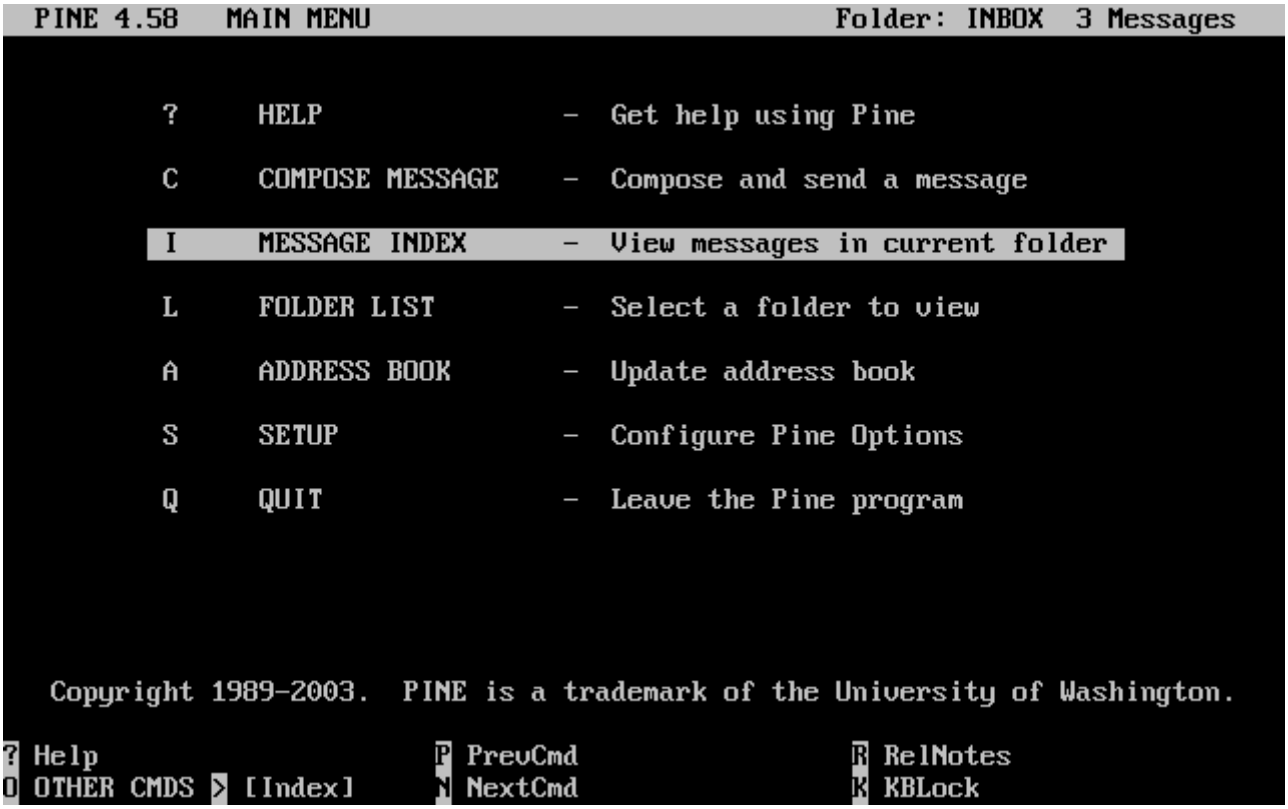
```
% alpine
```

Beim ersten Start von alpine, zeigt das Programm eine Seite mit einer kurzen Einführung an. Um die alpine-Benutzer zu zählen, bitten die Entwickler auf dieser Seite um eine anonyme E-Mail. Sie können diese anonyme E-Mail senden, indem Sie `Enter` drücken oder den Begrüßungsbildschirm mit der Taste `E` verlassen, ohne die anonyme E-Mail zu senden. Der Begrüßungsbildschirm sieht wie folgt aus:

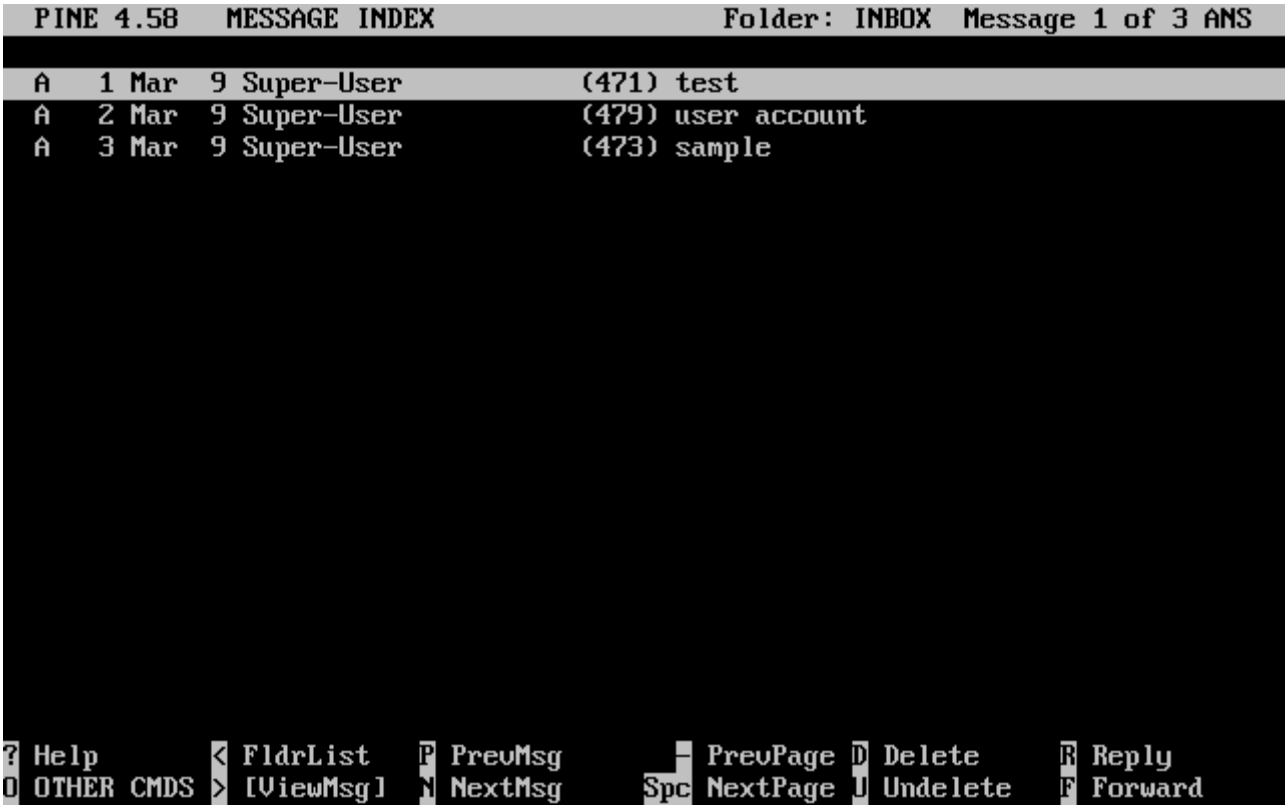



Nach dem Begrüßungsbildschirm wird das Hauptmenü dargestellt, das sich mit den Pfeiltasten bedienen lässt. Über Tastenkombinationen können aus dem Hauptmenü neue E-Mails erstellt, Postfächer angezeigt und das Adressbuch verwaltet werden. Unterhalb des Menüs werden die Tastenkombinationen für die verfügbaren Aktionen angezeigt.

In der Voreinstellung öffnet alpine das Verzeichnis inbox. Die Taste `I` oder der Menüpunkt MESSAGE INDEX führt zu einer Nachrichten-Liste:



Die Liste zeigt die Nachrichten im Arbeitsverzeichnis. Sie können Nachrichten mit den Pfeiltasten markieren. Um eine Nachricht zu lesen, drücken Sie `Enter`.

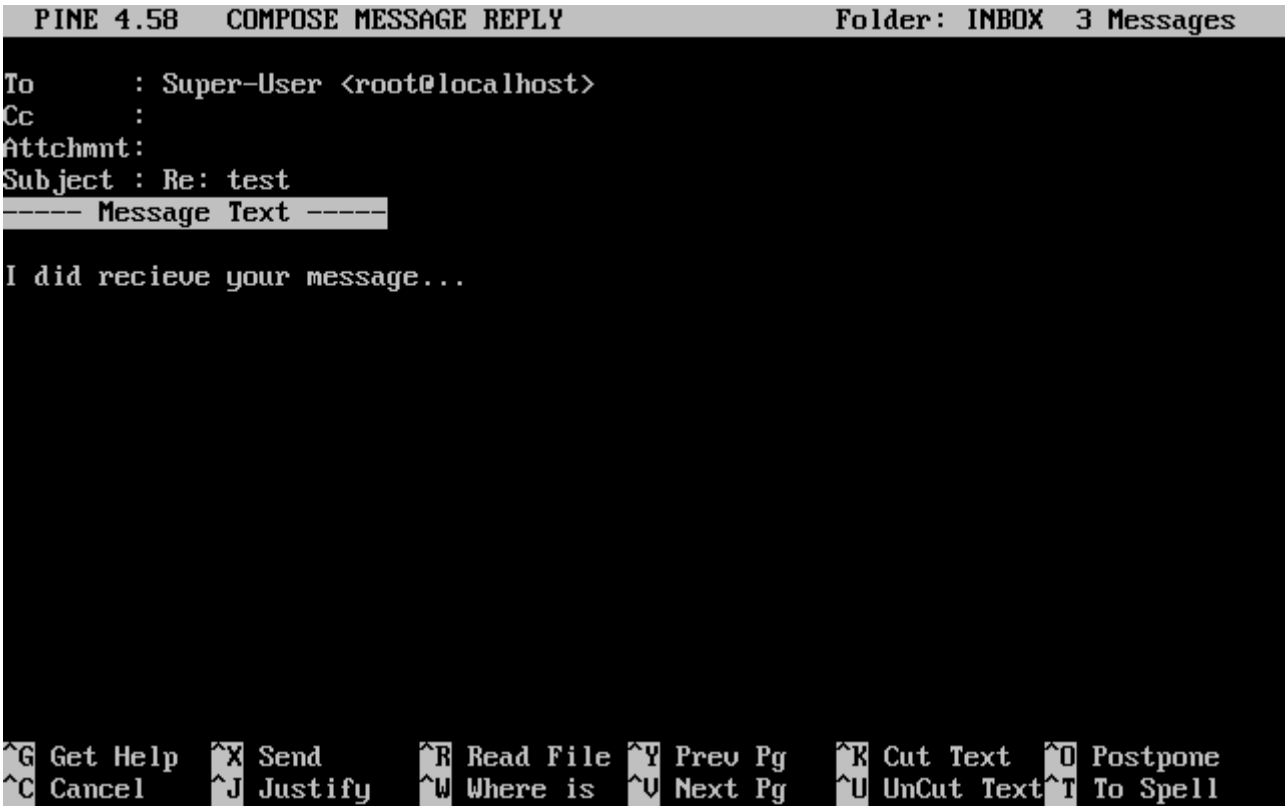



Im nächsten Bildschirmabzug sehen Sie, wie alpine eine Nachricht darstellt. Die unteren Bildschirmzeilen zeigen die verfügbaren Tastenkombinationen. Mit `r` können Sie zum Beispiel auf die gerade angezeigte Nachricht antworten.




Zum Antworten auf eine E-Mail wird in alpine der Editor pico, der mit installiert wird, benutzt. pico

ist leicht zu bedienen und gerade für Anfänger besser geeignet als vi(1) oder mail(1). Die Antwort wird mit der Tastenkombination Ctrl + X versendet. Vor dem Versand bittet alpine noch um eine Bestätigung.




Über den Menüpunkt SETUP des Hauptmenüs können Sie alpine an Ihre Bedürfnisse anpassen. Erläuterungen dazu finden Sie auf der Seite http://www.washington.edu/pine/.

## 51.12. E-Mails mit fetchmail abholen

fetchmail ist ein vollwertiger IMAP- und POP-Client. Mit fetchmail können Benutzer E-Mails von entfernten IMAP- und POP-Servern in leichter zugängliche lokale Postfächer laden. fetchmail wird aus dem Port mail/fetchmail installiert. Das Programm bietet unter anderem folgende Funktionen:

- fetchmail beherrscht die Protokolle POP3, APOP, KPOP, IMAP, ETRN und ODMR.
- E-Mails können mit SMTP weiterverarbeitet werden. Dadurch ist garantiert, dass Filter, Weiterleitungen und Aliase weiterhin funktionieren.
- Das Programm kann als Dienst laufen und periodisch neue Nachrichten abrufen.
- fetchmail kann mehrere Postfächer abfragen und je nach Konfiguration die E-Mails an verschiedene lokale Benutzer zustellen.

Dieser Abschnitt erklärt einige grundlegende Funktionen von fetchmail. Das Programm benötigt eine Konfigurationsdatei .fetchmailrc im Heimatverzeichnis des Benutzers. In dieser Datei werden Informationen über Server wie auch Benutzerdaten und Passwörter hinterlegt. Wegen des kritischen Inhalts dieser Datei ist es ratsam, diese nur für den Benutzer lesbar zu machen:

```
% chmod 600 .fetchmailrc
```

Die folgende .fetchmailrc zeigt, wie das Postfach eines einzelnen Benutzers mit POP heruntergeladen wird. fetchmail wird angewiesen, eine Verbindung zu `example.com` herzustellen und sich dort als Benutzer `joesoap` mit dem Passwort `XXX` anzumelden. Das Beispiel setzt voraus, dass der Benutzer `joesoap` auch auf dem lokalen System existiert.

```
poll example.com protocol pop3 username "joesoap" password "XXX"
```

Im folgenden Beispiel werden mehrere POP- und IMAP-Server benutzt. Wo notwendig, werden E-Mails auf andere lokale Konten umgeleitet:

```
poll example.com proto pop3:
user "joesoap", with password "XXX", is "jsoap" here;
user "andrea", with password "XXXX";
poll example2.net proto imap:
user "john", with password "XXXXX", is "myth" here;
```

fetchmail kann als Dämon gestartet werden. Verwendet wird dazu die Kommandozeilenoption `-d` gefolgt von einer Zeitspanne in Sekunden, die angibt, wie oft die Server aus .fetchmailrc abgefragt werden sollen. Mit dem nachstehenden Befehl fragt fetchmail die Server alle 600 Sekunden ab:

```
% fetchmail -d 600
```

Mehr über fetchmail erfahren Sie auf der Seite http://www.fetchmail.info/.

## 51.13. E-Mails mit procmail filtern

procmail ist ein mächtiges Werkzeug, mit dem sich eingehende E-Mails filtern lassen. Benutzer können Regeln für eingehende E-Mails definieren, die E-Mails zu anderen Postfächern oder anderen E-Mail-Adressen umleiten. procmail befindet sich im Port mail/procmail. procmail kann leicht in die meisten MTAs integriert werden. Lesen Sie dazu bitte die Dokumentation des verwendeten MTAs. Alternativ kann procmail in das E-Mail-System eingebunden werden, indem die nachstehende Zeile in die Datei .forward im Heimatverzeichnis eines Benutzers eingefügt wird:

```
"|exec /usr/local/bin/procmail || exit 75"
```

Der folgende Abschnitt zeigt einige einfache procmail-Regeln sowie eine kurze Beschreibung dessen, was sie tun. Regeln müssen in .procmailrc im Heimatverzeichnis des Benutzers eingefügt werden.

Den Großteil dieser Regeln finden Sie auch in procmailex(5).

Um E-Mails von user@example.com an die externe Adresse goodmail@example2.com weiterzuleiten:

```
:0
* ^From.*user@example.com
! goodmail@example2.com
```

Um E-Mails, die kürzer als 1000 Bytes sind, an goodmail@example2.com weiterzuleiten:

```
:0
* < 1000
! goodmail@example2.com
```

Um E-Mails, die an alternate@example.com geschickt werden, im Postfach alternate zu speichern:

```
:0
* ^TOalternate@example.com
alternate
```

Um E-Mails, die im Betreff Spam enthalten, nach /dev/null zu verschieben:

```
:0
^Subject:.*Spam
/dev/null
```

Zuletzt ein nützliches Rezept, das eingehende E-Mails von den FreeBSD.org-Mailinglisten in ein separates Postfach für jede Liste einsortiert:

```
:0
* ^Sender:.owner-freebsd-\/[^@]+@FreeBSD.ORG
{
    LISTNAME=${MATCH}
    :0
    * LISTNAME??^\/[^@]+
    FreeBSD-${MATCH}
}
```

# Kapitel 52. Netzwerkserver

## 52.1. Übersicht

Dieses Kapitel beschreibt einige der häufiger verwendeten Netzwerkdienste auf UNIX®-Systemen. Dazu zählen Installation und Konfiguration sowie Test und Wartung verschiedener Netzwerkdienste. Zusätzlich sind im ganzen Kapitel Beispielkonfigurationen als Referenz enthalten.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- Den inetd-Daemon konfigurieren können.
- Wissen, wie das Network File System (NFS) eingerichtet wird.
- Einen Network Information Server (NIS) einrichten können, um damit Benutzerkonten im Netzwerk zu verteilen.
- Wissen, wie Sie FreeBSD einrichten, um als LDAP-Server oder -Client zu agieren.
- Rechner durch Nutzung von DHCP automatisch für ein Netzwerk konfigurieren können.
- In der Lage sein, einen Domain Name Server (DNS) einzurichten.
- Den ApacheHTTP-Server konfigurieren können.
- Wissen, wie man einen File Transfer Protocol (FTP)-Server einrichtet.
- Mit Samba einen Datei- und Druckserver für Windows®-Clients konfigurieren können.
- Unter Nutzung des NTP-Protokolls Datum und Uhrzeit synchronisieren sowie einen Zeitserver installieren können.
- Wissen, wie iSCSI eingerichtet wird.

Dieses Kapitel setzt folgende Grundkenntnisse voraus:

- /etc/rc-Skripte.
- Netzwerkterminologie
- Installation zusätzlicher Software von Drittanbietern (Installieren von Anwendungen: Pakete und Ports).

## 52.2. Der inetd"Super-Server"

Der inetd(8)-Daemon wird manchmal auch als "Internet Super-Server" bezeichnet, weil er Verbindungen für viele Dienste verwaltet. Anstatt mehrere Anwendungen zu starten, muss nur der inetd-Dienst gestartet werden. Wenn eine Verbindung für einen Dienst eintrifft, der von inetd verwaltet wird, bestimmt inetd, welches Programm für die eingetroffene Verbindung zuständig ist, aktiviert den entsprechenden Prozess und reicht den Socket an ihn weiter. Der Einsatz von inetd an Stelle viele einzelner Daemonen kann auf nicht komplett ausgelasteten Servern zu einer Verringerung der Systemlast führen.

inetd wird vor allem dazu verwendet, andere Daemonen zu aktivieren, einige Protokolle werden aber auch intern verwaltet. Dazu gehören chargen, auth, time, echo, discard sowie daytime.

Dieser Abschnitt beschreibt die Konfiguration von inetd.

### 52.2.1. Konfigurationsdatei

Die Konfiguration von inetd erfolgt über /etc/inetd.conf Jede Zeile dieser Datei repräsentiert eine Anwendung, die von inetd gestartet werden kann. In der Voreinstellung beginnt jede Zeile mit einem Kommentar (#), was bedeutet dass inetd keine Verbindungen für Anwendungen akzeptiert. Entfernen Sie den Kommentar am Anfang der Zeile, damit inetd Verbindungen für diese Anwendung entgegennimmt.

Nachdem Sie die Änderungen gespeichert haben, fügen Sie folgende Zeile in /etc/rc.conf ein, damit inetd bei Booten automatisch gestartet wird:

```
inetd_enable="YES"
```

Starten Sie jetzt inetd, so dass er Verbindungen für die von Ihnen konfigurierten Dienste entgegennimmt:

```
# service inetd start
```

Sobald inetd gestartet ist, muss der Dienst benachrichtigt werden, wenn eine Änderung in /etc/inetd.conf gemacht wird:

*Beispiel 45. Die Konfigurationsdatei von inetd neu einlesen*

```
# service inetd reload
```

Normalerweise müssen Sie lediglich den Kommentar vor der Anwendung entfernen. In einigen Situationen kann es jedoch sinnvoll sein, den Eintrag weiter zu bearbeiten.

Als Beispiel dient hier der Standardeintrag für ftpd(8) über IPv4:

```
ftp     stream  tcp     nowait  root    /usr/libexec/ftpd       ftpd -l
```

Die sieben Spalten in diesem Eintrag haben folgende Bedeutung:

```
service-name
socket-type
protocol
{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]
user[:group][/login-class]
server-program
server-program-arguments
```

**service-name**

Der Dienstname eines bestimmten Daemons. Er muss einem in /etc/services aufgelisteten Dienst entsprechen. Hier wird festgelegt, auf welchen Port inetd eingehende Verbindungen für diesen Dienst entgegennimmt. Wenn ein neuer Dienst benutzt wird, muss er zuerst in /etc/services eingetragen werden.

**socket-type**

Entweder `stream`, `dgram`, `raw`, oder `seqpacket`. Nutzen Sie `stream` für TCP-Verbindungen und `dgram` für UDP-Dienste.

**protocol**

Benutzen Sie eines der folgenden Protokolle:

| Protokoll | Bedeutung |
|---|---|
| tcp oder tcp4 | TCP (IPv4) |
| udp oder udp4 | UDP (IPv4) |
| tcp6 | TCP (IPv6) |
| udp6 | UDP (IPv6) |
| tcp46 | TCP sowohl unter IPv4 als auch unter IPv6 |
| udp46 | UDP sowohl unter IPv4 als auch unter IPv6 |

**{wait|nowait}[/max-child[/max-connections-per-ip-per-minute[/max-child-per-ip]]]**

In diesem Feld muss `wait` oder `nowait` angegeben werden. `max-child`, `max-connections-per-ip-per-minute` sowie `max-child-per-ip` sind optional.

`wait|nowait` gibt an, ob der Dienst seinen eigenen Socket verwalten kann oder nicht. `dgram` -Sockets müssen `wait` verwenden, während Daemonen mit `stream`-Sockets, die normalerweise auch aus mehreren Threads bestehen, `nowait` verwenden sollten. `wait` gibt in der Regel mehrere Sockets an einen einzelnen Daemon weiter, während `nowait` für jeden neuen Socket einen Childdaemon erzeugt.

Die maximale Anzahl an Child-Daemonen, die inetd erzeugen kann, wird durch die Option `max-child` festgelegt. Wenn ein bestimmter Daemon 10 Instanzen benötigt, wird der Wert `/10` hinter die Option `nowait` gesetzt. Der Wert `/0` gibt an, das es keine Beschränkung gibt.

`max-connections-per-ip-per-minute` legt die maximale Anzahl von Verbindungsversuchen pro Minute fest, die von einer bestimmten IP-Adresse aus unternommen werden können. Sobald das Limit erreicht ist, werden weitere Verbindungen von dieser IP-Adresse geblockt, bis die Minute vorüber ist. Ein Wert von `/10` würde die maximale Anzahl der Verindungsversuche einer bestimmten IP-Adresse auf zehn Versuche in der Minute beschränken. `max-child-per-ip` legt fest, wie viele Child-Daemonen von einer bestimmten IP-Adresse aus gestartet werden können. Durch diese Optionen lassen sich Ressourcenverbrauch sowie die Auswirkungen eines `Denial of Service (DoS)`-Angriffs begrenzen.

Ein Beispiel finden Sie in den Voreinstellungen für fingerd(8):

```
finger stream  tcp     nowait/3/10 nobody /usr/libexec/fingerd fingerd -k -s
```

**user**

Der Benutzername, unter dem der jeweilige Daemon laufen soll. Meistens laufen Daemonen als `root`, `daemon` oder `nobody`.

**server-program**

Der vollständige Pfad des Daemons. Wird der Daemon von inetd intern bereitgestellt, verwenden Sie `internal`.

**server-program-arguments**

Dieser Eintrag legt die Argumente fest, die bei der Aktivierung an den Daemon übergeben werden. Wenn es sich beim Daemon um einen internen Dienst handelt, verwenden Sie wiederum `internal`.

### 52.2.2. Kommandozeilenoptionen

Wie die meisten anderen Server-Daemonen lässt sich auch inetd über verschiedene Optionen steuern. In der Voreinstellung wird inetd mit `-wW -C 60` gestartet. Durch das Setzen dieser Werte wird das TCP-Wrapping für alle inetd-Dienste aktiviert. Zudem wird verhindert, dass eine IP-Adresse eine Dienst öfter als 60 Mal pro Minute anfordern kann.

Um die Voreinstellungen für inetd zu ändern, fügen Sie einen Eintrag für `inetd_flags` in /etc/rc.conf hinzu. Wenn inetd bereits ausgeführt wird, starten Sie ihn mit `service inetd restart` neu.

Die verfügbaren Optionen sind:

**-c maximum**

Legt die maximale Anzahl von parallelen Aufrufen eines Dienstes fest; in der Voreinstellung gibt es keine Einschränkung. Diese Einstellung kann für jeden Dienst durch Setzen des Parameters `max-child` in /etc/inetd.conf festgelegt werden.

**-C rate**

Legt fest, wie oft ein Dienst von einer einzelnen IP-Adresse in einer Minute aufgerufen werden kann; in der Voreinstellung gibt es keine Einschränkung. Dieser Wert kann für jeden Dienst durch das Setzen des Parameters `max-connections-per-ip-per-minute` in /etc/inetd.conf festgelegt werden.

**-R rate**

Legt fest, wie oft ein Dienst in der Minute aktiviert werden kann; in der Voreinstellung sind dies `256` Aktivierungen pro Minute. Ein Wert von `0` erlaubt unbegrenzt viele Aktivierungen.

**-s maximum**

Legt fest, wie oft ein Dienst in der Minute von einer einzelnen IP-Adresse aus aktiviert werden kann; in der Voreinstellung gibt es hier keine Beschränkung. Diese Einstellung kann für jeden Dienst durch die Angabe von `max-child-per-ip` in /etc/inetd.conf angepasst werden.

Es sind noch weitere Optionen verfügbar. Eine vollständige Liste der Optionen finden Sie in inetd(8).

### 52.2.3. Sicherheitsbedenken

Viele Daemonen, die von inetd verwaltet werden, sind nicht auf Sicherheit bedacht. Einige Damonen, wie beispielsweise fingerd, liefern Informationen, die für einen Angreifer nützlich sein könnten. Aktivieren Sie nur erforderliche Dienste und überwachen Sie das System auf übermäßige Verbindungsversuche. `max-connections-per-ip-per-minute`, `max-child` und `max-child-per-ip` können verwendet werden, um solche Angriffe zu begrenzen.

TCP-Wrapper ist in der Voreinstellung aktiviert. Lesen Sie hosts_access(5), wenn Sie weitere Informationen zum Setzen von TCP-Beschränkungen für verschiedene von inetd aktivierte Daemonen benötigen.

## 52.3. Network File System (NFS)

FreeBSD unterstützt das Netzwerkdateisystem NFS, das es einem Server erlaubt, Dateien und Verzeichnisse über ein Netzwerk mit Clients zu teilen. Mit NFS können Benutzer und Programme auf Daten entfernter Systeme zugreifen, und zwar so, als ob es sich um lokal gespeicherte Daten handeln würde.

Die wichtigsten Vorteile von NFS sind:

- Daten, die sonst auf jeden Client dupliziert würden, können an einem zentralen Ort aufbewahrt, und von den Clients über das Netzwerk aufgerufen werden.
- Verschiedene Clients können auf ein gemeinsames Verzeichnis /usr/ports/distfiles zugreifen. Die gemeinsame Nutzung dieses Verzeichnisses ermöglicht einen schnellen Zugriff auf die Quelldateien, ohne sie auf jede Maschine zu kopieren zu müssen.
- In größeren Netzwerken ist es praktisch, einen zentralen NFS-Server einzurichten, auf dem die Heimatverzeichnisse der Benutzer gespeichert werden. Dadurch steht den Benutzern immer das gleiche Heimatverzeichnis zur Verfügung, unabhängig davon, an welchem Client im Netzwerk sie sich anmelden.
- Die Verwaltung der NFS-Exporte wird vereinfacht. Zum Beispiel gibt es dann nur noch ein Dateisystem, für das Sicherheits- oder Backup-Richtlinien festgelegt werden müssen.
- Wechselmedien können von anderen Maschinen im Netzwerk verwendet werden. Dies reduziert die Anzahl von Geräten im Netzwerk und bietet einen zentralen Ort für die Verwaltung. Oft ist es einfacher, über ein zentrales Installationsmedium Software auf mehreren Computern zu installieren.

NFS besteht aus einem Server und einem oder mehreren Clients. Der Client greift über das Netzwerk auf die Daten zu, die auf dem Server gespeichert sind. Damit dies korrekt funktioniert, müssen einige Prozesse konfiguriert und gestartet werden:

Folgende Daemonen müssen auf dem Server ausgeführt werden:

| Daemon | Beschreibung |
|---|---|
| nfsd | Der NFS-Daemon. Er bearbeitet Anfragen der NFS-Clients. |
| mountd | Der NFS-Mount-Daemon. Er bearbeitet die Anfragen von `nfsd`. |
| rpcbind | Der Portmapper-Daemon. Durch ihn erkennen die NFS-Clients, welchen Port der NFS-Server verwendet. |

Der Einsatz von nfsiod(8) ist nicht zwingend erforderlich, kann aber die Leistung auf dem Client verbessern.

### 52.3.1. Konfiguration des Servers

Die Dateisysteme, die der NFS-Server exportieren soll, werden in /etc/exports festgelegt. Jede Zeile in dieser Datei beschreibt ein zu exportierendes Dateisystem, Clients, die darauf Zugriff haben sowie alle Zugriffsoptionen. Die Optionen eines auf einen anderen Rechner exportierten Dateisystems müssen alle in einer Zeile stehen. Wird in einer Zeile kein Rechner festgelegt, dürfen alle Clients im Netzwerk das exportierte Dateisystem einhängen.

Wie Dateisysteme exportiert werden, ist in der folgenden /etc/exports zu sehen. Diese Beispiele müssen natürlich an die Arbeitsumgebung und die Netzwerkkonfiguration angepasst werden. Es existieren viele verschiedene Optionen, allerdings werden hier nur wenige von ihnen erwähnt. Eine vollständige Liste der Optionen finden Sie in exports(5).

Dieses Beispiel exportiert /cdrom für drei Clients, *alpha*, *bravo* und *charlie*:

```
/cdrom -ro alpha bravo charlie
```

Die Option `-ro` kennzeichnet das exportierte Dateisystem als schreibgeschützt. Dadurch sind Clients nicht in der Lage, das exportierte Dateisystem zu verändern. Dieses Beispiel geht davon aus, dass die Hostnamen entweder über DNS oder über /etc/hosts aufgelöst werden können. Lesen Sie hosts(5) falls das Netzwerk über keinen DNS-Server verfügt.

Das nächste Beispiel exportiert /home auf drei durch IP-Adressen bestimmte Clients. Diese Einstellung kann für Netzwerke ohne DNS-Server und /etc/hosts nützlich sein. Die Option `-alldirs` ermöglicht es, auch Unterverzeichnisse als Mountpunkte festzulegen. Dies bedeutet aber nicht, dass alle Unterverzeichnisse eingehängt werden, vielmehr wird es dem Client ermöglicht, nur diejenigen Verzeichnisse einzuhängen, die auch benötigt werden.

```
/usr/home  -alldirs  10.0.0.2 10.0.0.3 10.0.0.4
```

Das nächste Beispiel exportiert /a, damit Clients von verschiedenen Domänen auf das Dateisystem zugreifen können. Die Option `-maproot=root` erlaubt es dem Benutzer `root` des Clients, als `root` auf das exportierte Dateisystem zu schreiben. Wenn diese Option nicht gesetzt ist, wird der `root`

-Benutzer des Clients dem nobody-Konto des Servers zugeordnet und unterliegt somit den Zugriffsbeschränkungen dieses Kontos.

```
/a  -maproot=root  host.example.com box.example.org
```

Ein Client kann für jedes Dateisystem nur einmal definiert werden. Wenn beispielsweise /usr ein gesondertes Dateisystem ist, dann wären die folgenden Einträge falsch, da in beiden Einträgen der gleiche Rechner angegeben wird:

```
#Nicht erlaubt, wenn /usr ein einziges Dateisystem ist
/usr/src   client
/usr/ports client
```

Das richtige Format für eine solche Situation ist:

```
/usr/src /usr/ports  client
```

Das Folgende ist ein Beispiel für eine gültige Exportliste, in der /usr und /exports lokale Dateisysteme sind:

```
# Export src and ports to client01 and client02, but only
# client01 has root privileges on it
/usr/src /usr/ports -maproot=root    client01
/usr/src /usr/ports                  client02
# The client machines have root and can mount anywhere
# on /exports. Anyone in the world can mount /exports/obj read-only
/exports -alldirs -maproot=root      client01 client02
/exports/obj -ro
```

Damit die vom NFS-Server benötigen Prozesse beim Booten gestartet werden, fügen Sie folgende Optionen in /etc/rc.conf hinzu:

```
rpcbind_enable="YES"
nfs_server_enable="YES"
mountd_enable="YES"
```

Der Server kann jetzt mit diesem Kommando gestartet werden:

```
# service nfsd start
```

Wenn der NFS-Server startet, wird auch mountd automatisch gestartet. Allerdings liest mountd /etc/exports nur, wenn der Server gestartet wird. Um nachfolgende Änderungen an /etc/exports wirksam werden zu lassen, kann mountd angewiesen werden, die Datei neu einzulesen:

```
# service mountd reload
```

### 52.3.2. Konfiguration des Clients

Um den NFS-Client zu aktivieren, setzen Sie folgende Option in /etc/rc.conf auf jedem Client:

```
nfs_client_enable="YES"
```

Der Client ist nun in der Lage, ein entferntes Dateisystem einzuhängen. In diesen Beispielen ist der Name des Servers `server` und der Name des Clients `client`. Fügen Sie folgenden Befehl aus, um das Verzeichnis /home vom `server` auf dem `client` ins Verzeichnis /mnt einzuhängen:

```
# mount server:/home /mnt
```

Die Dateien und Verzeichnisse in /home stehen dem Rechner `client` nun im Verzeichnis /mnt zur Verfügung.

Um ein entferntes Dateisystem bei jedem Systemstart automatisch einzuhängen, fügen Sie das Dateisystem in /etc/fstab ein:

```
server:/home    /mnt    nfs rw  0   0
```

fstab(5) enthält eine Beschreibung aller Optionen.

### 52.3.3. Dateien sperren (Locking)

Einige Anwendungen erfordern die Sperrung von Dateien, damit sie korrekt arbeiten. Um diese Sperre zu aktivieren, müssen diese Zeilen in /etc/rc.conf sowohl auf dem Client als auch auf dem Server hinzugefügt werden:

```
rpc_lockd_enable="YES"
rpc_statd_enable="YES"
```

Danach starten Sie die beiden Anwendungen:

```
# service lockd start
# service statd start
```

Wenn keine Dateisperren zwischen den NFS-Clients und dem NFS-Server benötigt werden, können Sie den NFS-Client durch die Übergabe der Option `-L` an mount zu einer lokalen Sperrung von Dateien zwingen. Weitere Details finden Sie in mount_nfs(8).

### 52.3.4. Automatisches Einhängen mit autofs(5)



autofs(5) wird seit FreeBSD 10.1-RELEASE unterstützt. Um die Funktionalität des automatischen Einhängens in älteren FreeBSD-Versionen zu benutzen, verwenden Sie stattdessen amd(8). In diesem Kapitel wird nur das automatische Einhängen mit Hilfe von autofs(5) beschrieben.

autofs(5) ist eine gebräuchliche Bezeichnung für verschiedene Komponenten, welche es erlauben, lokale und entfernte Dateisysteme automatisch einzuhängen, sobald auf eine Datei oder ein Verzeichnis in diesem Dateisystem zugegriffen wird. Es besteht aus einer Kernel-Komponente autofs(5) und mehreren Benutzerprogrammen: automount(8), automountd(8) und autounmountd(8). autofs(5) ist eine Alternative für amd(8) aus früheren FreeBSD-Versionen. amd(8) steht nach wie vor zur Verfügung, da beide Programme ein unterschiedliches Format verwenden. Das Format welches autofs(5) verwendet ist das gleiche wie bei anderen SVR4 Automountern, beispielsweise denen aus Solaris™, Mac OS® X und Linux®.

Das virtuelle autofs(5)-Dateisystem wird von automount(8) in einen bestimmten Mountpunkt eingehängt. Dies geschieht gewöhnlich während des Bootens.

Jedes Mal, wenn ein Prozess versucht auf eine Datei unterhalb des autofs(5)-Mountpunkts zuzugreifen, wird der Kernel den automountd(8)-Daemon benachrichtigen und den aktuellen Prozess anhalten. Der automountd(8)-Daemon wird dann die Anfrage des Kernels bearbeiten und das entsprechende Dateisystem einhängen. Anschließend wird der Daemon den Kernel benachrichtigen, dass der angehaltene Prozess wieder freigegeben werden kann. Der autounmountd(8)-Daemon hängt automatisch Dateisysteme nach einiger Zeit ab, sofern sie nicht mehr verwendet werden.

Die primäre Konfigurationsdatei von autofs ist /etc/auto_master. Sie enthält die einzelnen Zuordnungen zu den Mountpunkten. Eine Erklärung zu auto_master und der Syntax für die Zuordnungen finden Sie in auto_master(5).

Eine spezielle Automounter Zuordnung wird in /net eingehängt. Wenn auf eine Datei in diesem Verzeichnis zugegriffen wird, hängt autofs(5) einen bestimmten, entfernen Mountpunkt ein. Wenn beispielsweise auf eine Datei unterhalb von /net/foobar/usr zugegriffen werden soll, würde automountd(8) das exportierte Dateisystem /usr von dem Rechner `foobar` einhängen.

*Beispiel 46. Ein exportiertes Dateisystem mit autofs(5) in den Verzeichnisbaum einhängen*

In diesem Beispiel zeigt `showmount -e` die exportierten Dateisysteme des NFS-Servers `foobar`:

```
% showmount -e foobar
Exports list on foobar:
/usr                               10.10.10.0
/a                                 10.10.10.0
% cd /net/foobar/usr
```

Die Ausgabe von `showmount` zeigt das exportierte Dateisystem /usr. Wenn in das Verzeichnis

/host/foobar/usr gewechselt wird, fängt automountd(8) die Anforderung ab und versucht, den Rechnernamen `foobar` aufzulösen. Gelingt dies, wird automountd(8) automatisch das exportierte Dateisystem einhängen.

Um autofs(5) beim Booten zu aktivieren, fügen Sie diese Zeile in /etc/rc.conf ein:

```
autofs_enable="YES"
```

Danach kann autofs(5) gestartet werden:

```
# service automount start
# service automountd start
# service autounmountd start
```

Obwohl das Format von autofs(5) das gleiche ist wie in anderen Betriebssystemen, kann es wünschenswert sein, Informationen von anderen Betriebssystemen zu Rate zu ziehen, wie dieses Mac OS X Dokument.

Weitere Informationen finden Sie in den Manualpages automount(8), automountd(8), autounmountd(8) und auto_master(5).

# 52.4. Network Information System (NIS)

Das Network Information System (NIS) wurde entwickelt, um UNIX®-Systeme zentral verwalten zu können. Dazu zählen beispielsweise Solaris™, HP-UX, AIX®, Linux®, NetBSD, OpenBSD und FreeBSD. NIS war ursprünglich als *Yellow Pages* bekannt, aus markenrechtlichen Gründen wurde der Name aber geändert. Dies ist der Grund, warum NIS-Kommandos mit `yp` beginnen.

Bei NIS handelt es sich um ein RPC-basiertes Client/Server-System. Eine Gruppe von Rechnern greift dabei innerhalb einer NIS-Domäne auf gemeinsame Konfigurationsdateien zu. Dies erlaubt es einem Systemadministrator, NIS-Clients mit minimalem Aufwand einzurichten, sowie Änderungen an der Systemkonfiguration von einem zentralen Ort aus durchzuführen.

FreeBSD verwendet die Version 2 des NIS-Protokolls.

## 52.4.1. NIS-Begriffe und -Prozesse

Tabelle 30.1 fasst die Begriffe und Anwenderprozesse zusammen, die von NIS verwendet werden:

*Tabelle 25. NIS Begriffe*

| Begriff | Beschreibung |
|---|---|
| NIS-Domänenname | NIS-Masterserver und Clients benutzen einen gemeinsamen NIS-Domänennamen. In der Regel hat dieser Name nichts mit DNS zu tun. |

| Begriff | Beschreibung |
| --- | --- |
| rpcbind(8) | Dieser Dienst aktiviert RPC und muss gestartet sein, damit ein NIS-Server oder -Client ausgeführt werden kann. |
| ypbind(8) | Dieser Dienst "bindet" einen NIS-Client an seinen NIS-Server. Der Client bezieht den NIS-Domänennamen vom System und stellt über das RPC-Protokoll eine Verbindung zum NIS-Server her. ypbind ist der zentrale Bestandteil der Client-Server-Kommunikation in einer NIS-Umgebung. Wird der Dienst auf einem Client beendet, ist dieser nicht mehr in der Lage, auf den NIS-Server zuzugreifen. |
| ypserv(8) | Dies ist der Prozess für den NIS-Server. Wenn dieser Dienst nicht mehr läuft, kann der Server nicht mehr auf NIS-Anforderungen reagieren. Wenn ein Slaveserver existiert, kann dieser als Ersatz fungieren. Einige NIS-Systeme (allerdings nicht das von FreeBSD) versuchen allerdings erst gar nicht, sich mit einem anderen Server zu verbinden, wenn der Masterserver nicht mehr reagiert. Die einzige Lösung besteht darin, den Serverprozess oder den ypbind-Prozess auf dem Client neu zu starten. |
| rpc.yppasswdd(8) | Dieser Prozess läuft nur auf dem NIS-Masterserver. Es handelt sich um einen Daemonprozess, der es NIS-Clients ermöglicht, ihre NIS-Passwörter zu ändern. Wenn dieser Daemon nicht läuft, müssen sich die Benutzer am NIS-Masterserver anmelden und ihre Passwörter dort ändern. |

### 52.4.2. Arten von NIS-Rechnern

- NIS-Masterserver

  Dieser Server dient als zentraler Speicherort für Rechnerkonfigurationen. Zudem verwaltet er die maßgebliche Kopie, der von den NIS-Clients gemeinsam verwendeten Dateien. passwd, group, sowie verschiedene andere von den Clients verwendete Dateien existieren auf dem Masterserver. Obwohl ein Rechner auch für mehrere NIS-Domänen als Masterserver fungieren kann, wird diese Art von Konfiguration nicht behandelt, da sich dieser Abschnitt auf eine relativ kleine NIS-Umgebung konzentriert.

- NIS-Slaveserver

  NIS-Slaveserver verwalten Kopien der Daten des NIS-Masterservers um Redundanz zu bieten. Zudem entlasten Slaveserver den Masterserver: NIS-Clients verbinden sich immer mit dem NIS-

Server, welcher zuerst reagiert. Dieser Server kann auch ein Slaveserver sein.

- NIS-Clients

  NIS-Clients identifizieren sich gegenüber dem NIS-Server während der Anmeldung.

Mit NIS können Informationen aus verschiedenen Dateien von mehreren Rechnern gemeinsam verwendet werden. master.passwd, group, und hosts werden oft gemeinsam über NIS verwendet. Immer, wenn ein Prozess auf einem Client auf Informationen zugreifen will, die normalerweise in lokalen Dateien vorhanden wären, wird stattdessen eine Anfrage an den NIS-Server gestellt, an den der Client gebunden ist.

### 52.4.3. Planung

Dieser Abschnitt beschreibt eine einfache NIS-Umgebung, welche aus 15 FreeBSD-Maschinen besteht, für die keine zentrale Verwaltung existiert. Jeder Rechner hat also eine eigene Version von /etc/passwd und /etc/master.passwd. Diese Dateien werden manuell synchron gehalten; wird ein neuer Benutzer angelegt, so muss dies auf allen fünfzehn Rechnern manuell erledigt werden.

In Zukunft soll die Konfiguration wie folgt aussehen:

| Rechnername | IP-Adresse | Rechneraufgabe |
|---|---|---|
| `ellington` | `10.0.0.2` | NIS-Master |
| `coltrane` | `10.0.0.3` | NIS-Slave |
| `basie` | `10.0.0.4` | Workstation der Fakultät |
| `bird` | `10.0.0.5` | Clientrechner |
| `cli[1-11]` | `10.0.0.[6-17]` | Verschiedene andere Clients |

Wenn erstmalig ein NIS-Schema eingerichtet wird, sollte es im Voraus sorgfältig geplant werden. Unabhängig von der Größe des Netzwerks müssen einige Entscheidungen im Rahmen des Planungsprozesses getroffen werden.

#### Einen NIS-Domänennamen wählen

Wenn ein Client Informationen anfordert, ist in dieser Anforderung der Name der NIS-Domäne enthalten. Dadurch weiß jeder Server im Netzwerk, auf welche Anforderung er antworten muss. Stellen Sie sich den NIS-Domänennamen als einen Namen einer Gruppe von Rechnern vor.

Manchmal wird der Name der Internetdomäne auch für die NIS-Domäne verwendet. Dies ist allerdings nicht empfehlenswert, da es bei der Behebung von Problemen verwirrend sein kann. Der Name der NIS-Domäne sollte innerhalb des Netzwerks eindeutig sein. Hilfreich ist es, wenn der Name die Gruppe der in ihr zusammengefassten Rechner beschreibt. Die Kunstabteilung von Acme Inc. hätte daher vielleicht die NIS-Domäne "acme-art". Für dieses Beispiel wird der Name `test-domain` verwendet.

Es gibt jedoch auch Betriebssysteme, die als NIS-Domänennamen den Namen der Internetdomäne verwenden. Wenn dies für einen oder mehrere Rechner des Netzwerks zutrifft, *muss* der Name der Internetdomäne als NIS-Domänennamen verwendet werden.

#### Anforderungen an den Server

Bei der Wahl des NIS-Servers müssen einige Dinge beachtet werden. Da die NIS-Clients auf die Verfügbarkeit des Servers angewiesen sind, sollten Sie einen Rechner wählen, der nicht regelmäßig neu gestartet werden muss. Der NIS-Server sollte idealerweise ein alleinstehender Rechner sein, dessen einzige Aufgabe es ist, als NIS-Server zu dienen. Wenn das Netzwerk nicht zu stark ausgelastet ist, ist es auch möglich, den NIS-Server als weiteren Dienst auf einem anderen Rechner laufen zu lassen. Wenn jedoch ein NIS-Server ausfällt, wirkt sich dies negativ auf *alle* NIS-Clients aus.

### 52.4.4. Einen NIS-Masterserver konfigurieren

Die verbindlichen Kopien aller NIS-Dateien befinden sich auf dem Masterserver. Die Datenbanken, in denen die Informationen gespeichert sind, bezeichnet man als NIS-Maps. Unter FreeBSD werden diese Maps unter /var/yp/[domainname] gespeichert, wobei [domainname] der Name der NIS-Domäne ist. Da ein NIS-Server mehrere Domänen verwalten kann, können auch mehrere Verzeichnisse vorhanden sein. Jede Domäne verfügt über ein eigenes Verzeichnis sowie einen eigenen, von anderen Domänen unabhängigen Satz von NIS-Maps.

NIS-Master- und Slaveserver verwenden ypserv(8), um NIS-Anfragen zu bearbeiten. Dieser Daemon ist für eingehende Anfragen der NIS-Clients verantwortlich. Er ermittelt aus der angeforderten Domäne und Map einen Pfad zur entsprechenden Datenbank und sendet die angeforderten Daten von der Datenbank zum Client.

Abhängig von den Anforderungen ist die Einrichtung eines NIS-Masterservers relativ einfach, da NIS von FreeBSD bereits in der Standardkonfiguration unterstützt wird. Es kann durch folgende Zeilen in /etc/rc.conf aktiviert werden:

```
nisdomainname="test-domain"  ①
nis_server_enable="YES"      ②
nis_yppasswdd_enable="YES" ③
```

① Diese Zeile setzt den NIS-Domänennamen auf `test-domain`.

② Dadurch werden die NIS-Serverprozesse beim Systemstart automatisch ausgeführt.

③ Durch diese Zeile wird der rpc.yppasswdd(8)-Daemon aktiviert, der die Änderung von NIS-Passwörtern von einem Client aus ermöglicht.

Wird ypserv in einer Multi-Serverdomäne verwendet, in der NIS-Server gleichzeitig als NIS-Clients arbeiten, ist es eine gute Idee, diese Server zu zwingen, sich an sich selbst zu binden. Damit wird verhindert, dass Bindeanforderungen gesendet werden und sich die Server gegenseitig binden. Sonst könnten seltsame Fehler auftreten, wenn ein Server ausfällt, auf den andere Server angewiesen sind. Letztlich werden alle Clients einen Timeout melden, und versuchen, sich an andere Server zu binden. Die dadurch entstehende Verzögerung kann beträchtlich sein. Außerdem kann der Fehler erneut auftreten, da sich die Server wiederum aneinander binden könnten.

Server, die auch als Client arbeiten, können durch das Hinzufügen der folgenden Zeilen in /etc/rc.conf zu gezwungen werden, sich an einen bestimmten Server zu binden:

```
nis_client_enable="YES" ①
nis_client_flags="-S test-domain,server" ②
```

① Ermöglicht die Aktivierung der Client-Komponenten.

② Diese Zeile setzt den NIS-Domain Namen `test-domain` und bindet sich an sich selbst.

Nachdem die Parameter konfiguriert wurden, muss noch `/etc/netstart` ausgeführt werden, um alles entsprechend den Vorgaben in /etc/rc.conf einzurichten. Bevor die NIS-Maps einrichtet werden können, muss der ypserv(8)-Daemon manuell gestartet werden:

```
# service ypserv start
```

**Die NIS-Maps initialisieren**

NIS-Maps Sie werden am NIS-Masterserver aus den Konfigurationsdateien unter /etc erzeugt. Einzige Ausnahme: /etc/master.passwd. Dies verhindert, dass die Passwörter für `root`- oder andere Administratorkonten an alle Server in der NIS-Domäne verteilt werden. Deshalb werden die primären Passwort-Dateien konfiguriert, bevor die NIS-Maps initialisiert werden:

```
# cp /etc/master.passwd /var/yp/master.passwd
# cd /var/yp
# vi master.passwd
```

Es ist ratsam, alle Einträge für Systemkonten sowie Benutzerkonten, die nicht an die NIS-Clients weitergegeben werden sollen, wie beispielsweise `root` und weitere administrative Konten, zu entfernen.



Stellen Sie sicher, dass /var/yp/master.passwd weder von der Gruppe noch von der Welt gelesen werden kann, indem Sie Zugriffsmodus auf `600` einstellen.

Nun können die NIS-Maps initialisiert werden. FreeBSD verwendet dafür das Skript ypinit(8). Geben Sie `-m` und den NIS-Domänennamen an, wenn Sie NIS-Maps für den Masterserver erzeugen:

```
ellington# ypinit -m test-domain
Server Type: MASTER Domain: test-domain
Creating an YP server will require that you answer a few questions.
Questions will all be asked at the beginning of the procedure.
Do you want this procedure to quit on non-fatal errors? [y/n: n] n
Ok, please remember to go back and redo manually whatever fails.
If not, something might not work.
At this point, we have to construct a list of this domains YP servers.
rod.darktech.org is already known as master server.
Please continue to add any slave servers, one per line. When you are
done with the list, type a <control D>.
master server   :  ellington
next host to add:  coltrane
```

```
next host to add:  ^D
The current list of NIS servers looks like this:
ellington
coltrane
Is this correct?  [y/n: y] y

[..output from map generation..]

NIS Map update completed.
ellington has been setup as an YP master server without any errors.
```

Dadurch erzeugt `ypinit`/var/yp/Makefile aus /var/yp/Makefile.dist. Diese Datei geht in der Voreinstellung davon aus, dass in einer NIS-Umgebung mit nur einem Server gearbeitet wird und dass alle Clients unter FreeBSD laufen. Da `test-domain` aber auch über einen Slaveserver verfügt, muss /var/yp/Makefile entsprechend angepasst werden, sodass es mit einem Kommentar (`#`) beginnt:

```
NOPUSH = "True"
```

**Neue Benutzer hinzufügen**

Jedes Mal, wenn ein neuer Benutzer angelegt wird, muss er am NIS-Masterserver hinzugefügt und die NIS-Maps anschließend neu erzeugt werden. Wird dieser Punkt vergessen, kann sich der neue Benutzer *nur* am NIS-Masterserver anmelden. Um beispielsweise den neuen Benutzer `jsmith` zur Domäne `test-domain` hinzufügen wollen, müssen folgende Kommandos auf dem Masterserver ausgeführt werden:

```
# pw useradd jsmith
# cd /var/yp
# make test-domain
```

Statt `pw useradd jsmith` kann auch `adduser jsmith` verwendet werden.

### 52.4.5. Einen NIS-Slaveserver einrichten

Um einen NIS-Slaveserver einzurichten, melden Sie sich am Slaveserver an und bearbeiten Sie /etc/rc.conf analog zum Masterserver. Erzeugen Sie aber keine NIS-Maps, da diese bereits auf dem Server vorhanden sind. Wenn `ypinit` auf dem Slaveserver ausgeführt wird, benutzen Sie `-s` (Slave) statt `-m` (Master). Diese Option benötigt den Namen des NIS-Masterservers und den Domänennamen, wie in diesem Beispiel zu sehen:

```
coltrane# ypinit -s ellington test-domain

Server Type: SLAVE Domain: test-domain Master: ellington

Creating an YP server will require that you answer a few questions.
```

```
Questions will all be asked at the beginning of the procedure.

Do you want this procedure to quit on non-fatal errors? [y/n: n]  n

Ok, please remember to go back and redo manually whatever fails.
If not, something might not work.
There will be no further questions. The remainder of the procedure
should take a few minutes, to copy the databases from ellington.
Transferring netgroup...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byuser...
ypxfr: Exiting: Map successfully transferred
Transferring netgroup.byhost...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byuid...
ypxfr: Exiting: Map successfully transferred
Transferring passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring group.bygid...
ypxfr: Exiting: Map successfully transferred
Transferring group.byname...
ypxfr: Exiting: Map successfully transferred
Transferring services.byname...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring rpc.byname...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.byname...
ypxfr: Exiting: Map successfully transferred
Transferring master.passwd.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byname...
ypxfr: Exiting: Map successfully transferred
Transferring networks.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring netid.byname...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byaddr...
ypxfr: Exiting: Map successfully transferred
Transferring protocols.bynumber...
ypxfr: Exiting: Map successfully transferred
Transferring ypservers...
ypxfr: Exiting: Map successfully transferred
Transferring hosts.byname...
ypxfr: Exiting: Map successfully transferred

coltrane has been setup as an YP slave server without any errors.
```

```
Remember to update map ypservers on ellington.
```

Hierbei wird auf dem Slaveserver ein Verzeichnis namens /var/yp/test-domain erstellt, welches Kopien der NIS-Masterserver-Maps enthält. Durch hinzufügen der folgenden Zeilen in /etc/crontab wird der Slaveserver angewiesen, seine Maps mit den Maps des Masterservers zu synchronisieren:

```
20      *       *       *       *       root    /usr/libexec/ypxfr passwd.byname
21      *       *       *       *       root    /usr/libexec/ypxfr passwd.byuid
```

Diese Einträge sind nicht zwingend notwendig, da der Masterserver automatisch versucht, alle Änderungen seiner NIS-Maps an seine Slaveserver weiterzugeben. Da Passwortinformationen aber auch für nur vom Slaveserver abhängige Systeme vital sind, ist es eine gute Idee, diese Aktualisierungen zu erzwingen. Besonders wichtig ist dies in stark ausgelasteten Netzen, in denen Map-Aktualisierungen unvollständig sein könnten.

Um die Konfiguration abzuschließen, führen Sie `/etc/netstart` auf dem Slaveserver aus, um die NIS-Dienste erneut zu starten.

### 52.4.6. Einen NIS-Client einrichten

Ein NIS-Client `bindet` sich unter Verwendung von `ypbind` an einen NIS-Server. Dieser Daemon sendet RPC-Anfragen auf dem lokalen Netzwerk. Diese Anfragen legen den Namen der Domäne fest, die auf dem Client konfiguriert ist. Wenn der Server der entsprechenden Domäne eine solche Anforderung erhält, schickt er eine Antwort an `ypbind`, das wiederum die Adresse des Servers speichert. Wenn mehrere Server verfügbar sind, verwendet der Client die erste erhaltene Adresse und richtet alle Anfragen an genau diesen Server. `ypbind` "pingt" den Server gelegentlich an, um sicherzustellen, dass der Server funktioniert. Antwortet der Server innerhalb eines bestimmten Zeitraums nicht (Timeout), markiert `ypbind` die Domäne als ungebunden und beginnt erneut, RPCs über das Netzwerk zu verteilen, um einen anderen Server zu finden.

Einen FreeBSD-Rechner als NIS-Client einrichten:

1. Fügen Sie folgende Zeilen in /etc/rc.conf ein, um den NIS-Domänennamen festzulegen, und um ypbind(8) bei der Initialisierung des Netzwerks zu starten:

   ```
   nisdomainname="test-domain"
   nis_client_enable="YES"
   ```

2. Um alle Passworteinträge des NIS-Servers zu importieren, löschen Sie alle Benutzerkonten in /etc/master.passwd mit `vipw`. Denken Sie daran, zumindest ein lokales Benutzerkonto zu behalten. Dieses Konto sollte außerdem Mitglied der Gruppe `wheel` sein. Wenn es mit NIS Probleme gibt, können Sie diesen Zugang verwenden, um sich als Superuser anzumelden und das Problem zu beheben. Bevor Sie die Änderungen speichern, fügen Sie folgende Zeile am Ende der Datei hinzu:

```
+::::::::::
```

Diese Zeile legt für alle gültigen Benutzerkonten der NIS-Server-Maps einen Zugang an. Es gibt verschiedene Wege, den NIS-Client durch Änderung dieser Zeile zu konfigurieren. Eine Methode wird in Netzgruppen verwenden beschrieben. Weitere detaillierte Informationen finden Sie im Buch `Managing NFS and NIS` vom O'Reilly Verlag.

3. Um alle möglichen Gruppeneinträge vom NIS-Server zu importieren, fügen Sie folgende Zeile in /etc/group ein:

```
+:*::
```

Um den NIS-Client direkt zu starten, führen Sie als Superuser die folgenden Befehle aus:

```
# /etc/netstart
# service ypbind start
```

Danach sollte bei der Eingabe von `ypcat passwd` auf dem Client die `passwd-Map` des NIS-Servers angezeigt werden.

### 52.4.7. Sicherheit unter NIS

Da RPC ein Broadcast-basierter Dienst ist, kann jedes System innerhalb der Domäne mittels ypbind den Inhalt der NIS-Maps abrufen. Um nicht autorisierte Transaktionen zu verhindern, unterstützt ypserv(8) eine Funktion namens "securenets", durch die der Zugriff auf bestimmte Rechner beschränkt werden kann. In der Voreinstellung sind diese Informationen in /var/yp/securenets gespeichert, es sei denn, ypserv(8) wurde mit der Option `-p` und einem alternativen Pfad gestartet. Diese Datei enthält Einträge, die aus einer Netzwerkadresse und einer Netzmaske bestehen. Kommentarzeilen beginnen mit "#". /var/yp/securnets könnte beispielsweise so aussehen:

```
# allow connections from local host -- mandatory
127.0.0.1     255.255.255.255
# allow connections from any host
# on the 192.168.128.0 network
192.168.128.0 255.255.255.0
# allow connections from any host
# between 10.0.0.0 to 10.0.15.255
# this includes the machines in the testlab
10.0.0.0      255.255.240.0
```

Wenn ypserv(8) eine Anforderung von einer zu diesen Regeln passenden Adresse erhält, wird die Anforderung bearbeitet. Gibt es keine passende Regel, wird die Anforderung ignoriert und eine Warnmeldung aufgezeichnet. Wenn securenets nicht existiert, erlaubt `ypserv` Verbindungen von jedem Rechner.

TCP Wrapper beschreibt eine alternative Methode zur Zugriffskontrolle. Obwohl beide Methoden einige Sicherheit gewähren, sind sie anfällig für "IP-Spoofing"-Angriffe. Der NIS-Verkehr sollte daher von einer Firewall blockiert werden.

Server, die securenets verwenden, können Schwierigkeiten bei der Anmeldung von NIS-Clients haben, die ein veraltetes TCP/IP-Subsystem besitzen. Einige dieser TCP/IP-Subsysteme setzen alle Rechnerbits auf Null, wenn sie einen `Broadcast` durchführen oder können die Subnetzmaske nicht auslesen, wenn sie die Broadcast-Adresse berechnen. Einige Probleme können durch Änderungen der Clientkonfiguration behoben werden. Andere hingegen lassen sich nur durch das Entfernen des betreffenden Rechners aus dem Netzwerk oder den Verzicht auf securenets umgehen.

Die Verwendung der TCP-Wrapper verlangsamt die Reaktion des NIS-Servers. Diese zusätzliche Reaktionszeit kann in Clientprogrammen zu Timeouts führen. Dies vor allem in Netzwerken, die stark ausgelastet sind, oder nur über langsame NIS-Server verfügen. Wenn ein oder mehrere Clients dieses Problem aufweisen, sollten Sie die betreffenden Clients in NIS-Slaveserver umwandeln, und diese an sich selbst binden.

**Bestimmte Benutzer an der Anmeldung hindern**

In diesem Beispiel gibt es innerhalb der NIS-Domäne den Rechner `basie`, der nur für Mitarbeiter der Fakultät bestimmt ist. Die passwd Datenbank des NIS-Masterservers enthält Benutzerkonten sowohl für Fakultätsmitarbeiter als auch für Studenten. Dieser Abschnitt beschreibt, wie Sie den Mitarbeitern der Fakultät die Anmeldung am System ermöglichen, während den Studenten die Anmeldung verweigert wird.

Es gibt eine Möglichkeit, bestimmte Benutzer an der Anmeldung an einem bestimmten Rechner zu hindern, selbst wenn diese in der NIS-Datenbank vorhanden sind. Dazu kann mit `vipw` der Eintrag `-Benutzername` und die richtige Anzahl von Doppelpunkten an das Ende von /etc/master.passwd gesetzt werden, wobei *Benutzername* der zu blockierende Benutzername ist. Die Zeile mit dem geblockten Benutzer muss dabei vor der `+` Zeile, für zugelassene Benutzer stehen. In diesem Beispiel wird die Anmeldung für den Benutzer `bill` am Rechner `basie` blockiert:

```
basie# cat /etc/master.passwd
root:[password]:0:0::0:0:The super-user:/root:/bin/csh
toor:[password]:0:0::0:0:The other super-user:/root:/bin/sh
daemon:*:1:1::0:0:Owner of many system processes:/root:/usr/sbin/nologin
operator:*:2:5::0:0:System &:/:/usr/sbin/nologin
bin:*:3:7::0:0:Binaries Commands and Source,,,:/:/usr/sbin/nologin
tty:*:4:65533::0:0:Tty Sandbox:/:/usr/sbin/nologin
kmem:*:5:65533::0:0:KMem Sandbox:/:/usr/sbin/nologin
games:*:7:13::0:0:Games pseudo-user:/usr/games:/usr/sbin/nologin
news:*:8:8::0:0:News Subsystem:/:/usr/sbin/nologin
man:*:9:9::0:0:Mister Man Pages:/usr/shared/man:/usr/sbin/nologin
bind:*:53:53::0:0:Bind Sandbox:/:/usr/sbin/nologin
uucp:*:66:66::0:0:UUCP pseudo-user:/var/spool/uucppublic:/usr/libexec/uucp/uucico
xten:*:67:67::0:0:X-10 daemon:/usr/local/xten:/usr/sbin/nologin
pop:*:68:6::0:0:Post Office Owner:/nonexistent:/usr/sbin/nologin
nobody:*:65534:65534::0:0:Unprivileged user:/nonexistent:/usr/sbin/nologin
-bill:::::::::
```

```
+:::::::::

basie#
```

### 52.4.8. Netzgruppen verwenden

Bestimmten Benutzern die Anmeldung an einzelnen Systemen zu verweigern, kann in großen Netzwerken schnell unübersichtlich werden. Dadurch verlieren Sie den Hauptvorteil von NIS: die *zentrale* Verwaltung.

Netzgruppen wurden entwickelt, um große, komplexe Netzwerke mit Hunderten Benutzern und Rechnern zu verwalten. Ihre Aufgabe ist vergleichbar mit UNIX® Gruppen. Die Hauptunterschiede sind das Fehlen einer numerischen ID sowie die Möglichkeit, Netzgruppen zu definieren, die sowohl Benutzer als auch andere Netzgruppen enthalten.

Um das Beispiel in diesem Kapitel fortzuführen, wird die NIS-Domäne um zusätzliche Benutzer und Rechner erweitert:

*Tabelle 26. Zusätzliche Benutzer*

| Benutzername(n) | Beschreibung |
|---|---|
| alpha, beta | Mitarbeiter der IT-Abteilung |
| charlie, delta | Lehrlinge der IT-Abteilung |
| echo, foxtrott, golf, ... | Mitarbeiter |
| able, baker, ... | Praktikanten |

*Tabelle 27. Zusätzliche Rechner*

| Rechnername(n) | Beschreibung |
|---|---|
| war, death, famine, pollution | Nur Mitarbeiter der IT-Abteilung dürfen sich an diesen Rechnern anmelden. |
| pride, greed, envy, wrath, lust, sloth | Nur Mitarbeiter und Lehrlinge der IT-Abteilung dürfen sich auf diesen Rechnern anmelden. |
| one, two, three, four, ... | Gewöhnliche Arbeitsrechner für Mitarbeiter. |
| trashcan | Ein sehr alter Rechner ohne kritische Daten. Sogar Praktikanten dürfen diesen Rechner verwenden. |

Bei der Verwendung von Netzgruppen wird jeder Benutzer einer oder mehreren Netzgruppen zugewiesen und die Anmeldung wird dann für die Netzgruppe erlaubt oder verwehrt. Wenn ein neuer Rechner hinzugefügt wird, müssen die Zugangsbeschränkungen nur für die Netzgruppen festgelegt werden. Wird ein neuer Benutzer angelegt, muss er einer oder mehreren Netzgruppen zugewiesen werden. Wenn die Einrichtung von NIS sorgfältig geplant wurde, muss nur noch eine zentrale Konfigurationsdatei bearbeitet werden, um den Zugriff auf bestimmte Rechner zu erlauben oder zu verbieten.

Dieses Beispiel erstellt vier Netzgruppen: IT-Mitarbeiter, IT-Lehrlinge, normale Mitarbeiter sowie Praktikanten:

```
IT_EMP  (,alpha,test-domain)    (,beta,test-domain)
IT_APP  (,charlie,test-domain)  (,delta,test-domain)
USERS   (,echo,test-domain)     (,foxtrott,test-domain) \
        (,golf,test-domain)
INTERNS (,able,test-domain)     (,baker,test-domain)
```

Jede Zeile konfiguriert eine Netzgruppe. Die erste Spalte der Zeile bezeichnet den Namen der Netzgruppe. Die Einträge in den Klammern stehen entweder für eine Gruppe von einem oder mehreren Benutzern, oder für den Namen einer weiteren Netzgruppe. Wenn ein Benutzer angegeben wird, haben die drei Felder in der Klammer folgende Bedeutung:

1. Der Name des Rechner(s), auf dem die weiteren Felder für den Benutzer gültig sind. Wird kein Rechnername festgelegt, ist der Eintrag auf allen Rechnern gültig.
2. Der Name des Benutzerkontos, der zu dieser Netzgruppe gehört.
3. Die NIS-Domäne für das Benutzerkonto. Benutzerkonten können von anderen NIS-Domänen in eine Netzgruppe importiert werden.

Wenn eine Gruppe mehrere Benutzer enthält, müssen diese durch Leerzeichen getrennt werden. Darüber hinaus kann jedes Feld Wildcards enthalten. Weitere Einzelheiten finden Sie in netgroup(5).

Netzgruppennamen sollten nicht länger als 8 Zeichen sein. Es wird zwischen Groß- und Kleinschreibung unterschieden. Die Verwendung von Großbuchstaben für Netzgruppennamen ermöglicht eine leichte Unterscheidung zwischen Benutzern, Rechnern und Netzgruppen.

Einige NIS-Clients (dies gilt nicht für FreeBSD) können keine Netzgruppen mit mehr als 15 Einträgen verwalten. Diese Grenze kann umgangen werden, indem mehrere Subnetzgruppen mit weniger als fünfzehn Benutzern angelegt werden und diese Subnetzgruppen wiederum in einer Netzgruppe zusammengefasst wird, wie in diesem Beispiel zu sehen:

```
BIGGRP1  (,joe1,domain)  (,joe2,domain)  (,joe3,domain) [...]
BIGGRP2  (,joe16,domain)  (,joe17,domain) [...]
BIGGRP3  (,joe31,domain)  (,joe32,domain)
BIGGROUP  BIGGRP1 BIGGRP2 BIGGRP3
```

Wiederholen Sie diesen Vorgang, wenn mehr als 225 (15*15) Benutzer in einer einzigen Netzgruppe existieren.

Die neue NIS-Map aktivieren und verteilen:

```
ellington# cd /var/yp
ellington# make
```

Dadurch werden die NIS-Maps netgroup, netgroup.byhost und netgroup.byuser erzeugt. Prüfen Sie die Verfügbarkeit der neuen NIS-Maps mit ypcat(1):

```
ellington% ypcat -k netgroup
ellington% ypcat -k netgroup.byhost
ellington% ypcat -k netgroup.byuser
```

Die Ausgabe des ersten Befehls gibt den Inhalt von /var/yp/netgroup wieder. Der zweite Befehl erzeugt nur dann eine Ausgabe, wenn rechnerspezifische Netzgruppen erzeugt wurden. Der dritte Befehl gibt die Netzgruppen nach Benutzern sortiert aus.

Wenn Sie einen Client einrichten, verwenden Sie vipw(8) um den Namen der Netzgruppe anzugeben. Ersetzen Sie beispielsweise auf dem Server namens `war` die folgende Zeile:

```
+:::::::::
```

durch

```
+@IT_EMP:::::::::
```

ersetzt werden.

Diese Zeile legt fest, dass nur noch Benutzer der Netzgruppe `IT_EMP` in die Passwortdatenbank dieses Systems importiert werden. Nur diese Benutzer dürfen sich an diesem Server anmelden.

Diese Konfiguration gilt auch für die `~`-Funktion der Shell und für alle Routinen, die auf Benutzernamen und numerische Benutzer-IDs zugreifen. Oder anders formuliert, `cd ~Benutzer` ist nicht möglich, `ls -l` zeigt die numerische Benutzer-ID statt dem Benutzernamen und `find . -user joe -print` erzeugt die Fehlermeldung `No such user`. Um dieses Problem zu beheben, müssen alle Benutzereinträge importiert werden, ohne ihnen jedoch zu erlauben, sich am Server anzumelden. Dies kann durch das Hinzufügen einer zusätzlichen Zeile erreicht werden:

```
+:::::::::/usr/sbin/nologin
```

Diese Zeile weist den Client an, alle Einträge zu importieren, aber die Shell in diesen Einträgen durch /usr/sbin/nologin zu ersetzen.

Stellen Sie sicher, dass die zusätzliche Zeile *nach* der Zeile `+@IT_EMP:::::::::` eingetragen ist. Andernfalls haben alle via NIS importierten Benutzerkonten /usr/sbin/nologin als Loginshell und niemand wird sich mehr am System anmelden können.

Um die weniger wichtigen Server zu konfigurieren, ersetzen Sie den alten Eintrag `+:::::::::` auf den Servern mit diesen Zeilen:

```
+@IT_EMP:::::::::
```

```
+@IT_APP:::::::::
+:::::::::/usr/sbin/nologin
```

Die entsprechenden Zeilen für Arbeitsplätze lauten:

```
+@IT_EMP:::::::::
+@USERS:::::::::
+:::::::::/usr/sbin/nologin
```

NIS ist in der Lage, Netzgruppen aus anderen Netzgruppen zu bilden. Dies kann nützlich sein, wenn sich die Firmenpolitik ändert. Eine Möglichkeit ist die Erzeugung rollenbasierter Netzgruppen. Sie könnten eine Netzgruppe `BIGSRV` erzeugen, um den Zugang zu den wichtigsten Servern zu beschränken, eine weitere Gruppe `SMALLSRV` für die weniger wichtigen Server und eine dritte Netzgruppe `USERBOX` für die Arbeitsplatzrechner. Jede dieser Netzgruppen enthält die Netzgruppen, die sich auf diesen Rechnern anmelden dürfen. Die Einträge der Netzgruppen in der NIS-Map sollten ähnlich den folgenden aussehen:

```
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP  ITINTERN
USERBOX   IT_EMP  ITINTERN USERS
```

Diese Methode funktioniert besonders gut, wenn Rechner in Gruppen mit identischen Beschränkungen eingeteilt werden können. Unglücklicherweise ist dies die Ausnahme und nicht die Regel. Meistens wird die Möglichkeit zur rechnerspezischen Zugangsbeschränkung benötigt.

Rechnerspezifische Netzgruppen sind eine weitere Möglichkeit, um mit den oben beschriebenen Änderungen umzugehen. In diesem Szenario enthält /etc/master.passwd auf jedem Rechner zwei mit "+" beginnende Zeilen. Die erste Zeile legt die Netzgruppe mit den Benutzern fest, die sich auf diesem Rechner anmelden dürfen. Die zweite Zeile weist allen anderen Benutzern /usr/sbin/nologin als Shell zu. Verwenden Sie auch hier (analog zu den Netzgruppen) Großbuchstaben für die Rechnernamen:

```
+@BOXNAME:::::::::
+:::::::::/usr/sbin/nologin
```

Sobald dies für alle Rechner erledigt ist, müssen die lokalen Versionen von /etc/master.passwd nie mehr verändert werden. Alle weiteren Änderungen geschehen über die NIS-Maps. Nachfolgend ein Beispiel für eine mögliche Netzgruppen-Map:

```
# Define groups of users first
IT_EMP    (,alpha,test-domain)    (,beta,test-domain)
IT_APP    (,charlie,test-domain)  (,delta,test-domain)
DEPT1     (,echo,test-domain)     (,foxtrott,test-domain)
DEPT2     (,golf,test-domain)     (,hotel,test-domain)
DEPT3     (,india,test-domain)    (,juliet,test-domain)
```

```
ITINTERN  (,kilo,test-domain)     (,lima,test-domain)
D_INTERNS (,able,test-domain)     (,baker,test-domain)
#
# Now, define some groups based on roles
USERS     DEPT1   DEPT2     DEPT3
BIGSRV    IT_EMP  IT_APP
SMALLSRV  IT_EMP  IT_APP    ITINTERN
USERBOX   IT_EMP  ITINTERN  USERS
#
# And a groups for a special tasks
# Allow echo and golf to access our anti-virus-machine
SECURITY  IT_EMP  (,echo,test-domain)  (,golf,test-domain)
#
# machine-based netgroups
# Our main servers
WAR       BIGSRV
FAMINE    BIGSRV
# User india needs access to this server
POLLUTION  BIGSRV  (,india,test-domain)
#
# This one is really important and needs more access restrictions
DEATH     IT_EMP
#
# The anti-virus-machine mentioned above
ONE       SECURITY
#
# Restrict a machine to a single user
TWO       (,hotel,test-domain)
# [...more groups to follow]
```

Es ist nicht immer ratsam, rechnerbasierte Netzgruppen zu verwenden. Wenn Dutzende oder Hunderte identische Rechner eingerichtet werden müssen, sollten rollenbasierte Netzgruppen verwendet werden, um die Größe der NIS-Maps in Grenzen zu halten.

### 52.4.9. Passwortformate

Alle Rechner innerhalb der NIS-Domäne müssen für die Verschlüsselung von Passwörtern das gleiche Format benutzen. Wenn Benutzer Schwierigkeiten bei der Authentifizierung auf einem NIS-Client haben, liegt dies möglicherweise an einem anderen Passwort-Format. In einem heterogenen Netzwerk muss das verwendete Format von allen Betriebssystemen unterstützt werden, wobei DES der kleinste gemeinsame Standard ist.

Welches Format die Server und Clients verwenden, steht in /etc/login.conf:

```
default:\
    :passwd_format=des:\
    [weitere Einträge]
```

In diesem Beispiel verwendet das System das Format DES. Weitere mögliche Werte sind unter

anderem `blf` und `md5` (mit Blowfish und MD5 verschlüsselte Passwörter).

Wird auf einem Rechner das Format entsprechend der NIS-Domäne geändert, muss anschließend die Login-Capability Datenbank neu erstellt werden:

```
# cap_mkdb /etc/login.conf
```



Das Format der schon bestehenden Passwörter wird erst aktualisiert, wenn ein Benutzer sein Passwort ändert, *nachdem* die Datenbank neu erstellt wurde.

## 52.5. Lightweight Access Directory Protocol (LDAP)

Das Lightweight Directory Access Protocol (LDAP) ist ein Protokoll der Anwendungsschicht, das verwendet wird um Objekte mithilfe eines verteilten Verzeichnisdienstes abzurufen, zu verändern und zu authentifizieren. Betrachten Sie es als ein Telefonbuch, das homogene Informationen in mehreren hierarchischen Ebenen speichert. Es wird in Active Directory und OpenLDAP-Netzwerken eingesetzt, in denen Benutzer unter Verwendung eines einzigen Kontos auf diverse interne Informationen zugreifen. Beispielsweise kann E-Mail-Authentifizierung, Abfrage von Kontaktinformationen und Website-Authentifizierung über ein einzelnes Benutzerkonto aus der Datenbank des LDAP-Servers erfolgen.

Dieser Abschnitt enthält eine kompakte Anleitung, um einen LDAP-Server auf einem FreeBSD-System zu konfigurieren. Es wird vorausgesetzt, dass der Administrator bereits einen Plan erarbeitet hat, der verschiedene Punkte umfasst, unter anderem die Art der zu speichernden Informationen, für was die Informationen verwendet werden, welche Benutzer Zugriff auf die Informationen haben und wie die Informationen vor unbefugtem Zugriff geschützt werden.

### 52.5.1. LDAP Terminologie und Struktur

LDAP verwendet mehrere Begriffe die Sie verstehen sollten bevor Sie die Konfiguration beginnen. Alle Verzeichniseinträge bestehen aus einer Gruppe von *Attributen*. Jede Attributgruppe enthält einen eindeutigen Bezeichner, der als distinguished name (DN) bekannt ist. Dieser setzt sich normalerweise aus mehreren anderen Attributen, wie dem Relative Distinguished Name (RDN) zusammen. Wie bei Verzeichnissen gibt es auch hier absolute und relative Pfade. Betrachten Sie DN als absoluten Pfad und RDN als relativen Pfad.

Beispielsweise könnte ein LDAP-Eintrag wie folgt aussehen. Dieses Beispiel sucht nach dem Eintrag für das angegebene Benutzerkonto (`uid`), Organisationseinheit (`ou` und Organisation (`o`):

```
% ldapsearch -xb "uid=trhodes,ou=users,o=example.com"
# extended LDIF

#
# LDAPv3
# base <uid=trhodes,ou=users,o=example.com> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
```

```
#

# trhodes, users, example.com
dn: uid=trhodes,ou=users,o=example.com
mail: trhodes@example.com
cn: Tom Rhodes
uid: trhodes
telephoneNumber: (123) 456-7890

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries:1
```

Die Einträge in diesem Beispiel zeigen die Werte für die Attribute `dn`, `mail`, `cn`, `uid` und `telephoneNumber`. Das Attribut `cn` ist der RDN.

Weitere Informationen über LDAP und dessen Terminologie finden Sie unter http://www.openldap.org/doc/admin24/intro.html.

### 52.5.2. Konfiguration eines LDAP-Servers

FreeBSD integriert keinen LDAP-Server. Beginnen Sie die Konfiguration mit der Installation des Ports oder Pakets net/openldap-server:

```
# pkg install openldap-server
```

Im Paket sind eine große Anzahl an Optionen aktiviert. Mit dem Befehl `pkg info openldap-server` können diese überprüft werden. Falls die Optionen nicht ausreichend sind (weil bspw. SQL-Unterstützung benötigt wird), sollten Sie in Betracht ziehen, den Port mit dem entsprechenden Framework neu zu übersetzen.

Während der Installation wird für die Daten das Verzeichnis /var/db/openldap-data erstellt. Das Verzeichnis für die Ablage der Zertifikate muss manuell angelegt werden:

```
# mkdir /usr/local/etc/openldap/private
```

Im nächsten Schritt wird die Zertifizierungsstelle konfiguriert. Die folgenden Befehle müssen in /usr/local/etc/openldap/private ausgeführt werden. Dies ist wichtig, da die Dateiberechtigungen restriktiv gesetzt werden und Benutzer keinen direkten Zugriff auf diese Daten haben sollten. Weitere Informationen über Zertifikate und deren Parameter finden Sie im OpenSSL. Geben Sie folgenden Befehl ein, um die Zertifizierungsstelle zu erstellen und folgen Sie den Anweisungen:

```
# openssl req -days 365 -nodes -new -x509 -keyout ca.key -out ../ca.crt
```

Diese Einträge sind frei wählbar, *mit Ausnahme* von *Common Name*. Hier muss etwas anderes als der Hostname des Systems eingetragen werden. Wenn ein selbstsigniertes Zertifikat verwendet wird, stellen Sie dem Hostnamen einfach das Präfix `CA` für die Zertifizierungsstelle voran.

Die nächste Aufgabe besteht darin, einen Zertifikatsregistrierungsanforderung (CSR) sowie einen privaten Schlüssel zu erstellen. Geben Sie folgenden Befehl ein und folgen Sie den Anweisungen:

```
# openssl req -days 365 -nodes -new -keyout server.key -out server.csr
```

Stellen Sie hierbei sicher, dass `Common Name` richtig eingetragen wird. Die Zertifikatsregistrierungsanforderung muss mit dem Schlüssel der Zertifizierungsstelle unterschrieben werden, um als gültiges Zertifikat verwendet zu werden:

```
# openssl x509 -req -days 365 -in server.csr -out ../server.crt -CA ../ca.crt -CAkey
ca.key -CAcreateserial
```

Der letzte Schritt für die Erstellung der Zertifikate besteht darin, die Client-Zertifikate zu erstellen und zu signieren:

```
# openssl req -days 365 -nodes -new -keyout client.key -out client.csr
# openssl x509 -req -days 3650 -in client.csr -out ../client.crt -CAkey ca.key
```

Achten Sie wieder auf das Attribut `Common name`. Stellen Sie außerdem sicher, dass bei diesem Verfahren acht (8) neue Dateien erzeugt worden sind.

Der Daemon, auf dem der OpenLDAP-Server läuft, heißt slapd. Die Konfiguration erfolgt über slapd.ldif. Die alte slapd.conf wird von OpenLDAP nicht mehr verwendet.

Konfigurationsbeispiele für slapd.ldif finden sich auch in /usr/local/etc/openldap/slapd.ldif.sample. Optionen sind in slapd-config(5) dokumentiert. Jeder Abschnitt in slapd.ldif wird, wie alle anderen LDAP-Attributgruppen, durch einen DN eindeutig identifiziert. Achten Sie darauf, dass keine Leerzeilen zwischen der Anweisung `dn:` und dem gewünschten Ende des Abschnitts verbleiben. Im folgenden Beispiel wird TLS verwendet, um einen sicheren Kanal zu implementieren. Der erste Abschnitt stellt die globale Konfiguration dar:

```
#
# See slapd-config(5) for details on configuration options.
# This file should NOT be world readable.
#
dn: cn=config
objectClass: olcGlobal
cn: config
#
#
# Define global ACLs to disable default read access.
#
```

```
olcArgsFile: /var/run/openldap/slapd.args
olcPidFile: /var/run/openldap/slapd.pid
olcTLSCertificateFile: /usr/local/etc/openldap/server.crt
olcTLSCertificateKeyFile: /usr/local/etc/openldap/private/server.key
olcTLSCACertificateFile: /usr/local/etc/openldap/ca.crt
#olcTLSCipherSuite: HIGH
olcTLSProtocolMin: 3.1
olcTLSVerifyClient: never
```

Hier müssen die Zertifizierungsstelle, das Serverzertifikat und die privaten Schlüssel des Servers angegeben werden. Es wird empfohlen, den Clients die Wahl der Sicherheits-Chiffre zu überlassen und die Option `olcTLSCipherSuite` wegzulassen (inkompatibel mit anderen TLS-Clients als openssl). Mit der Option `olcTLSProtocolMin` benötigt der Server nur eine minimale Sicherheitsstufe. Diese Option wird empfohlen. Während die Verfizierung für den Server verpflichtend ist, ist sie es nicht für den Client: `olcTLSVerifyClient: never`.

Der zweite Abschnitt behandelt die Backend-Module und kann wie folgt konfiguriert werden:

```
#
# Load dynamic backend modules:
#
dn: cn=module,cn=config
objectClass: olcModuleList
cn: module
olcModulepath:  /usr/local/libexec/openldap
olcModuleload:  back_mdb.la
#olcModuleload: back_bdb.la
#olcModuleload: back_hdb.la
#olcModuleload: back_ldap.la
#olcModuleload: back_passwd.la
#olcModuleload: back_shell.la
```

Der dritte Abschnitt widmet sich dem Laden der benötigten ldif-Schemata, die von den Datenbanken verwendet werden sollen. Diese Dateien sind essentiell.

```
dn: cn=schema,cn=config
objectClass: olcSchemaConfig
cn: schema

include: file:///usr/local/etc/openldap/schema/core.ldif
include: file:///usr/local/etc/openldap/schema/cosine.ldif
include: file:///usr/local/etc/openldap/schema/inetorgperson.ldif
include: file:///usr/local/etc/openldap/schema/nis.ldif
```

Als nächstes folgt der Abschnitt zur Frontend-Konfiguration:

```
# Frontend settings
```

```
#
dn: olcDatabase={-1}frontend,cn=config
objectClass: olcDatabaseConfig
objectClass: olcFrontendConfig
olcDatabase: {-1}frontend
olcAccess: to * by * read
#
# Sample global access control policy:
#   Root DSE: allow anyone to read it
#   Subschema (sub)entry DSE: allow anyone to read it
#   Other DSEs:
#       Allow self write access
#       Allow authenticated users read access
#       Allow anonymous users to authenticate
#
#olcAccess: to dn.base="" by * read
#olcAccess: to dn.base="cn=Subschema" by * read
#olcAccess: to *
#   by self write
#   by users read
#   by anonymous auth
#
# if no access controls are present, the default policy
# allows anyone and everyone to read anything but restricts
# updates to rootdn.  (e.g., "access to * by * read")
#
# rootdn can always read and write EVERYTHING!
#
olcPasswordHash: {SSHA}
# {SSHA} is already the default for olcPasswordHash
```

Ein weiterer Abschnitt ist dem Konfigurations-Backend gewidmet, der einzige Weg, später auf die OpenLDAP-Serverkonfiguration zuzugreifen, ist als globaler Superuser.

```
dn: olcDatabase={0}config,cn=config
objectClass: olcDatabaseConfig
olcDatabase: {0}config
olcAccess: to * by * none
olcRootPW: {SSHA}iae+lrQZILpiUdf16Z9KmDmSwT77Dj4U
```

Der voreingestellte Benutzername für den Administrator lautet `cn=config`. Geben Sie slappasswd in eine Shell ein, wählen Sie ein Passwort und verwenden Sie seinen Hash in `olcRootPW`. Wenn diese Option jetzt nicht angegeben ist, kann vor dem Import der slapd.ldif niemand später den Abschnitt *global configuration* ändern.

Der letzte Abschnitt befasst sich mit dem Datenbank-Backend:

```
#######################################################################
```

```
# LMDB database definitions
#######################################################################
#
dn: olcDatabase=mdb,cn=config
objectClass: olcDatabaseConfig
objectClass: olcMdbConfig
olcDatabase: mdb
olcDbMaxSize: 1073741824
olcSuffix: dc=domain,dc=example
olcRootDN: cn=mdbadmin,dc=domain,dc=example
# Cleartext passwords, especially for the rootdn, should
# be avoided.  See slappasswd(8) and slapd-config(5) for details.
# Use of strong authentication encouraged.
olcRootPW: {SSHA}X2wHvIWDk6G76CQyCMS1vDCvtICWgn0+
# The database directory MUST exist prior to running slapd AND
# should only be accessible by the slapd and slap tools.
# Mode 700 recommended.
olcDbDirectory: /var/db/openldap-data
# Indices to maintain
olcDbIndex: objectClass eq
```

Diese Datenbank enthält den *eigentlichen Inhalt* des LDAP-Verzeichnisses. Neben `mdb` sind weitere Versionen verfügbar. Dessen Superuser, nicht zu verwechseln mit dem globalen, wird hier konfiguriert: ein Benutzername in `olcRootDN` und der Passworthash in `olcRootPW`; slappasswd kann wie zuvor benutzt werden.

Dieses Repository enthält vier Beispiele für slapd.ldif. Lesen Sie diese Seite, um eine bestehende slapd.conf in slapd.ldif zu konvertieren. Beachten Sie, dass dies einige unbrauchbare Optionen einführen kann.

Wenn die Konfiguration abgeschlossen ist, muss slapd.ldif in ein leeres Verzeichnis verschoben werden. Folgendes ist die empfohlene Vorgehensweise:

```
# mkdir /usr/local/etc/openldap/slapd.d/
```

Importieren Sie die Konfigurationsdatenbank:

```
# /usr/local/sbin/slapadd -n0 -F /usr/local/etc/openldap/slapd.d/ -l
/usr/local/etc/openldap/slapd.ldif
```

Starten Sie den slapd-Daemon:

```
# /usr/local/libexec/slapd -F /usr/local/etc/openldap/slapd.d/
```

Die Option `-d` kann, wie in slapd(8) beschrieben, zur Fehlersuche benutzt werden. Stellen Sie sicher, dass der Server läuft und korrekt arbeitet:

```
# ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
# extended LDIF
#
# LDAPv3
# base <> with scope baseObject
# filter: (objectclass=*)
# requesting: namingContexts
#

#
dn:
namingContexts: dc=domain,dc=example

# search result
search: 2
result: 0 Success

# numResponses: 2
# numEntries: 1
```

Dem Server muss noch vertraut werden. Wenn dies noch nie zuvor geschehen ist, befolgen Sie diese Anweisungen. Installieren Sie das Paket oder den Port OpenSSL:

```
# pkg install openssl
```

Aus dem Verzeichnis, in dem ca.crt gespeichert ist (in diesem Beispiel /usr/local/etc/openldap), starten Sie:

```
# c_rehash .
```

Sowohl die CA als auch das Serverzertifikat werden nun in ihren jeweiligen Rollen korrekt erkannt. Um dies zu überprüfen, führen die folgenden Befehl aus dem Verzeichnis der server.crt aus:

```
# openssl verify -verbose -CApath . server.crt
```

Falls slapd ausgeführt wurde, muss der Daemon neu gestartet werden. Wie in /usr/local/etc/rc.d/slapd angegeben, müssen die folgenden Zeilen in /etc/rc.conf eingefügt werden, um slapd beim Booten ordnungsgemäß auszuführen:

```
lapd_enable="YES"
slapd_flags='-h "ldapi://%2fvar%2frun%2fopenldap%2fldapi/
ldap://0.0.0.0/"'
slapd_sockets="/var/run/openldap/ldapi"
slapd_cn_config="YES"
```

slapd bietet beim Booten keine Möglichkeit zur Fehlersuche. Überprüfen Sie dazu /var/log/debug.log, `dmesg -a` und /var/log/messages.

Das folgende Beispiel fügt die Gruppe `team` und den Benutzer `john` zur LDAP-Datenbank `domain.example` hinzu, die bislang leer ist. Erstellen Sie zunächst die Datei domain.ldif:

```
# cat domain.ldif
dn: dc=domain,dc=example
objectClass: dcObject
objectClass: organization
o: domain.example
dc: domain

dn: ou=groups,dc=domain,dc=example
objectClass: top
objectClass: organizationalunit
ou: groups

dn: ou=users,dc=domain,dc=example
objectClass: top
objectClass: organizationalunit
ou: users

dn: cn=team,ou=groups,dc=domain,dc=example
objectClass: top
objectClass: posixGroup
cn: team
gidNumber: 10001

dn: uid=john,ou=users,dc=domain,dc=example
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount
cn: John McUser
uid: john
uidNumber: 10001
gidNumber: 10001
homeDirectory: /home/john/
loginShell: /usr/bin/bash
userPassword: secret
```

Weitere Informationen finden Sie in der OpenLDAP-Dokumentation. Benutzen Sie slappasswd, um das Passwort durch einen Hash in `userPassword` zu ersetzen. Der in `loginShell` angegebene Pfad muss in allen Systemen existieren, in denen `john` sich anmelden darf. Benutzen Sie schließlich den `mdb`-Administrator, um die Datenbank zu ändern:

```
# ldapadd -W -D "cn=mdbadmin,dc=domain,dc=example" -f domain.ldif
```

Änderungen im Bereich *global configuration* können nur vom globalen Superuser vorgenommen werden. Angenommen die Option `olcTLSCipherSuite: HIGH:MEDIUM:SSLv3` wurde ursprünglich definiert und soll nun gelöscht werden. Dazu erstellen Sie zunächst eine Datei mit folgendem Inhalt:

```
# cat global_mod
dn: cn=config
changetype: modify
delete: olcTLSCipherSuite
```

Übernehmen Sie dann die Änderungen:

```
# ldapmodify -f global_mod -x -D "cn=config" -W
```

Geben Sie bei Aufforderung das im Abschnitt *configuration backend* gewählte Passwort ein. Der Benutzername ist nicht erforderlich: Hier repräsentiert `cn=config` den DN des zu ändernden Datenbankabschnitts. Alternativ können Sie mit `ldapmodify` eine einzelne Zeile der Datenbank löschen, mit `ldapdelete` einen ganzen Eintrag.

Wenn etwas schief geht oder der globale Superuser nicht auf das Konfigurations-Backend zugreifen kann, ist es möglich, die gesamte Konfiguration zu löschen und neu zu schreiben:

```
# rm -rf /usr/local/etc/openldap/slapd.d/
```

slapd.ldif kann dann bearbeitet und erneut importiert werden. Bitte folgenden Sie dieser Vorgehensweise nur, wenn keine andere Lösung verfügbar ist.

Dies ist nur die Konfiguration des Servers. Auf demselben Rechner kann auch ein LDAP-Client mit eigener, separater Konfiguration betrieben werden.

## 52.6. Dynamic Host Configuration Protocol (DHCP)

Das Dynamic Host Configuration Protocol (DHCP) ermöglicht es einem System, sich mit einem Netzwerk zu verbinden und die für die Kommunikation mit diesem Netzwerk nötigen Informationen zu beziehen. FreeBSD verwendet den von OpenBSD stammenden `dhclient`, um die Adressinformationen zu beziehen. FreeBSD installiert keinen DHCP-Server, aber es stehen einige Server in der FreeBSD Ports-Sammlung zu Verfügung. Das DHCP-Protokoll wird vollständig im RFC 2131 beschrieben. Eine weitere, lehrreiche Informationsquelle existiert unter isc.org/downloads/dhcp/.

In diesem Abschnitt wird beschrieben, wie der integrierte DHCP-Client verwendet wird. Anschließend wird erklärt, wie ein DHCP-Server zu installieren und konfigurieren ist.



Unter FreeBSD wird das Gerät bpf(4) für den DHCP-Server und den DHCP-Client benötigt. Das Gerät ist bereits im GENERIC-Kernel enthalten. Benutzer, die es vorziehen einen angepassten Kernel zu erstellen, müssen dieses Gerät behalten,

wenn DHCP verwendet wird.

Es sei darauf hingewiesen, dass bpf es priviligierten Benutzern ermöglicht einen Paket-Sniffer auf dem System auszuführen.

### 52.6.1. Einen DHCP-Client konfigurieren

Die Unterstützung für den DHCP-Client ist im Installationsprogramm von FreeBSD enthalten, sodass ein neu installiertes System automatisch die Adressinformationen des Netzwerks vom DHCP-Server erhält. In Benutzerkonten, Zeitzone, Dienste und Sicherheitsoptionen finden Sie Beispiele für eine Netzwerkkonfiguration.

`dhclient` beginnt von einem Clientrechner aus über den UDP-Port 68 Konfigurationsinformationen anzufordern. Der Server antwortet auf dem UDP-Port 67, indem er dem Client eine IP-Adresse zuweist und ihm weitere relevante Informationen über das Netzwerk, wie Netzmasken, Router und DNS-Server mitteilt. Diese Informationen werden als DHCP-Lease bezeichnet und sind nur für bestimmte Zeit, die vom Administrator des DHCP-Servers vorgegeben wird, gültig. Dadurch fallen verwaiste IP-Adressen, deren Clients nicht mehr mit dem Netzwerk verbunden sind, automatisch an den Server zurück. DHCP-Clients können sehr viele Informationen von einem DHCP-Server erhalten. Eine ausführliche Liste finden Sie in dhcp-options(5).

Das Gerät bpf ist im GENERIC-Kernel bereits enthalten. Für die Nutzung von DHCP muss also kein angepasster Kernel erzeugt werden. In einer angepassten Kernelkonfigurationsdatei muss das Gerät enthalten sein, damit DHCP ordnungsgemäß funktioniert.

Standardmässig läuft die DHCP-Konfiguration bei FreeBSD im Hintergrund oder auch *asynchron*. Andere Startskripte laufen weiter, während DHCP fertig abgearbeitet wird, was den Systemstart beschleunigt.

DHCP im Hintergrund funktioniert gut, wenn der DHCP-Server schnell auf Anfragen der Clients antwortet. Jedoch kann DHCP eine lange Zeit benötigen, um auf manchen Systemen fertig zu werden. Falls Netzwerkdienste gestartet werden, bevor DHCP die Informationen und Netzwerkadressen gesetzt hat, werden diese fehlschlagen. Durch die Verwendung von DHCP im *asynchronen* Modus wird das Problem verhindert, so dass die Startskripte pausiert werden, bis die DHCP-Konfiguration abgeschlossen ist.

Diese Zeile wird in /etc/rc.conf verwendet, um den asynchronen Modus zu aktivieren:

```
ifconfig_fxp0="DHCP"
```

Die Zeile kann bereits vorhanden sein, wenn bei der Installation des Systems DHCP konfiguriert wurde. Ersetzen Sie *fxp0* durch die entsprechende Schnittstelle. Die dynamische Konfiguration von Netzwerkkarten wird in “Einrichten von Netzwerkkarten” beschrieben.

Um stattdessen den synchronen Modus zu verwenden, der während des Systemstarts pausiert bis die DHCP-Konfiguration abgeschlossen ist, benutzen Sie "SYNCDHCP":

```
ifconfig_fxp0="SYNCDHCP"
```

Es stehen weitere Optionen für den Client zur Verfügung. Suchen Sie in rc.conf(5) nach `dhclient`, wenn Sie an Einzelheiten interessiert sind.

Der DHCP-Client verwendet die folgenden Dateien:

- /etc/dhclient.conf

  Die Konfigurationsdatei von `dhclient`. Diese Datei enthält normalerweise nur Kommentare, da die Vorgabewerte zumeist ausreichend sind. Diese Konfigurationsdatei wird in dhclient.conf(5) beschrieben.

- /sbin/dhclient

  Weitere Informationen über dieses Kommando finden Sie in dhclient(8).

- /sbin/dhclient-script

  Das FreeBSD-spezifische Konfigurationsskript des DHCP-Clients. Es wird in dhclient-script(8) beschrieben und kann meist unverändert übernommen werden.

- /var/db/dhclient.leases.interface

  Der DHCP-Client verfügt über eine Datenbank, die alle derzeit gültigen Leases enthält und als Logdatei erzeugt wird. Diese Datei wird in dhclient.leases(5) beschrieben.

### 52.6.2. Einen DHCP-Server installieren und einrichten

Dieser Abschnitt beschreibt die Einrichtung eines FreeBSD-Systems als DHCP-Server. Dazu wird die DHCP-Implementation von ISC (Internet Systems Consortium) verwendet. Diese Implementation und die Dokumentation können als Port oder Paket net/isc-dhcp44-server installiert werden.

Der Port net/isc-dhcp44-server installiert eine Beispiel-Konfigurationsdatei. Kopieren Sie /usr/local/etc/dhcpd.conf.example nach /usr/local/etc/dhcpd.conf und nehmen Sie die Änderungen an der neuen Datei vor.

Diese Konfigurationsdatei umfasst Deklarationen für Subnetze und Rechner, die den DHCP-Cleints zur Verfügung gestellt wird. Die folgenden Zeilen konfigurieren Folgendes:

```
option domain-name "example.org";①
option domain-name-servers ns1.example.org;②
option subnet-mask 255.255.255.0;③

default-lease-time 600;④
max-lease-time 72400;⑤
ddns-update-style none;⑥

subnet 10.254.239.0 netmask 255.255.255.224 {
```

```
  range 10.254.239.10 10.254.239.20;⑦
  option routers rtr-239-0-1.example.org;⑧
}

host fantasia {
  hardware ethernet 08:00:07:26:c0:a5;⑨
  fixed-address fantasia.fugue.com;⑩
}
```

① Diese Option beschreibt die Standardsuchdomäne, die den Clients zugewiesen wird. Weitere Informationen finden Sie in resolv.conf(5).

② Diese Option legt eine, durch Kommata getrennte Liste von DNS-Servern fest, die von den Clients verwendet werden sollen. Die Server können über den Namen (FQDN) oder die IP-Adresse spezifiziert werden.

③ Die den Clients zugewiesene Subnetzmaske.

④ Die Voreinstellung für die Ablaufzeit des Lease in Sekunden. Ein Client kann diesen Wert in der Konfiguration überschreiben.

⑤ Die maximale Zeitdauer, für die der Server Leases vergibt. Sollte ein Client eine längere Zeitspanne anfordern, wird dennoch nur der Wert `max-lease-time` zugewiesen.

⑥ Die Voreinstellung `none` deaktiviert dynamische DNS-Updates. Bei der Einstellung `interim` aktualisiert der DHCP-Server den DNS-Server, wenn ein Lease vergeben oder zurückgezogen wurde. Ändern Sie die Voreinstellung nicht, wenn der Server so konfiguriert wurde, dynamische DNS-Updates zu unterstützen.

⑦ Diese Zeile erstellt einen Pool der verfügbaren IP-Adressen, die für die Zuweisung der DHCP-Clients reserviert sind. Der Bereich muss für das angegebene Netz oder Subnetz aus der vorherigen Zeile gültig sein.

⑧ Legt das Standard-Gateway für das Netz oder Subnetz fest, das nach der öffnenden Klammer `{` gültig ist.

⑨ Bestimmt die Hardware-MAC-Adresse eines Clients, durch die der DHCP-Server den Client erkennt, der eine Anforderung an ihn stellt.

⑩ Einem Rechner soll immer die gleiche IP-Adresse zugewiesen werden. Hier ist auch ein Rechnername gültig, da der DHCP-Server den Rechnernamen auflöst, bevor er das Lease zuweist.

Die Konfigurationsdatei unterstützt viele weitere Optionen. Lesen Sie dhcpd.conf(5), die mit dem Server installiert wird, für Details und Beispiele.

Nachdem dhcpd.conf konfiguriert ist, aktivieren Sie den DHCP-Server in /etc/rc.conf:

```
dhcpd_enable="YES"
dhcpd_ifaces="dc0"
```

Dabei müssen Sie `dc0` durch die Gerätedatei (mehrere Gerätedateien müssen durch Leerzeichen getrennt werden) ersetzen, die der DHCP-Server auf Anfragen von DHCP-Clients hin überwachen

soll.

Starten Sie den Server mit folgenden Befehl:

```
# service isc-dhcpd start
```

Künftige Änderungen an der Konfiguration des Servers erfordern, dass der Dienst `dhcpd` gestoppt und anschließend mit service(8) gestartet wird.

- /usr/local/sbin/dhcpd

  Weitere Informationen zu dhcpd finden Sie in dhcpd(8).

- /usr/local/etc/dhcpd.conf

  Die Konfigurationsdatei des Servers muss alle Informationen enthalten, die an die Clients weitergegeben werden soll. Außerdem sind hier Informationen zur Konfiguration des Servers enthalten. Diese Konfigurationsdatei wird in dhcpd.conf(5) beschrieben.

- /var/db/dhcpd.leases

  Der DHCP-Server hat eine Datenbank, die alle vergebenen Leases enthält. Diese wird als Logdatei erzeugt. dhcpd.leases(5) enthält eine ausführliche Beschreibung.

- /usr/local/sbin/dhcrelay

  Dieser Daemon wird in komplexen Umgebungen verwendet, in denen ein DHCP-Server eine Anfrage eines Clients an einen DHCP-Server in einem separaten Netzwerk weiterleitet. Wenn Sie diese Funktion benötigen, müssen Sie net/isc-dhcp44-relay installieren. Weitere Informationen zu diesem Thema finden Sie in dhcrelay(8).

## 52.7. Domain Name System (DNS)

DNS ist das für die Umwandlung von Rechnernamen in IP-Adressen zuständige Protokoll. Im Internet wird DNS durch ein komplexes System von autoritativen Root-Nameservern, Top Level Domain-Servern (TLD) sowie anderen kleineren Nameservern verwaltet, die individuelle Domaininformationen speichern und untereinander abgleichen. Für einfache DNS-Anfragen wird auf dem lokalen System kein Nameserver benötigt.

Die folgende Tabelle beschreibt einige mit DNS verbundenen Begriffe:

*Tabelle 28. DNS-Begriffe*

| Begriff | Bedeutung |
|---|---|
| Forward-DNS | Rechnernamen in IP-Adressen umwandeln. |
| Origin (Ursprung) | Die in einer bestimmten Zonendatei beschriebene Domäne. |

| Begriff | Bedeutung |
|---|---|
| Resolver | Ein Systemprozess, durch den ein Rechner Zoneninformationen von einem Nameserver anfordert. |
| Reverse-DNS | die Umwandlung von IP-Adressen in Rechnernamen |
| Root-Zone | Der Beginn der Internet-Zonenhierarchie. Alle Zonen befinden sich innerhalb der Root-Zone. Dies ist analog zu einem Dateisystem, in dem sich alle Dateien und Verzeichnisse innerhalb des Wurzelverzeichnisses befinden. |
| Zone | Eine individuelle Domäne, Unterdomäne, oder ein Teil von DNS, der von der gleichen Autorität verwaltet wird. |

Es folgen nun einige Zonenbeispiele:

- Innerhalb der Dokumentation wird die Root-Zone in der Regel mit `.` bezeichnet.
- `org.` ist eine Top level Domain (TLD) innerhalb der Root-Zone.
- `example.org.` ist eine Zone innerhalb der `org.`-TLD.
- `1.168.192.in-addr.arpa.` ist die Zone mit allen IP-Adressen des Bereichs `192.168.1.*`.

Wie man an diesen Beispielen erkennen kann, befindet sich der spezifischere Teil eines Rechnernamens auf der linken Seite der Adresse. `example.org.` beschreibt einen Rechner also genauer als `org.`, während `org.` genauer als die Root-Zone ist. Jeder Teil des Rechnernamens hat Ähnlichkeiten mit einem Dateisystem, in dem etwa /dev dem Wurzelverzeichnis untergeordnet ist.

### 52.7.1. Gründe für die Verwendung eines Nameservers

Es gibt zwei Arten von Nameservern: Autoritative Nameserver sowie zwischenspeichernde (cachende, auch bekannt als auflösende) Nameserver.

Ein autoritativer Nameserver ist notwendig, wenn

- Sie anderen verbindliche DNS-Auskünfte erteilen wollen.
- eine Domain, beispielsweise `example.org`, registriert wird, und den zu dieser Domain gehörenden Rechnern IP-Adressen zugewiesen werden müssen.
- ein IP-Adressblock reverse-DNS-Einträge benötigt, um IP-Adressen in Rechnernamen auflösen zu können.
- ein Backup-Nameserver (auch Slaveserver genannt) oder ein zweiter Nameserver auf Anfragen antworten soll.

Ein cachender Nameserver ist notwendig, weil

- ein lokaler DNS-Server Daten zwischenspeichern und daher schneller auf Anfragen reagieren

kann als ein entfernter Server.

Wird nach www.FreeBSD.org gesucht, leitet der Resolver diese Anfrage an den Nameserver des ISPs weiter und nimmt danach das Ergebnis der Abfrage entgegen. Existiert ein lokaler, zwischenspeichernder DNS-Server, muss dieser die Anfrage nur einmal nach außen weitergeben. Für alle weiteren Anfragen ist dies nicht mehr nötig, da diese Information nun lokal gespeichert ist.

### 52.7.2. DNS-Server Konfiguration

Unbound ist im Basissystem von FreeBSD enthalten. In der Voreinstellung bietet es nur die DNS-Auflösung auf dem lokalen Rechner. Obwohl das im Basissystem enthaltene Unbound konfiguriert werden kann, um Namensauflösung über den lokalen Rechner hinweg bereitzustellen, ist es empfehlenswert für solche Anforderungen Unbound aus der FreeBSD Ports-Sammlung zu installieren.

Um Unbound zu aktivieren, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
local_unbound_enable="YES"
```

Alle vorhandenen Nameserver aus /etc/resolv.conf werden als Forwarder in der neuen Unbound-Konfiguration benutzt.



Wenn einer der aufgeführten Nameserver kein DNSSEC unterstützt, wird die lokale DNS-Auflösung nicht funktionieren. Testen Sie jeden Server und entfernen Sie die Server, die den Test nicht bestehen. Das folgende Beispiel zeigt einen Trust Tree beziehungsweise einen Fehler für den Nameserver auf 192.168.1.1:

```
# drill -S FreeBSD.org @192.168.1.1
```

Nachdem jeder Server für DNSSEC konfiguriert ist, starten Sie Unbound:

```
# service local_unbound onestart
```

Dieses Kommando sorgt für die Aktualisierung von /etc/resolv.conf, so dass Abfragen für DNSSEC gesicherte Domains jetzt funktionieren. Führen Sie folgenden Befehl aus, um den DNSSECTrust Tree für FreeBSD.org zu überprüfen:

```
% drill -S FreeBSD.org
;; Number of trusted keys: 1
;; Chasing: freebsd.org. A

DNSSEC Trust tree:
freebsd.org. (A)
|---freebsd.org. (DNSKEY keytag: 36786 alg: 8 flags: 256)
    |---freebsd.org. (DNSKEY keytag: 32659 alg: 8 flags: 257)
    |---freebsd.org. (DS keytag: 32659 digest type: 2)
```

```
        |---org. (DNSKEY keytag: 49587 alg: 7 flags: 256)
            |---org. (DNSKEY keytag: 9795 alg: 7 flags: 257)
            |---org. (DNSKEY keytag: 21366 alg: 7 flags: 257)
            |---org. (DS keytag: 21366 digest type: 1)
            |   |---. (DNSKEY keytag: 40926 alg: 8 flags: 256)
            |       |---. (DNSKEY keytag: 19036 alg: 8 flags: 257)
            |---org. (DS keytag: 21366 digest type: 2)
                |---. (DNSKEY keytag: 40926 alg: 8 flags: 256)
                    |---. (DNSKEY keytag: 19036 alg: 8 flags: 257)
;; Chase successful
```

# 52.8. Apache HTTP-Server

Der Open Source Apache HTTP-Server ist der am weitesten verbreitete Webserver. Dieser Webserver ist nicht im Basissystem von FreeBSD enthalten, kann aber als Paket oder Port www/apache24 installiert werden.

Dieser Abschnitt beschreibt die Konfiguration der Version 2.*x* des Apache HTTP-Server. Weiterführende Informationen und Konfigurationsanweisungen für Apache 2.X finden Sie unter httpd.apache.org.

## 52.8.1. Apache konfigurieren und starten

Der Apache HTTP-Server wird unter FreeBSD primär in /usr/local/etc/apache2x/httpd.conf konfiguriert, wobei das *x* die Versionsnummer darstellt. In dieser Textdatei leitet ein `#` einen Kommentar ein. Die am häufigsten verwendeten Optionen sind:

`ServerRoot "/usr/local"`

Legt das Standardwurzelverzeichnis für die Apache-Installation fest. Binärdateien werden in die Verzeichnisse bin und sbin unterhalb des Serverwurzelverzeichnisses installiert, während sich Konfigurationsdateien im Unterverzeichnis etc/apache2x befinden.

`ServerAdmin you@example.com`

Die E-Mail-Adresse, an die Mitteilungen über Serverprobleme geschickt werden. Diese Adresse erscheint auf vom Server erzeugten Seiten, beispielsweise auf Fehlerseiten.

`ServerName www.example.com:80`

Erlaubt dem Administrator, einen Rechnernamen festzulegen, den der Server an die Clients sendet. Beispielsweise könnte `www` statt des richtigen Rechnernamens verwendet werden. Wenn das System keinen eingetragenen DNS-Namen hat, kann stattdessen die IP-Adresse eingetragen werden. Lauscht der Server auf einem anderen Port, tauschen Sie die `80` gegen eine entsprechende Portnummer.

`DocumentRoot "/usr/local/www/apache2x/data"`

Das Verzeichnis, in dem die Dokumente abgelegt sind. In der Voreinstellung befinden sich alle Seiten in diesem Verzeichnis, durch symbolische Links oder Aliase lassen sich aber auch andere Orte festlegen.

Es ist empfehlenswert, eine Sicherungskopie der Apache-Konfigurationsdatei anzulegen, bevor Änderungen durchgeführt werden. Wenn die Konfiguration von Apache abgeschlossen ist, speichern Sie die Datei und überprüfen Sie die Konfiguration mit `apachectl`. Der Befehl `apachectl configtest` sollte `Syntax OK` zurückgeben.

Um den Apache beim Systemstart zu starten, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
apache24_enable="YES"
```

Wenn Sie während des Systemstarts weitere Parameter an den Apache übergeben wollen, können Sie diese durch eine zusätzliche Zeile in rc.conf angeben:

```
apache24_flags=""
```

Wenn apachectl keine Konfigurationsfehler meldet, starten Sie `httpd`:

```
# service apache24 start
```

Sie können den `httpd`-Dienst testen, indem Sie `http://localhost` in einen Browser eingeben, wobei Sie *localhost* durch den vollqualifizierten Domainnamen der Maschine ersetzen, auf dem der `httpd` läuft. Die Standard Webseite, die angezeigt wird, ist /usr/local/www/apache24/data/index.html.

Die Konfiguration von Apache kann bei nachfolgenden Änderungen an der Konfigurationsdatei bei laufendem `httpd`, auf Fehler überprüft werden. Geben Sie dazu folgendes Kommando ein:

```
# service apache24 configtest
```



Es ist wichitg zu beachten, dass `configtest` kein rc(8)-Standard ist, und somit nicht zwingend mit anderen rc(8)-Startskripten funktioniert.

### 52.8.2. Virtual Hosting

Virtual Hosting ermöglicht es, mehrere Webseiten auf einem Apache-Server laufen zu lassen. Die virtuellen Hosts können *IP-basiert* oder *namensbasiert* sein. IP-basiertes virtual Hosting verwendet eine IP-Adresse für jede Webseite. Beim namensbasierten virtual Hosting wird der HTTP/1.1-Header der Clients dazu verwendet, den Rechnernamen zu bestimmen. Dadurch wird es möglich, mehrere Domains unter der gleichen IP-Adresse zu betreiben.

Damit der Apache namenbasierte virtuelle Domains verwalten kann, fügen Sie für jede Webseite einen separaten `VirtualHost`-Block ein. Wenn der Webserver beispielsweise `www.domain.tld` heißt und die virtuelle Domain `www.someotherdomain.tld` einrichtet werden soll, ergänzen Sie httpd.conf um folgende Einträge:

```
<VirtualHost *>
```

```
    ServerName www.domain.tld
    DocumentRoot /www/domain.tld
</VirtualHost>

<VirtualHost *>
    ServerName www.someotherdomain.tld
    DocumentRoot /www/someotherdomain.tld
</VirtualHost>
```

Setzen Sie für jeden virtuellen Host die entsprechenden Werte für `ServerName` und `DocumentRoot`.

Ausführliche Informationen zum Einrichten von virtuellen Hosts finden Sie in der offiziellen Apache-Dokumentation unter http://httpd.apache.org/docs/vhosts/.

### 52.8.3. Häufig verwendete Apache-Module

Apache verwendet Module, die den Server um zusätzliche Funktionen erweitern. Eine vollständige Auflistung der zur Verfügung stehenden Module und Konfigurationsdetails finden Sie unter http://httpd.apache.org/docs/current/mod/.

In FreeBSD können einige Module mit dem Port www/apache24 kompiliert werden. Geben Sie in /usr/ports/www/apache24`make config` ein, um zu sehen, welche Module zur Verfügung stehen und welche Module in der Voreinstellung aktiviert sind. Wenn ein Modul nicht zusammen mit dem Port kompiliert wird, bietet die Ports-Sammlung die Möglichkeit viele Module zu installieren. Dieser Abschnitt beschreibt drei der am häufigsten verwendeten Module.

#### SSL-Unterstützung

Zu einem bestimmten Zeitpunkt erforderte die Unterstützung von SSL innerhalb von Apache ein separates Modul namens mod_ssl. Dies ist nicht mehr der Fall und die Installation des Apache-Webservers wird im Standard mit SSL-Unterstützung ausgeliefert. Ein Beispiel, wie Sie SSL-Unterstützung für einen Webserver aktivieren können, finden Sie in der Datei httpd-ssl.conf im Verzeichnis /usr/local/etc/apache24/extra. In diesem Verzeichnis befindet sich auch eine Beispieldatei namens ssl.conf-sample. Es wird empfohlen, beide Dateien zu überprüfen, um sichere Webseiten auf dem Apache-Webserver einzurichten.

Nachdem die Konfiguration von SSL abgeschlossen ist, muss die folgende Zeile in httpd.conf auskommentiert werden, um die Änderungen beim nächsten Neustart oder erneuten Laden der Konfiguration zu aktivieren:

```
#Include etc/apache24/extra/httpd-ssl.conf
```



SSL in Version 2 und 3 haben bekannte Schwachstellen. Es wird dringend empfohlen, TLS Version 1.2 und 1.3 anstelle der älteren SSL-Optionen zu aktivieren. Dies kann durch die Einstellung der folgenden Optionen in ssl.conf erreicht werden:

```
SSLProtocol all -SSLv3 -SSLv2 +TLSv1.2 +TLSv1.3
SSLProxyProtocol all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1
```

Um die Konfiguration von SSL im Webserver abzuschließen, entfernen Sie den Kommentar in der folgenden Zeile, um sicherzustellen, dass die Konfiguration bei einem Neustart oder beim erneuten laden der Konfiguration von Apache übernommen wird:

```
# Secure (SSL/TLS) connections
Include etc/apache24/extra/httpd-ssl.conf
```

Diese Zeilen müssen in httpd.conf ebenfalls auskommentiert bleiben, um SSL in Apache vollständig zu unterstützen:

```
LoadModule authn_socache_module libexec/apache24/mod_authn_socache.so
LoadModule socache_shmcb_module libexec/apache24/mod_socache_shmcb.so
LoadModule ssl_module libexec/apache24/mod_ssl.so
```

Der nächste Schritt ist die Kooperation mit einer Zertifizierungsstelle, um die entsprechenden Zertifikate auf dem System installieren zu lassen. Dadurch wird eine Vertrauenskette für die Webseite etabliert und jegliche Warnungen vor selbstsignierten Zertifikaten verhindert.

#### mod_perl

Das Modul mod_perl macht es möglich, vollständig in Perl geschriebene Apache-Module zu erzeugen. Da der Perl-Interpreter in den Server eingebettet wird, muss weder ein externer Interpreter noch Perl zusätzlich aufgerufen werden.

mod_perl wird über den Port oder das Paket www/mod_perl2 installiert. Dokumentation für dieses Modul finden Sie unter http://perl.apache.org/docs/2.0/index.html.

#### mod_php

*PHP: Hypertext Preprocessor* (PHP) ist eine vielseitig verwendbare Skriptsprache, die besonders für die Web-Entwicklung geeignet ist. PHP kann in HTML eingebettet werden und ähnelt von der Syntax her Sprachen wie C, Java™ und Perl. Das Hauptanliegen von PHP ist es, Web-Entwicklern die rasche Erstellung von dynamisch erzeugten Internetseiten zu ermöglichen.

PHP und weitere in PHP geschriebene Funktionen unterstützt, muss das entsprechende Paket installiert werden.

Sie können mit pkg die Paketdatenbank nach allen unterstützten PHP-Versionen durchsuchen:

```
# pkg search php
```

Die Ausgabe ist eine Liste mit Versionen und Funktionen des jeweiligen Pakets. Die Komponenten sind vollständig modular, d.h. die Funktionen werden durch die Installation des entsprechenden

Pakets aktiviert. Geben Sie folgenden Befehl ein, um PHP-Version 7.4 für Apache zu installieren:

```
# pkg install mod_php74
```

Falls irgendwelche Pakete Abhängigkeiten besitzen, werden diese zusätzlichen Pakete ebenfalls installiert.

Standardmäßig ist PHP nicht aktiviert. Die folgenden Zeilen müssen in der Apache-Konfigurationsdatei unterhalb von /usr/local/etc/apache24 hinzugefügt werden, um PHP zu aktivieren:

```
<FilesMatch "\.php$">
    SetHandler application/x-httpd-php
</FilesMatch>
<FilesMatch "\.phps$">
    SetHandler application/x-httpd-php-source
</FilesMatch>
```

Zusätzlich muss auch der `DirectoryIndex` in der Konfigurationsdatei aktualisiert werden und Apache muss entweder neu gestartet, oder die Konfiguration neu geladen werden, damit die Änderungen wirksam werden.

Mit `pkg` kann die Unterstützung für viele weitere PHP-Funktionen installiert werden. Um beispielsweise die Unterstützung für XML oder SSL zu erhalten, installieren Sie die entsprechenden Pakete:

```
# pkg install php74-xml php74-openssl
```

Wie zuvor muss die Konfiguration von Apache neu geladen werden, damit die Änderungen wirksam werden. Dies gilt auch für Fälle, in denen lediglich ein Modul installiert wurde.

Geben Sie folgenden Befehl ein, um einen geordneten Neustart durchzuführen und die Konfiguration neu zu laden:

```
# apachectl graceful
```

Sobald die Installation abgeschlossen ist, gibt es zwei Möglichkeiten, um eine Liste der installierten PHP-Module und Informationen über die Umgebung der Installation zu erhalten. Die erste Möglichkeit besteht darin, die vollständige PHP-Binärdatei zu installieren und den Befehl auszuführen, um die Informationen zu erhalten:

```
# pkg install php74
```

```
# php -i | less
```

Da die Ausgabe des Befehls sehr umfangreich ist, ist die Weiterleitung an einen Pager, wie beispielsweise `more` oder `less`, sinnvoll.

Um Änderungen an der globalen Konfiguration von PHP vorzunehmen, gibt es schließlich eine gut dokumentierte Datei, die in /usr/local/etc/php.ini installiert ist. Zum Zeitpunkt der Installation wird diese Datei nicht existieren, da zwei Versionen zur Auswahl stehen. Eine php.ini-development und eine php.ini-production. Diese Dateien sind Ansatzpunkte, die Administratoren bei der Implementierung unterstützen sollen.

### 52.8.4. Dynamische Webseiten

Neben mod_perl und mod_php stehen noch weitere Sprachen zur Erstellung von dynamischen Inhalten zur Verfügung. Dazu gehören auch Django und Ruby on Rails.

#### Django

Bei Django handelt es sich um ein unter der BSD-Lizenz verfügbares Framework zur schnellen Erstellung von mächtigen Internet-Applikationen. Es beinhaltet einen objekt-relationalen Mapper (wodurch Datentypen als Phyton-Objekte entwickelt werden können) sowie eine API für den dynamischen Datenbankzugriff auf diese Objekte, ohne dass Entwickler jemals SQL-Code schreiben müssen. Zusätzlich existiert ein umfangreiches Template-System, wodurch die Programmlogik von der HTML-Präsentation getrennt werden kann.

Django setzt das Modul mod_python und eine SQL-Datenbank voraus. In FreeBSD wird bei der Installation von www/py-django automatisch mod_python installiert. Als Datenbanken werden PostgreSQL, MySQL und SQLite unterstützt, wobei SQLite die Voreinstellung ist. Wenn Sie die Datenbank ändern möchten, geben Sie in /usr/ports/www/py-django`make config` ein und installieren Sie den Port neu.

Nachdem Django installiert ist, benötigt die Anwendung ein Projektverzeichnis und die Apache-Konfiguration, um den eingebetteten Python-Interpreter zu nutzen. Dieser Interpreter wird verwendet um die Anwendung für spezifische URLs der Seite aufrufen.

Damit Apache Anfragen für bestimmte URLs an die Web-Applikation übergeben kann, müssen Sie den vollständigen Pfad zum Projektverzeichnis in httpd.conf festlegen:

```
<Location "/">
    SetHandler python-program
    PythonPath "['/pfad/zu/den/django/paketen/'] + sys.path"
    PythonHandler django.core.handlers.modpython
    SetEnv DJANGO_SETTINGS_MODULE mysite.settings
    PythonAutoReload On
    PythonDebug On
</Location>
```

Weitere Informationen zur Verwendung von Django finden Sie unter

https://docs.djangoproject.com/en/1.6/.

**Ruby on Rails**

Ruby on Rails ist ein weiteres, als Open Source verfügbares Webframework. Es bietet einen kompletten Entwicklungsstack und erlaubt es Webentwicklern, umfangreiche und mächtige Applikationen in kurzer Zeit zu programmieren. Unter FreeBSD kann das Framework über den Port oder das Paket www/rubygem-rails installiert werden.

Weitere Informationen zur Verwendung von Ruby on Rails finden Sie unter http://rubyonrails.org/documentation.

# 52.9. File Transfer Protocol (FTP)

Das File Transfer Protocol (FTP) ermöglicht auf einfache Art und Weise den Dateiaustausch mit einem FTP-Server. Der FTP-Server ftpd ist bei FreeBSD bereits im Basisystem enthalten.

FreeBSD verwendet mehrere Konfigurationsdateien, um den Zugriff auf den FTP zu kontrollieren. Dieser Abschnitt fasst diese Dateien zusammen. In ftpd(8) finden Sie weitere Inforamtionen über den integrierten FTP-Server.

## 52.9.1. Konfiguration

Der wichtigste Punkt ist hier die Entscheidung darüber, welche Benutzer auf den FTP-Server zugreifen dürfen. Ein FreeBSD-System verfügt über diverse Systembenutzerkonten, die jedoch nicht auf den FTP-Server zugreifen sollen. Die Datei /etc/ftpusers enthält alle Benutzer, die vom FTP-Zugriff ausgeschlossen sind. In der Voreinstellung gilt dies auch die gerade erwähnten Systembenutzerkonten. Sie können über diese Datei weitere Benutzer vom FTP-Zugriff ausschließen.

In einigen Fällen kann es wünschenswert sein, den Zugang für manche Benutzer einzuschränken, ohne dabei FTP komplett zu verbieten. Dazu passen Sie /etc/ftpchroot, wie in ftpchroot(5) beschrieben, entsprechend an. Diese Datei enthält Benutzer und Gruppen sowie die für sie geltenden Einschränkungen für FTP.

Um anonymen FTP-Zugriff auf dem Server zu aktivieren, muss ein Benutzer `ftp` auf dem FreeBSD-System angelegt werden. Danach können sich Benutzer mit dem Benutzernamen `ftp` oder `anonymous` am FTP-Server anmelden. Das Passwort ist dabei beliebig, allerdings wird dazu in der Regel eine E-Mail-Adresse verwendet. Meldet sich ein anonymer Benutzer an, aktiviert der FTP-Server chroot(2), um den Zugriff auf das Heimatverzeichnis des Benutzers `ftp` zu beschränken.

Es gibt zwei Textdateien, deren Inhalt den FTP-Clients bei der Anmeldung angezeigt wird. Der Inhalt von /etc/ftpwelcome wird angezeigt, bevor der Login-Prompt erscheint. Nach einer erfolgreichen Anmeldung wird der Inhalt von /etc/ftpmotd angezeigt. Beachten Sie aber, dass es dabei um einen Pfad relativ zur Umgebung des anzumeldenden Benutzers handelt. Bei einer anonymen Anmeldung würde also der Inhalt von ~ftp/etc/ftpmotd angezeigt.

Sobald der FTP-Server konfiguriert ist, setzen Sie die entsprechende Variable in /etc/rc.conf, damit der Dienst beim Booten gestartet wird:

```
ftpd_enable="YES"
```

Starten Sie den Dienst:

```
# service ftpd start
```

Testen Sie die Verbindung zum FTP-Server, indem Sie folgendes eingeben:

```
% ftp localhost
```

### 52.9.2. Wartung

Der ftpd-Daemon verwendet syslog(3), um Protokolldateien zu erstellen. In der Voreinstellung werden alle FTP betreffenden Nachrichten nach /var/log/xferlog geschrieben. Dies lässt sich aber durch das Einfügen der folgenden Zeile in /etc/syslog.conf ändern:

```
ftp.info      /var/log/xferlog
```



Beachten Sie, dass mit dem Betrieb eines anonymen FTP-Servers verschiedene Sicherheitsrisiken verbunden sind. Problematisch ist hier vor allem die Erlaubnis zum anonymen Upload von Dateien. Dadurch könnte der Server zur Verbreitung von illegaler oder nicht lizensierter Software oder noch Schlimmeren missbraucht werden. Wenn anonyme FTP-Uploads dennoch erforderlich sind, sollten Sie die Zugriffsrechte so setzen, dass solche Dateien erst nach Zustimmung eines Administrators von anderen Benutzern heruntergeladen werden können.

## 52.10. Datei- und Druckserver für Microsoft® Windows®-Clients (Samba)

Samba ist ein beliebtes Open Source Softwarepaket, das Datei- und Druckdienste über das SMB/CIFS-Protokoll zur Verfügung stellt. Dieses Protokoll ist in Microsoft® Windows®-Systemen enthalten und kann über die Installation der Samba-Client-Bibliotheken in andere Betriebssysteme integriert werden. Das Protokoll ermöglicht es Clients auf freigegebene Daten und Drucker zuzugreifen, so als ob es sich um lokale Drucker und Festplatten handeln würde.

Unter FreeBSD können die Samba-Client-Bibliotheken über den Port oder das Paket net/samba410 installiert werden. Der Client ermöglicht es einem FreeBSD-System auf SMB/CIFS-Freigaben in einem Microsoft® Windows®-Netzwerk zuzugreifen.

Ein FreeBSD-System kann auch als Samba-Server agieren, wenn Sie den Port oder das Paket net/samba410 installieren. Dies erlaubt es dem Administrator SMB/CIFS-Freigaben auf dem FreeBSD-System einzurichten, auf welche dann Clients mit Microsoft® Windows® oder den Samba-Client-Bibliotheken zugreifen können.

### 52.10.1. Konfiguration des Servers

Samba wird in /usr/local/etc/smb4.conf konfiguriert. Diese Datei muss erstellt werden, bevor Samba benutzt werden kann.

Eine einfache smb4.conf, wie hier gezeigt, stellt den Zugriff auf Verzeichnisse und Drucker für Windows®-Clients in einer Arbeitsgruppe (engl. Workgroup) zur Verfügung. In aufwendigeren Installationen, in denen LDAP oder Active Directory zum Einsatz kommt, ist es einfacher die smb4.conf mit dem Werkzeug samba-tool(8) zu erstellen.

```
[global]
workgroup = WORKGROUP
server string = Samba Server Version %v
netbios name = ExampleMachine
wins support = Yes
security = user
passdb backend = tdbsam

# Example: share /usr/src accessible only to 'developer' user
[src]
path = /usr/src
valid users = developer
writable  = yes
browsable = yes
read only = no
guest ok = no
public = no
create mask = 0666
directory mask = 0755
```

#### Globale Einstellungen

Einstellungen für das Netzwerk werden in /usr/local/etc/smb4.conf definiert:

`workgroup`

Der Name der Arbeitsgruppe.

`netbios name`

Der NetBIOS-Namen fest, unter dem der Samba-Server bekannt ist. In der Regel handelt es sich dabei um den ersten Teil des DNS-Namens des Servers.

`server string`

Legt die Beschreibung fest, die angezeigt wird, wenn mit `net view` oder anderen Netzwerkprogrammen Informationen über den Server angefordert werden.

`wins support`

Legt fest, ob Samba als WINS-Server fungieren soll. Aktivieren Sie die Unterstützung für WINS auf maximal einem Server im Netzwerk.

#### Samba absichern

Die wichtigsten Einstellungen in /usr/local/etc/smb4.conf betreffen das zu verwendende Sicherheitsmodell sowie das Backend-Passwortformat. Die folgenden Direktiven steuern diese Optionen:

`security`

Die häufigsten Optionen sind `security = share` und `security = user`. Wenn die Clients Benutzernamen verwenden, die den Benutzernamen auf dem FreeBSD-Rechner entsprechen, dann sollte die Einstellung `user level` verwendet werden. Dies ist die Standardeinstellung. Allerdings ist es dazu erforderlich, dass sich die Clients auf dem Rechner anmelden, bevor sie auf gemeinsame Ressourcen zugreifen können.

In der Einstellung `share level` müssen sich Clients nicht unter Verwendung eines gültigen Logins auf dem Rechner anmelden, bevor sie auf gemeinsame Ressourcen zugreifen können. In früheren Samba-Versionen war dies die Standardeinstellung.

`passdb backend`

Samba erlaubt verschiedene Backend-Authentifizierungsmodelle. Clients können sich durch LDAP, NIS+, eine SQL-Datenbank oder eine Passwortdatei authentifizieren. Die empfohlene Authentifizierungsmethode, `tdbsam`, ist ideal für einfache Netzwerke und wird hier vorgestellt. Für größere oder komplexere Netzwerke wird `ldapsam` empfohlen. `smbpasswd` war der frühere Standard und gilt mittlerweile als veraltet.

#### Samba Benutzer

Damit Windows®-Clients auf die Freigaben zugreifen können, müssen die FreeBSD-Benutzerkonten in der `SambaSAMAccount`-Datenbank zugeordnet werden. Für bereits vorhandene Benutzerkonten kann dazu pdbedit(8) benutzt werden:

```
# pdbedit -a username
```

Dieser Abschnitt beschreibt lediglich die am häufigsten verwendeten Einstellungen. Ausführliche Informationen zur Konfiguration von Samba finden Sie im Official Samba HOWTO.

### 52.10.2. Samba starten

Damit Samba beim Systemstart automatisch aktiviert wird, fügen Sie die folgende Zeile in /etc/rc.conf ein:

```
samba_server_enable="YES"
```

Jetzt kann Samba direkt gestartet werden:

```
# service samba_server start
Performing sanity check on Samba configuration: OK
Starting nmbd.
```

```
Starting smbd.
```

Samba verwendet drei Daemonen. Sowohl nmbd als auch smbd werden durch `samba_enable` gestartet. Wenn eine Namensauflösung über winbind benötigt wird, setzen Sie zusätzlich:

```
winbindd_enable="YES"
```

Samba kann jederzeit durch folgenden Befehl beendet werden:

```
# service samba_server stop
```

Samba ist ein komplexes Softwarepaket mit umfassenden Funktionen, die eine weitreichende Integration von Microsoft® Windows®-Netzwerken ermöglichen. Für eine Beschreibung dieser Zusatzfunktionen sollten Sie sich auf http://www.samba.org umsehen.

# 52.11. Die Uhrzeit mit NTP synchronisieren

Die interne Uhrzeit eines Computers ist nie ganz exakt. Dies ist problematisch, da viele Dienste darauf angewiesen sind, dass die Computer im Netzwerk die exakte Uhrzeit übermitteln. Die exakte Uhrzeit ist auch erforderlich um sicherzustellen, dass die Zeitstempel der Dateien konsistent bleiben. Das Network Time Protocol (NTP) bietet die Möglichkeit, die exakte Uhrzeit in einem Netzwerk zur Verfügung zu stellen.

FreeBSD enthält ntpd(8), das andere NTP-Server abfragen kann um die Uhrzeit auf diesem Computer zu synchronisieren, oder um selbst die Uhrzeit für andere Computer im Netzwerk bereitzustellen.

Dieser Abschnitt beschreibt die Konfiguration von ntpd unter FreeBSD. Zusätzliche Dokumentation im HTML-Format finden Sie in /usr/shared/doc/ntp/.

## 52.11.1. NTP konfigurieren

FreeBSD enthält mit ntpd ein Werkzeug, das zur Synchronisation der Uhrzeit verwendet werden kann. Die Konfiguration von Ntpd erfolgt über Variablen in rc.conf(5) und /etc/ntp.conf, und wird in den folgenden Abschnitten beschrieben.

Ntpd kommuniziert über UDP mit mit seinen Peers. Sämtliche Firewalls zwischen Ihrem Rechner und seinen NTP-Peers müssen so konfiguriert sein, dass UDP-Pakete auf Port 123 ein- und ausgehen können.

**/etc/ntp.conf**

Ntpd liest /etc/ntp.conf um herauszufinden, welche NTP-Server abgefragt werden sollen. Die Auswahl mehrerer NTP-Server wird empfohlen, falls einer der Server nicht erreichbar ist oder sich seine Uhr als unzuverlässig erweist. Wenn ntpd Antworten erhält, bevorzugt es zuverlässige Server gegenüber weniger zuverlässigen. Die abgefragten Server können lokal im Netzwerk, von einem

ISP bereitgestellt oder aus einer Liste öffentlich zugänglicher NTP-Server ausgewählt werden. Wenn Sie einen öffentlichen NTP-Server auswählen, wählen Sie einen geografisch nahen NTP-Server und überprüfen Sie dessen Nutzungsrichtlinien. Das Schlüsselwort `pool` wählt einen oder mehrere Server aus einem Pool von Servern aus. Eine Liste mit öffentlich zugänglichen NTP-Pools ist ebenfalls verfügbar, sortiert nach geografischen Gebieten. Darüber hinaus bietet FreeBSD einen vom Projekt gespendeten Pool, `0.freebsd.pool.ntp.org`.

*Beispiel 47. Beispiel für /etc/ntp.conf*

Dies ist ein einfaches Beispiel für eine ntp.conf-Datei. Die Einträge können so übernommen werden, wie sie sind. Die Datei enthält die notwendigen Einschränkungen für den Betrieb an einer öffentlich zugänglichen Netzwerkverbindung.

```
# Disallow ntpq control/query access.  Allow peers to be added only
# based on pool and server statements in this file.
restrict default limited kod nomodify notrap noquery nopeer
restrict source  limited kod nomodify notrap noquery

# Allow unrestricted access from localhost for queries and control.
restrict 127.0.0.1
restrict ::1

# Add a specific server.
server ntplocal.example.com iburst

# Add FreeBSD pool servers until 3-6 good servers are available.
tos minclock 3 maxclock 6
pool 0.freebsd.pool.ntp.org iburst

# Use a local leap-seconds file.
leapfile "/var/db/ntpd.leap-seconds.list"
```

Das Format dieser Datei ist in ntp.conf(5) beschrieben. Die folgenden Erläuterungen geben einen Überblick über die Schlüsselwörter, die in dem obigen Beispiel benutzt werden.

In der Voreinstellung ist ein NTP-Server für jeden Host im Netzwerk zugänglich. Das Schlüsselwort `restrict` steuert, welche Systeme auf den Server zugreifen dürfen. Es werden mehrere `restrict` -Einträge unterstützt, die jeweils die vorherigen Anweisungen verfeinern. Die im Beispiel gezeigten Werte gewährem dem lokalen System vollen Abfrage- und Kontrollzugriff, während entfernte Systemen nur die Möglichkeit gegeben wird, die Zeit abzufragen. Weitere Details finden Sie im Abschnitt `Access Control Support` von ntp.conf(5).

Das Schlüsselwort `server` gibt einen einzelnen Server zur Abfrage der Zeit an. Die Datei kann das Schlüsselwort `server` mehrmals enthalten, wobei pro Zeile jeweils ein Server aufgeführt ist. Das Schlüsselwort `pool` gibt einen Pool von Servern an. Ntpd fügt bei Bedarf einen oder mehrere Server aus diesem Pool hinzu, um die Anzahl der mit dem Wert `tos minclock` Peers zu erreichen. Das Schlüsselwort `iburst` weist ntpd an, einen Burst von acht schnellen Paketen mit dem Server auszutauschen, wenn der Kontakt zum ersten Mal hergestellt wird, um so die Systemzeit schneller

zu synchronisieren.

Das Schlüsselwort `leapfile` gibt den Pfad einer Datei an, die Informationen über Schaltsekunden enthält. Die Datei wird automatisch durch periodic(8) aktualisiert. Der angegebene Pfad muss mit dem in der Variable `ntp_db_leapfile` aus /etc/rc.conf übereinstimmen.

#### NTP-Einträge in /etc/rc.conf

Um ntpd beim Booten zu starten, Sie in /etc/rc.conf den Eintrag `ntpd_enable="YES"` hinzu. Danach kann ntpd direkt gestartet werden:

```
# service ntpd start
```

Lediglich `ntpd_enable` wird benötigt um ntpd benutzen zu können. Die unten aufgeführten rc.conf -Variablen können bei Bedarf ebenfalls verwendet werden.

Ist `ntpd_sync_on_start="YES"` konfiguriert, setzt ntpd die Uhrzeit beim Systemstart, unabhängig davon wie hoch die Abweichung ist. Normalerweise protokolliert ntpd eine Fehlermeldung und beendet sich selbst, wenn die Uhr um mehr als 1000 Sekunden abweicht. Diese Option ist besonders auf Systemem ohne batteriegepufferte Echtzeituhr nützlich.

Setzen Sie `ntpd_oomprotect="YES"`, um ntpd-Daemon davor zu schützen, vom System beendet zu werden, das versucht, sich von einer Out of Memory (OOM) Situation zu retten.

Mit `ntpd_config=` setzen Sie den Pfad auf eine alternative ntp.conf-Datei.

In `ntpd_flags=` können bei Bedarf weitere Werte enthalten sein. Vermeiden Sie jedoch die Werte, die intern von /etc/rc.d/ntpd verwaltet werden:

- `-p` (Pfad zur PID-Datei)
- `-c` (Setzen Sie stattdessen `ntpd_config=`)

#### Ntpd und der nicht privilegierte `ntpd`-Benutzer

In FreeBSD kann Ntpd als nicht privilegierter Benutzer gestartet und ausgeführt werden. Dies erfordert das Modul mac_ntpd(4). Das Startskript /etc/rc.d/ntpd untersucht zunächst die NTP Konfiguration. Wenn möglich, lädt es das `mac_ntpd`-Modul und startet dann ntpd als nicht privilegierten Benutzer `ntpd` (Benutzer-ID 123). Um Probleme mit dem Datei- und Verzeichniszugriff zu vermeiden, wird das Startskript ntpd nicht automatisch als Benutzer `ntpd` starten, falls die Konfiguration irgendwelche Datei-bezogenen Optionen enthält.

Falls einer der folgenden Werte in `ntpd_flags` vorhanden ist, muss eine manuelle Konfiguration vorgenommen werden, damit der Daemon vom `ntpd`-Benutzer ausgeführt werden kann:

- -f oder --driftfile
- -i oder --jaildir
- -k oder --keyfile
- -l oder --logfile

- -s oder --statsdir

Wenn einer der folgenden Schlüsselwörter in ntp.conf vorhanden ist, muss eine manuelle Konfiguration vorgenommen werden, damit der Daemon vom `ntpd`-Benutzer ausgeführt werden kann:

- crypto
- driftfile
- key
- logdir
- statsdir

Um ntpd so zu konfigurieren, dass der Daemon als Benutzer `ntpd` läuft, müssen folgende Voraussetzungen erfüllt sein:

- Stellen Sie sicher, dass der `ntpd`-Benutzer Zugriff auf alle in der Konfiguration angegebenen Dateien und Verzeichnisse hat.
- Stellen Sie sicher, dass das Modul `mac_ntpd` in den Kernel geladen oder kompiliert wird. mac_ntpd(4) enthält weitere Details.
- Setzen Sie `ntpd_user="ntpd"` in /etc/rc.conf.

### 52.11.2. NTP mit einer PPP-Verbindung verwenden

ntpd benötigt keine ständige Internetverbindung. Wenn Sie sich über eine PPP-Verbindung ins Internet einwählen, sollten Sie verhindern, dass NTP-Verkehr eine Verbindung aufbauen oder aufrechterhalten kann. Dies kann in den `filter`-Direktiven von /etc/ppp/ppp.conf festgelegt werden. Ein Beispiel:

```
set filter dial 0 deny udp src eq 123
# Prevent NTP traffic from initiating dial out
set filter dial 1 permit 0 0
set filter alive 0 deny udp src eq 123
# Prevent incoming NTP traffic from keeping the connection open
set filter alive 1 deny udp dst eq 123
# Prevent outgoing NTP traffic from keeping the connection open
set filter alive 2 permit 0/0 0/0
```

Weitere Informationen finden Sie im Abschnitt `PACKET FILTERING` von ppp(8) sowie in den Beispielen unter /usr/shared/examples/ppp/.



Einige Internetprovider blockieren Ports mit niedrigen Nummern. In solchen Fällen funktioniert NTP leider nicht, da Antworten eines NTP-Servers den Rechner nicht erreichen werden.

# 52.12. iSCSI Initiator und Target Konfiguration

iSCSI bietet die Möglichkeit, Speicherkapazitäten über ein Netzwerk zu teilen. Im Gegensatz zu NFS, das auf Dateisystemebene arbeitet, funktioniert iSCSI auf Blockgerätebene.

In der iSCSI-Terminologie wird das System, das den Speicherplatz zur Verfügung stellt, als *Target* bezeichnet. Der Speicherplatz selbst kann aus einer physischen Festplatte bestehen, oder auch aus einem Bereich, der mehrere Festplatten, oder nur Teile einer Festplatte, repräsentiert. Wenn beispielsweise die Festplatte(n) mit ZFS formatiert ist, kann ein zvol erstellt werden, welches dann als iSCSI-Speicher verwendet werden kann.

Die Clients, die auf den iSCSI-Speicher zugreifen, werden *Initiator* genannt. Ihnen steht der verfügbare Speicher als rohe, nicht formatierte Festplatte, die auch als LUN bezeichnet wird, zur Verfügung. Die Gerätedateien für die Festplatten erscheinen in /dev/ und müssen separat formatiert und eingehangen werden.

FreeBSD enthält einen nativen, kernelbasierten iSCSI *Target* und *Initiator*. Dieser Abschnitt beschreibt, wie ein FreeBSD-System als Target oder Initiator konfiguriert wird.

## 52.12.1. Ein iSCSI-Target konfigurieren

Um ein iSCSI-Target zu konfigurieren, erstellen Sie die Konfigurationsdatei /etc/ctl.conf und fügen Sie eine Zeile in /etc/rc.conf hinzu, um sicherzustellen, dass ctld(8) automatisch beim Booten gestartet wird. Starten Sie dann den Daemon.

Das folgende Beispiel zeigt eine einfache /etc/ctl.conf. Eine vollständige Beschreibung dieser Datei und der verfügbaren Optionen finden Sie in ctl.conf(5).

```
portal-group pg0 {
    discovery-auth-group no-authentication
    listen 0.0.0.0
    listen [::]
}

target iqn.2012-06.com.example:target0 {
    auth-group no-authentication
    portal-group pg0

    lun 0 {
        path /data/target0-0
        size 4G
    }
}
```

Der erste Eintrag definiert die Portalgruppe `pg0`. Portalgruppen legen fest, auf welchen Netzwerk-Adressen der ctld(8)-Daemon Verbindungen entgegennehmen wird. Der Eintrag `discovery-auth-group no-authentication` zeigt an, dass jeder Initiator iSCSI-Targets suchen darf, ohne sich authentifizieren zu müssen. Die dritte und vierte Zeilen konfigurieren ctld(8) so, dass er auf allen

IPv4- (`listen 0.0.0.0`) und IPv6-Adressen (`listen [::]`) auf dem Standard-Port 3260 lauscht.

Es ist nicht zwingend notwendig eine Portalgruppe zu definieren, da es bereits eine integrierte Portalgruppe namens `default` gibt. In diesem Fall ist der Unterschied zwischen `default` und `pg0` der, dass bei `default` eine Authentifizierung nötig ist, während bei `pg0` die Suche nach Targets immer erlaubt ist.

Der zweite Eintrag definiert ein einzelnes Target. Ein Target hat zwei mögliche Bedeutungen: eine Maschine die iSCSI bereitstellt, oder eine Gruppe von LUNs. Dieses Beispiel verwendet die letztere Bedeutung, wobei `iqn.2012-06.com.example:target0` der Name des Targets ist. Dieser Name ist nur für Testzwecke geeignet. Für den tatsächlichen Gebrauch ändern Sie `com.example` auf einen echten, rückwärts geschriebenen Domainnamen. `2012-06` steht für das Jahr und den Monat, an dem die Domain erworben wurde. `target0` darf einen beliebigen Wert haben und in der Konfigurationsdatei darf eine beliebige Anzahl von Targets definiert werden.

Der Eintrag `auth-group no-authentication` erlaubt es allen Initiatoren sich mit dem angegebenen Target zu verbinden und `portal-group pg0` macht das Target über die Portalgruppe `pg0` erreichbar.

Die nächste Sektion definiert die LUN. Jede LUN wird dem Initiator als separate Platte präsentiert. Für jedes Target können mehrere LUNs definiert werden. Jede LUN wird über eine Nummer identifiziert, wobei LUN 0 verpflichtend ist. Die Zeile mit dem Pfad `path /data/target0-0` definiert den absoluten Pfad zu der Datei oder des zvols für die LUN. Der Pfad muss vorhanden sein, bevor ctld(8) gestartet wird. Die zweite Zeile ist optional und gibt die Größe der LUN an. Als nächstes fügen Sie folgende Zeile in /etc/rc.conf ein, um ctld(8) automatisch beim Booten zu starten:

```
ctld_enable="YES"
```

Um ctld(8) jetzt zu starten, geben Sie dieses Kommando ein:

```
# service ctld start
```

Der ctld(8)-Daemon liest beim Start /etc/ctl.conf. Wenn diese Datei nach dem Starten des Daemons bearbeitet wird, verwenden Sie folgenden Befehl, damit die Änderungen sofort wirksam werden:

```
# service ctld reload
```

#### Authentifizierung

Die vorherigen Beispiele sind grundsätzlich unsicher, da keine Authentifizierung verwendet wird und jedermann vollen Zugriff auf alle Targets hat. Um für den Zugriff auf die Targets einen Benutzernamen und ein Passwort vorauszusetzen, ändern Sie die Konfigurationsdatei wie folgt:

```
auth-group ag0 {
    chap username1 secretsecret
    chap username2 anothersecret
}
```

```
portal-group pg0 {
    discovery-auth-group no-authentication
    listen 0.0.0.0
    listen [::]
}

target iqn.2012-06.com.example:target0 {
    auth-group ag0
    portal-group pg0
    lun 0 {
        path /data/target0-0
        size 4G
    }
}
```

Die Sektion `auth-group` definiert die Benutzernamen und Passwörter. Um sich mit `iqn.2012-06.com.example:target0` zu verbinden, muss ein Initiator zuerst einen Benutzernamen und ein Passwort angeben. Eine Suche nach Targets wird jedoch immer noch ohne Authentifizierung gestattet. Um eine Authentifizierung zu erfordern, setzen Sie `discovery-auth-group` auf eine definierte `auth-group` anstelle von `no-autentication`.

In der Regel wird für jeden Initiator ein einzelnes Target exportiert. In diesem Beispiel wird der Benutzername und das Passwort direkt im Target-Eintrag festgelegt:

```
target iqn.2012-06.com.example:target0 {
    portal-group pg0
    chap username1 secretsecret

    lun 0 {
        path /data/target0-0
        size 4G
    }
}
```

### 52.12.2. Einen iSCSI-Initiator konfigurieren



Der in dieser Sektion beschriebene iSCSI-Initiator wird seit FreeBSD 10.0-RELEASE unterstützt. Lesen Sie iscontrol(8), wenn Sie den iSCSI-Initiator mit älteren Versionen benutzen möchten.

Um den Initiator zu verwenden, muss zunächst ein iSCSI-Daemon gestartet sein. Der Daemon des Initiators benötigt keine Konfigurationsdatei. Um den Daemon automatisch beim Booten zu starten, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
iscsid_enable="YES"
```

Um iscsid(8) jetzt zu starten, geben Sie dieses Kommando ein:

```
# service iscsid start
```

Die Verbindung mit einem Target kann mit, oder ohne eine Konfigurationsdatei /etc/iscsi.conf durchgeführt werden. Dieser Abschnitt beschreibt beide Möglichkeiten.

**Verbindung zu einem Target herstellen - ohne Konfigurationsdatei**

Um einen Initiator mit einem Target zu verbinden, geben Sie die IP-Adresse des Portals und den Namen des Ziels an:

```
# iscsictl -A -p 10.10.10.10 -t iqn.2012-06.com.example:target0
```

Um zu überprüfen, ob die Verbindung gelungen ist, rufen Sie `iscsictl` ohne Argumente auf. Die Ausgabe sollte in etwa wie folgt aussehen:

```
Target name                                    Target portal   State
iqn.2012-06.com.example:target0                10.10.10.10     Connected: da0
```

In diesem Beispiel wurde die iSCSI-Sitzung mit der LUN /dev/da0 erfolgreich hergestellt. Wenn das Target `iqn.2012-06.com.example:target0` mehr als nur eine LUN exportiert, werden mehrere Gerätedateien in der Ausgabe angezeigt:

```
Connected: da0 da1 da2.
```

Alle Fehler werden auf die Ausgabe und in die Systemprotokolle geschrieben. Diese Meldung deutet beispielsweise darauf hin, dass der iscsid(8)-Daemon nicht ausgeführt wird:

```
Target name                                    Target portal   State
iqn.2012-06.com.example:target0                10.10.10.10     Waiting for iscsid(8)
```

Die folgende Meldung deutet auf ein Netzwerkproblem hin, zum Beispiel eine falsche IP-Adresse oder einen falschen Port:

```
Target name                                    Target portal   State
iqn.2012-06.com.example:target0                10.10.10.11     Connection refused
```

Diese Meldung bedeutet, dass der Name des Targets falsch angegeben wurde:

```
Target name                                    Target portal   State
iqn.2012-06.com.example:target0                10.10.10.10     Not found
```

Diese Meldung bedeutet, dass das Target eine Authentifizierung erfordert:

```
Target name                                     Target portal   State
iqn.2012-06.com.example:target0                 10.10.10.10     Authentication failed
```

Verwenden Sie diese Syntax, um einen CHAP-Benutzernamen und ein Passwort anzugeben:

```
# iscsictl -A -p 10.10.10.10 -t iqn.2012-06.com.example:target0 -u user -s
secretsecret
```

**Verbindung mit einem Target herstellen - mit Konfigurationsdatei**

Wenn Sie für die Verbindung eine Konfigurationsdatei verwenden möchten, erstellen Sie /etc/iscsi.conf mit etwa folgendem Inhalt:

```
t0 {
    TargetAddress   = 10.10.10.10
    TargetName      = iqn.2012-06.com.example:target0
    AuthMethod      = CHAP
    chapIName       = user
    chapSecret      = secretsecret
}
```

`t0` gibt den Namen der Sektion in der Konfigurationsdatei an. Diser Name wird vom Initiator benutzt, um zu bestimmen, welche Konfiguration verwendet werden soll. Die anderen Einträge legen die Parameter fest, die während der Verbindung verwendet werden. `TargetAddress` und `TargetName` müssen angegeben werden, die restlichen sind optional. In diesen Beispiel wird der CHAP-Benuztername und das Passwort angegeben.

Um sich mit einem bestimmten Target zu verbinden, geben Sie dessen Namen an:

```
# iscsictl -An t0
```

Um sich stattdessen mit allen definierten Targets aus der Konfigurationsdatei zu verbinden, verwenden Sie:

```
# iscsictl -Aa
```

Damit sich der Initiator automatisch mit allen Targets aus /etc/iscsi.conf verbindet, fügen Sie folgendes in /etc/rc.conf hinzu:

```
iscsictl_enable="YES"
iscsictl_flags="-Aa"
```

# Kapitel 53. Firewalls

## 53.1. Einführung

Firewalls ermöglichen es, den ein- und ausgehenden Netzwerkverkehr eines Systems zu filtern. Dazu verwendet eine Firewall eine oder mehrere Gruppen von "Regeln", um ankommende Netzwerkpakete zu untersuchen und entweder durchzulassen oder zu blockieren. Die Regeln einer Firewall untersuchen charakteristische Eigenschaften von Datenpaketen, darunter den Protokolltyp, die Quell- und Zieladresse sowie den Quell- und Zielport.

Firewalls können die Sicherheit eines Rechners oder eines Netzwerks erhöhen, indem sie folgende Aufgaben übernehmen:

- Den Schutz der Anwendungen, Dienste und Rechner eines internen Netzwerks vor unerwünschtem Datenverkehr aus dem Internet.
- Die Beschränkung des Zugriffs von Rechnern des internen Netzwerks auf Rechner oder Dienste des öffentlichen Internets.
- Den Einsatz von Network Address Translation (NAT), welches es durch die Verwendung von privaten IP-Adressen ermöglicht, eine einzige gemeinsame Internetverbindung für mehrere Rechner zu nutzen. Dies geschieht entweder über eine einzige IP-Adresse oder über eine Gruppe von jeweils automatisch zugewiesenen öffentlichen Adressen.

Das Basissystem von FreeBSD enthält drei Firewalls: PF, IPFW und IPFILTER (auch als IPF bekannt). FreeBSD enthält ebenfalls zwei Traffic-Shaper zur Kontrolle der Bandbreite: altq(4) und dummynet(4). ALTQ ist traditionell eng an PF gebunden, während dummynet zusammen mit IPFW verwendet wird. Gemeinsam ist allen Firewalls, dass sie Regeln einsetzen, um den Transfer von ein- und ausgehenden Datenpaketen des Systems zu steuern. Unterschiedlich ist aber die Art und Weise, wie dies realisiert wird. Auch die für diese Regeln verwendete Syntax ist unterschiedlich.

FreeBSD besitzt mehrere Firewalls, um den unterschiedlichen Anforderungen und Vorlieben von Benutzern gerecht zu werden. Jeder Benutzer sollte selbst beurteilen, welche Firewall seinen Bedürfnissen am besten entspricht.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie wissen:

- Wie man Paketfilterregeln erstellt.
- Was die Unterschiede zwischen den in FreeBSD eingebauten Firewalls sind.
- Wie die PF-Firewall konfiguriert und einsetzt wird.
- Wie die IPFW-Firewall konfiguriert und einsetzt wird.
- Wie die IPFILTER-Firewall konfiguriert und einsetzt wird.

Bevor Sie dieses Kapitel lesen, sollten Sie:

- Die grundlegenden Konzepte von FreeBSD und dem Internet verstehen.



Da alle Firewalls auf der Inspektion ausgewählter Kontrollfelder in Datenpaketen

basieren, muss für die Erstellung von Firewallregeln ein grundlegendes Verständnis von TCP/IP vorhanden sein. Eine gute Einführung finden Sie in Daryl's TCP/IP Primer.

## 53.2. Firewallkonzepte

Ein Regelsatz besteht aus einer Gruppe von Regeln, die Pakete basierend auf ihren Inhalt entweder blockieren oder durchlassen. Der bidirektionale Austausch von Paketen zwischen zwei Rechnern wird als Sitzung (Session) bezeichnet. Der Regelsatz verarbeitet sowohl ankommende Pakete aus dem Internet, als auch die vom System erzeugten Antwortpakete. Jeder TCP/IP-Dienst hat ein festgelegtes Protokoll und einen vorgegebenen Port. Pakete für einen bestimmten Dienst stammen von einer Quelladresse und einem unprivilegierten Port und gehen an einen spezifischen Port auf der Zieladresse. Alle oben genannten Parameter können als Selektionskriterien verwendet werden, um einen Regelsatz zu erstellen, der den Zugriff auf bestimmte Dienste gewährt oder blockiert.

Unbekannte Portnummern können Sie in /etc/services nachschlagen. Alternativ finden Sie die Portnummern und deren Verwendungszweck auf http://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers.

Unter diesem Link finden Sie Portnummern, die auch von Trojanern benutzt werden.

FTP hat zwei Modi: Aktiv und Passiv. Unterschied liegt in der Bestimmung des Datenkanals. Der Passiv-Modus ist sicherer, da der Datenkanal vom Client bestimmt wird. Eine ausführliche Erklärung von FTP und den verschiedenen Modi finden Sie unter http://www.slacksite.com/other.ftp.html.

Ein Firewall-Regelsatz kann entweder "einschließend" (inclusive firewall) oder "ausschließend" (exclusive Firewall) sein. Eine ausschließende Firewall lässt jeden Datenverkehr durch, der nicht durch eine Regel ausgeschlossen wurde. Eine einschließende Firewall macht das genaue Gegenteil. Sie lässt Datenverkehr nur dann passieren, wenn dieser einer der definierten Regeln entspricht.

Eine einschließende Firewall bietet eine wesentlich bessere Kontrolle des ausgehenden Verkehrs, was sie zur besseren Wahl für Systeme macht, welche Dienste für das Internet anbieten. Sie kontrolliert auch den Verkehr aus dem öffentlichen Internet zum privaten Netzwerk. Jeder Verkehr, der keiner Regel entspricht wird geblockt und protokolliert. Einschließende Firewalls sind generell sicherer als ausschließende Firewalls, da sie das Risiko, dass unerwünschter Verkehr hindurch geht, drastisch reduzieren.



Wenn nicht anders vermerkt, verwenden alle Konfigurationen und Regelsätze in diesem Kapitel einschließende Firewalls.

Die Sicherheit kann durch den Einsatz einer "zustandsorientierten Firewall" (stateful firewall) weiter erhöht werden. Dieser Typ Firewall überwacht alle offenen Verbindungen und erlaubt nur Datenverkehr von bereits bestehenden Verbindungen oder wenn eine neue Verbindung aufgebaut wird.

Eine zustandsorientierte Firewall behandelt den Verkehr als einen bidirektionalen Austausch von Paketen während einer Session. Wenn ein Zustand für eine passende Regel angegeben wird, erstellt

die Firewall dynamisch interne Regeln für jedes Paket, das während dieser Session ausgetauscht wird. Die Firewall hat ausreichend Möglichkeiten, um zu bestimmen, ob ein Paket zu einer Session gehört. Alle Pakete, die nicht zu dieser Session passen, werden automatisch abgelehnt.

Sobald die Session beendet ist, wird sie aus der dynamischen Zustandstabelle entfernt.

Eine zustandsorientierte Filterung erlaubt es, sich auf die Sperrung bzw. Freigabe von neuen Sessions zu konzentrieren. Wenn eine neue Session genehmigt wird, werden alle nachfolgenden Pakete dieser Session automatisch erlaubt und betrügerische Pakete werden automatisch abgelehnt. Wenn eine neue Session nicht genehmigt wird, werden alle nachfolgenden Pakete dieser Session abgelehnt. Die zustandsorientierte Filterung bietet fortgeschrittene Fähigkeiten zur Abwehr von verschiedensten Angriffsmethoden, die von Angreifern eingesetzt werden.

NAT steht für *Network Address Translation*. Die NAT-Funktion ermöglicht es einem privaten LAN hinter einer Firewall, sich eine einzelne vom ISP zugewiesene IP-Adresse zu teilen, auch wenn die Adresse dynamisch zugewiesen wird. NAT ermöglicht den Internetzugriff für jeden Rechner im LAN, ohne dass der ISP für mehrere Internet-Konten bezahlt wird.

NAT übersetzt automatisch die private IP-Adresse auf die öffentliche IP-Adresse, sobald ein Paket für das öffentliche Internet die Firewall passiert. Zusätzlich führt es auch die Übersetzung der Anwortpakete durch.

Gemäß RFC 1918 sind die folgenden IP-Adressbereiche für private Netzwerke reserviert und werden nie ins öffentliche Internet weitergeleitet. Daher sind diese Bereiche für den Einsatz mit NAT geeignet:

- `10.0.0.0/8`
- `172.16.0.0/12`
- `192.168.0.0/16`



Seien Sie *äußerst vorsichtig* wenn Sie mit Firewallregeln arbeiten. Durch eine falsche Konfiguration kann der Administrator den Zugriff auf den Server verlieren. Um auf der sicheren Seite zu sein, sollten Sie die anfängliche Konfiguration der Firewall von der lokalen Konsole durchführen, anstatt dass Sie dies aus der Ferne über ssh tun.

## 53.3. PF

In FreeBSD 5.3 wurde PF von OpenBSD in das Basissystem integriert. Bei PF handelt es sich um eine komplette, voll ausgestattete Firewall, die optional auch ALTQ (Alternatives Queuing) unterstützt. ALTQ stellt Quality of Service (QoS) zur Verfügung.

Das OpenBSD-Projekt pflegt die maßgebliche Referenz von PF in der PF FAQ. Peter Hansteen betreut ein sehr ausführliches PF-Tutorial unter http://home.nuug.no/~peter/pf/.



Bedenken Sie beim Studium der PF FAQ, dass die PF-Version von FreeBSD im Laufe der Jahre erheblich von der Version in OpenBSD abgewichen ist. Nicht alle Eigenschaften funktionieren unter FreeBSD genauso wie unter OpenBSD und

> umgekehrt.

Die FreeBSD packet filter mailing list ist ein guter Anlaufpunkt für Fragen zur Konfiguration und dem Einsatz der PF-Firewall. Überprüfen Sie aber zunächst die Archive der Mailingliste, bevor Sie eine Frage stellen. Vielleicht wurde die Frage dort schon beantwortet.

Dieser Abschnitt konzentriert sich auf PF in FreeBSD. Es wird beschrieben, wie PF und ALTQ aktiviert werden. Zusätzlich wird demonstriert, wie Regelsätze auf einem FreeBSD-System erstellt werden.

### 53.3.1. PF aktivieren

Um PF zu benutzen, muss zunächst das Kernelmodul geladen werden. Dieser Abschnitt beschreibt die Einträge für /etc/rc.conf, die verwendet werden können um PF zu aktivieren.

Beginnen Sie damit `pf_enable=yes` in /etc/rc.conf hinzuzufügen:

```
# sysrc pf_enable=yes
```

pfctl(8) beschreibt zusätzliche Optionen, die beim Start an PF übergeben werden können. Fügen Sie diesen Eintrag in /etc/rc.conf hinzu und schreiben Sie die benötigten Optionen zwischen die Anführungszeichen:

```
pf_flags=""                   # additional flags for pfctl startup
```

PF kann nicht gestartet werden, wenn es seine Konfigurationsdatei nicht findet. In der Voreinstellung existiert unter FreeBSD kein Regelsatz namens /etc/pf.conf. Beispiel-Regelsätze finden Sie in /usr/shared/examples/pf/. Wenn bereits ein Regelsatz an anderer Stelle gespeichert wurde, fügen Sie in /etc/rc.conf einen Eintrag mit dem vollständigen Pfad zur Datei ein:

```
pf_rules="/path/to/pf.conf"
```

Protokollierungsfunktionen für PF werden von pflog(4) zur Verfügung gestellt. Fügen Sie `pflog_enable=yes` in /etc/rc.conf ein, um diese Funktion zu aktivieren:

```
# sysrc pflog_enable=yes
```

Die folgenden Zeilen können zusätzlich hinzugefügt werden, um den Speicherort der Protokolldatei zu bestimmen und weitere Optionen beim Start an pflog(4) zu übergeben:

```
pflog_logfile="/var/log/pflog"  # where pflogd should store the logfile
pflog_flags=""                  # additional flags for pflogd startup
```

Falls ein LAN hinter der Firewall existiert und die Pakete an die Rechner im LAN weitergeleitet

werden müssen, oder wenn NAT benötigt wird, aktivieren Sie die folgende Option:

```
gateway_enable="YES"            # Enable as LAN gateway
```

Nachdem die Änderungen gespeichert wurden, kann PF mit Unterstützung für Protokollierung gestartet werden:

```
# service pf start
# service pflog start
```

In der Voreinstellung liest PF seine Konfiguration aus /etc/pf.conf und modifiziert, verwirft oder akzeptiert Pakete anhand der Definitionen in dieser Datei. FreeBSD enthält mehrere Beispieldateien unter /usr/shared/examples/pf/. Auch die PF FAQ enthält sehr ausführliche Beispiele für PF-Regeln.

Zur Steuerung von PF wird `pfctl` verwendet. Nützliche `pfctl` Optionen fasst einige nützliche Optionen für diesen Befehl zusammen. Eine Beschreibung aller verfügbaren Optionen finden Sie in pfctl(8).

*Tabelle 29. Nützliche `pfctl` Optionen*

| Kommando | Aufgabe |
|---|---|
| `pfctl -e` | PF aktivieren |
| `pfctl -d` | PF deaktivieren |
| `pfctl -F all -f /etc/pf.conf` | Alle Filterregeln zurücksetzen (NAT, Filter, Zustandstabelle) und /etc/pf.conf erneut einlesen. |
| `pfctl -s [ rules \| nat \| states ]` | Zusammenfassung der Filterregeln, NAT-Regeln, oder der Zustandstabelle. |
| `pfctl -vnf /etc/pf.conf` | Überprüft /etc/pf.conf auf Fehler, lädt aber die Filterregeln nicht neu. |



security/sudo ist nützlich um Kommandos mit erhöhten Berechtigungen auszuführen, wie beispielsweise `pfctl`. Das Programm kann aus der Ports-Sammlung installiert werden.

Um den ein- und ausgehenden Verkehr im Auge zu behalten, können Sie ein Werkzeug wie sysutils/pftop benutzen. Sobald das Programm installiert ist, können Sie pftop ausführen, um einen Snapshot des Datenverkehrs zu sehen. Das Format der Ausgabe ist der von top(1) sehr ähnlich.

### 53.3.2. PF Regelsätze

Dieser Abschnitt beschreibt die Erstellung von angepassten Regelsätzen. Es wird mit dem einfachsten Regelsatz begonnen auf dem dann weitere aufgebaut werden, um die Konzepte und Funktionen von PF an einigen konkreten Beispielen zu verdeutlichen.

Der einfachste Regelsatz gilt für einen Rechner, der keine Dienste anbietet und Zugriff auf das Internet haben soll. Für diesen minimalen Regelsatz wird /etc/pf.conf wie folgt konfiguriert:

```
block in all
pass out all keep state
```

Die erste Regel blockiert jeglichen eingehenden Datenverkehr. Die zweite Regel erlaubt ausgehende Verbindungen von diesem Rechner, während die Zustandsinformationen dieser Verbindungen gespeichert werden. Diese Zustandsinformationen machen es möglich, den Antwortverkehr für diese Verbindungen zu erlauben. Der Regelsatz wird mit dem folgenden Befehl geladen:

```
# pfctl -e ; pfctl -f /etc/pf.conf
```

Neben den Zustandsinformationen verfügt PF über *Listen* und *Makros*. Diese können bei der Erstellung der Regeln definiert werden. Makros können Listen enthalten und sie müssen vor ihrer ersten Benutzung definiert sein. Fügen Sie beispielsweise folgende Zeilen an den Anfang des Regelsatzes:

```
tcp_services = "{ ssh, smtp, domain, www, pop3, auth, pop3s }"
udp_services = "{ domain }"
```

PF versteht sowohl Portnamen als auch Portnummern, solange die Namen in /etc/services aufgeführt sind. Dieses Beispiel erstellt zwei Makros. Das erste ist eine Liste mit sieben TCP-Portnamen, die zweite Liste enthält einen UDP-Portnamen. Sobald ein Makro definiert ist, kann es in den Regeln verwendet werden. In diesem Beispiel wird der gesamte Datenverkehr geblockt, mit Ausnahme der Verbindungen die von diesem Rechner initiiert wurden und sich auf einen der angegebenen TCP-Dienste oder den UDP-Dienst beziehen:

```
tcp_services = "{ ssh, smtp, domain, www, pop3, auth, pop3s }"
udp_services = "{ domain }"
block all
pass out proto tcp to any port $tcp_services keep state
pass proto udp to any port $udp_services keep state
```

Obwohl UDP als zustandsloses Protokoll betrachtet wird, ist PF in der Lage einige Zustandsinformationen zu verfolgen. Wenn beispielsweise eine UDP-Abfrage für einen Nameserver das System verlässt, wird PF nach der Antwort Ausschau halten und das Antwortpaket durch lassen.

Nachdem der Regelsatz verändert wurde, muss er neu geladen werden:

```
# pfctl -f /etc/pf.conf
```

Wenn keine Syntaxfehler festgestellt werden, wird `pfctl` keine Ausgabe erzeugen. Die Syntax kann

auch getestet werden, bevor der Regelsatz geladen wird:

```
# pfctl -nf /etc/pf.conf
```

Die Option `-n` bewirkt, dass die Regeln nur interpretiert, jedoch nicht geladen werden. Dies bietet die Möglichkeit, alle Fehler zu korrigieren. Es wird immer der letzte gültige Regelsatz geladen, bis PF entweder deaktiviert, oder ein neuer Regelsatz geladen wird.



Wenn Sie beim Laden oder Prüfen des Regelsatzes noch die Option `-v` hinzufügen, wird `pfctl` den komplett interpretierten Regelsatz anzeigen. Dies ist äußerst nützlich, wenn Sie versuchen Fehler im Regelsatz zu finden.

**Einfaches Gateway mit NAT**

Dieser Abschnitt zeigt wie ein FreeBSD-System mit PF als Gateway konfiguriert wird. Das Gateway muss über mindestens zwei Netzwerkkarten verfügen, die jeweils mit einem separaten Netzwerk verbunden sind. In diesem Beispiel ist xl0 mit dem Internet verbunden und xl1 ist mit dem internen Netzwerk verbunden.

Aktivieren Sie zunächst das Gateway, damit der Rechner den Netzwerkverkehr von einer Schnittstelle zur nächsten weiterleiten kann. Diese sysctl-Einstellung sorgt dafür, dass IPv4-Pakete weitergeleitet werden:

```
# sysctl net.inet.ip.forwarding=1
```

So leiten Sie IPv6-Datenverkehr weiter:

```
# sysctl net.inet6.ip6.forwarding=1
```

Um diese Einstellungen beim Systemstart zu aktivieren, fügen Sie sie mit Hilfe von sysrc(8) in /etc/rc.conf ein:

```
# sysrc gateway_enable=yes
# sysrc ipv6_gateway_enable=yes
```

Prüfen Sie mit `ifconfig`, dass beide Schnittstellen vorhanden und aktiv sind.

Als nächstes erstellen Sie die nötigen PF-Regeln, damit das Gateway den Datenverkehr weiterleiten kann. Die folgende Regel erlaubt den zustandsorientierten Verkehr aus dem Internet zu den Rechnern im Netzwerk:

```
pass in on xl1 from xl1:network to xl0:network port $ports keep state
```

Diese Regel erlaubt lediglich den Datenverkehr über das Gateway auf der internen Schnittstelle.

Damit die Pakete noch weiter gehen, wird eine passende Regel benötigt:

```
pass out on xl0 from xl1:network to xl0:network port $ports keep state
```

Obwohl diese beiden Regeln funktionieren, werden sie in der Praxis so spezifisch selten benötigt. Ein lesbarer Regelsatz ist oft ein sicherer Regelsatz. Der Rest dieses Abschnitts zeigt, wie Sie die Regeln so einfach und lesbar wie möglich halten. Zum Beispiel könnten die beiden Regeln zu einer Regel zusammengefasst werden:

```
pass from xl1:network to any port $ports keep state
```

Die Notation `interface:network` kann durch ein Makro ersetzt werden, um den Regelsatz besser lesbar zu machen. Zum Beispiel könnte für das Netzwerk an der internen Schnittstelle (`xl0:network`) ein Makro namens `$localnet` definiert werden. Alternativ könnte für die Definition von `$localnet` auch eine *IP-Adresse/Netzmaske* Notation verwendet werden, um ein Netzwerk zu bezeichnen, beispielsweise `192.168.100.1/24` für ein privates Subnetz.

Bei Bedarf kann für `$localnet` auch eine Liste von Netzwerken definiert werden. Abhängig von den Bedürfnissen kann `$localnet` auch für eine typische Regel wie folgt verwendet werden:

```
pass from $localnet to any port $ports keep state
```

Der folgende Regelsatz erlaubt sämtlichen Verkehr, der von den Rechnern im internen Netzwerk initiiert wird. Zunächst werden zwei Makros definiert, die die externen und internen 3COM-Schnittstellen repräsentieren.



Bei Einwählverbindungen wird tun0 für die externe Schnittstelle verwendet. Bei ADSL-Verbindungen, insbesondere denen die PPP over Ethernet (PPPoE) verwenden, ist die richtige externe Schnittstelle tun0 und nicht die physische Ethernet-Schnittstelle.

```
ext_if = "xl0"  # macro for external interface - use tun0 for PPPoE
int_if = "xl1"  # macro for internal interface
localnet = $int_if:network
# ext_if IP address could be dynamic, hence ($ext_if)
nat on $ext_if from $localnet to any -> ($ext_if)
block all
pass from { lo0, $localnet } to any keep state
```

Dieser Regelsatz führt die NAT-Regel ein, die verwendet wird, um die Übersetzung der Netzwerkadressen von den nicht-routebaren Adressen im internen Netzwerk auf die IP-Adresse der externen Schnittstelle zu handhaben. Die Klammern im letzten Teil der NAT-Regel `($ext_if)` werden angegeben, wenn die IP-Adresse der externen Schnittstelle dynamisch zugewiesen wird. Damit wird sichergestellt, dass der Netzwerkverkehr ohne schwerwiegende Unterbrechungen weiterläuft, auch wenn sich die externe IP-Adresse ändert.

Beachten Sie, dass dieser Regelsatz wahrscheinlich mehr Verkehr aus dem Netzwerk zulässt, als eigentlich nötig ist. Bei einem angemessenen Aufbau könnte folgendes Makro erstellt werden:

```
client_out = "{ ftp-data, ftp, ssh, domain, pop3, auth, nntp, http, \
    https, cvspserver, 2628, 5999, 8000, 8080 }"
```

Dieses Makro wird dann in der Filterregel benutzt:

```
pass inet proto tcp from $localnet to any port $client_out \
    flags S/SA keep state
```

Weitere `pass` Regeln werden vielleicht noch benötigt. Diese Regel aktiviert SSH auf der externen Schnittstelle:

```
pass in inet proto tcp to $ext_if port ssh
```

Dieses Makrodefinition und Regel erlaubt DNS und NTP für interne Clients:

```
udp_services = "{ domain, ntp }"
pass quick inet proto { tcp, udp } to any port $udp_services keep state
```

Beachten Sie das Schlüsselwort `quick` in dieser Regel. Da der Regelsatz aus mehreren Regeln besteht, ist es wichtig, die Beziehungen zwischen den einzelnen Regeln zu verstehen. Die Regeln werden von oben nach unten ausgewertet, in der Reihenfolge wie sie geschrieben sind. Für jedes Paket oder jede Verbindung, das PF ausgewertet, wird die letzte übereinstimmende Regel im Regelsatz angewendet. Wenn jedoch ein Paket auf eine Regel passt, welche das Schlüsselwort `quick` enthält, wird das Paket entsprechend dieser Regel behandelt und die Regelverarbeitung wird gestoppt. Diese Vorgehensweise ist sehr nützlich, wenn eine Ausnahme von den allgemeinen Regeln erforderlich ist.

#### Einen FTP-Proxy einrichten

Die Konfiguration einer funktionierenden Regel für FTP kann aufgrund der Beschaffenheit des FTP-Protokolls problematisch sein. FTP ist sehr viel älter als Firewalls und schon vom Design her unsicher. Die häufigsten Argumente gegen eine Verwendung von FTP sind:

- Passwörter werden im Klartext übertragen.
- Das Protokoll erfordert die Verwendung von mindestens zwei TCP-Verbindungen (Steuerung und Daten) auf separaten Ports.
- Wenn eine Sitzung aufgebaut wird, werden die Daten auf zufällig ausgewählten Ports übermittelt.

All diese Punkte stellen Herausforderungen dar, noch bevor die Client- oder Server-Software auf potenzielle Sicherheitslücken überprüft wurde. Es existieren aber auch sichere Alternativen für die

Dateiübertragung, wie sftp(1) oder scp(1), wo die Authentifizierung und die Datenübertragung über eine verschlüsselte Verbindung erfolgt.

Für Situationen, in denen FTP erforderlich ist, kann PF den FTP-Datenverkehr an ein kleines Proxy-Programm namens ftp-proxy(8) weiterleiten. Dieses Programm ist im Basissystem von FreeBSD enthalten. Die Aufgabe des Proxies ist das dynamische Einfügen und Entfernen von Regeln im Regelsatz. Dies wird durch den Einsatz von Ankern erreicht, damit der FTP-Verkehr korrekt verarbeitet werden kann.

Fügen Sie folgende Zeilen in /etc/rc.conf ein, um den Proxy zu aktivieren:

```
ftpproxy_enable="YES"
```

Danach kann der Proxy mit `service ftp-proxy start` gestartet werden.

Für die Grundkonfiguration müssen drei weitere Einträge in /etc/pf.conf hinzugefügt werden. Zunächst werden die Anker hinzugefügt, die der Proxy für die FTP-Sitzungen verwendet:

```
nat-anchor "ftp-proxy/*"
rdr-anchor "ftp-proxy/*"
```

Dann wird eine `pass`-Regel benötigt, damit der FTP-Datenverkehr durch den Proxy geleitet werden kann.

Die Regeln für Umleitung und NAT müssen vor den eigentlichen Filterregeln definiert werden. Fügen Sie diese `rdr`-Regel unmittelbar nach der NAT-Regel ein:

```
rdr pass on $int_if proto tcp from any to any port ftp -> 127.0.0.1 port 8021
```

Zum Schluss muss der umgeleitete Verkehr die Firewall passieren dürfen:

```
pass out proto tcp from $proxy to any port ftp
```

`$poxy` enthält die Adresse, an dem der Proxy-Daemon gebunden ist.

Speichern Sie /etc/pf.conf und laden Sie die Regeln neu. Prüfen Sie von einem Client, ob die FTP-Verbindungen funktionieren:

```
# pfctl -f /etc/pf.conf
```

Dieses Beispiel umfasst eine Grundkonfiguration, in der die Rechner im lokalen Netzwerk Zugriff auf entfernte FTP-Server benötigen. Diese Konfiguration sollte mit den meisten FTP-Clients und -Servern gut funktionieren. Das Verhalten von ftp-proxy(8) kann durch diverse Optionen in `ftpproxy_flags` beeinflusst werden. Einige Clients und Server haben bestimmte Marotten, die bei der Konfiguration berücksichtigt werden müssen. Es kann zum Beispiel notwendig sein, den FTP-

Datenverkehr für den Proxy einer bestimmten Warteschlange zuzuweisen.

Es besteht auch die Möglichkeit einen FTP-Server mit PF und ftp-proxy(8) zu schützen. Konfigurieren Sie einen separaten `ftp-proxy` mit `-R` für den Reverse-Modus auf einem separaten Port und einer eigenen Umleitungsregel.

#### ICMP verwalten

Viele Werkzeuge zur Fehlerbehebung in TCP/IP-Netzwerken verlassen sich auf das Internet Control Message Protocol (ICMP), das speziell für diese Zwecke entwickelt wurde.

Das ICMP-Protokoll sendet und empfängt Kontrollnachrichten zwischen Rechnern und Gateways, hauptsächlich um ungewöhnliche Bedingungen auf dem Weg zum Zielrechner zu berichten. Router verwenden ICMP um Paketgrößen und andere Übertragungsparameter zu ermitteln. Dieser Prozess ist auch als *Path MTU Discovery* bekannt.

Aus der Sicht einer Firewall sind einige ICMP-Kontrollnachrichten anfällig für bekannte Angriffsmethoden. Zwar ist die Fehlerbehebung einfacher, wenn alle ICMP-Pakete bedingungslos durch gelassen werden, aber dass macht es auch für Angreifer leichter, Informationen über das Netzwerk zu extrahieren. Aus diesen Gründen ist die folgende Regel nicht optimal:

```
pass inet proto icmp from any to any
```

Eine Lösung besteht darin, nur den ICMP-Verkehr aus dem lokalen Netz zu akzeptieren, während ICMP-Pakete von außerhalb des Netzwerks verworfen werden:

```
pass inet proto icmp from $localnet to any keep state
pass inet proto icmp from any to $ext_if keep state
```

Es stehen noch weitere Optionen zur Verfügung, die die Flexibilität von PF demonstrieren. Anstatt beispielsweise alle ICMP-Nachrichten zu erlauben, kann man die Nachrichten angeben, die von ping(8) und traceroute(8) verwendet werden. Beginnen Sie damit, ein Makro für diese Art von Nachrichten zu definieren:

```
icmp_types = "echoreq"
```

Erstellen Sie dann eine Regel, die das eben erstellte Makro benutzt:

```
pass inet proto icmp all icmp-type $icmp_types keep state
```

Wenn weitere Arten von ICMP-Nachrichten benötigt werden, kann die Liste `icmp_types` einfach erweitert werden. Geben Sie `more /usr/src/sbin/pfctl/pfctl_parser.c` ein, um eine Liste der von PF unterstützten ICMP-Nachrichten zu sehen. Die Webseite http://www.iana.org/assignments/icmp-parameters/icmp-parameters.xhtml enthält eine Erklärung für jeden Nachrichtentyp.

Da UNIX® `traceroute` in der Voreinstellung UDP verwendet, wird eine weitere Regel benötigt:

```
# allow out the default range for traceroute(8):
pass out on $ext_if inet proto udp from any to any port 33433 >< 33626 keep state
```

Da `TRACERT.EXE` unter Microsoft® Windows®-Systemen ICMP Echo Request Meldungen verwendet, ist nur die erste Regel notwendig um Traces für solche Systeme zu ermöglichen. UNIX® `traceroute` kann aber auch andere Protokolle verwenden, zum Beispiel ICMP Echo Request, wenn der Schalter `-I` benutzt wird. Details finden Sie in traceroute(8).

#### Path MTU Discovery

Internet-Protokolle sind so ausgelegt, dass sie geräteunabhängig sind. Eine Folge davon ist, dass die optimale Paketgröße nicht immer zuverlässig vorhergesagt werden kann. Das größte Hindernis ist hier die *Maximum Transmission Unit* (`MTU`), welche die Obergrenze für die Paketgröße festlegt. Die MTU für die Schnittstelle des Systems können Sie sich mit `ifconfig` anzeigen lassen.

TCP/IP benutzt ein Verfahren, das als path MTU discovery bekannt ist, um die korrekte Paketgröße für eine Verbindung zu bestimmen. Dieses Verfahren sendet Pakete unterschiedlicher Größe mit dem Flag "do not fragment" und erwartet ein ICMP-Antwortpaket vom Typ "type 3, code 4", wenn die Obergrenze erreicht worden ist. Typ 3 bedeutet "Ziel nicht erreichbar" und Code 4 ist die Abkürzung für "Fragmentierung nötig, aber Do-not-Fragment Flag ist gesetzt". Um path MTU discovery zu erlauben und damit Verbindungen zu anderen MTUs zu unterstützen, fügen Sie dem Makro `icmp_types` den Typ `destination unreachable` hinzu:

```
icmp_types = "{ echoreq, unreach }"
```

Da die `pass`-Regel bereits das Makro verwendet, braucht es nicht geändert werden um den neuen ICMP-Typ zu unterstützen:

```
pass inet proto icmp all icmp-type $icmp_types keep state
```

PF kann alle Variationen von ICMP-Typen und Codes filtern. Eine Liste der verfügbaren Typen und Codes ist in icmp(4) und icmp6(4) dokumentiert.

### Tabellen benutzen

Manchmal sind bestimmte Daten für die Filterung und Weiterleitung interessant, jedoch wäre eine Definition einer solchen Filterregel für einen Regelsatz viel zu lang. PF unterstützt die Verwendung von Tabellen. Dies sind definierte Listen, die verändert werden können, ohne den gesamten Regelsatz neu laden zu müssen. Zudem können diese Listen sehr schnell durchsucht werden. Tabellennamen sind immer in `< >` eingeschlossen und sehen wie folgt aus:

```
table <clients> { 192.168.2.0/24, !192.168.2.5 }
```

In diesem Beispiel ist das Netzwerk `192.168.2.0/24` Teil der Tabelle. `192.168.2.5` wurde im dem Operator `!` ausgeschlossen und ist somit nicht Teil der Tabelle. Es ist auch möglich Tabellen aus

Dateien zu laden, wo jeder Eintrag in einer separaten Zeile steht. Dieses Beispiel verwendet dazu die Datei /etc/clients:

```
192.168.2.0/24
!192.168.2.5
```

Um sich auf diese Datei zu beziehen, definieren Sie die Tabelle wie folgt:

```
table <clients> persist file "/etc/clients"
```

Sobald die Tabelle definiert ist, kann eine Filterregel Bezug darauf nehmen:

```
pass inet proto tcp from <clients> to any port $client_out flags S/SA keep state
```

Die Inhalte einer Tabelle können mit `pfctl` direkt verändert werden. Dieses Beispiel fügt ein weiteres Netzwerk zur Tabelle hinzu:

```
# pfctl -t clients -T add 192.168.1.0/16
```

Beachten Sie, dass auf diese Weise vorgenommene Änderungen direkt übernommen werden, jedoch bei einem Neustart des Systems oder bei einem Stromausfall verloren gehen. Um die Änderungen dauerhaft zu speichern, müssen sie in der Definition der Tabelle oder in der Datei, auf die sich die Tabelle bezieht, bearbeitet werden. Mit einem cron(8) Job und einem Befehl wie `pfctl -t clients -T show >/etc/clients` können Sie auch eine Kopie der Tabelle auf Platte speichern und dann in regelmäßigen Abständen aktualisieren. Alternativ kann /etc/clients auch mit den Tabelleneinträgen, die sich aktuell im Speicher befinden, aktualisiert werden.

```
# pfctl -t clients -T replace -f /etc/clients
```

**Verwendung von Tabellen zum Schutz von SSH**

Benutzer, die SSH auf einer externen Schnittstelle ausführen, haben wahrscheinlich schon einmal ähnliche Meldungen in den Protokolldateien gesehen:

```
Sep 26 03:12:34 skapet sshd[25771]: Failed password for root from 200.72.41.31 port
40992 ssh2
Sep 26 03:12:34 skapet sshd[5279]: Failed password for root from 200.72.41.31 port
40992 ssh2
Sep 26 03:12:35 skapet sshd[5279]: Received disconnect from 200.72.41.31: 11: Bye Bye
Sep 26 03:12:44 skapet sshd[29635]: Invalid user admin from 200.72.41.31
Sep 26 03:12:44 skapet sshd[24703]: input_userauth_request: invalid user admin
Sep 26 03:12:44 skapet sshd[24703]: Failed password for invalid user admin from
200.72.41.31 port 41484 ssh2
```

Diese Meldungen deuten auf einen Brute-Force-Angriff hin, bei dem ein Angreifer oder ein Programm versucht, den Benutzernamen und das Passwort zu erraten, um Zugriff auf das System zu bekommen.

Wenn der Zugriff über SSH für berechtigte Benutzer erforderlich ist, kann eine Änderung des Standard-Ports für SSH einen gewissen Schutz bieten. Allerdings bietet PF eine elegantere Lösung für dieses Problem. `pass`-Regeln können Einschränkungen für Dinge enthalten, die ein verbindender Rechner tun kann. Bei einem Verstoß gegen diese Einschränkungen kann dann dem betroffenen Rechner der Zugriff teilweise oder ganz entzogen werden. Es ist sogar möglich, alle bestehenden Verbindungen zu trennen, falls die Grenze überschritten wird.

Um dies zu konfigurieren, erstellen Sie folgende Tabelle im Regelsatz:

```
table <bruteforce> persist
```

Fügen Sie dann ziemlich am Anfang der Filterregeln folgende Regeln hinzu, um die Brute-Force-Angriffe zu blocken und gleichzeitig berechtigte Verbindungen zu erlauben:

```
block quick from <bruteforce>
pass inet proto tcp from any to $localnet port $tcp_services \
    flags S/SA keep state \
    (max-src-conn 100, max-src-conn-rate 15/5, \
    overload <bruteforce> flush global)
```

Der Teil in Klammern definiert die Grenzwerte. Die Zahlen sollten an die lokalen Anforderungen angepasst werden. Die Zeilen können wie folgt interpretiert werden:

`max-src-conn` definiert die maximal erlaubte Anzahl gleichzeitiger Verbindungen von einem Rechner.

`max-src-conn-rate` definiert die maximal erlaubte Anzahl neuer Verbindungen eines einzelnen Rechners (*15*) pro Anzahl von Sekunden (*5*).

`overload <bruteforce>` bedeutet, dass jeder Rechner, der diesen Grenzwert überschreitet, zur Tabelle `bruteforce` hinzugefügt wird. Diese Filterregel blockiert jeglichen Datenverkehr von Adressen aus der Tabelle `bruteforce`.

`flush global` besagt, dass alle (`global`) Verbindungen dieses Rechners getrennt (`flush`) werden, wenn der Grenzwert erreicht wird.



Diese Filterregeln helfen nicht bei langsamen Brute-Force-Angriffen, wie sie in http://home.nuug.no/~peter/hailmary2013/ beschrieben sind.

Dieser Beispielregelsatz dient lediglich als Illustration. Wenn Sie allgemein eine große Anzahl an Verbindungen erlauben wollen, aber gleichzeitig bei SSH etwas restriktiver vorgehen möchten, können Sie die obige Regel ergänzen:

```
pass quick proto { tcp, udp } from any to any port ssh \
    flags S/SA keep state \
    (max-src-conn 15, max-src-conn-rate 5/3, \
    overload <bruteforce> flush global)
```



*Es ist möglicherweise nicht notwendig, alle aggressiven Rechner zu blockieren*

Es ist zu erwähnen, dass der `overlaod`-Mechanismus eine allgemeine Technik darstellt, die nicht auf SSH beschränkt ist. Außerdem ist es nicht immer optimal, Datenverkehr von aggressiven Rechnern zu blockieren.

Eine `overload`-Regel kann beispielsweise benutzt werden, um einen Mail- oder Webserver zu schützen. Die `overload`-Tabelle könnte dann in einer Regel verwendet werden, um aggressive Rechner einer Warteschlange mit geringerer Bandbreite zuzuweisen, oder den Rechner auf eine bestimtme Webseite umzuleiten.

Im Laufe der Zeit werden die Tabellen durch die `overload`-Regeln immer größer und belegen immer mehr Speicher. Manchmal wird eine geblockte IP-Adresse einem Rechner dynamisch zugewiesen, der eigentlich berechtigt ist, mit den Rechnern im lokalen Netzwerk zu kommunizieren.

Für solche Situationen bietet `pfctl` die Möglichkeit, Tabelleneinträge auslaufen zu lassen. Dieses Kommando würde beispielsweise Einträge aus der Tabelle `<bruteforce>` löschen, die seit `86400` Sekunden nicht mehr referenziert wurden:

```
# pfctl -t bruteforce -T expire 86400
```

Eine ähnliche Funktionalität bietet security/expiretable, welches Einträge entfernt, die für einen bestimmten Zeitraum nicht referenziert wurden.

Nach der Installation kann expiretable benutzt werden, um Einträge aus der Tabelle `<bruteforce>` nach einer bestimmten Zeit zu entfernen. Dieses Beispiel entfernt alle Einträge, die älter sind als 24 Stunden:

```
/usr/local/sbin/expiretable -v -d -t 24h bruteforce
```

**Schutz vor SPAM**

Im Gegensatz zum spamd-Daemon von spamassassin, kann mail/spamd zusammen mit PF den SPAM direkt an der Firewall abwehren. Dieser spamd wird in PF über einen Satz von Umleitungen konfiguriert.

Spammer neigen dazu, eine große Anzahl von Nachrichten zu versenden. Dabei nutzten Sie SPAM-freundliche Netzwerke und gekaperte Rechner, welche dann ziemlich schnell bei sogenannten Blacklists gemeldet werden.

Wenn eine SMTP-Verbindung von einer Adresse in der Blacklist empfangen wird, präsentiert

spamd einen Banner und schaltet sofort in einen Modus, in dem die Antworten auf den SMTP-Verkehr jeweils ein Byte groß sind. Diese Technik, die möglichst viel Zeit des Spammers verschwenden soll, wird Tarpitting genannt. Die spezifische Implementierung, welche ein Byte SMTP-Antworten verwendet, wird als Stuttering bezeichnet.

Dieses Beispiel zeigt das grundlegende Verfahren zur Konfiguration von spamd mit automatisch aktualisierten Blacklists. Für weitere Informationen lesen die Manualpages, die zusammen mit mail/spamd installiert werden.

**Procedure: Konfiguration von spamd**

1. Installieren Sie das Paket oder den Port mail/spamd. Um spamd's Greylisting-Funktion zu nutzen, muss fdescfs(5) in /dev/fd eingehängt werden. Fügen Sie folgende Zeile in /etc/fstab ein:

   ```
   fdescfs /dev/fd fdescfs rw 0 0
   ```

   Danach hängen Sie das Dateisystem ein:

   ```
   # mount fdescfs
   ```

2. Fügen Sie folgende Zeilen in den PF-Regelsatz ein:

   ```
   table <spamd> persist
   table <spamd-white> persist
   rdr pass on $ext_if inet proto tcp from <spamd> to \
       { $ext_if, $localnet } port smtp -> 127.0.0.1 port 8025
   rdr pass on $ext_if inet proto tcp from !<spamd-white> to \
       { $ext_if, $localnet } port smtp -> 127.0.0.1 port 8025
   ```

   Die beiden Tabellen `<spamd>` und `<spam-white>` sind von großer Bedeutung. SMTP-Verkehr von einer Adresse, die in `<spamd>` aber nicht in `<spamd-white>` ist, wird an den spamd-Daemon auf Port 8025 umgeleitet.

3. Im nächsten Schritt wird spamd in /usr/local/etc/spamd.conf konfiguriert und einige Parameter werden in /etc/rc.conf hinzugefügt.

   Die Installation von mail/spamd enthält eine Beispielkonfiguration (/usr/local/etc/spamd.conf.sample) und eine Manualpage für spamd.conf. Beziehen Sie sich für zusätzliche Konfigurationsoptionen auf diese Dokumentation.

   Die Konfigurationsdatei enthält einen Block, in dem die `all`-Liste definiert ist, die wiederum weitere Listen spezifiziert:

   ```
   all:\
   ```

```
    :traplist:whitelist:
```

Dieser Eintrag fügt die gewünschten Blacklists, getrennt durch einen Doppelpunkt (`:`), hinzu. Um auch eine Whitelist zu verwenden, fügen Sie den Namen unmittelbar hinter dem Namen der Blacklist ein. Zum Beispiel: `:Blacklist:Whitelist:`.

Danach folgt die Definition der verwendeten Blacklist:

```
traplist:\
    :black:\
    :msg="SPAM. Your address %A has sent spam within the last 24 hours":\
    :method=http:\
    :file=www.openbsd.org/spamd/traplist.gz
```

In der ersten Zeile steht der Name der Blacklist und die zweite Zeile gibt den Typ an. Das Feld `msg` enthält die Nachricht, die dem Absender während des SMTP-Dialogs angezeigt wird. Das Feld `mehtod` legt fest, wie spamd-setup die Listen bezieht; unterstützte Methoden sind `http`, `ftp`, `file` und ein externes Programm via `exec`. Im letzten Feld gibt `file` den Namen der Datei an, die spamd erwartet.

Die Definition der Whitelist ist ähnlich. Das Feld `msg` wird jedoch nicht definiert, da eine Meldung hier nicht erforderlich ist:

```
whitelist:\
    :white:\
    :method=file:\
    :file=/var/mail/whitelist.txt
```



*Wählen Sie die Datenquellen mit Sorgfalt*

Bei der Verwendung von sämtlichen Blacklists aus der Beispieldatei spamd.conf würden große Teile des Internets geblockt. Der Administrator muss diese Datei bearbeiten, um eine optimale Konfiguration zu erzielen. Dazu gehört auch die Auswahl von geeigneten Blacklists und, wenn nötig, die Erstellung von benutzerdefinierten Listen.

Als nächstes fügen Sie folgenden Eintrag in /etc/rc.conf hinzu. Zusätzliche Optionen sind in der Manualpage beschrieben:

```
spamd_flags="-v" # use "" and see spamd-setup(8) for flags
```

Wenn Sie fertig sind, starten Sie spamd durch die Eingabe von `service obspamd start`. Führen Sie die weitere Konfiguration mit `spamd-setup` durch. Erstellen Sie zum Schluss einen cron(8) -Job, der `spamd-setup` in regelmäßigen Abständen aufruft, um die Listen zu aktualisieren.

Auf einem typischen Gateway vor dem Mailserver, werden Rechner innerhalb von wenigen

Minuten geblockt.

PF unterstützt auch *Greylisting*, das Nachrichten von unbekannten Rechnern vorübergehend mit *45n*-Codes ablehnt. Nachrichten von diesen Rechnern werden bei einem erneuten Versuch nach einer angemessenen Zeit durchgelassen. Nachrichten von Rechnern, die nach RFC 1123 und RFC 2821 konfiguriert sind, werden sofort durchgelassen.

Weitere Informationen über Greylisting finden Sie unter greylisting.org. Das Erstaunlichste an Greylisting ist, neben der einfachen Benutzung, dass es immer noch funktioniert. Spammer und Malware-Autoren gelingt es bislang nur schwer, diese Technik zu umgehen.

Die grundsätzliche Vorgehensweise zur Konfiguration von Greylisting ist wie folgt:

**Procedure: Konfiguration von Greylisting**

1. Stellen Sie sicher, dass fdescfs(5) eingehängt ist. Dies wird in Schritt 1 der vorherigen Prozedur beschrieben.
2. Um spamd im Greylisting-Modus auszuführen, fügen Sie folgende Zeilen in /etc/rc.conf ein:

   ```
   spamd_grey="YES"  # use spamd greylisting if YES
   ```

   Lesen Sie die Manualpage von spamd für Beschreibungen von zusätzlichen Parametern.
3. Starten Sie die Dienste, um die Konfiguration von Greylisting abzuschließen:

   ```
   # service obspamd restart
   # service spamlogd start
   ```

Hinter den Kulissen führen die spamdb-Datenbank und spamlogd wesentliche Aufgaben der Greylisting-Funktion aus. spamdb ist die Schnittstelle für den Administrator, der über den Inhalt der Datenbank /var/db/spamdb Blaklists, Whitelists und Greylists verwaltet.

### Netzwerk-Hygiene

Dieser Abschnitt beschreibt die Verwendung von `block-policy`, `scrub` und `antispoof`, mit denen das Verhalten des Regelsatzes weiter optimiert werden kann.

Die Option `block-policy` kann im Teil `options` des Regelwerks konfiguriert werden, vor den Umleitungen und den eigentlichen Filterregeln. Diese Option legt fest, welche Rückmeldung PF an einen geblockten Rechner sendet. Es existieren zwei mögliche Werte: `drop` verwirft das Paket ohne Rückmeldung und `return` gibt eine Statusmeldung, wie etwa `Connection refused` zurück.

Die Voreinstellung ist `drop`. Geben Sie den gewünschten Wert ein, um die `block-policy`-Richtlinie zu ändern:

```
set block-policy return
```

`scrub` ist ein Schlüsselwort in PF, das die Paket-Normalisierung aktiviert. Dieser Prozess fügt fragmentierte Pakete wieder zusammen und blockt TCP-Pakete mit ungültigen Flag-Kombinationen. Ein aktiviertes `scrub` bietet einen gewissen Schutz gegen Angriffe, die auf die falsche Handhabung von fragmentierten Paketen aufbauen. Es stehen viele Optionen zur Verfügung, jedoch sollte die einfachste Form für die meisten Konfigurationen ausreichend sein:

```
scrub in all
```

Einige Dienste, wie beispielsweise NFS, erfordern eine bestimmte Handhabung von fragmentierten Paketen. Weitere Informationen finden Sie unter https://home.nuug.no/~peter/pf/en/scrub.html.

Dieses Beispiel fügt fragmentierte Pakete wieder zusammen, löscht das "do not fragment"-Bit und setzt die maximale Segmentgröße auf 1440 Bytes:

```
scrub in all fragment reassemble no-df max-mss 1440
```

Der `antispoof`-Mechanismus bietet einen Schutz gegen gefälschte IP-Adressen. Dabei werden hauptsächlich Pakete verworfen, die auf der falschen Schnittstellen ankommen.

Folgende Regeln verwerfen gefälschte Adressen, wenn sie aus dem Internet oder dem lokalen Netzwerk stammen:

```
antispoof for $ext_if
antispoof for $int_if
```

**Handhabung von nicht-routebaren Adressen**

Sogar bei einem richtig konfigurierten NAT-Gateway müssen Sie vielleicht die Fehlkonfiguration anderer Personen ausgleichen. Ein typischer Fehler besteht darin, nicht-routebare Adressen ins Internet zu lassen. Da der Verkehr von nicht-routebaren Adressen Teil eines DoS-Angriffs sein kann, sollten Sie in Betracht ziehen, diesen Verkehr explizit an der externen Schnittstelle des Netzwerks zu blockieren.

In diesem Beispiel wird ein Makro erstellt, das die nicht-routebaren Adressen enthält. Datenverkehr von und zu diesen Adressen wird dann an der externen Schnittstelle des Gateways verworfen.

```
martians = "{ 127.0.0.0/8, 192.168.0.0/16, 172.16.0.0/12, \
          10.0.0.0/8, 169.254.0.0/16, 192.0.2.0/24, \
          0.0.0.0/8, 240.0.0.0/4 }"

block drop in quick on $ext_if from $martians to any
```

```
block drop out quick on $ext_if from any to $martians
```

### 53.3.3. ALTQ aktivieren

Unter FreeBSD kann ALTQ zusammen mit PF benutzt werden, um Quality of Service (QoS) bereitzustellen. Sobald ALTQ aktiviert ist, können Warteschlangen definiert werden, mit denen Sie die Priorität für ausgehende Pakete festlegen können.

Bevor Sie ALTQ aktivieren, sollten Sie altq(4) lesen und sicherstellen, das der Treiber der Netzwerkkarte diese Funktion unterstützt.

ALTQ steht nicht als ladbares Kernelmodul zur Verfügung. Wenn die Netzwerkkarte des Systems ALTQ unterstützt, erstellen Sie nach den Anweisungen in Konfiguration des FreeBSD-Kernels einen angepassten Kernel. Als erstes muss ALTQ aktiviert werden. Zudem ist mindestens eine weitere Option nötig, um den Algorithmus für die Warteschlange zu bestimmen:

```
options         ALTQ
options         ALTQ_CBQ        # Class Based Queuing (CBQ)
options         ALTQ_RED        # Random Early Detection (RED)
options         ALTQ_RIO        # RED In/Out
options         ALTQ_HFSC       # Hierarchical Packet Schedule (HFSC)
options         ALTQ_PRIQ       # Priority Queuing (PRIQ)
```

Die folgenden Algorithmen stehen zur Verfügung:

**CBQ**

Class Based Queuing (CBQ) erlaubt es, die Bandbreite einer Verbindung in verschiedene Klassen oder Warteschlangen zu unterteilen, um die Priorität von Datenpaketen basierend auf Filterregeln zu beeinflussen.

**RED**

Random Early Detection (RED) wird eingesetzt, um eine Überlastung des Netzwerks zu vermeiden. Dazu ermittelt RED die Größe der Warteschlange und vergleicht diesen Wert mit den minimalen und maximalen Grenzwerten der Warteschlange. Ist die Warteschlange größer als das erlaubte Maximum, werden alle neuen Pakete nach dem Zufallsprinzip verworfen.

**RIO**

Random Early Detection In and Out (RIO). Dieser Modus verwaltet mehrere Warteschlangen durchschnittlicher Größe mit mehreren Schwellwerten, eine für jedes QoS-Level.

**HFSC**

Hierachical Fair Service Curve Packet Scheduler (HFSC) wird in http://www-2.cs.cmu.edu/~hzhang/HFSC/main.html beschrieben.

**PRIQ**

Priority Queuing (PRIQ) lässt den Verkehr einer Warteschlange mit höherer Priorität zuerst durch.

Weitere Informationen über diese Algorithmen und Beispiele für Regelsätze finden Sie in den OpenBSD Archiven.

## 53.4. IPFW

IPFW ist eine Stateful-Firewall für FreeBSD, die sowohl IPv4 als auch IPv6 unterstützt. Die Firewall setzt sich aus mehreren Komponenten zusammen: dem Kernel Firewall Filter-Prozessor mit integriertem Paket-Accounting, Protokollfunktionen, NAT, dem dummynet(4) Traffic-Shaper, sowie Weiterleitungs-, Bridge- und ipstealth-Funktionen.

FreeBSD enthält mit /etc/rc.firewall ein Beispielregelwerk, welches mehrere Firewall-Typen für gebräuchliche Szenarien definiert und unerfahrene Anwender dabei unterstützen soll, ein geeignetes Regelwerk zu erstellen. IPFW besitzt eine leistungsstarke Syntax, mit der erfahrene Benutzer ihre eigenen Regeln anfertigen können, um den Sicherheitsanforderungen der jeweiligen Umgebung gerecht zu werden.

Diser Abschnitt beschreibt, wie IPFW aktiviert wird und bietet einen Überblick über die Regelsyntax. Zudem werden mehrere Regelsätze für gebräuchliche Konfigurationsszenarien vorgestellt.

### 53.4.1. IPFW aktivieren

Das FreeBSD Basissystem enthält für IPFW ein ladbares Kernelmodul, was bedeutet, dass kein angepasster Kernel benötigt wird, um IPFW zu benutzen.

Wenn Sie eine statische Unterstützung für IPFW in den Kernel kompilieren wollen, lesen Sie IPFW Kerneloptionen.

Um IPFW beim Systemstart zu aktivieren, fügen Sie `firewall_enable="YES"` in /etc/rc.conf ein:

```
# sysrc firewall_enable="YES"
```

Wenn Sie einen der von FreeBSD zur Verfügung gestellten Firewall-Profile benutzen möchten, fügen Sie eine weitere Zeile hinzu, in der Sie das Profil bestimmen:

```
# sysrc firewall_type="open"
```

Folgende Profile stehen zur Verfügung:

- `open`: gestattet jeglichen Datenverkehr.
- `client`: schützt lediglich diesen Rechner.
- `simple`: schützt das gesamte Netzwerk.
- `closed`: blockiert den gesamten IP-Datenverkehr, mit Ausnahme des Verkehrs über die Loopback-Schnittstelle.
- `workstation`: schützt lediglich diesen Rechner und verwendet zustandsorientierte Regeln.

- `UNKNOWN`: deaktiviert das Laden von Firewallregeln.
- filename: absoluter Pfad zu einer Datei, in der die Firewallregeln definiert sind.

Wenn Sie `firewall_type` auf `client` oder `simple` setzen, müssen Sie die voreingestellten Regeln in /etc/rc.firewall anpassen, damit sie der Konfiguration des Systems entsprechen.

Beachten Sie, dass das Profil `filename` verwendet wird, um ein benutzerdefiniertes Regelwerk zu laden.

Eine alternative Möglichkeit, um ein benutzerdefiniertes Regelwerk zu laden, bietet die Variable `firewall_script`. Setzen Sie die Variable auf den absoluten Pfad eines *ausführbaren Skripts*, welches die Befehle für IPFW enthält. Die Beispiele in diesem Abschnitt gehen davon aus, dass `firewall_script` auf /etc/ipfw.rules gesetzt ist.

```
# sysrc firewall_script="/etc/ipfw.rules"
```

Die Protokollierung wird mit diesem Befehl aktiviert:

```
# sysrc firewall_logging="YES"
```



Es werden nur Firewallregeln mit der Option `log` protokolliert. Die voreingestellten Regeln enthalten diese Option nicht und müssen manuell hinzugefügt werden. Daher ist es ratsam, diese Regeln zu bearbeiten. Außerdemkann eine Rotation der Protokolle erwünscht sein, wenn die Protokolle in einer separaten Datei gespeichert werden.

Es existiert keine Variable für /etc/rc.conf, um die Protokollierung zu begrenzen. Um die Anzahl der Protokoll-Nachrichten pro Verbindungsversuch zu begrenzen, legen Sie die Anzahl der Einträge in /etc/sysctl.conf fest:

```
# echo "net.inet.ip.fw.verbose_limit=5" >> /etc/sysctl.conf
```

Um die Protokollierung über die spezielle Schnittstelle `ipfw0` zu aktivieren, fügen Sie stattdessen folgende Zeile in /etc/rc.conf hinzu:

```
# sysrc firewall_logif="YES"
```

Benutzen Sie dann tcpdump, um zu sehen, was protokolliert wird:

```
# tcpdump -t -n -i ipfw0
```



Durch die Protokollierung entsteht kein Aufwand, es sei denn, tcpdump wird an die Schnittstelle angebunden.

Nachdem Sie die Änderungen vorgenommen haben, können Sie die Firewall starten. Um auch die Anzahl der Protokoll-Nachrichten zu konfigurieren, setzen Sie mit `sysctl` den gewünschten Wert:

```
# service firewall start
# sysctl net.inet.ip.fw.verbose_limit=5
```

### 53.4.2. IPFW Regel-Syntax

Wenn ein Paket die Firewall "betritt", also von der Firewall geprüft und verarbeitet wird, wird die erste Regel des Regelwerkes auf das Paket angewandt. Auf diese Weise wird in aufsteigender Reihenfolge der Regelnummer mit allen weiteren Regeln verfahren. Falls die Selektionsparameter einer Regel auf ein Paket zutreffen, wird das Aktionsfeld der Regel ausgeführt und die Prüfung des Pakets beendet, nachfolgende Regeln werden also nicht mehr geprüft. Diese Suchmethode wird als "erster Treffer gewinnt" bezeichnet. Falls keine Regel auf das betreffende Paket zutrifft, wird die obligatorische IPFW-Rückfallregel mit der Nummer 65535 angewendet und das Paket wird ohne Rückantwort verworfen. Wenn das Paket jedoch einer Regel mit dem Schlüsselwort `count`, `skipto` oder `tee` entspricht, wird die Prüfung des Pakets weiter fortgeführt. Weitere Details darüber, wie diese Schlüsselwörter die Regelverarbeitung beeinflussen, finden Sie in ipfw(8).

Bei der Erstellung der IPFW-Regeln müssen die Schlüsselwörter in der folgenden Reihenfolge geschrieben werden. Einige Schlüsselwörter müssen zwingend angegeben werden, während andere optional sind. Die Wörter in Großbuchstaben repräsentieren Variablen und die Wörter in Kleinbuchstaben müssen den Variablen vorangestellt werden. Das Zeichen `#` wird benutzt, um einen Kommentar einzuleiten und kann am Ende einer Regel oder in einer eigenen Zeile stehen. Leerzeilen werden ignoriert.

```
CMD RULE_NUMBER set SET_NUMBER ACTION log LOG_AMOUNT PROTO from SRC SRC_PORT to DST DST_PORT OPTIONS
```

Dieser Abschnitt bietet einen Überblick über diese Schlüsselwörter und deren Optionen. Es ist keine vollständige Liste aller verfügbaren Optionen. Eine vollständige Beschreibung der Regel-Syntax, die Sie verwenden können um IPFW-Regeln zu erstellen, finden Sie in ipfw(8).

**CMD**

Jede Regel muss mit ipfw add beginnen.

**RULE_NUMBER**

Jede Regel gehört zu einer Nummer zwischen `1` und `65534`. Die Nummer wird verwendet, um die Reihenfolge der Regelverarbeitung zu kennzeichnen. Es ist möglich, dass mehrere Regeln dieselbe Nummer haben. In diesem Fall werden sie entsprechend der Reihenfolge angewendet, in der sie aufgenommen wurden.

**SET_NUMBER**

Jede Regel ist einer *Set*-Nummer zwischen `0` und `31` zugeordnet. Sets können einzeln aktiviert oder deaktiviert werden. Dies macht es möglich, eine Reihe von Regeln schnell hinzuzufügen oder zu löschen. Wenn `SET_NUMBER` nicht angegeben ist, wird die Regel zu Set `0` hinzugefügt.

**ACTION**

Eine Regel kann mit einer der folgenden Aktionen verknüpft werden. Die festgelegte Aktion wird ausgeführt, wenn das Paket den Selektionskriterien der Regel entspricht.

allow | accept | pass | permit: All diese Aktionen sind gleichbedeutend und erlauben Pakete, die mit der Regel übereinstimmen.

check-state: Diese Aktion überprüft die Regel in der dynamischen Zustandstabelle. Bei einer Übereinstimmung wird die mit der dynamischen Regel verknüpfte Aktion ausgeführt, andernfalls wird mit der Prüfung gegen die nächste Regel fortgefahren. Die Regel `check-state` hat selbst kein Selektionskriterium. Sollte keine `check-state`-Regel im Regelwerk vorhanden sein, wird die dynamische Zustandstabelle beim ersten Vorkommen einer `keep-state`- oder `limit` -Regel überprüft.

count: Aktualisiert die Zähler für alle Pakete, die mit dieser Regel übereinstimmen. Die Prüfung wird mit der nächsten Regel fortgesetzt.

deny | drop: Diese Aktionen sind gleichbedeutend und verwerfen Pakete, die mit dieser Regel übereinstimmen.

Es stehen noch weitere Aktionen zur Verfügung. Einzelheiten finden Sie in ipfw(8).

**LOG_AMOUNT**

Erfüllt ein Paket die Selektionskriterien mit dem Schlüsselwort `log`, wird dies von syslogd(8) mit der Annotation `SECURITY` protokolliert. Dies erfolgt allerdings nur, wenn die Anzahl der protokollierten Pakete der betreffenden Regel die definierte `LOG_AMOUNT`-Grenze nicht übersteigt. Wenn `LOG_AMOUNT` nicht definiert ist, wird die Grenze aus dem Wert von `net.inet.ip.fw.verbose_limit` benutzt. Ein Wert von `0` bedeutet eine unbegrenzte Protokollierung. Wird eine definierte Grenze erreicht, wird die Protokollierung für diese Regel deaktiviert. Um die Protokollierung zu reaktivieren, können Sie den Protokoll- oder Paketzähler mit `ipfw resetlog` zurücksetzen.



Die Protokollierung findet statt, nachdem alle Selektionskriterien geprüft und bevor die endgültige Aktion auf das Paket angewendet wird. Der Administrator entscheidet, welche Regel protokolliert werden soll.

**PROTO**

Dieser optionale Wert wird verwendet, um einen beliebigen Protokollnamen oder -nummer aus /etc/protocols gegen das Paket zu prüfen.

**SRC**

Nach dem Schlüsslwort `from` muss die Quelladresse stehen, oder ein Schlüsselwort, das die Quelladresse darstellt. Eine Adresse wird dargestellt duch `any`, `me` (jede Adresse dieses Systems), `me6` (jede IPv6-Adresse dieses Systems), oder `table` gefolgt von der Nummer der Tabelle, welche die Adressen enthält. IP-Adressen können in CIDR-Notation geschrieben werden. Beispielsweise `1.2.3.4/25` oder `1.2.3.4:255.255.255.128`.

**SRC_PORT**

Optional kann ein Quellport über eine Nummer oder einen Namen aus /etc/services spezifiziert werden.

**DST**

Nach dem Schlüsselwort `to` muss die Zieladresse stehen, oder ein Schlüsselwort, das die Zieladresse darstellt. Es können die gleichen Schlüsselwörter und Adressen benutzt werden, die bereits im SRC-Abschnitt beschrieben wurden.

**DST_PORT**

Optional kann ein Zielport über eine Nummer oder einen Namen aus /etc/services spezifiziert werden.

**OPTIONS**

Nach der Quell- und Zieladresse können noch weitere Optionen angegeben werden. Wie der Name bereits sagt, sind `OPTIONS` optional. Häufig verwendete Optionen sind `in` oder `out`, mit denen die Richtug des Pakets bestimmt wird, `icmptypes` gefolgt vom Typ der ICMP-Nachricht, sowie `keep-state`.

Wenn ein Paket auf eine keep-state-Regel zutrifft, wird die Firewall eine dynamische Regel erstellen, die dem bidirektionalen Datenverkehr zwischen den gleichen Quell- und Zieladressen mit dem gleichen Protokoll entspricht.

Dynamische Regeln sind für einen sogenannten SYN-flood-Angriff anfällig, bei dem eine riesige Anzahl an dynamischen Regeln erzeugt wird. Verwenden Sie die Option `limit`, um einen solchen Angriff entgegenzuwirken. Diese Option begrenzt die Anzahl der gleichzeitig möglichen Sitzungen. Es handelt sich dabei um einen Zähler, der die Anzahl von dynamischen Regeln in Kombination mit der Quelladresse verfolgt. Übersteigt der Zähler den durch `limit` definierten Wert, wird das Paket verworfen.

Es stehen noch viele weitere Optionen zur Verfügung. ipfw(8) enthält eine Beschreibung der einzelnen Optionen.

### 53.4.3. Beispiel für einen Regelsatz

Dieser Abschnitt die Erstellung eines Firewall-Skripts namens /etc/ipfw.rules mit zustandsorientierten (stateful Regeln. Alle Regeln in diesem Beispiel verwenden die Optionen `in` und `out`, um die Richtung des Pakets zu verdeutlichen. Zusätzlich wird `via` *interface-name* benutzt, um die Schnittstelle für das Paket zu prüfen.



Bei den anfänglichen Tests mit dem Firewall-Regelsatz sollten Sie vielleicht folgende Einstellung vornehmen:

```
net.inet.ip.fw.default_to_accept="1"
```

Dies legt die Standardregel von ipfw(8) etwas großzügiger fest, als das voreingestellte `default deny ip from any to any`. Dadurch sinkt die Gefahr, sich

> nach einem Neustart des Systems auszusperren.

Das Firewall-Skript beginnt mit einem Hinweis, dass es sich um ein Bourne Shell-Skript handelt. Danach werden alle vorhandenen Filterregeln gelöscht. Anschließend wird die Variable `cmd` erstellt, sodass `ipfw add` nicht jedes mal von Hand eingegeben werden muss. Die Variable `pif` repräsentiert die mit dem Internet verbundene Schnittstelle.

```
#!/bin/sh
# Flush out the list before we begin.
ipfw -q -f flush

# Set rules command prefix
cmd="ipfw -q add"
pif="dc0"     # interface name of NIC attached to Internet
```

Jetzt folgen die eigentlichen Filterregeln. Diese ersten beiden Regeln erlauben den Datenverkehr aus dem internen Netzwerk und über die Loopback-Schnittstelle:

```
# Change xl0 to LAN NIC interface name
$cmd 00005 allow all from any to any via xl0

# No restrictions on Loopback Interface
$cmd 00010 allow all from any to any via lo0
```

Die nächste Regel erlaubt Pakete, für die ein Eintrag in der dynamischen Zustandstabelle existiert:

```
$cmd 00101 check-state
```

Die nächsten Regeln definieren, welche internen Rechner Verbindungen zu anderen Rechnern im Internet aufbauen dürfen. Hier werden wieder zustandsorientierte Regeln verwendet:

```
# Allow access to public DNS
# Replace x.x.x.x with the IP address of a public DNS server
# and repeat for each DNS server in /etc/resolv.conf
$cmd 00110 allow tcp from any to x.x.x.x 53 out via $pif setup keep-state
$cmd 00111 allow udp from any to x.x.x.x 53 out via $pif keep-state

# Allow access to ISP's DHCP server for cable/DSL configurations.
# Use the first rule and check log for IP address.
# Then, uncomment the second rule, input the IP address, and delete the first rule
$cmd 00120 allow log udp from any to any 67 out via $pif keep-state
#$cmd 00120 allow udp from any to x.x.x.x 67 out via $pif keep-state

# Allow outbound HTTP and HTTPS connections
$cmd 00200 allow tcp from any to any 80 out via $pif setup keep-state
$cmd 00220 allow tcp from any to any 443 out via $pif setup keep-state
```

```
# Allow outbound email connections
$cmd 00230 allow tcp from any to any 25 out via $pif setup keep-state
$cmd 00231 allow tcp from any to any 110 out via $pif setup keep-state

# Allow outbound ping
$cmd 00250 allow icmp from any to any out via $pif keep-state

# Allow outbound NTP
$cmd 00260 allow udp from any to any 123 out via $pif keep-state

# Allow outbound SSH
$cmd 00280 allow tcp from any to any 22 out via $pif setup keep-state

# deny and log all other outbound connections
$cmd 00299 deny log all from any to any out via $pif
```

Die folgenden Regeln steuern die Verbindungen von Rechern aus dem Internet ins interne Netzwerk. Zuerst werden Pakete verworfen, die typischerweise im Zusammenhang mit Angriffen stehen. Danach werden bestimmte Arten von Verbindungen erlaubt. Alle Dienste aus dem öffentlichen Internet beinhalten die Option `limit`, um Flooding zu unterbinden.

```
# Deny all inbound traffic from non-routable reserved address spaces
$cmd 00300 deny all from 192.168.0.0/16 to any in via $pif     #RFC 1918 private IP
$cmd 00301 deny all from 172.16.0.0/12 to any in via $pif      #RFC 1918 private IP
$cmd 00302 deny all from 10.0.0.0/8 to any in via $pif         #RFC 1918 private IP
$cmd 00303 deny all from 127.0.0.0/8 to any in via $pif        #loopback
$cmd 00304 deny all from 0.0.0.0/8 to any in via $pif          #loopback
$cmd 00305 deny all from 169.254.0.0/16 to any in via $pif     #DHCP auto-config
$cmd 00306 deny all from 192.0.2.0/24 to any in via $pif       #reserved for docs
$cmd 00307 deny all from 204.152.64.0/23 to any in via $pif    #Sun cluster
interconnect
$cmd 00308 deny all from 224.0.0.0/3 to any in via $pif        #Class D & E multicast

# Deny public pings$
$cmd 00310 deny icmp from any to any in via $pif$
$
# Deny ident$
$cmd 00315 deny tcp from any to any 113 in via $pif$
$
# Deny all Netbios services.$
$cmd 00320 deny tcp from any to any 137 in via $pif$
$cmd 00321 deny tcp from any to any 138 in via $pif$
$cmd 00322 deny tcp from any to any 139 in via $pif$
$cmd 00323 deny tcp from any to any 81 in via $pif$

# Deny fragments
$cmd 00330 deny all from any to any frag in via $pif

# Deny ACK packets that did not match the dynamic rule table
```

```
$cmd 00332 deny tcp from any to any established in via $pif

# Allow traffic from ISP's DHCP server.
# Replace x.x.x.x with the same IP address used in rule 00120.
#$cmd 00360 allow udp from any to x.x.x.x 67 in via $pif keep-state

# Allow HTTP connections to internal web server
$cmd 00400 allow tcp from any to me 80 in via $pif setup limit src-addr 2

# Allow inbound SSH connections
$cmd 00410 allow tcp from any to me 22 in via $pif setup limit src-addr 2

# Reject and log all other incoming connections
$cmd 00499 deny log all from any to any in via $pif
```

Die letzte Regel protokolliert alle Pakete, die mit keiner Regel im Regelsatz übereinstimmen:

```
# Everything else is denied and logged
$cmd 00999 deny log all from any to any
```

### 53.4.4. In-Kernel NAT

Die IPFW-Firewall von FreeBSD hat zwei NAT-Implementierungen: die Userland-Implementierung natd(8) und die neuere, kernelinterne NAT-Implementierung. Beide arbeiten in Verbindung mit IPFW, um die Übersetzung von Netzwerkadressen zu ermöglichen. Damit kann eine Lösung zur gemeinsamen Nutzung der Internetverbindung bereitgestellt werden, so dass mehrere interne Rechner unter Verwendung einer einzigen öffentlichen IP-Adresse eine Verbindung zum Internet herstellen können.

Um dies zu tun, muss der mit dem Internet verbundene FreeBSD-Rechner als Gateway eingerichtet sein. Das System muss über zwei Netzwerkschnittstellen verfügen, wobei eine Schnittstelle mit dem Internet verbunden ist und die andere mit dem internen Netzwerk. Jeder Rechner im internen Netzwerk sollte eine RFC 1918 konforme Adresse zugewiesen bekommen.

Es ist noch ein wenig Konfiguration nötig, um die In-Kernel NAT-Funktion von IPFW zu aktivieren. Um die In-Kernel NAT-Unterstützung beim Booten zu aktivieren, müssen folgende Einträge in /etc/rc.conf vorhanden sein:

```
gateway_enable="YES"
firewall_enable="YES"
firewall_nat_enable="YES"
```



Wenn `firewall_nat_enable` gesetzt ist, `firewall_enable` jedoch nicht, hat dies keine Auswirkung, da die NAT-Implementierung im Kernel nur mit IPFW kompatibel ist.

Wenn der Regelsatz zustandsorientierte Regeln enthält, ist die Position der NAT-Regel kritisch und die `skipto`-Aktion wird benutzt. Die Aktion `skipto` benötigt eine Regelnummer, damit IPFW weiß, zu

welcher Regel es springen muss. Das folgende Beispiel baut auf den im vorherigen Abschnitt gezeigten Firewall-Relgelsatz auf. Es werden einige neue Einträge hinzugefügt und bestehende Regeln modifiziert, um In-Kernel NAT zu konfigurieren. Zunächst werden einige Variablen hinzugefügt, darunter Regelnummern, die `keep-state`-Option und eine Liste mit TCP-Ports um die Anzahl der Regeln zu reduzieren:

```
#!/bin/sh
ipfw -q -f flush
cmd="ipfw -q add"
skip="skipto 1000"
pif=dc0
ks="keep-state"
good_tcpo="22,25,37,53,80,443,110"
```

Bei In-Kernel NAT muss aufgrund der Architektur von libalias(3), einer Bibliothek, die als Kernel-Modul implementiert ist, um die In-Kernel NAT-Funktion für IPFW bereitzustellen, TCP segment offloading (TSO) deaktiviert werden. TSO kann pro Netzwerkschnittstelle mit ifconfig(8), oder systemweit mit sysctl(8) deaktiviert werden. Um TSO systemweit zu deaktivieren, muss folgende Zeile in /etc/sysctl.conf enthalten sein:

```
net.inet.tcp.tso="0"
```

Danach wird eine NAT-Instanz konfiguriert. Mit In-Kernel NAT ist es möglich, mehrere NAT-Instanzen mit jeweils eigener Konfiguration zu betreiben. In diesem Beispiel wird jedoch nur eine NAT-Instanz mit der Nummer 1 benötigt. Die Konfiguration kann ein paar Optionen enthalten, zum Beispiel: `if`, dass die öffentliche Netzwerkschnittstelle angibt, `same_ports`, das dafür sorgt, dass Alias-Ports und lokale Portnummern identisch zugeordnet werden, `unreg_only` führt dazu, dass nur unregistrierte (private) Adressräume von der NAT-Instanz verarbeitet werden, und `reset`, was dazu beiträgt, dass eine NAT-Instanz auch dann erhalten bleibt, wenn sich die öffentliche IP-Adresse des Rechners ändert. Weitere mögliche Optionen, die an einzelne NAT-Instanzen übergeben werden können, finden Sie in ipfw(8). Wenn eine zustandsorientierte NAT-Firewall konfiguriert wird, ist es notwendig, dass übersetzte Pakete zur weiteren Verarbeitung in die Firewall eingespielt werden können, was durch die Deaktivierung des `one_pass`-Verhaltens beim Start des Firewall-Skripts erreicht werden kann.

```
ipfw disable one_pass
ipfw -q nat 1 config if $pif same_ports unreg_only reset
```

Die NAT-Regel für eingehende Pakete wird *nach* den beiden Regeln, die das interne Netzwerk und die Loopback-Schnittstelle erlauben, und nach der Reassamble-Regel, aber *vor* der `check-state` -Regel eingefügt. Es ist wichtig, dass die Nummer der NAT-Regel (in diesem Beispiel `100`) höher ist, als die drei vorherigen Regeln und niedriger, als die `check-state`-Regel. Darüber hinaus wird aufgrund des Verhaltens von In-Kernel NAT empfohlen, eine Reassamble-Regel kurz vor der ersten NAT-Regel, aber hinter den Regeln zu platzieren, die den Datenverkehr auf einer vertrauenswürdigen Schnittstelle erlauben. In der Regel sollte es nicht zu einer Fragmentierung

kommen, aber bei getunnelten IPSEC/ESP/GRE-Verkehr kann es vorkommen, und das Zusammensetzen von Fragmenten ist notwendig, bevor das komplette Paket an das In-Kernel NAT übergeben werden kann.



Die Reassamble-Regel wird beim Userland natd(8) nicht benötigt, da die Aktion `divert` von IPFW dies bereits automatisch übernimmt, bevor das Paket an den Socket ausgeliefert wird. Dies ist auch in ipfw(8) dokumentiert.

Beachten Sie, dass die aktuelle NAT-Instanznummer und NAT-Regelnummer in diesem Beispiel nicht mit der voreingestellten NAT-Instanznummer und Regelnummer übereinstimmt, wenn sie mit dem rc.firewall-Skript von FreeBSD erstellt wurde.

```
$cmd 005 allow all from any to any via xl0  # exclude LAN traffic
$cmd 010 allow all from any to any via lo0  # exclude loopback traffic
$cmd 099 reass all from any to any in       # reassemble inbound packets
$cmd 100 nat 1 ip from any to any in via $pif # NAT any inbound packets
# Allow the packet through if it has an existing entry in the dynamic rules table
$cmd 101 check-state
```

Die Regeln für den ausgehenden Verkehr werden ebenfalls modifiziert, um Aktionen mit der `$skipto`-Variable zu erlauben und anzuzeigen, dass die Prüfung mit der Regel `1000` fortgesetzt wird. Die sieben Regeln für TCP wurden durch die Regel `125` ersetzt, da die sieben erlaubten ausgehenden Ports in der Variable `$good_tcp0` enthalten sind.



Beachten Sie, dass die Leistung von IPFW weitgehend von der Anzahl der im Regelsatz vorhandenen Regeln bestimmt wird.

```
# Authorized outbound packets
$cmd 120 $skip udp from any to x.x.x.x 53 out via $pif $ks
$cmd 121 $skip udp from any to x.x.x.x 67 out via $pif $ks
$cmd 125 $skip tcp from any to any $good_tcpo out via $pif setup $ks
$cmd 130 $skip icmp from any to any out via $pif $ks
```

Die eingehenden Regeln bleiben unverändert, mit Ausnahme der letzten Regel, in der das `via $pif` entfernt wird, um ein- und ausgehende Pakete prüfen zu können. Nach der letzten Regel für ausgehende Pakete muss die NAT-Regel folgen. Die Regel muss eine höhere Nummer als die letzte Regel haben und die Nummer muss über die `skipto`-Aktion referenziert werden. In diesem Regelsatz leitet die Regel mit der Nummer `1000` alle ausgehenden Pakete zur konfigurierten NAT-Instanz weiter. Die darauf folgende Regel lässt alle von NAT verarbeiteten Pakete passieren.

```
$cmd 999 deny log all from any to any
$cmd 1000 nat 1 ip from any to any out via $pif # skipto location for outbound
stateful rules
$cmd 1001 allow ip from any to any
```

In diesem Beispiel steuern die Regeln `100`, `101`, `125`, `1000` und `1001` die Adressübersetzung der ein- und ausgehende Pakete, so dass immer die private LANIP-Adresse in der dynamische Zustandstabelle registriert werden.

Nehmen wir beispielsweise einen Web-Browser, der neue HTTP-Sitzungen über Port 80 aufbaut. Wenn nun das erste ausgehende Paket von der Firewall geprüft wird, trifft es nicht auf Regel `100` zu, da das Paket nach außen geleitet wird und nicht nach innen. Das Paket trifft auch nicht auf Regel `101` zu, da es das erste ist und somit noch nicht in der dynamischen Zustandstabelle enthalten ist. Das Paket entspricht schließlich Regel `125`, da es ausgehend auf einem erlaubten Port gesendet wird und von einer IP-Adresse aus dem internen LAN stammt. Für Pakete, die auf diese Regel zutreffen, werden zwei Aktionen ausgeführt. Zuerst wird durch die Aktion `keep-state` ein dynamischer Eintrag in der Statustabelle erstellt und die angegebene Aktion `skipto 1000` ausgeführt. Als nächstes durchläuft das Paket NAT und wird dann an das Internet gesendet. Nachdem dieses Paket am Webserver angekommen ist, wird dort eine Antwort erzeugt und zurückgeschickt. Dieses Paket wird wieder von oben nach unten durch das Regelwerk geprüft. Dieses Mal trifft Regel `100` auf das Paket zu und die Zieladresse wird auf die zugehörige (lokale) LAN-Adresse abgebildet. Danach wird das Paket von der Regel `check-state` verarbeitet. Die Zustandstabelle erkennt, dass eine zugehörige aktive Sitzung vorliegt und das Paket wird freigegeben und in das LAN geleitet.

Für den eingehenden Datenverkehr muss der Regelsatz unerwünschte Pakete blockieren und Pakete für autorisierte Dienste durchlassen. Ein Paket, das mit einer Regel für den eingehenden Datenverkehr übereinstimmt, wird in der dynamischen Zustandstabelle eingetragen und dann an das LAN freigegeben. Das Antwortpaket wird von der Regel `check-state` als Paket einer aktiven Sitzung erkannt. Das Paket wird dann von Regel `1000` per NAT verarbeitet, bevor es über die externe Schnittstelle verschickt wird.



Der Wechsel vom Userland natd(8) zu In-Kernel NAT mag zunächst nahtlos erscheinen, aber es gibt einen kleinen Haken. Bei Verwendung des GENERIC -Kernels wird IPFW das Kernelmodul libalias.ko laden, wenn `firewall_nat_enable` in rc.conf aktiviert ist. Das Kernelmodul libalias.ko stellt nur grundlegende NAT-Funktionalität bereit, während die Userland-Implementierung natd(8) alle Funktionalitäten ohne zusätzliche Konfiguration zur Verfügung stellt. Die gesamte Funktionalität bezieht sich auf die folgenden Kernelmodule, die bei Bedarf zusätzlich zu libalias.ko geladen werden können: alias_cuseeme.ko, alias_ftp.ko, alias_bbt.ko, skinny.ko, irc.ko, alias_pptp.ko und alias_smedia.ko unter Verwendung der `kld_list` Direktive in rc.conf. Wenn ein angepasster Kernel benutzt wird, kann die volle Funktionalität der Userland-Bibliothek im Kernel mit `options LIBALIAS` gebaut werden.

### Weiterleitung von Ports

Der Nachteil von NAT ist, dass die Rechner im LAN nicht aus dem Internet zugänglich sind. Diese Rechner können zwar ausgehende Verbindungen zur Außenwelt aufbauen, jedoch keine eingehenden Verbindungen empfangen. Dies stellt ein Problem dar, wenn Sie auf einem Rechner im LAN Dienste anbieten möchten, die aus dem Internet erreichbar sein sollen. In diesem Fall können Sie die Ports, welche über das Internet erreichbar sein sollen, über die NAT-Maschine an den Rechner im LAN weiterleiten.

Angenommen es gibt einen IRC-Server auf Rechner A und einen Webserver auf Rechner B. Damit dies funktioniert, müssen die Verbindungen auf den Ports 6667 (IRC) und 80 (HTTP) an die jeweiligen Rechner weitergeleitet werden.

Bei In-Kernel NAT wird die gesamte Konfiguration in der NAT-Instanz selbst vorgenommen. Alle Optionen, die in einer NAT-Instanz benutzt werden können, sind in ipfw(8) dokumentiert. Die Syntax für IPFW folgt dabei der von natd. Die Syntax für `-redirect_port` lautet:

```
redirect_port proto targetIP:targetPORT[-targetPORT]
  [aliasIP:]aliasPORT[-aliasPORT]
  [remoteIP[:remotePORT[-remotePORT]]]
```

Für das obige Beispiel sollten die Argumente wie folgt aussehen:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_port tcp 192.168.0.3:80 80
```

Nachdem diese Argumente der Konfiguration der NAT-Instanz 1 im obigen Regelsatz hinzugefügt wurden, werden die TCP-Ports an die Rechner im LAN weitergeleitet, auf denen IRC- und HTTP-Dienste laufen.

```
ipfw -q nat 1 config if $pif same_ports unreg_only reset \
  redirect_port tcp 192.168.0.2:6667 6667 \
  redirect_port tcp 192.168.0.3:80 80
```

Portbereiche können über `redirect_port` festgelegt werden. Zum Beispiel würde *tcp 192.168.0.2:2000-3000 2000-3000* alle Verbindungen auf die Ports 2000 bis 3000 an die Ports 2000 bis 3000 an Rechner A weiterleiten.

#### Weiterleiten von Adressen

Das Weiterleiten von Adressen ist nützlich, wenn mehr als eine IP-Adresse zur Verfügung steht. Jeder Rechner im LAN kann über ipfw(8) seine eigene externe IP-Adresse zugewiesen bekommen. IPFW wird dann den ausgehenden Datenverkehr der Rechner aus dem LAN mit der entsprechenden externen IP-Adresse umschreiben. Auch der eingehenden Datenverkehr über die externe IP-Adresse wird an die entsprechenden Rechner im LAN weitergeleitet. Diese Methode ist auch als statisches NAT bekannt. Wenn Ihnen beispielsweise die IP-Adressen 128.1.1.1, 128.1.1.2 und 128.1.1.3 zur Verfügung stehen, kann 128.1.1.1 als externe Adresse der ipfw(8)-Maschine verwendet werden, während 128.1.1.2 und 128.1.1.3 an Rechner A und Rechner B im LAN weitergeleitet werden.

Die Syntax für `redirect_address` lautet wie im Folgenden, wobei `localIP` die interne IP-Adresse des Rechners im LAN, und `publicIP` die externe IP-Adresse ist, die dem Rechner im LAN entspricht.

```
redirect_address localIP publicIP
```

Auf das Beispiel bezogen, würden die Argumente so lauten:

```
redirect_address 192.168.0.2 128.1.1.2
redirect_address 192.168.0.3 128.1.1.3
```

Genau wie bei `redirect_port`, werden diese Argumente in der Konfiguration der NAT-Instanz gesetzt. Bei der Weiterleitung von Adressen ist keine Portumleitung notwendig, da alle Daten, die auf einer bestimmten IP-Adresse empfangen werden, weitergeleitet werden.

Die externe IP-Adresse der ipfw(8)-Maschine muss auf der externen Schnittstelle aktiv und mit einem Alias versehen sein. Weitere Einzelheiten sind in rc.conf(5); beschrieben.

**Userland NAT**

Zunächst sei gesagt, dass natd(8), die Userland-Implementierung aufwändiger ist als In-Kernel NAT. Damit natd(8) Pakete übersetzen kann, müssen die Pakete vom Kernel ins Userland und zurück kopiert werden, was zusätzlichen Aufwand mit sich bringt. Dieser Aufwand entfällt bei In-Kernel NAT.

Um den Userland NAT-Daemon natd(8) beim Systemstart zu aktivieren, ist etwas Konfiguration in /etc/rc.conf nötig. `natd_interface` wird auf den Namen der mit dem Internet verbundenen Schnittstelle gesetzt. Das rc(8)-Skript von natd(8) wird selbstständig prüfen, ob eine dynamische IP-Adresse benutzt wird und sich selbst so konfigurieren, dass es damit umgehen kann.

```
gateway_enable="YES"
natd_enable="YES"
natd_interface="rl0"
```

Generell kann der obige Regelsatz, wie er für In-Kernel NAT erklärt wurde, auch zusammen mit natd(8) benutzt werden. Die Ausnahmen sind die Konfiguration der In-Kernel NAT-Instanz (`ipfw -q nat 1 config …`), die nicht zusammen mit der Regel 99 benötigt wird, da die `divert`-Aktion sich um die Fragmentierung kümmert. Die Regeln 100 und 1000 müssen leicht modifiziert werden, wie unten gezeigt.

```
$cmd 100 divert natd ip from any to any in via $pif
$cmd 1000 divert natd ip from any to any out via $pif
```

Um eine Port- oder Adressumleitung zu konfigurieren, wird eine ähnliche Syntax wie bei In-Kernel NAT verwendet. Anstatt die Konfiguration in unserem Regelsatz-Skript wie bei In-Kernel NAT anzugeben, wird die Konfiguration von natd(8) am besten in einer Konfigurationsdatei vorgenommen. Dazu muss eine zusätzliche Option in /etc/rc.conf übergeben werden, welche den Pfad zur Konfigurationsdatei angibt.

```
natd_flags="-f /etc/natd.conf"
```



Die Konfigurationsdatei muss eine Liste von Optionen enthalten, eine pro Zeile. Weitere Informationen über die Konfigurationsdatei und mögliche Variablen finden Sie in natd(8). Hier zwei Beispieleinträge, einer pro Zeile:

```
redirect_port tcp 192.168.0.2:6667 6667
redirect_address 192.168.0.3 128.1.1.3
```

### 53.4.5. Das IPFW Kommando

`ipfw` kann benutzt werden, um einzelne Regeln im laufenden Betrieb hinzuzufügen oder zu entfernen. Problematisch ist jedoch, dass diese Änderungen bei einem Neustart des Systems verloren gehen. Daher ist es empfehlenswert, eigene Regeln in einer Datei zu definieren und diese zu laden, um die Regeln der Firewall im laufenden Betrieb anzupassen.

`ipfw` ist auch hilfreich, um die geladenen Regeln der auf der Konsole auszugeben. IPFW erzeugt dynamisch einen Zähler, der jedes Paket, auf das eine Regel zutrifft, zählt. Dadurch ist es möglich, die Funktion einer Regel zu überprüfen.

Eine Auflistung aller geladenen Regeln erhalten Sie mit:

```
# ipfw list
```

Eine Auflistung aller Regeln inklusive des letzten Treffers erhalten Sie mit:

```
# ipfw -t list
```

Das nächste Beispiel zeigt Informationen über die Anzahl der Pakete, die von einer Regel gefiltert wurden sowie die Regel selbst. Der erste Spalte zeigt die Nummer der Regel, gefolgt von der Anzahl der gefilterten Pakete und der Anzahl der Pakete in Bytes. Zum Schluss steht die Regel selbst:

```
# ipfw -a list
```

Das folgende Kommando zeigt zusätzlich alle dynamischen Regeln an:

```
# ipfw -d list
```

Um diese Auflistung um die "abgelaufenen" Regeln zu erweitern, geben Sie folgendes Kommando ein:

```
# ipfw -d -e list
```

Hiermit werden alle Zähler auf Null zurückgesetzt:

```
# ipfw zero
```

Es ist auch möglich, einen spezifischen Zähler zurückzusetzen:

```
# ipfw zero NUM
```

#### Protokollierung von Firewall-Nachrichten

Auch bei aktivierter Protokollierung wird IPFW von selbst keine Regeln protokollieren. Der Administrator muss entscheiden, welche Regeln aus dem Regelwerk protokolliert werden sollen. In diesen Regeln muss dann das Schlüsselwort `log` hinzugefügt werden. Normalerweise werden nur geblockte Pakete protokolliert. Es ist üblich, die "ipfw default deny everything"-Regel am Ende des Regelwerks mit dem Schlüsselwort `log` zu duplizieren. Dadurch ist es möglich, alle Pakete zu sehen, auf die keine Regel zutraf.

Protokollierung ist allerdings ein zweischneidiges Schwert. Bei mangelnder Vorsicht oder einem DoS-Angriff wird die Festplatte mit einer enormen Flut von Protokolldaten belastet. Protokoll-Nachrichten werden nicht nur an syslogd(8) geschickt, sondern auch auf der Konsole angezeigt, was dann schnell lästig werden kann.

Die Kerneloption `IPFIREWALL_VERBOSE_LIMIT=5` begrenzt die Anzahl identischer Nachrichten an syslogd(8) für eine gegebene Regel auf fünf Nachrichten. Ist diese Option im Kernel aktiviert, wird nach Erreichen den festgelegten Anzahl die Protokollierung von aufeinanderfolgenden Nachrichten auf den festgelegten Wert begrenzt, da beispielsweise die Speicherung von 200 gleichen Protokoll-Nachrichten sinnlos ist. Daher werden durch diese Option nur fünf gleichartige Nachrichten protokolliert. Alle weiteren Nachrichten werden nur gezählt und deren Gesamtzahl wird schließlich von syslogd(8) wie folgt ausgegeben:

```
Last message repeated 45 times
```

Alle protokollierten Pakete werden in der Voreinstellung in /var/log/security gespeichert. Dies wird in /etc/syslog.conf definiert.

#### Ein Firewall-Regelwerk erstellen

Die meisten fortgeschrittenen IPFW-Benutzer erzeugen eine Datei, welche die Regeln für die Firewall enthält, um diese als Skript ausführen zu können. Der Vorteil einer derartigen Konfiguration besteht darin, dass dadurch mehrere Regeln gleichzeitig geändert und aktiviert werden können, ohne dass dazu das System neu gestartet werden muss. Dies ist zudem beim Testen von Regeländerungen sehr hilfreich. Weil es sich bei der Datei um ein Skript handelt, ist es auch möglich, häufig verwendete Befehle durch Aliase zu ersetzen und diese dann in mehreren Regeln zu nutzen.

Die Syntax des folgenden Skripts entspricht der Syntax von sh(1), csh(1) sowie tcsh(1). Felder, die symbolisch substituiert werden, haben das Präfix $ (Dollarzeichen). Symbolische Felder haben das $-Präfix nicht. Der Wert, mit dem das symbolische Feld belegt wird, muss in doppelten

Anführungszeichen ("") stehen.

Die Datei mit den Regeln könnte wie folgt aufgebaut sein:

```
############### start of example ipfw rules script #############
#
ipfw -q -f flush       # Delete all rules
# Set defaults
oif="tun0"             # out interface
odns="192.0.2.11"      # ISP's DNS server IP address
cmd="ipfw -q add "     # build rule prefix
ks="keep-state"        # just too lazy to key this each time
$cmd 00500 check-state
$cmd 00502 deny all from any to any frag
$cmd 00501 deny tcp from any to any established
$cmd 00600 allow tcp from any to any 80 out via $oif setup $ks
$cmd 00610 allow tcp from any to $odns 53 out via $oif setup $ks
$cmd 00611 allow udp from any to $odns 53 out via $oif $ks
################### End of example ipfw rules script #############
```

Die Regeln in diesem Beispiel sind nicht wichtig. Wichtig ist es, zu zeigen, wie die symbolische Substitution innerhalb der Regeln verwendet wird.

Wenn dieses Beispiel in etc/ipfw.rules gespeichert wurde, so könnten alle Regeln durch die Ausführung des folgenden Kommandos neu geladen werden:

```
# sh /etc/ipfw.rules
```

Anstelle von /etc/ipfw.rules kann ein beliebig anderer Name oder Speicherort verwendet werden.

Alternativ können die einzelnen Befehle dieses Skripts auch von Hand eingegeben werden:

```
# ipfw -q -f flush
# ipfw -q add check-state
# ipfw -q add deny all from any to any frag
# ipfw -q add deny tcp from any to any established
# ipfw -q add allow tcp from any to any 80 out via tun0 setup keep-state
# ipfw -q add allow tcp from any to 192.0.2.11 53 out via tun0 setup keep-state
# ipfw -q add 00611 allow udp from any to 192.0.2.11 53 out via tun0 keep-state
```

### 53.4.6. IPFW Kerneloptionen

Um die Unterstützung für IPFW statisch in den Kernel zu kompilieren, lesen Sie die Anweisungen in Konfiguration des FreeBSD-Kernels. Die folgenden Optionen können in der Kernelkonfigurationsdatei verwendet werden:

```
options    IPFIREWALL          # enables IPFW
```

```
options    IPFIREWALL_VERBOSE      # enables logging for rules with log keyword to
syslogd(8)
options    IPFIREWALL_VERBOSE_LIMIT=5   # limits number of logged packets per-entry
options    IPFIREWALL_DEFAULT_TO_ACCEPT # sets default policy to pass what is not
explicitly denied
options    IPFIREWALL_NAT       # enables basic in-kernel NAT support
options    LIBALIAS         # enables full in-kernel NAT support
options    IPFIREWALL_NAT64     # enables in-kernel NAT64 support
options    IPFIREWALL_NPTV6     # enables in-kernel IPv6 NPT support
options    IPFIREWALL_PMOD      # enables protocols modification module support
options    IPDIVERT         # enables NAT through natd(8)
```



IPFW kann auch als Kernelmodul geladen werden: Die oben genannten Optionen werden standardmäßig als Module erstellt, oder können zur Laufzeit über Parameter festgelegt werden.

# 53.5. IPFILTER (IPF)

IPFILTER, auch als IPF bekannt, ist eine plattformübergreifende Open Source Firewall, die auf mehrere Betriebssysteme portiert wurde, einschließlich FreeBSD, NetBSD, OpenBSD und Solaris™.

IPFILTER basiert auf einer kernelseitigen Firewall und einem NAT-Mechanismus, der durch Anwenderprogramme gesteuert und überwacht werden kann. Firewallregeln werden mit ipf gesetzt oder gelöscht. Für die Manipulation der NAT-Regeln wird ipnat benutzt. Mit ipfstat werden Laufzeitstatistiken der kernelseitigen Anteile von IPFILTER aufgelistet. Mit ipmon können die Aktionen von IPFILTER in Protokolldateien gespeichert werden.

IPF wurde ursprünglich mit der Verarbeitungslogik "die letzte passende Regel gewinnt" geschrieben und verwendete ausschließlich Regeln ohne feste Zustände. Inzwischen wurde IPF modernisiert und unterstützt nun auch die Optionen `quick` und `keep state`.

Antworten auf häufige Fragen finden Sie unter http://www.phildev.net/ipf/index.html. Ein Archiv der IPFILTER Mailingliste steht unter http://marc.info/?l=ipfilter zur Verfügung.

Dieser Abschnitt des Handbuchs konzentriert sich auf IPF unter FreeBSD. Es werden auch Firewallregeln mit den Optionen `quick` und `keep state` vorgestellt.

## 53.5.1. IPF aktivieren

IPF ist in FreeBSD als ladbares Kernelmodul enthalten. Das bedeutet, dass Sie keinen angepassten Kernel erzeugen müssen um IPF zu aktivieren.

Benutzer, die IPF lieber statisch in den Kernel kompilieren, sollten den Anweisungen in Konfiguration des FreeBSD-Kernels folgen. Die folgenden Kerneloptionen stehen zur Verfügung:

```
options IPFILTER
options IPFILTER_LOG
options IPFILTER_LOOKUP
```

```
options IPFILTER_DEFAULT_BLOCK
```

`options IPFILTER` aktiviert die Unterstützung für IPFILTER. `options IPFILTER_LOG` aktiviert die Protokollierung über die Pseudo-Schnittstelle ipl für Firewallrelgen, die das Schlüsselwort `log` enthalten. `IPFILTER_LOOKUP` aktiviert IP-Pools, um die Suche nach IP-Adressen zu beschleunigen. `IPFILTER_DEFAULT_BLOCK` ändert das Verhalten der Firewall dahingehend, dass jedes Paket, das nicht explizit von einer `pass`-Regel Zugang erhält, geblockt wird.

Um IPF während des Bootens zu aktivieren, müssen folgende Einträge in /etc/rc.conf hinzugefügt werden. Diese Einträge aktivieren ebenfalls die Protokollierung und die Regel `default pass all`. Um diese Voreinstellung zu ändern, ohne einen neuen Kernel zu übersetzen, müssen Sie am Ende der Firewallregeln eine `block all` Regel hinzufügen.

```
ipfilter_enable="YES"             # Start ipf firewall
ipfilter_rules="/etc/ipf.rules"   # loads rules definition text file
ipv6_ipfilter_rules="/etc/ipf6.rules" # loads rules definition text file for IPv6
ipmon_enable="YES"                # Start IP monitor log
ipmon_flags="-Ds"                 # D = start as daemon
                                  # s = log to syslog
                                  # v = log tcp window, ack, seq
                                  # n = map IP & port to names
```

Wenn die NAT-Funktionalität benötigt wird, müssen auch diese Zeilen hinzugefügt werden:

```
gateway_enable="YES"              # Enable as LAN gateway
ipnat_enable="YES"                # Start ipnat function
ipnat_rules="/etc/ipnat.rules"    # rules definition file for ipnat
```

Jetzt können Sie IPF starten:

```
# service ipfilter start
```

Um die Firewallregeln zu laden, übergeben Sie den Namen des Regelwerks an `ipf`. Mit dem folgenden Kommando ersetzen Sie alle aktuell geladenen Regeln:

```
# ipf -Fa -f /etc/ipf.rules
```

`-Fa` löscht zunächst alle internen Regeln und mit `-f` wird die Datei angegeben, welche die zu ladenen Regeln enthält.

Damit haben Sie die Möglichkeit, Änderungen an der laufenden Firewall zu machen, ohne dass das System neu gestartet werden muss. Da dieser Vorgang beliebig oft wiederholt werden kann, ist es ein sehr bequemer Weg neue Regeln zu testen.

Diese und weitere Optionen sind in ipf(8) beschrieben.

### 53.5.2. IPF Regel-Syntax

Mit der hier beschriebenen Regel-Syntax können zustandsorientierte Regeln erstellt werden. Beim Erstellen von Regeln ist zu beachten, dass Regeln ohne das Schlüsselwort `quick` der Reihe nach geprüft werden und "die letzte zutreffende Regel" angewendet wird. Das bedeutet, dass selbst dann, wenn die erste zutreffende Regel eine `pass`-Regel ist, das Paket dennoch geblockt wird, falls später eine `block`-Regel zutrifft. Beispielregelsätze finden Sie in /usr/shared/examples/ipfilter.

Beim Erstellen von Regeln wird das Zeichen `#` verwendet, um einen Kommentar bis zum Ende der Zeile einzuleiten. Leere Zeilen werden ignoriert.

Die Schlüsselwörter, die in den Regeln verwendet werden, müssen in einer bestimmten Reihenfolge geschrieben werden, von links nach rechts. Einige Schlüsselwörter sind verbindlich, andere sind optional. Einige Schlüsselwörter haben Unteroptionen, die wiederum selbst Schlüsselwörter sind und ebenfalls weitere Unteroptionen einschließen können. Die Reihenfolge der Schlüsselwörter ist wie folgt, wobei die Wörter in Großbuchstaben eine Variable darstellen und die Wörter in Kleinbuchstaben der Variable vorangestellt werden müssen:

```
ACTION DIRECTION OPTIONS proto PROTO_TYPE from SRC_ADDR SRC_PORT to DST_ADDR DST_PORT
TCP_FLAG|ICMP_TYPE keep state STATE
```

Dieser Abschnitt beschreibt jedes dieser Schlüsselwörter und ihre Optionen. Es ist jedoch keine vollständige Liste aller möglichen Optionen. ipf(5) enthält eine vollständige Beschreibung der Syntax und einige Beispiele zur Erstellung von IPF-Regeln.

**ACTION**

Dieses Schlüsselwort bestimmt, was mit dem Paket zu tun ist, wenn es auf eine Regel zutrifft. Jede Regel *muss* dieses Schlüsselwort enthalten. Die folgenden Aktionen werden erkannt:

`block`: Das Paket wird verworfen.

`pass`: Das Paket wird durchgelassen.

`log`: Das Paket wird protokolliert.

`count`: Zählt die Anzahl der Pakete und die Bytes. Die kann einen Hinweis darauf geben, wie oft Pakete auf diese Regel zutreffen.

`auth`: Das Paket geht in eine Warteschlange zur Weiterverarbeitung durch ein anderes Programm.

`call`: Ermöglicht den Zugriff auf eingebaute IPF-Funktionen, die komplexere Aktionen ermöglichen.

`decapsulate`: Entfernt alle Header, um den Inhalt des Pakets zu verarbeiten.

**DIRECTION**

Als nächstes muss für jede Regel explizit die Richtung mit einem der folgenden Schlüsselwörter angegeben werden:

`in`: Die Regel wird auf ein eingehendes Paket angewendet.

`out`: Die Regel wird auf ein ausgehendes Paket angewendet.

`all`: Die Regel gilt für beide Richtungen.

Wenn das System mehrere Schnittstellen ausweist, kann die Schnittstelle zusammen mit der Richtung angegeben werden. Ein Beispiel wäre `in on fxp0`.

**OPTIONS**

Optionen müssen nicht zwingend angegeben werden. Falls jedoch mehrere Optionen angegeben werden, müssen sie in der hier gezeigten Reihenfolge verwendet werden.

`log`: Wenn die Firewall die angegebene Aktion durchführt, werden die Kopfdaten des Pakets auf der Pseudo-Schnittstelle ipl(4) protokolliert.

`quick`: Wenn ein Paket mit dieser Regel übereinstimmt, wird die Aktion für diese Regel ausgeführt und die Regelprüfung stoppt an dieser Stelle.

`on`: Auf dieses Schlüsselwort muss der Name der Schnittstelle folgen. Die Regel trifft nur dann zu, wenn das Paket auf der angegebenen Schnittstelle in die angegebene Richtung geht.

Wenn das Schlüsselwort `log` verwendet wird, können die folgenden Ausdrücke in dieser Reihenfolge benutzt werden:

`body`: die ersten 128 Bytes des Paketinhaltes werden zusätzlich zu den Kopfdaten protokolliert.

`first`: trifft nur zu, wenn das Schlüsselwort `log` zusammen mit `keep-state` verwendet wird. Es bestimmt, dass nur das auslösende Paket protokolliert wird und nicht jedes weitere Paket, dass von der gespeicherten Status-Regel betroffen ist.

Es stehen noch weitere Optionen zur Rückmeldung von Fehlern verfügbar. Ausführliche Details finden Sie in ipf(5).

**PROTO_TYPE**

Der Protokolltyp ist optional. Er ist jedoch zwingend erforderlich, falls die Regel einen SRC_PORT oder DST_PORT angeben muss da es den Typ des Protokolls bestimmt. Wenn Sie das Protokoll angeben, verwenden Sie das Schlüsselwort `proto`, gefolgt von der Protokollnummer oder dem Namen aus /etc/protocols. Zum Beispiel `tcp`, `udp`, oder `icmp`. Wenn PROTO_TYPE angegeben wird und SCR_PORT oder DST_PORT ausgelassen werden, stimmen alle Portnummern für dieses Protokoll mit dieser Regel überein.

**SRC_ADDR**

Das Schlüsselwort `from` ist verpflichtend und darauf folgt das Schlüsselwort, das die Quelle des Pakets darstellt. Die Quelle kann ein Rechnername, eine IP-Adresse gefolgt von der CIDR-Maske, ein Adresspool oder das Schlüsselwort `all` sein. ipf(5) enthält einige Beispiele.

IP-Bereiche können nur in der CIDR-Notation angegeben werden. Der Port oder das Paket net-mgmt/ipcalc hilft bei der Berechnung der richtigen CIDR-Maske. Weiterführende Informationen finden Sie auf der Webseite http://jodies.de/ipcalc.

**SCR_PORT**

Die Portnummer der Quelle ist optional. Wenn sie jedoch verwendet wird, muss in der Regel zuerst PROTO_TYPE angegeben werden. Die Portnummer muss auch auf das Schlüsselwort `proto` folgen.

Es werden verschiedene Vergleichsoperatoren unterstützt: `=` (gleich), `!=` (nicht gleich), `<` (kleiner als), `>` (größer als), ⇐ (kleiner als oder gleich) `>=` (größer als oder gleich).

Um Portbereiche anzugeben, schreiben Sie zwei Portnummern zwischen `<>` (kleiner als und größer als), `><` (größer als und kleiner als), oder `:` (größer als oder gleich und kleiner als oder gleich).

**DST_ADDR**

Das Schlüsselwort `to` ist verpflichtend und darauf folgt das Schlüsselwort, welches das Ziel des Pakets darstellt. Dieses Ziel kann ein Rechnername, eine IP-Adresse gefolgt von der CIDR-Maske, ein Adresspool oder das Schlüsselwort `all` sein.

**DST_PORT**

Die Portnummer des Ziels ist optional. Wenn sie jedoch verwendet wird, muss in der Regel zuerst PROTO_TYPE angegeben werden. Die Portnummer muss auch auf das Schlüsselwort `proto` folgen.

**TCP_FLAG|ICMP_TYPE**

Wenn `tcp` als PROTO_TYPE verwendet wird, können bestimmte TCP-Flags angegeben werden, die den Zustand einer Verbindung bestimmen. Mögliche Flags sind: `S` (SYN), `A` (ACK), `P` (PSH), `F` (FIN), `U` (URG), `R` (RST), `C` (CWN) und `E` (ECN).

Wenn `icmp` als PROTO_TYPE verwendet wird, kann der ICMP-Typ mit angegeben werden. ipf(5) enthält eine Auflistung der zulässigen Typen.

**STATE**

Wenn eine `pass`-Regel das Schlüsselwort `keep state` enthält, wird IPF einen Eintrag in der dynamischen Zustandstabelle hinzufügen, damit nachfolgende Pakete dieser Verbindung ebenfalls durchgelassen werden. IPF kann den Zustand für TCP, UDP und ICMP-Sitzungen verfolgen. IPF wird jedes Paket, das zu einer aktiven Sitzung gehört, durchlassen, auch wenn ein anderes Protokoll verwendet wird.

Pakete, die über die Schnittstelle zum öffentlichen Internet raus gehen, werden von IPF zuerst gegen die dynamische Zustandstabelle geprüft. Wenn das nächste Paket dieser aktiven Sitzung mit dem vorherigen Paket übereinstimmt, verlässt dieses Paket die Firewall und der Status wird in der dynamischen Zustandstabelle aktualisiert. Pakete, die nicht zu einer aktiven Sitzung gehören, werden gegen ausgehende Regeln geprüft. Eingehende Pakete von der Schnittstelle zum öffentlichen Internet werden gegen die dynamische Zustandstabelle geprüft. Wenn das nächste Paket mit der aktiven Sitzung übereinstimmt, verlässt dieses Paket die Firewall und der Status wird in der dynamischen Zustandstabelle aktualisiert. Pakete, die nicht zu einer aktiven Sitzung gehören, werden gegen eingehende Regeln geprüft.

Mehrere Schlüsselwörter können an `keep state` angefügt werden. Bei der Verwendung dieser Schlüsselwörter werden verschiedene Optionen gesetzt, um die zustandsorientierte Filterung zu

steuern. ipf(5) enthält eine Liste der verfügbaren Optionen und deren Beschreibungen.

### 53.5.3. Beispielregelsatz

Dieser Abschnitt beschreibt die Erstellung eines Regelsatzes, welcher nur entsprechende Dienste erlaubt und alle anderen Verbindungen blockiert.

FreeBSD verwendet die Loopback-Schnittstelle (lo0) und die IP-Adresse `127.0.0.1` zur internen Kommunikation. Der Regelsatz muss Regeln enthalten, die Pakete für diesen internen Verkehr ermöglichen:

```
# no restrictions on the loopback interface
pass in quick on lo0 all
pass out quick on lo0 all
```

Die mit dem Internet verbundene Schnittstelle wird für die Autorisierung und den Zugriff aller ein- und ausgehenden Verbindungen verwendet. Wenn eine oder mehrere Schnittstellen mit privaten Netzwerken verbunden sind, müssen Regeln existieren, die den Datenverkehr aus dem LAN zwischen den internen Netzwerken oder ins Internet erlauben. Der Regelsatz sollte in drei Bereiche unterteilt werden: vertrauenswürdige interne Schnittstellen, ausgehende Verbindungen über die öffentlichen Schnittstellen und eingehende Verbindungen über die öffentliche Schnittstelle.

Diese beiden Regeln erlauben den gesamten Datenverkehr über eine vertrauenswürdige LAN-Schnittstelle namens xl0:

```
# no restrictions on inside LAN interface for private network
pass out quick on xl0 all
pass in quick on xl0 all
```

Die Regeln für den ein- und ausgehenden Verkehr der öffentlichen Schnittstelle sollten in einer bestimmten Reihenfolge geschrieben werden. Zuerst Regeln, die häufiger übereinstimmen, danach Regeln, die seltener übereinstimmen. Die letzte Regel blockiert und protokolliert alle Pakete auf der Schnittstelle.

Der folgende Regelsatz definiert die ausgehenden Regeln der öffentlichen Schnittstelle dc0. Die Regeln prüfen den Zustand und identifizieren bestimmte Dienste, auf die die internen Systeme zugreifen dürfen. Alle Regeln verwenden das Schlüsselwort `quick` und geben die passenden Portnummern und ggf. auch die Zieladressen an.

```
# interface facing Internet (outbound)
# Matches session start requests originating from or behind the
# firewall, destined for the Internet.

# Allow outbound access to public DNS servers.
# Replace x.x.x. with address listed in /etc/resolv.conf.
# Repeat for each DNS server.
pass out quick on dc0 proto tcp from any to x.x.x. port = 53 flags S keep state
```

```
pass out quick on dc0 proto udp from any to xxx port = 53 keep state

# Allow access to ISP's specified DHCP server for cable or DSL networks.
# Use the first rule, then check log for the IP address of DHCP server.
# Then, uncomment the second rule, replace z.z.z.z with the IP address,
# and comment out the first rule
pass out log quick on dc0 proto udp from any to any port = 67 keep state
#pass out quick on dc0 proto udp from any to z.z.z.z port = 67 keep state

# Allow HTTP and HTTPS
pass out quick on dc0 proto tcp from any to any port = 80 flags S keep state
pass out quick on dc0 proto tcp from any to any port = 443 flags S keep state

# Allow email
pass out quick on dc0 proto tcp from any to any port = 110 flags S keep state
pass out quick on dc0 proto tcp from any to any port = 25 flags S keep state

# Allow NTP
pass out quick on dc0 proto tcp from any to any port = 37 flags S keep state

# Allow FTP
pass out quick on dc0 proto tcp from any to any port = 21 flags S keep state

# Allow SSH
pass out quick on dc0 proto tcp from any to any port = 22 flags S keep state

# Allow ping
pass out quick on dc0 proto icmp from any to any icmp-type 8 keep state

# Block and log everything else
block out log first quick on dc0 all
```

Die folgenden Beispielregeln für den eingehenden Verkehr auf der öffentlichen Schnittstelle blockieren zuerst alle unerwünschten Pakete. Dies reduziert die Anzahl der Pakete, die durch die letzte Regel protokolliert werden.

```
# interface facing Internet (inbound)
# Block all inbound traffic from non-routable or reserved address spaces
block in quick on dc0 from 192.168.0.0/16 to any    #RFC 1918 private IP
block in quick on dc0 from 172.16.0.0/12 to any     #RFC 1918 private IP
block in quick on dc0 from 10.0.0.0/8 to any        #RFC 1918 private IP
block in quick on dc0 from 127.0.0.0/8 to any       #loopback
block in quick on dc0 from 0.0.0.0/8 to any         #loopback
block in quick on dc0 from 169.254.0.0/16 to any    #DHCP auto-config
block in quick on dc0 from 192.0.2.0/24 to any      #reserved for docs
block in quick on dc0 from 204.152.64.0/23 to any   #Sun cluster interconnect
block in quick on dc0 from 224.0.0.0/3 to any       #Class D & E multicast

# Block fragments and too short tcp packets
block in quick on dc0 all with frags
```

```
block in quick on dc0 proto tcp all with short

# block source routed packets
block in quick on dc0 all with opt lsrr
block in quick on dc0 all with opt ssrr

# Block OS fingerprint attempts and log first occurrence
block in log first quick on dc0 proto tcp from any to any flags FUP

# Block anything with special options
block in quick on dc0 all with ipopts

# Block public pings and ident
block in quick on dc0 proto icmp all icmp-type 8
block in quick on dc0 proto tcp from any to any port = 113

# Block incoming Netbios services
block in log first quick on dc0 proto tcp/udp from any to any port = 137
block in log first quick on dc0 proto tcp/udp from any to any port = 138
block in log first quick on dc0 proto tcp/udp from any to any port = 139
block in log first quick on dc0 proto tcp/udp from any to any port = 81
```

Wenn eine Regel mit der Option `log first` protokolliert wird, können Sie mit `ipfstat -hio` prüfen, wie viele Übereinstimmungen es für diese Regel gibt. Eine große Anzahl von Übereinstimmungen kann darauf hindeuten, dass das System angegriffen wird.

Die restlichen Regeln definieren, welche Verbindungen aus dem Internet kommend hergestellt werden dürfen. Die letzte Regel blockiert alle Verbindungen, die nicht ausdrücklich von vorhergehenden Regeln erlaubt wurden.

```
# Allow traffic in from ISP's DHCP server. Replace z.z.z.z with
# the same IP address used in the outbound section.
pass in quick on dc0 proto udp from z.z.z.z to any port = 68 keep state

# Allow public connections to specified internal web server
pass in quick on dc0 proto tcp from any to x.x.x.x port = 80 flags S keep state

# Block and log only first occurrence of all remaining traffic.
block in log first quick on dc0 all
```

### 53.5.4. NAT Konfiguration

Um NAT zu aktivieren, fügen Sie folgende Zeilen in /etc/rc.conf hinzu. Geben Sie den Namen der Datei an, welche die NAT-Regeln enthält:

```
gateway_enable="YES"
ipnat_enable="YES"
```

```
ipnat_rules="/etc/ipnat.rules"
```

NAT-Regeln sind sehr flexibel, um den Bedürfnissen von kommerziellen Anwendern und Heimanwendern gerecht zu werden. Die hier vorgestellte Regelsyntax wurde vereinfacht, um die gemeinsame Nutzung zu demonstrieren. Eine vollständige Beschreibung der Syntax finden Sie in ipnat(5).

Die grundlegende Syntax für eine NAT-Regel ist wie folgt. `map` leitet die Regel ein und `IF` sollte durch den Namen der externen Schnittstelle ersetzt werden:

```
map IF LAN_IP_RANGE -> PUBLIC_ADDRESS
```

`LAN_IP_RANGE` ist ein Bereich von IP-Adressen, der von den internen Rechnern verwendet wird. In der Regel ist dies ein privater Bereich, beispielsweise `192.168.1.0/24`. `PUBLIC_ADDRESS` kann entweder eine statische externe IP-Adresse sein, oder das Schlüsselwort `0/32`, welches der zugewiesenen IP-Adresse für *IF* entspricht.

Wenn ein Paket aus dem LAN mit einem öffentlichen Ziel an der IPF Firewall ankommt, werden zunächst die Regeln für den ausgehenden Verkehr geprüft. Danach wird das Paket an das NAT-Regelwerk geleitet, wo es von oben nach unten gelesen und geprüft wird, wobei die erste übereinstimmende Regel gewinnt. IPF testet jede NAT-Regel gegen die Schnittstelle und die Quell-IP-Adresse des Pakets. Wenn der Schnittstellenname des Pakets mit einer NAT-Regel übereinstimmt, wird geprüft, ob die Quell-IP-Adresse des Pakets auf den Bereich in `LAN_IP_RANGE` passt. Wenn dies der Fall ist, wird die Quell-IP-Adresse des Pakets mit der Adresse aus `PUBLIC_ADDRESS` überschrieben. IPF speichert die Einträge in seiner internen NAT-Tabelle, so dass wenn das Paket aus dem Internet zurückkehrt, es der ursprünglichen privaten IP-Adresse zugeordnet werden kann, bevor es von den weiteren Firewallregeln geprüft wird.

Bei Netzwerken mit einer großen Anzahl von Systemen oder mehreren Subnetzen, steigert sich der Ressourcenverbrauch für das Umschreiben der IP-Adressen. Es existieren zwei Methoden, um dieses Problem zu umgehen.

Bei der ersten Methode wird ein Portbereich definiert, der für die Quell-Ports verwendet wird. Durch das Hinzufügen des Schlüsselworts `portmap` kann NAT angewiesen werden, nur Quell-Ports aus dem angegebenen Bereich zu benutzen:

```
map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp 20000:60000
```

Alternativ kann das Schlüsselwort `auto` verwendet werden. Dadurch ermittelt NAT selbstständig die zur Verfügung stehenden Ports:

```
map dc0 192.168.1.0/24 -> 0/32 portmap tcp/udp auto
```

Mit der zweiten Methode wird ein Pool von öffentlichen Adressen verwendet. Dies ist nützlich, wenn es viele Systeme im Netzwerk gibt und ein Block öffentlicher IP-Adressen verfügbar ist. Aus diesem Pool kann NAT dann IP-Adressen für die ausgehenden Pakete auswählen.

Der Bereich der öffentlichen IP-Adressen kann mit einer Netzmaske oder der CIDR-Notation festgelegt werden. Die folgenden Regeln sind identisch:

```
map dc0 192.168.1.0/24 -> 204.134.75.0/255.255.255.0
map dc0 192.168.1.0/24 -> 204.134.75.0/24
```

Es ist gängige Praxis, öffentlich zugängliche Web- oder Mail-Server getrennt von den internen Netzwerksegmenten zu betreiben. Der Verkehr von diesen Servern muss dennoch von NAT bearbeitet werden und die Portumleitung ist erforderlich, um den eingehenden Datenverkehr an den richtigen Server zu leiten. Verwenden Sie beispielsweise folgende Regel, um den eingehenden Verkehr auf der öffentlichen IP-Adresse `20.20.20.5` dem internen Server mit der Adresse `10.0.10.25` zuzuordnen:

```
rdr dc0 20.20.20.5/32 port 80 -> 10.0.10.25 port 80
```

Wenn dies der einzige Webserver im Netz ist, würde auch folgende Regel funktionieren, die alle HTTP-Anfragen an `10.0.10.25` umleitet:

```
rdr dc0 0.0.0.0/0 port 80 -> 10.0.10.25 port 80
```

IPF enthält einen FTP-Proxy, der zusammen mit NAT benutzt werden kann. Dieser Proxy überwacht den ausgehenden Datenverkehr für aktive und passive Verbindungsanfragen und erstellt dynamische Filterregeln, welche die Portnummern des jeweiligen FTP-Datenkanal enthalten. Dadurch entfällt die Notwendigkeit, viele Ports für FTP-Verbindungen zu öffnen.

In diesem Beispiel verwendet die erste Regel den Proxy für ausgehende FTP-Verbindungen aus dem internen LAN. Die zweite Regel übergibt den FTP-Datenverkehr von der Firewall an das Internet und die dritte Regel handhabt den restlichen Datenverkehr aus dem internen LAN:

```
map dc0 10.0.10.0/29 -> 0/32 proxy port 21 ftp/tcp
map dc0 0.0.0.0/0 -> 0/32 proxy port 21 ftp/tcp
map dc0 10.0.10.0/29 -> 0/32
```

FTP `map`-Regeln stehen vor den NAT-Regeln. Wenn ein Paket mit der FTP-Regel übereinstimmt, erstellt der FTP-Proxy eine temporäre Filterregel, damit die Pakete durchgelassen und von NAT verarbeitet werden können. Alle Pakte aus dem LAN, die nicht für FTP bestimmt sind, werden von NAT verarbeitet, wenn sie mit der dritten Regel übereinstimmen.

Ohne den FTP-Proxy würden stattdessen folgende Regeln benötigt. Beachten Sie, dass ohne den Proxy alle Ports oberhalb von `1024` freigegeben werden müssen:

```
# Allow out LAN PC client FTP to public Internet
# Active and passive modes
pass out quick on rl0 proto tcp from any to any port = 21 flags S keep state
```

```
# Allow out passive mode data channel high order port numbers
pass out quick on rl0 proto tcp from any to any port > 1024 flags S keep state$
# Active mode let data channel in from FTP server
pass in quick on rl0 proto tcp from any to any port = 20 flags S keep state
```

Nachdem die Datei mit den NAT-Regeln bearbeitet wurde, führen Sie `ipnat` mit `-CF` aus, um die aktuellen NAT-Regeln und den Inhalt der dynamischen Zuordnungstabelle zu löschen. Geben Sie `-f` zusammen mit dem NAT-Regelsatz an:

```
# ipnat -CF -f /etc/ipnat.rules
```

Statistiken zu NAT lassen sich wie folgt anzeigen:

```
# ipnat -s
```

Die aktuellen Zuordnungen der NAT-Tabelle geben Sie mit diesem Kommando aus:

```
# ipnat -l
```

Ausführliche Informationen erhalten Sie mit:

```
# ipnat -v
```

### 53.5.5. IPF Statistiken

IPF enthält mit ipfstat(8) ein Werkzeug, mit dem Statistiken abgerufen und angezeigt werden können. Die Zahlen beziehen sich auf den Zeitpunkt, an dem die Firewall zuletzt gestartet wurde, beziehungsweise die Statistik mit `ipf -Z` zurückgesetzt wurde.

Die Ausgabe von `ifstat` sieht in etwa wie folgt aus:

```
input packets: blocked 99286 passed 1255609 nomatch 14686 counted 0
output packets: blocked 4200 passed 1284345 nomatch 14687 counted 0
input packets logged: blocked 99286 passed 0
output packets logged: blocked 0 passed 0
packets logged: input 0 output 0
log failures: input 3898 output 0
fragment state(in): kept 0 lost 0
fragment state(out): kept 0 lost 0
packet state(in): kept 169364 lost 0
packet state(out): kept 431395 lost 0
ICMP replies: 0 TCP RSTs sent: 0
Result cache hits(in): 1215208 (out): 1098963
IN Pullups succeeded: 2 failed: 0
OUT Pullups succeeded: 0 failed: 0
```

```
Fastroute successes: 0 failures: 0
TCP cksum fails(in): 0 (out): 0
Packet log flags set: (0)
```

Es stehen viele Optionen zur Verfügung. Wird entweder `-i` (eingehend) oder `-o` (ausgehend) angegeben, wird der Befehl die entsprechende Liste mit den derzeit vom Kernel benutzten Filterregeln anzeigen. Um auch die Regelnummern zu sehen, muss `-n` angegeben werden. Zum Beispiel zeigt `ipfstat -on` die Tabelle für ausgehende Regeln und die Regelnummer an:

```
@1 pass out on xl0 from any to any
@2 block out on dc0 from any to any
@3 pass out quick on dc0 proto tcp/udp from any to any keep state
```

Wenn Sie der Regel ein `-h` voranstellen, wird der Zähler für die jeweilige Regel ausgegeben. Zum Beispiel gibt `ipfstat -oh` die ausgehenden Regeln inklusive der Zähler aus:

```
2451423 pass out on xl0 from any to any
354727 block out on dc0 from any to any
430918 pass out quick on dc0 proto tcp/udp from any to any keep state
```

Benutzen Sie `ipfstat -t` um die Zustandstabelle in einem top(1) ähnlichen Format anzuzeigen. Unterliegt die Firewall einem Angriff, bietet diese Option die Möglichkeit, die entsprechenden Pakete zu identifizieren. Mit den optionalen Flags können IP-Adressen, Ports oder Protokolle in Echtzeit überwacht werden. Lesen Sie ipfstat(8) für weitere Informationen.

### 53.5.6. IPF Protokollierung

IPF enthält mit `ipmon` ein Werkzeug, mit dem die Protokolle der Firewall in menschenlesbarer Form gespeichert werden können. Dies erfordert jedoch, dass `options IPFILTER_LOG` in die Kernelkonfigurationsdatei hinzugefügt wird. Folgen Sie dazu den Anweisungen in Konfiguration des FreeBSD-Kernels.

Um eine kontinuierliche Protokolldatei bereitzustellen, läuft dieses Kommando normalerweise im Daemon-Modus, damit auch ältere Ereignisse nachverfolgt werden können. Da FreeBSD mit syslogd(8) ein Werkzeug besitzt, das automatisch Protokolldateien rotiert, wird in der Voreinstellung für `ipmon_flags -Ds` in rc.conf verwendet:

```
ipmon_flags="-Ds" # D = start as daemon
                  # s = log to syslog
                  # v = log tcp window, ack, seq
                  # n = map IP & port to names
```

Protokollierung bietet die im Nachhinein die Möglichkeit festzustellen, welche Pakete verworfen wurden, von welchen Adressen diese Pakete kamen und wohin sie gehen sollten. Diese Informationen sind hilfreich beim Aufspüren von Angreifern.

Nachdem die Protokollierung in /etc/rc.conf aktiviert und mit `service ipmon start` gestartet wurde, wird IPF Regeln aufzeichnen, welche das Schlüsselwort `log` enthalten. Der Firewalladministrator entscheidet, welche Regeln protokolliert werden. In der Regel werden nur geblockte Pakete protokolliert. Es ist üblich, das Schlüsselwort `log` in der letzten Regel des Regelsatzes mit aufzunehmen. Dies macht es möglich, alle Pakete zu sehen, die mit keiner Regel des Regelsatzes übereinstimmten.

In der Voreinstellung verwendet `ipmon -Ds local0` als Protokoll-Facility. Die folgenden Level können verwendet werden, um die erfassten Daten weiter aufzuspalten:

```
LOG_INFO - packets logged using the "log" keyword as the action rather than pass or
block.
LOG_NOTICE - packets logged which are also passed
LOG_WARNING - packets logged which are also blocked
LOG_ERR - packets which have been logged and which can be considered short due to an
incomplete header
```

Damit IPF alle Daten protokolliert, legen Sie zunächst eine neue Datei /var/log/ipfilter.log an:

```
# touch /var/log/ipfilter.log
```

Um alle Nachrichten in der angegebenen Datei zu protokollieren, fügen Sie den folgenden Eintrag in /etc/syslog.conf ein:

```
local0.* /var/log/ipfilter.log
```

Führen Sie `service syslogd reload` aus, damit syslogd(8) /etc/syslog.conf neu einliest, um die Änderungen zu aktivieren.

Denken Sie daran, auch /etc/newsyslog.conf anzupassen, damit das neue Protokoll rotiert wird.

Die von `ipmon` generierten Nachrichten bestehen aus Daten, welche durch Leerzeichen getrennt sind. Alle Nachrichten enthalten die folgenden Felder:

1. Das Datum, an dem das Paket empfangen wurde.
2. Die Uhrzeit, wann das Paket empfangen wurde. Das Format ist HH:MM:SS.F (Stunden, Minuten, Sekunden und Sekundenbruchteile).
3. Der Name der Schnittstelle, die das Paket verarbeitet hat.
4. Die Gruppen- und Regelnummer im Format `@0:17`.
5. Die Aktion: `p` für durchgelassene Pakete, `b` für blockierte Pakete, `S` für kurze Pakete, `n` für Pakete auf die keine Regel zutraf und `L` für Pakete die protokolliert wurden.
6. Die Adressen werden in drei Felder unterteilt: die Quelladresse und der Port getrennt durch Komma, das Zeichen →, sowie die Zieladresse und Port. Zum Beispiel `209.53.17.22,80 → 198.72.220.17,1722`.

7. `PR`, gefolgt vom Namen oder Nummer des Protokolls. Zum Beispiel `PR tcp`.
8. `len`, gefolgt von der Größe des Headers und der Gesamtgröße des Pakets. Zum Beispiel `len 20 40`.

Wenn es sich beim dem Paket um ein TCP-Paket handelt, gibt es ein zusätzliches Feld, das mit einem Bindestrich beginnt und die Buchstaben der entsprechenden Flags enthält. Eine Liste der Flags und deren Buchstaben finden Sie in ipf(5).

Wenn es sich beim dem Paket um ein ICMP-Paket handelt, gibt es zwei zusätzliche Felder: das erste Feld ist immer "icmp" und das zweite Feld enthält die ICMP-Nachricht und den Nachrichten-Code, getrennt durch einen Schrägstrich. Beispielswiese `icmp 3/3` für die Nachricht Port unreachable.

# 53.6. Blacklistd

Blacklistd ist ein Daemon der auf Sockets lauscht, um Benachrichtigungen von anderen Daemons über fehlgeschlagene oder erfolgreiche Verbindungsversuche zu erhalten. Dieser Daemon wird häufig verwendet, um zu viele Verbindungsversuche auf offenen Ports zu blockieren. Ein Beispiel ist SSH, das viele Anfragen von Bots oder Skripten erhält, die versuchen, Passwörter zu erraten und Zugriff zu erhalten. Mit Hilfe von blacklistd kann der Daemon die Firewall benachrichtigen, eine Filterregel zu erstellen, um übermäßige Verbindungsversuche einer einzigen Quelle nach einer Reihe von Versuchen zu blockieren. Blacklistd wurde ursprünglich auf NetBSD entwickelt und erschien dort in der Version 7. FreeBSD 11 hat blacklistd von NetBSD importiert.

In diesem Kapitel wird die Einrichtung und Konfiguration von blacklistd besprochen. Sie finden aber auch Beispiele für die Verwendung von blacklistd. Sie sollten allerdings mit grundlegenden Firewall-Konzepten wie Filterregeln vertraut sein. Weitere Informationen finden Sie im Kapitel Firewalls. In diesen Beispielen wird PF benutzt, aber auch andere unter FreeBSD verfügbare Firewalls sollten in der Lage sein mit blacklistd zusammen zu arbeiten.

## 53.6.1. Blacklistd aktivieren

Die Konfiguration für blacklistd wird in blacklistd.conf(5) gespeichert. Um das Laufzeitverhalten von blacklistd zu beeinflussen, sind verschiedene Kommandozeilenoptionen verfügbar. Die permanente Konfiguration über Neustarts hinweg sollte in /etc/blacklistd.conf gespeichert werden. Um den Daemon während des Systemstarts zu aktivieren, fügen Sie eine Zeile `blacklistd_enable` in /etc/rc.conf hinzu:

```
# sysrc blacklistd_enable=yes
```

Sie können den Daemon auch manuell starten:

```
# service blacklistd start
```

## 53.6.2. Erstellen von Blacklistd-Regeln

Die Regeln für blacklistd werden in blacklistd.conf(5) mit einem Eintrag pro Zeile konfiguriert. Jede

Regel enthält ein Tupel, das durch Leerzeichen oder Tabulator getrennt ist. Eine Regel gilt entweder für einen lokalen oder einen entfernten Rechner.

**Lokale Regeln**

Ein typischer Eintrag für eine lokale Regel in /etc/blacklistd.conf sieht wie folgt aus:

```
[local]
ssh             stream  *       *               *       3       24h
```

Alle Regeln, die dem Abschnitt `[local]` folgen, werden als lokale Regeln behandelt, die für den lokalen Rechner gelten. In einem `[remote]`-Abschnitt gelten alle Regeln für entfernte Maschinen.

Die sieben Felder einer Regel werden entweder durch Tabulator oder Leerzeichen getrennt. Die ersten vier Felder identifizieren den Netzwerkverkehr, welcher geblockt werden soll. Die drei folgenden Felder definieren das Verhalten von blacklistd. Wildcards werden mit einem Sternchen (`*`) gekennzeichnet und stimmen mit allen anderen in diesem Feld überein. Das erste Feld definiert den Standort. In den lokalen Regeln sind dies die Ports. Die Syntax ist wie folgt:

```
[address|interface][/mask][:port]
```

Adressen können als IPv4 im numerischen Format oder IPv6 in eckigen Klammern angegeben werden. Ebenfalls kann der Name der Schnittstelle wie `em0` verwendet werden.

Im zweiten Feld wird der Socket-Typ definiert. TCP-Sockets sind vom Typ `stream`, wohingegen UDP als `dgram` bezeichnet wird. Das obige Beispiel verwendet TCP, weil SSH dieses Protokoll benutzt.

Im dritten Feld kann ein Protokoll definiert werden. Die folgenden Protokolle können verwendet werden: `tcp`, `udp`, `tcp6`, `udp6` oder numerisch. Eine Wildcard, wie im Beispiel, wird typischerweise verwendet, um alle Protokolle abzubilden, es sei denn, es gibt einen Grund, den Verkehr nach einem bestimmten Protokoll zu differenzieren.

Im vierten Feld wird der effektive Benutzer oder Eigentümer des Daemon-Prozesses definiert, welcher das Ereignis meldet. Hier kann der Benutzer oder die UID sowie eine Wildcard verwendet werden (siehe Beispiel oben).

Der Name der Firewallregel wird im fünften Feld definiert. In der Voreinstellung setzt blacklistd alle geblockten Pakete unter einen pf-Anker namens `blacklistd` in pf.conf wie folgt:

```
anchor "blacklistd/*" in on $ext_if
block in
pass out
```

Für separate Blacklists kann in diesem Feld ein Ankername benutzt werden. In anderen Fällen genügt eine Wildcard. Ein Name mit vorangestelltem Bindestrich (`-`) bedeutet, das ein Anker mit dem voreingestellten Regelnamen verwendet werden sollte. Ein modifiziertes Beispiel von oben mit dem Bindestrich würde so aussehen:

```
ssh             stream  *       *               -ssh       3       24h
```

Mit einer solchen Regel werden alle neuen Blacklistregeln zu einem Anker namens `blacklistd-ssh` hinzugefügt.

Um ganze Subnetze für eine einzelne Regelverletzung zu blockieren, kann ein `/` im Regelnamen benutzt werden. Dadurch wird der verbleibende Teil des Namens als Maske interpretiert, die auf die in der Regel angegebene Adresse angewendet wird. Diese Regel würde beispielsweise jede Adresse blockieren, die an `/24` angrenzt:

```
22              stream  tcp       *               */24    3       24h
```



Es ist wichtig, hier das richtige Protokoll anzugeben. IPv4 und IPv6 behandeln `/24` unterschiedlich, deshalb kann `*` im dritten Feld für diese Regel nicht benutzt werden.

Diese Regel bewirkt, dass, wenn ein Rechner in diesem Netzwerk wegen seines Verhaltens blockiert wird, auch alle anderen Rechner aus diesem Netzwerk blockiert werden.

Das sechste Feld, genannt `nfail`, legt die Anzahl der Anmeldeversuche fest, die erforderlich sind, um die betreffende IP auf die Blacklist zu setzen. Eine Wildcard an dieser Stelle bedeutet, dass niemals geblockt wird. Im obigen Beispiel ist eine Anzahl von 3 definiert, was bedeutet, dass die IP nach drei fehlgeschlagenen Anmeldeversuchen über SSH gesperrt wird.

Das letzte Feld in der Regel gibt an, wie lange ein Rechner auf der Blacklist steht. Die Standardeinheit ist Sekunden, aber Suffixe wie `m` (Minuten), `h` (Stunden) und `d` (Tage) können auch angegeben werden.

Die Regel im Beispiel besagt, dass nach dreimaliger Authentifizierung über SSH eine neue PF-Regel für diesen Rechner angelegt wird. Beim Überprüfen der Regeln werden zuerst lokale Regeln, von sehr spezifisch bis am wenigsten spezifisch, geprüft. Wenn eine Übereinstimmung auftritt, werden die `remote`-Regeln angewendet und die Felder `name`, `nfail` und `disable` werden durch die entsprechende `remote`-Regel geändert.

**Remote-Regeln**

Mit Remote-Regeln wird das Verhalten von blacklistd, in Abhängigkeit vom aktuell ausgewerteten Remote-Rechner, festgelegt. Die einzelnen Felder einer Remote-Regel sind identisch mit den Feldern einer lokalen Regel. Der einzige Unterschied besteht darin, wie blacklistd sie verwendet. Zur besseren Verständlichkeit wird folgende Regel benutzt:

```
[remote]
203.0.113.128/25 *      *       *               =/25    =       48h
```

Das Adressfeld kann eine IP-Adresse (entweder v4 oder v6), einen Port oder beides beinhalten. Dies ermöglicht es, wie in diesem Beispiel, spezielle Regeln für einen bestimmten entfernten

Adressbereich festzulegen. Die Felder für den Socket-Typ, Protokoll und Besitzer werden genauso wie in den lokalen Regeln interpretiert.

Die Felder für den Namen sind jedoch unterschiedlich. Das Gleichheitszeichen (`=`) in einer Remote-Regel weist blacklistd an, den Wert aus der entsprechenden lokalen Regel zu verwenden. Das bedeutet, dass der Eintrag der Firewall-Regel übernommen und das Präfix `/25` (eine Netzmaske von `255.255.255.128`) hinzugefügt wird. Wenn eine Verbindung aus diesem Adressbereich geblockt wird, ist das gesamte Subnetz betroffen. Ein PF-Ankername kann auch hier verwendet werden. In diesem Fall fügt blacklistd Regeln für diesen Adressbereich dem Namen des Ankers hinzu. Die Standardtabelle wird verwendet, wenn eine Wildcard angegeben wird.

Für eine Adresse kann im Feld `nfail` die Anzahl von Fehlversuchen definiert werden. Dies ist nützlich für Ausnahmen, um weniger strenge Anwendungen zu ermöglichen, oder um Anmeldeversuche ein wenig nachsichtiger zu gestalten. Die Sperrung wird aufgehoben, wenn im sechsten Feld eine Wildcard benutzt wird.

Remote-Regeln ermöglichen eine strengere Durchsetzung der Beschränkungen bei Anmeldeversuchen im Vergleich zu Anmeldeversuchen die aus dem lokalen Netzwerk kommen.

### 53.6.3. Blacklistd Client Konfiguration

Es gibt einige Softwarepakete in FreeBSD, die die Funktionalität von blacklistd nutzen können. Die beiden bekanntesten sind ftpd(8) und sshd(8). Beide Programme nutzen blacklistd, um übermäßige Verbindungsversuche zu unterbinden. Um blacklistd im SSH-Daemon zu aktivieren, muss folgend Zeile in /etc/ssh/sshd_config hinzugefügt werden:

```
UseBlacklist yes
```

Damit die Änderungen wirksam werden, muss sshd im Anschluss neu gestartet werden.

Für ftpd(8) wird blacklistd mit dem Schalter `-B` aktiviert. Entweder in /etc/inetd.conf oder in /etc/rc.conf:

```
ftpd_flags="-B"
```

Das ist alles, was benötigt wird, damit diese Programme mit blacklist kommunizieren.

### 53.6.4. Blacklistd Verwaltung

Blacklistd stellt dem Benutzer das Verwaltungswerkzeug blacklistctl(8) zur Verfügung. Es zeigt blockierte Adressen und Netzwerke an, die nach den in blacklistd.conf(5) definierten Regeln auf der Blacklist stehen. Um die Liste der aktuell blockierten Rechner anzuzeigen, benutzen Sie `dump` zusammen mit der Option `-b`:

```
# blacklistctl dump -b
      address/ma:port id      nfail   last access
```

```
213.0.123.128/25:22   OK      6/3     2019/06/08 14:30:19
```

Dieses Beispiel zeigt, dass es sechs von drei erlaubten Anmeldeversuchen auf Port 22 aus dem Adressbereich `213.0.123.128/25` gab. Es sind mehr Versuche aufgelistet, als erlaubt sind, da SSH es einem Client erlaubt, mehrere Anmeldungen über eine einzige TCP-Verbindung zu tätigen. Eine derzeit laufende Verbindung wird nicht von blacklistd unterbunden. Der letzte Verbindungsversuch ist in der letzten Spalte der Ausgabe aufgeführt.

Um die verbleibende Zeit zu sehen, die sich dieser Rechner auf der Blacklist befindet, fügen Sie `-r` zum vorherigen Befehl hinzu:

```
# blacklistctl dump -br
      address/ma:port id      nfail   remaining time
213.0.123.128/25:22   OK      6/3     36s
```

In diesem Beispiel bleiben noch 36 Sekunden, bis dieser Rechner nicht mehr blockiert wird.

### 53.6.5. Rechner aus der Blocklist entfernen

Manchmal ist es notwendig, einen Rechner aus der Blocklist zu entfernen, bevor die verbleibende Zeit abgelaufen ist. Leider bietet blacklistd keine Möglichkeit dies zu tun. Es ist jedoch möglich, die Adresse mit `pfctl` aus der PF-Tabelle zu entfernen. Für den blockierten Port gibt es einen untergeordneten Anker innerhalb des definierten blacklistd-Ankers in /etc/pf.conf. Wenn es beispielsweise einen untergeordneten Anker zum Blockieren von Port 22 gibt, wird dieser als `blacklistd/22` bezeichnet. In diesem untergeordneten Anker befindet sich eine Tabelle, die die blockierten Adressen enthält. Diese Tabelle wird Port genannt, gefolgt von der Portnummer. In diesem Beispiel würde es `port22` heißen. Mit diesen Informationen und pfctl(8) ist es nun möglich, alle geblockten Adressen anzuzeigen:

```
# pfctl -a blacklistd/22 -t port22 -T show
...
213.0.123.128/25
...
```

Nachdem Sie die entsprechende Adresse ermittelt wurde, kann sie mit folgendem Befehl aus der Liste entfernt werden:

```
# pfctl -a blacklistd/22 -t port22 -T delete 213.0.123.128/25
```

Die Adresse ist nun aus PF entfernt, erscheint aber immer noch in der Liste von `blacklistctl`, da dieser keine Kenntnis von Änderungen an PF hat. Der Eintrag in blacklist's Datenbank wird irgendwann ablaufen und dann aus der Ausgabe entfernt werden. Der Eintrag wird wieder hinzugefügt, falls der Rechner erneut gegen eine der Regeln von blacklistd verstößt.

# Kapitel 54. Weiterführende Netzwerkthemen

## 54.1. Übersicht

Dieses Kapitel beschreibt verschiedene weiterführende Netzwerkthemen.

Nachdem Sie dieses Kapitel gelesen haben, werden Sie

- Die Grundlagen von Gateways und Routen kennen.
- Wissen, wie man USB Tethering einrichtet.
- Bluetooth®- sowie drahtlose, der Norm IEEE® 802.11 entsprechende, Geräte mit FreeBSD verwenden können.
- Eine Bridge unter FreeBSD einrichten können.
- Wissen, wie man mithilfe von PXE über ein Netzwerk von einem NFS Root-Dateisystem bootet.
- IPv6 auf einem FreeBSD-Rechner einrichten können.
- Das Common Address Redundancy Protocol (CARP) unter FreeBSD einsetzen können.
- Wissen, wie VLANs unter FreeBSD konfiguriert werden.
- Wissen, wie Bluetooth-Kopfhörer konfiguriert werden.

Bevor Sie dieses Kapitel lesen, sollten Sie

- Die Grundlagen der /etc/rc-Skripte verstanden haben.
- Mit der grundlegenden Netzwerkterminologie vertraut sein.
- Einen neuen FreeBSD-Kernel konfigurieren und installieren können (Konfiguration des FreeBSD-Kernels).
- Wissen, wie man zusätzliche Software von Drittherstellern installiert (Installieren von Anwendungen: Pakete und Ports).

## 54.2. Gateways und Routen

Der Mechanismus mit dem ein Rechner einen Rechner über ein Netzwerk finden kann, wird als *Routing* bezeichnet. Eine "Route" besteht aus einem definierten Adresspaar: Einem "Ziel" und einem "Gateway". Die Route zeigt an, dass Pakete über das *Gateway* zum *Ziel* gelangen können. Es gibt drei Arten von Zielen: Einzelne Rechner (Hosts), Subnetze und das "Standard"ziel. Die "Standardroute" wird verwendet, wenn keine andere Route zutrifft. Außerdem gibt es drei Arten von Gateways: Einzelne Rechner (Hosts), Schnittstellen (Interfaces, auch als "Links" bezeichnet), sowie Ethernet Hardware-Adressen (MAC). Bekannte Adressen werden in einer Routingtabelle gespeichert.

Dieser Abschnitt bietet einen Überblick über die Grundlagen des Routings. Er demonstriert, wie ein FreeBSD-System als Router konfiguriert werden kann und bietet einige Tipps zur Fehlerbehebung.

### 54.2.1. Grundlagen des Routings

netstat(1) zeigt die Routingtabellen eines FreeBSD-Systems an:

```
% netstat -r
Routing tables

Internet:
Destination      Gateway            Flags     Refs     Use     Netif Expire
default          outside-gw         UGS         37     418       em0
localhost        localhost          UH           0     181       lo0
test0            0:e0:b5:36:cf:4f   UHLW         5   63288       re0     77
10.20.30.255     link#1             UHLW         1    2421
example.com      link#1             UC           0       0
host1            0:e0:a8:37:8:1e    UHLW         3    4601       lo0
host2            0:e0:a8:37:8:1e    UHLW         0       5       lo0 =>
host2.example.com link#1            UC           0       0
224              link#1             UC           0       0
```

Die Einträge in diesem Beispiel sind wie folgt:

**default**

Die erste Route in der Ausgabe gibt die Standardroute (`default`) an. Wenn sich der lokale Rechner mit einem entfernten Rechner verbinden will, wird die Routingtabelle überprüft, um festzustellen, ob bereits ein bekannter Pfad vorhanden ist. Wird für den entfernten Rechner ein Eintrag in der Routingtabelle gefunden, so prüft das System ob es sich über die angegebene Schnittstelle verbinden kann.

Wenn das Zielsystem mit keinem Eintrag übereinstimmt, oder wenn alle bekannten Routen fehlschlagen, verwendet das System die Standardroute. Für die Rechner im lokalen Netzwerk ist das Feld `Gateway` auf das System gesetzt, welches direkt mit dem Internet verbunden ist. `UG` in der Spalte `Flags` zeigt an, dass das Gateway einsatzbereit ist.

Die Standardroute für einen Rechner, der selbst als Gateway zur Außenwelt fungiert, ist der Gateway-Rechner des Internetanbieters (ISP).

**localhost**

Die zweite Route zeigt die `localhost` Route. Die festgelegte Schnittstelle in der `Netif`-Spalte für `localhost` ist `lo0`, das auch als loopback-Gerät bekannt ist. Das bedeutet, dass der gesamte Datenverkehr für dieses Ziel intern bleibt, anstatt ihn über ein Netzwerk zu versenden.

**MAC-Adresse**

Bei den mit `0:e0:` beginnenden Adressen handelt es sich um MAC-Adressen. FreeBSD identifiziert Rechner im lokalen Netz, im Beispiel `test0`, automatisch und fügt eine direkte Route über die Ethernet-Schnittstelle re0 zu diesem Rechner hinzu. Außerdem existiert in der Spalte `Expire` ein Timeout, der verwendet wird, wenn dieser Rechner in einem definierten Zeitraum nicht reagiert. Wenn dies passiert, wird die Route zu diesem Rechner automatisch gelöscht. Rechner im lokalen Netz werden über das Routing Information Protocol (RIP) identifiziert,

welches den kürzesten Weg zu den jeweiligen Rechnern berechnet.

**Subnetz**

FreeBSD wird automatisch Subnetzrouten für das lokale Subnetz hinzufügen. In diesem Beispiel ist `10.20.30.255` die Broadcast-Adresse für das Subnetz `10.20.30`, und `example.com` ist der zu diesem Subnetz gehörige Domainname. Das Ziel `link#1` bezieht sich auf die erste Ethernet-Karte im Rechner.

Routen für Rechner im lokalen Netz und lokale Subnetze werden automatisch durch den routed(8) Daemon konfiguriert. Ist dieser nicht gestartet, existieren nur statische Routen, die vom Administrator definiert werden.

**Host**

Die Zeile `host1` bezieht sich auf den Rechner, der durch seine Ethernetadresse bekannt ist. Da es sich um den sendenden Rechner handelt, verwendet FreeBSD automatisch das Loopback-Gerät (lo0), anstatt den Datenverkehr über die Ethernet-Schnittstelle zu senden.

Die zwei `host2` Zeilen repräsentieren Aliase, die mit ifconfig(8) erstellt wurden. Das Symbol ⇒ nach der lo0-Schnittstelle sagt aus, dass zusätzlich zur Loopback-Adresse auch ein Alias eingestellt ist. Solche Routen sind nur auf Rechnern vorhanden, die den Alias bereitstellen. Alle anderen Rechner im lokalen Netz haben für solche Routen nur eine `link#1` Zeile.

**224**

Die letzte Zeile (Zielsubnetz `224`) behandelt Multicasting.

Schließlich gibt es für Routen noch verschiedene Attribute, die sich in der Spalte `Flags` befinden. Allgemeine Attribute in Routingtabellen fasst einige dieser Flags und deren Bedeutung zusammen:

*Tabelle 30. Allgemeine Attribute in Routingtabellen*

| Attribut | Bedeutung |
|---|---|
| U | Die Route ist aktiv (up). |
| H | Das Ziel der Route ist ein einzelner Rechner (Host). |
| G | Alle Daten, die an dieses Ziel gesendet werden, werden von dem Gateway an ihr jeweiliges Ziel weitergeleitet. |
| S | Diese Route wurde statisch konfiguriert. |
| C | Erzeugt eine neue Route, basierend auf der Route für den Rechner, mit dem wir uns verbinden. Diese Routenart wird normalerweise für lokale Netzwerke verwendet. |
| W | Eine Route, die automatisch konfiguriert wurde. Sie basiert auf einer lokalen Netzwerkroute (Clone). |

| Attribut | Bedeutung |
| --- | --- |
| L | Die Route beinhaltet einen Verweis auf eine Ethernetkarte (Link). |

In FreeBSD kann die Standardroute durch die Angabe der IP-Adresse des Standard-Gateways in /etc/rc.conf definiert werden:

```
defaultrouter="10.20.30.1"
```

Die Standardroute kann mit `route` auch manuell gesetzt werden:

```
# route add default 10.20.30.1
```

Beachten Sie, dass manuell hinzugefügte Routen bei einem Neustart des Systems verloren gehen. Weitere Informationen zum Bearbeiten von Netzwerk-Routingtabellen finden Sie in route(8).

### 54.2.2. Statische Routen einrichten

Ein FreeBSD-System kann als Standard-Gateway bzw. Router für ein Netzwerk konfiguriert werden, wenn es sich um einen Dual-Homed-Host handelt. Ein Dual-Homed-Host ist ein Rechner, der sich in mindestens zwei verschiedenen Netzwerken befindet. Typischerweise ist jedes Netzwerk über eine separate Netzwerkschnittstelle verbunden. Mit IP Aliasing können mehrere Adressen, die jeweils zu einem andren Subnetz gehören, an eine physikalische Schnittstelle gebunden werden.

Damit Pakete zwischen den Schnittstellen weitergeleitet werden können, muss das FreeBSD-System als Router konfiguriert werden. Internetstandards und gute Ingenieurspraxis sorgten dafür, dass diese Funktion in FreeBSD in der Voreinstellung deaktiviert ist. Sie kann jedoch aktiviert werden, indem folgende Zeile in /etc/rc.conf hinzugefügt wird:

```
gateway_enable="YES"          # Auf YES setzen, wenn der Rechner als Gateway arbeiten
soll
```

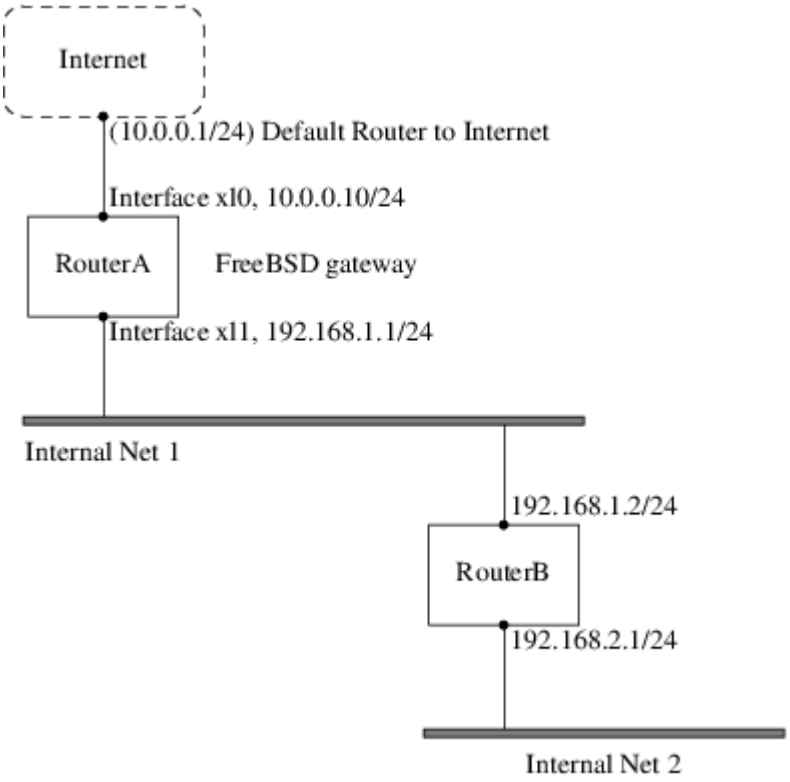
Um das Routing zu aktivieren, setzen Sie die sysctl(8)-Variable `net.inet.ip.forwarding` auf `1`. Um das Routing zu stoppen, muss die Variable wieder auf `0` gesetzt werden.

Die Routingtabelle eines Routers benötigt zusätzliche Routen, damit er weiß, wie er andere Netzwerke erreichen kann. Die Routen können entweder manuell als statische Routen hinzugefügt werden, oder aber der Router lernt automatisch die Routen anhand des Routing-Protokolls. Statische Routen eignen sich für kleine Netzwerke und dieser Abschnitt beschreibt, wie Sie eine statische Route für ein kleines Netzwerk hinzufügen.



In großen Netzwerken sind statische Routen schlecht skalierbar. FreeBSD beinhaltet den BSD-Routing-Daemon routed(8), der die Protokolle RIP (Version 1 und Version 2) sowie IRDP unterstützt. Die Routing-Protokolle BGP und OSPF können über den Port oder das Paket net/zebra installiert werden.

Nehmen wir an, dass wir über folgendes Netzwerk verfügen:




`RouterA`, ein FreeBSD-Rechner, dient als Router für den Zugriff auf das Internet. Die Standardroute ist auf `10.0.0.1` gesetzt, damit ein Zugriff auf das Internet möglich wird. `RouterB` ist bereits konfiguriert, da er `192.168.1.1` als Standard-Gateway benutzt.

Bevor die statischen Routen hinzugefügt werden, sieht die Routingtabelle auf `RouterA` in etwa so aus:

```
% netstat -nr
Routing tables

Internet:
Destination        Gateway            Flags    Refs      Use  Netif  Expire
default            10.0.0.1           UGS         0    49378    xl0
127.0.0.1          127.0.0.1          UH          0        6    lo0
10.0.0/24          link#1             UC          0        0    xl0
192.168.1/24       link#2             UC          0        0    xl1
```

Mit dieser Routingtabelle hat `RouterA` keine Route zum Netzwerk `192.168.2.0/24`. Der folgende Befehl wird das interne Netz 2 in die Routingtabelle von `RouterA` aufnehmen und dabei `192.168.1.2` als nächsten Zwischenschritt (Hop) verwenden:

```
# route add -net 192.168.2.0/24 192.168.1.2
```

Ab sofort kann `RouterA` alle Rechner des Netzwerks `192.168.2.0/24` erreichen. Allerdings gehen die Routing-Informationen verloren, wenn das FreeBSD-System neu gestartet wird. Um statische Routen dauerhaft einzurichten, müssen diese in /etc/rc.conf eingetragen werden:

```
# Add Internal Net 2 as a persistent static route
static_routes="internalnet2"
route_internalnet2="-net 192.168.2.0/24 192.168.1.2"
```

Die Variable `static_routes` enthält eine Reihe von Strings, die durch Leerzeichen getrennt sind. Jeder String bezieht sich auf den Namen einer Route. Die Variable `route__internalnet2_` enthält die statische Route.

Wird mit der Variablen `static_routes` mehr als eine Variable angegeben, so werden auch mehrere Routen angelegt. Im folgenden Beispiel werden statische Routen zu den Netzwerken `192.168.0.0/24` und `192.168.1.0/24` angelegt.

```
static_routes="net1 net2"
route_net1="-net 192.168.0.0/24 192.168.0.1"
route_net2="-net 192.168.1.0/24 192.168.1.1"
```

### 54.2.3. Problembehandlung

Wenn ein Adressraum einem Netzwerk zugeordnet wird, konfiguriert der Dienstanbieter seine Routing-Tabellen, so dass der gesamte Verkehr für das Netzwerk über die Verbindung zu der Seite gesendet wird. Aber woher wissen externe Webseiten, dass sie die Daten an das Netzwerk des ISP senden sollen?

Es gibt ein System, das alle zugewiesenen Adressräume verwaltet und die Verbindung zum Internet-Backbone definiert. Der "Backbone" ist das Netz aus Hauptverbindungen, die den Internetverkehr in der ganzen Welt transportieren und verteilen. Jeder Backbone-Rechner verfügt über eine Kopie von Master-Tabellen, die den Verkehr für ein bestimmtes Netzwerk hierarchisch vom Backbone über eine Kette von Dienstanbietern bis hin zu einem bestimmten Netzwerk leiten.

Es ist die Aufgabe des Dienstanbieters, den Backbone-Seiten mitzuteilen, dass sie mit einer Seite verbunden wurden. Dieser Vorgang wird als *Bekanntmachung von Routen* (routing propagation) bezeichnet.

Manchmal kommt es zu Problemen bei der Bekanntmachung von Routen, und einige Seiten sind nicht in der Lage, sich zu verbinden. Der vielleicht nützlichste Befehl, um festzustellen wo das Routing nicht funktioniert, ist `traceroute`. Das Programm ist nützlich, falls `ping` fehlschlägt.

Rufen Sie `traceroute` mit dem Namen des entfernten Rechners auf, mit dem eine Verbindung aufgebaut werden soll. Die Ausgabe zeigt die Gateway-Rechner entlang des Verbindungspfades an. Schließlich wird der Zielrechner erreicht oder es kommt zu einem Verbindungsabbruch. Weitere Informationen finden Sie in traceroute(8).

### 54.2.4. Multicast-Routing

FreeBSD unterstützt sowohl Multicast-Anwendungen als auch Multicast-Routing. Multicast-Anwendungen benötigen keine spezielle Konfiguration, um auf FreeBSD lauffähig zu sein. Damit Multicast-Routing unterstützt wird, muss die folgende Option in der Kernelkonfiguration aktiviert

werden:

```
options MROUTING
```

Der Multicast-Routing-Daemon mrouted kann als Port oder Paket net/mroute installiert werden. Dieser Daemon implementiert das DVMRP Multicast-Routing-Protokoll. Um die Tunnel und DVMRP einzurichten, muss /usr/local/etc/mrouted.conf bearbeitet werden. Bei der Installation von mrouted wird auch map-mbone und mrinfo sowie die zugehörigen Manualpages installiert, in denen Sie auch Konfigurationsbeispiele finden können.



DVMRP wurde in vielen Multicast-Installationen weitgehend durch das PIM-Protokoll ersetzt. Weitere Informationen finden Sie in pim(4).

# 54.3. Drahtlose Netzwerke

## 54.3.1. Grundlagen

Die meisten drahtlosen Netzwerke basieren auf dem Standard IEEE® 802.11. Ein einfaches drahtloses Netzwerk besteht aus Stationen, die im 2,4 GHz- oder im 5 GHz-Band miteinander kommunizieren. Es ist aber auch möglich, dass regional andere Frequenzen, beispielsweise im 2,3 GHz- oder 4,9 GHz-Band, verwendet werden.

802.11-Netzwerke können auf zwei verschiedene Arten aufgebaut sein: Im *Infrastruktur-Modus* agiert eine Station als Master, mit dem sich alle anderen Stationen verbinden. Die Summe aller Stationen wird als Basic Service Set (BSS), die Master-Station hingegen als Access Point (AP) bezeichnet. In einem BSS läuft jedwede Kommunikation über den Access Point. Die zweite Form drahtloser Netzwerke sind die sogenannten *Ad-hoc-Netzwerke* (auch als IBSS bezeichnet), in denen es keinen Access Point gibt und in denen die Stationen direkt miteinander kommunizieren.

Die ersten 802.11-Netzwerke arbeiteten im 2,4 GHz-Band und nutzten dazu Protokolle der IEEE®-Standards 802.11 sowie 802.11b. Diese Standards legen unter anderem Betriebsfrequenzen sowie Merkmale des MAC-Layers (wie Frames und Transmissionsraten) fest. Später kam der Standard 802.11a hinzu, der im 5 GHz-Band, im Gegensatz zu den ersten beiden Standards aber mit unterschiedlichen Signalmechanismen und höheren Transmissionsraten arbeitet. Der neueste Standard 802.11g implementiert die Signal- und Transmissionsmechanismen von 802.11a im 2,4 GHz-Band, ist dabei aber abwärtskompatibel zu 802.11b-Netzwerken.

Unabhängig von den zugrundeliegenden Transportmechanismen verfügen 802.11-Netzwerke über diverse Sicherheitsmechanismen. Der ursprüngliche 802.11-Standard definierte lediglich ein einfaches Sicherheitsprotokoll namens WEP. Dieses Protokoll verwendet einen fixen, gemeinsam verwendeten Schlüssel sowie die RC4-Kryptografie-Chiffre, um Daten verschlüsselt über das drahtlose Netzwerk zu senden. Alle Stationen des Netzwerks müssen sich auf den gleichen fixen Schlüssel einigen, um miteinander kommunizieren zu können. Dieses Schema ist sehr leicht zu knacken und wird deshalb heute kaum mehr eingesetzt. Aktuelle Sicherheitsmechanismen bauen auf dem Standard IEEE® 802.11i auf, der neue kryptographische Schlüssel (Chiffren), ein neues Protokoll für die Anmeldung von Stationen an einem Access Point, sowie Mechanismen zum Austausch von Schlüsseln als Vorbereitung der Kommunikation zwischen verschiedenen Geräten

festlegt. Kryptografische Schlüssel werden in regelmäßigen Abständen aktualisiert. Außerdem gibt es Mechanismen zur Feststellung und Prävention von Einbruchsversuchen. Ein weiteres häufig verwendetes Sicherheitsprotokoll ist WPA. Dabei handelt es sich um einen Vorläufer von 802.11i, der von einem Industriekonsortium als Zwischenlösung bis zur endgültigen Verabschiedung von 802.11i entwickelt wurde. WPA definiert eine Untergruppe der Anforderungen des 802.11i-Standards und ist für den Einsatz in älterer Hardware vorgesehen. WPA benötigt nur den TKIP-Chiffre, welcher auf dem ursprünglichen WEP-Code basiert. 802.11i erlaubt zwar auch die Verwendung von TKIP, benötigt aber zusätzlich eine stärkere Chiffre (AES-CCM) für die Datenverschlüsselung. AES war für WPA nicht vorgesehen, weil man es als zu rechenintensiv für den Einsatz in älteren Geräten ansah.

Ein weiterer zu beachtender Standard ist 802.11e. Dieser definiert Protokolle zur Übertragung von Multimedia-Anwendungen, wie das Streaming von Videodateien oder Voice-over-IP (VoIP) in einem 802.11-Netzwerk. Analog zu 802.11i verfügt auch 802.11e über eine vorläufige Spezifikation namens WMM (ursprünglich WME), die von einem Industriekonsortium als Untergruppe von 802.11e spezifiziert wurde, um Multimedia-Anwendungen bereits vor der endgültigen Verabschiedung des 802.11e-Standards implementieren zu können. 802.11e sowie WME/WMM erlauben eine Prioritätenvergabe beim Datentransfer in einem drahtlosen Netzwerk. Möglich wird dies durch den Einsatz von Quality of Service-Protokollen (QoS) und erweiterten Medienzugriffsprotokollen. Werden diese Protokolle korrekt implementiert, erlauben sie hohe Datenübertragungsraten und einen priorisierten Datenfluss.

FreeBSD unterstützt die Standards 802.11a, 802.11b und 802.11g. Ebenfalls unterstützt werden WPA sowie die Sicherheitsprotokolle gemäß 802.11i (sowohl für 11a, 11b als auch 11g). QoS und Verkehrspriorisierung, die von den WME/WMM-Protokollen benötigt werden, werden für einen begrenzten Satz von drahtlosen Geräten unterstützt.

### 54.3.2. Schnellstartanleitung

Häufig soll ein Computer an ein vorhandenes Drahtlosnetzwerk angeschlossen werden. Diese Prozedur zeigt die dazu erforderlichen Schritte.

1. Besorgen Sie sich vom Netzwerkadministrator die SSID (Service Set Identifier) und den PSK (Pre Shared Key) für das Drahtlosnetzwerk.
2. Ermitteln Sie den drahtlosen Adapter. Der GENERIC-Kernel von FreeBSD enthält Treiber für viele gängige Adapter. Wenn der drahtlose Adapter eines dieser Modelle ist, wird das in der Ausgabe von ifconfig(8) angezeigt:

   ```
   % ifconfig | grep -B3 -i wireless
   ```

   In FreeBSD 11 und neueren Versionen verwenden Sie stattdessen diesen Befehl:

   ```
   % sysctl net.wlan.devices
   ```

   Wenn der drahtlose Adapter nicht aufgeführt wird, könnte ein zusätzliches Kernelmodul erforderlich sein. Es besteht jedoch auch die Möglichkeit, dass der Adapter von FreeBSD nicht

unterstützt wird.

Dieses Beispiel verwendet einen drahtlosen Atheros-Adapter `ath0`.

3. Fügen Sie in /etc/wpa_supplicant.conf einen Eintrag für das Netzwerk hinzu. Wenn die Datei nicht existiert, müssen Sie diese erstellen. Ersetzen Sie *myssid* und *psk* durch die SSID und den PSK. Diese Informationen werden vom Netzwerkadministrator zur Verfügung gestellt.

```
network={
    ssid="myssid"
    psk="mypsk"
}
```

4. Fügen Sie die entsprechenden Einträge in /etc/rc.conf ein, um das Netzwerk beim Start zu konfigurieren:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA SYNCDHCP"
```

5. Starten Sie den Computer oder den Netzwerkdienst neu, um sich mit dem Netzwerk zu verbinden:

```
# service netif restart
```

### 54.3.3. Basiskonfiguration

#### Kernelkonfiguration

Um ein drahtloses Netzwerk zu nutzen, wird eine drahtlose Netzwerkkarte benötigt und ein Kernel, der drahtlose Netzwerke unterstützt. Der Kernel unterstützt den Einsatz von Kernelmodulen. Daher muss nur die Unterstützung für die verwendeten Geräte aktiviert werden.

Die meisten drahtlosen Geräte verwenden Bauteile von Atheros und werden deshalb vom ath(4) -Treiber unterstützt. Um diesen Treiber zu verwenden, muss die folgende Zeile in /boot/loader.conf hinzugefügt werden:

```
if_ath_load="YES"
```

Der Atheros-Treiber besteht aus drei Teilen: dem Treiber selbst (ath(4)), dem Hardware-Support-Layer für die chip-spezifischen Funktionen (ath_hal(4)) sowie einem Algorithmus zur Auswahl der Frame-Übertragungsrate (ath_rate_sample). Wenn diese Unterstützung als Kernelmodul geladen wird, kümmert sich das Modul automatisch um Abhängigkeiten. Um die Unterstützung für ein anderes drahtloses Gerät zu laden, geben Sie das entsprechende Modul für dieses Gerät an. Dieses Beispiel zeigt die Verwendung von Geräten, die auf Bauteilen von Intersil Prism basieren und den Treiber wi(4) benötigen:

```
if_wi_load="YES"
```



Die Beispiele in diesem Abschnitt verwenden den ath(4)-Treiber. Verwenden Sie ein anderes Gerät, muss der Gerätename an die Konfiguration angepasst werden. Eine Liste aller verfügbaren Treiber und unterstützten drahtlosen Geräte finden sich in den FreeBSD Hardware Notes unter Release Information der FreeBSD Homepage. Gibt es keinen nativen FreeBSD-Treiber für das drahtlose Gerät, kann möglicherweise mit NDIS ein Windows®-Treiber verwendet werden.

Zusätzlich müssen die Module zur Verschlüsselung des drahtlosen Netzwerks geladen werden. Diese werden normalerweise dynamisch vom wlan(4)-Modul geladen. Im folgenden Beispiel erfolgt allerdings eine manuelle Konfiguration. Folgende Module sind verfügbar: wlan_wep(4), wlan_ccmp(4) und wlan_tkip(4). Sowohl wlan_ccmp(4) als auch wlan_tkip(4) werden nur benötigt, wenn WPA und/oder die Sicherheitsprotokolle von 802.11i verwendet werden. Wenn das Netzwerk keine Verschlüsselung verwendet, wird die wlan_wep(4)-Unterstützung nicht benötigt. Um diese Module beim Systemstart zu laden, fügen Sie folgende Zeilen in /boot/loader.conf ein:

```
wlan_wep_load="YES"
wlan_ccmp_load="YES"
wlan_tkip_load="YES"
```

Sobald diese Einträge in /boot/loader.conf vorhanden sind, muss das FreeBSD-System neu gestartet werden. Alternativ können die Kernelmodule auch manuell mit kldload(8) geladen werden.



Benutzer, die keine Kernelmodule verwenden wollen, können die benötigten Treiber auch in den Kernel kompilieren. Dazu müssen die folgenden Zeilen in die Kernelkonfigurationsdatei aufgenommen werden:

```
device wlan              # 802.11 support
device wlan_wep          # 802.11 WEP support
device wlan_ccmp         # 802.11 CCMP support
device wlan_tkip         # 802.11 TKIP support
device wlan_amrr         # AMRR transmit rate control algorithm
device ath               # Atheros pci/cardbus NIC's
device ath_hal           # pci/cardbus chip support
options AH_SUPPORT_AR5416 # enable AR5416 tx/rx descriptors
device ath_rate_sample   # SampleRate tx rate control for ath
```

Mit diesen Informationen in der Kernelkonfigurationsdatei kann der Kernel neu gebaut, und das FreeBSD-System anschließend neu gestartet werden.

Informationen über das drahtlose Gerät sollten in den Boot-Meldungen folgendermaßen angezeigt werden:

```
ath0: <Atheros 5212> mem 0x88000000-0x8800ffff irq 11 at device 0.0 on cardbus1
```

```
ath0: [ITHREAD]
ath0: AR2413 mac 7.9 RF2413 phy 4.5
```

#### Konfiguration der entsprechenden Region

Da die rechtliche Situation in verschiedenen Teilen der Welt unterschiedlich ist, ist es notwendig, die für Ihre Region geltenden Domänen korrekt einzustellen, um die richtigen Informationen darüber zu erhalten, welche Kanäle benutzt werden können.

Die verfügbaren Definitionen der Regionen finden Sie in /etc/regdomain.xml. Um die Daten zur Laufzeit einzustellen, benutzen Sie `ifconfig`:

```
# ifconfig wlan0 regdomain ETSI country AT
```

Um die Einstellungen beizubehalten, fügen Sie folgende Zeile in /etc/rc.conf hinzu:

```
# sysrc create_args_wlan0="country AT regdomain ETSI"
```

### 54.3.4. Infrastruktur-Modus

Drahtlose Netzwerke werden in der Regel im Infrastruktur-Modus (BSS) betrieben. Dazu werden mehrere drahtlose Access Points zu einem gemeinsamen drahtlosen Netzwerk verbunden. Jedes dieser drahtlosen Netzwerke hat einen eigenen Namen, der als >SSID> bezeichnet wird. Alle Clients eines drahtlosen Netzwerks verbinden sich in diesem Modus mit einem Access Point.

#### FreeBSD-Clients

##### Einen Access Point finden

Um nach verfügbaren drahtlosen Netzwerken zu suchen verwenden Sie ifconfig(8). Dieser Scanvorgang kann einen Moment dauern, da jede verfügbare Frequenz auf verfügbare Access Points hin überprüft werden muss. Nur der Super-User kann einen Scanvorgang starten:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
dlinkap         00:13:46:49:41:76   11   54M -90:96   100 EPS  WPA WME
freebsdap       00:11:95:c3:0d:ac    1   54M -83:96   100 EPS  WPA
```



Die Netzwerkkarte muss in den Status `up` versetzt werden, bevor der erste Scanvorgang gestartet werden kann. Für spätere Scans ist dies aber nicht mehr erforderlich.

Als Ergebnis erhalten Sie eine Liste mit allen gefundenen BSS/IBSS-Netzwerken. Zusätzlich zum Namen des Netzwerks, der `SSID`, wird auch die `BSSID` ausgegeben. Dabei handelt es sich um die MAC-Adresse des Access Points. Das Feld `CAPS` gibt den Typ des Netzwerks sowie die Fähigkeiten der

Stationen innerhalb des Netzwerks an:

*Tabelle 31. Station Capability Codes*

| Capability Code | Bedeutung |
|---|---|
| E | Extended Service Set (ESS). Zeigt an, dass die Station Teil eines Infrastruktur-Netzwerks ist, und nicht eines IBSS/Ad-hoc-Netzwerks. |
| I | IBSS/Ad-hoc-Netzwerk. Die Station ist Teil eines Ad-hoc-Netzwerks und nicht eines ESS-Netzwerks. |
| P | Privacy. Alle Datenframes, die innerhalb des BSS ausgetauscht werden, sind verschlüsselt. Dieses BSS verwendet dazu kryptographische Verfahren wie WEP, TKIP oder AES-CCMP. |
| S | Short Preamble. Das Netzwerk verwendet eine kurze Präambel (definiert in 802.11b High Rate/DSSS PHY). Eine kurze Präambel verwendet ein 56 Bit langes Sync-Feld, im Gegensatz zu einer langen Präambel, die ein 128 Bit langes Sync-Feld verwendet. |
| s | Short slot time. Das 802.11g-Netzwerk verwendet eine kurze Slotzeit, da es in diesem Netzwerk keine veralteten (802.11b) Geräte gibt. |

Um eine Liste der bekannten Netzwerke auszugeben, verwenden Sie den folgenden Befehl:

```
# ifconfig wlan0 list scan
```

Diese Liste kann entweder automatisch durch das drahtlose Gerät oder manuell durch eine `scan` -Aufforderung aktualisiert werden. Veraltete Informationen werden dabei automatisch entfernt.

#### Basiseinstellungen

Dieser Abschnitt beschreibt, wie Sie eine drahtlose Netzwerkkarte ohne Verschlüsselung unter FreeBSD einrichten. Nachdem Sie sich mit den Informationen dieses Abschnitts vertraut gemacht haben, sollten Sie das drahtlose Netzwerk mit WPA verschlüsseln.

Das Einrichten eines drahtlosen Netzwerks erfolgt in drei Schritten: Der Auswahl eines Access Points, die Anmeldung der Station sowie der Konfiguration der IP-Adresse.

### Einen Access Point auswählen

Im Normalfall wird sich die Station automatisch mit einem der zur Verfügung stehenden Access Points verbinden. Dazu muss lediglich das drahtlose Gerät aktiviert, oder in /etc/rc.conf eingetragen sein:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

Stehen mehrere Access Points zur Verfügung, kann ein spezifischer durch Angabe der SSID gewählt werden:

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid Ihre_SSID DHCP"
```

Gibt es in einem Netzwerk mehrere Access Points mit der gleichen SSID, was das Routing vereinfacht, kann es notwendig sein, dass ein bestimmtes Gerät verbunden werden muss. Dazu muss lediglich die BSSID des Access Points angeben werden. Die Angabe der SSID ist hierbei nicht zwingend notwendig:

```
wlans_ath0="wlan0"
ifconfig_wlan0="ssid Ihre_SSID bssid xx:xx:xx:xx:xx:xx DHCP"
```

Es gibt noch weitere Möglichkeiten, den Zugriff auf bestimmte Access Point zu beschränken, beispielsweise durch die Begrenzung der Frequenzen, auf denen eine Station nach einem Access Point sucht. Sinnvoll ist ein solches Vorgehen beispielsweise, wenn das drahtlose Gerät in verschiedenen Frequenzbereichen arbeiten kann, da in diesem Fall das Prüfen aller Frequenzen sehr zeitintensiv sein kann. Um nur innerhalb eines bestimmten Frequenzbereichs nach einem Access Point zu suchen, verwenden Sie die Option `mode`:

```
wlans_ath0="wlan0"
ifconfig_wlan0="mode 11g ssid Ihre_SSID DHCP"
```

In diesem Beispiel sucht das drahtlose Gerät nur im 2,4 GHz-Band (802.11g), aber nicht innerhalb des 5 GHz-Bandes nach einem Access Point. Mit der Option `channel` kann eine bestimmte Frequenz vorgegeben werden, auf der gesucht werden soll. Die Option `chanlist` erlaubt die Angabe mehrerer erlaubter Frequenzen. Eine umfassende Beschreibung dieser Optionen finden Sie in ifconfig(8).

#### Authentifizierung

Sobald ein Access Point gefunden wurde, muss sich die Station am Access Point authentifizieren, bevor Daten übertragen werden können. Dazu gibt es verschiedene Möglichkeiten. Am häufigsten wird die sogenannte *offene Authentifizierung* verwendet. Dabei wird es jeder Station erlaubt, sich mit einem Netzwerk zu verbinden und Daten zu übertragen. Aus Sicherheitsgründen sollte diese Methode allerdings nur zu Testzwecken bei der erstmaligen Einrichtung eines drahtlosen Netzwerks verwendet werden. Andere Authentifizierungsmechanismen erfordern den Austausch kryptographischer Informationen, bevor sie die Übertragung von Daten erlauben. Dazu gehören der Austausch fixer (vorher vereinbarter) Schlüssel oder Kennwörter, sowie der Einsatz komplexerer Verfahren mit Backend-Diensten wie RADIUS. Die offene Authentifizierung ist die Voreinstellung. Am zweithäufigsten kommt das im WPA-PSK beschriebene WPA-PSK zum Einsatz, welches auch als WPA Personal bezeichnet wird.



Kommt eine Apple® AirPort® Extreme-Basisstation als Access Point zum Einsatz, muss sowohl die Shared-Key-Authentifizierung als auch ein WEP-Schlüssel konfiguriert werden. Die entsprechende Konfiguration erfolgt entweder in /etc/rc.conf oder über das Programm wpa_supplicant(8). Für eine einzelne AirPort®-Basisstation kann der Zugriff wie folgt konfiguriert werden:

```
wlans_ath0="wlan0"
ifconfig_wlan0="authmode shared wepmode on weptxkey 1 wepkey 01234567
DHCP"
```

Normalerweise sollte Shared-Key-Authentifizierung nicht verwendet werden, da diese die Sicherheit des WEP-Schlüssel noch weiter verringert. Wenn WEP für Kompatibilität mit älteren Geräten verwendet werden muss, ist es besser, WEP mit offener Authentifizierung zu verwenden. Weitere Informationen zu WEP finden Sie im WEP.

**Eine IP-Adresse über DHCP beziehen**

Sobald ein Access Point ausgewählt ist und die Authentifizierungsparameter festgelegt sind, wird eine IP-Adresse benötigt. In der Regel wird die IP-Adresse über DHCP bezogen. Um dies zu erreichen, bearbeiten Sie /etc/rc.conf und fügen Sie DHCP für das drahtlose Gerät in die Konfiguration hinzu:

```
wlans_ath0="wlan0"
ifconfig_wlan0="DHCP"
```

Das drahtlose Gerät kann nun gestartet werden:

```
# service netif start
```

Nachdem das Gerät aktiviert wurde, kann mit ifconfig(8) der Status des Geräts ath0 abgefragt werden:

```
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        ether 00:11:95:d5:43:62
        inet 192.168.1.100 netmask 0xffffff00 broadcast 192.168.1.255
        media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
        status: associated
        ssid dlinkap channel 11 (2462 Mhz 11g) bssid 00:13:46:49:41:76
        country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
        scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
        roam:rate 5 protmode CTS wme burst
```

`status: associated` besagt, dass sich das Gerät mit dem drahtlosen Netzwerk verbunden hat. `bssid`

`00:13:46:49:41:76` ist die MAC-Adresse des Access Points und `authmode OPEN` zeigt an, dass die Kommunikation nicht verschlüsselt wird.

### Statische IP-Adressen

Wenn eine IP-Adresse nicht von einem DHCP-Server bezogen werden kann, vergeben Sie eine statische IP-Adresse. Ersetzten Sie dazu das oben gezeigte Schlüsselwort `DHCP` durch die entsprechende IP-Adresse. Beachten Sie dabei, dass Sie die anderen Konfigurationsparameter nicht versehentlich verändern:

```
wlans_ath0="wlan0"
ifconfig_wlan0="inet 192.168.1.100 netmask 255.255.255.0 ssid your_ssid_here"
```

#### WPA

Wi-Fi Protected Access (WPA) ist ein Sicherheitsprotokoll, das in 802.11-Netzwerken verwendet wird, um die fehlende Authentifizierung und Schwächen von WEP zu vermeiden. WPA stellt das aktuelle 802.1X-Authentifizierungsprotokoll dar und verwendet eine von mehreren Chiffren, um die Datensicherheit zu gewährleisten. Die einzige Chiffre, die von WPA verlangt wird, ist Temporary Key Integrity Protocol (TKIP). TKIP ist eine Chiffre, die die von WEP verwendete RC4-Chiffre um Funktionen zur Prüfung der Datenintegrität und zur Erkennung und Bekämpfung von Einbruchsversuchen erweitert. TKIP ist durch Softwaremodifikationen auch unter veralteter Hardware lauffähig. Im Vergleich zu WEP ist WPA zwar sehr viel sicherer, es ist aber dennoch nicht völlig immun gegen Angriffe. WPA definiert mit AES-CCMP noch eine weitere Chiffre als Alternative zu TKIP. AES-CCMP, welches häufig als WPA2 oder RSN bezeichnet wird, sollte bevorzugt eingesetzt werden.

WPA definiert Authentifizierungs- und Verschlüsselungsprotokolle. Die Authentifizierung erfolgt in der Regel über eine der folgenden Techniken: 802.1X gemeinsam mit einem Backend-Authentifizierungsdienst wie RADIUS, oder durch einen Minimal-Handshake zwischen der Station und dem Access Point mit einem vorher vereinbarten gemeinsamen Schlüssel. Die erste Technik wird als WPA Enterprise, die zweite hingegen als WPA Personal bezeichnet. Da sich der Aufwand für das Aufsetzen eines RADIUS-Backend-Servers für die meisten drahtlosen Netzwerke nicht lohnt, wird WPA in der Regel als WPA-PSK konfiguriert.

Die Kontrolle der drahtlosen Verbindung sowie das Aushandeln des Schlüssel, oder die Authentifizierung mit einem Server, erfolgt über wpa_supplicant(8). Dieses Programm benötigt eine Konfigurationsdatei, /etc/wpa_supplicant.conf. Weitere Informationen finden Sie in wpa_supplicant.conf(5).

#### WPA-PSK

WPA-PSK, das auch als WPA-Personal bekannt ist, basiert auf einem gemeinsamen, vorher vereinbarten Schlüssel (PSK), der aus einem Passwort generiert und danach als Master-Key des drahtlosen Netzwerks verwendet wird. Jeder Benutzer des drahtlosen Netzwerks verwendet daher *den gleichen* Schlüssel. WPA-PSK sollte nur in kleinen Netzwerken eingesetzt werden, in denen die Konfiguration eines Authentifizierungsservers nicht möglich oder erwünscht ist.



Achten Sie darauf, immer starke Passwörter zu verwenden, die ausreichend lang sind und auch Sonderzeichen enthalten, damit diese nicht leicht erraten oder umgangen werden können.

Der erste Schritt zum Einsatz von WPA-PSK ist die Konfiguration der SSID und des gemeinsamen Schlüssels des Netzwerks in /etc/wpa_supplicant.conf:

```
network={
  ssid="freebsdap"
  psk="freebsdmall"
}
```

Danach wird in /etc/rc.conf definiert, dass WPA zur Verschlüsselung eingesetzt werden soll und dass die IP-Adresse über DHCP bezogen wird:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Nun kann das drahtlose Gerät aktiviert werden:

```
# service netif start
Starting wpa_supplicant.
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 5
DHCPDISCOVER on wlan0 to 255.255.255.255 port 67 interval 6
DHCPOFFER from 192.168.0.1
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Alternativ kann das drahtlose Gerät manuell, mit Hilfe der Informationen aus /etc/wpa_supplicant.conf konfiguriert werden:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:11:95:c3:0d:ac (SSID='freebsdap' freq=2412 MHz)
Associated with 00:11:95:c3:0d:ac
WPA: Key negotiation completed with 00:11:95:c3:0d:ac [PTK=CCMP GTK=CCMP]
```

```
CTRL-EVENT-CONNECTED - Connection to 00:11:95:c3:0d:ac completed (auth) [id=0 id_str=]
```

Im zweiten Schritt starten Sie nun dhclient(8), um eine IP-Adresse vom DHCP-Server zu beziehen:

```
# dhclient wlan0
DHCPREQUEST on wlan0 to 255.255.255.255 port 67
DHCPACK from 192.168.0.1
bound to 192.168.0.254 -- renewal in 300 seconds.
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```



Enthält /etc/rc.conf bereits die Zeile `ifconfig_wlan0="DHCP"`, wird dhclient(8) automatisch gestartet, nachdem wpa_supplicant(8) sich mit dem Access Point verbunden hat.

Sollte der Einsatz von DHCP nicht möglich oder nicht gewünscht sein, konfigurieren Sie eine statische IP-Adresse, nachdem wpa_supplicant(8) die Station authentifiziert hat:

```
# ifconfig wlan0 inet 192.168.0.100 netmask 255.255.255.0
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.100 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/36Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Falls DHCP nicht verwendet wird, müssen zusätzlich noch das Standard-Gateway sowie der Nameserver manuell festgelegt werden:

```
# route add default your_default_router
# echo "nameserver your_DNS_server" >> /etc/resolv.conf
```

### WPA und EAP-TLS

Die zweite Möglichkeit, WPA einzusetzen, ist die Verwendung eines 802.1X-Backend-Authentifizierungsservers. Diese Variante wird als WPA-Enterprise bezeichnet, um sie vom weniger sicheren WPA-Personal abzugrenzen. Die bei WPA-Enterprise verwendete Authentifizierung basiert auf dem Extensible Authentication Protocol (EAP).

EAP selbst bietet keine Verschlüsselung, sondern operiert in einem verschlüsselten Tunnel. Es gibt verschiedene auf EAP basierende Authentifizierungsmethoden, darunter EAP-TLS, EAP-TTLS und EAP-PEAP.

EAP mit Transport Layers Security (EAP-TLS) ist ein sehr gut unterstütztes Authentifizierungsprotokoll, da es sich dabei um die erste EAP-Methode handelt, die von der Wi-Fi Alliance zertifiziert wurde. EAP-TLS erfordert drei Zertifikate: Das auf allen Rechnern installierte CA-Zertifikat, das Server-Zertifikat des Authentifizierungsservers, sowie ein Client-Zertifikat für jeden drahtlosen Client. Sowohl der Authentifizierungsservers als auch die drahtlosen Clients authentifizieren sich gegenseitig über Zertifikate, wobei sie überprüfen, ob diese Zertifikate auch von der Zertifizierungs-Authorität (CA) des jeweiligen Unternehmens signiert wurden.

Die Konfiguration erfolgt (analog zu WPA-PSK) über /etc/wpa_supplicant.conf:

```
network={
  ssid="freebsdap" ①
  proto=RSN  ②
  key_mgmt=WPA-EAP ③
  eap=TLS ④
  identity="loader" ⑤
  ca_cert="/etc/certs/cacert.pem" ⑥
  client_cert="/etc/certs/clientcert.pem" ⑦
  private_key="/etc/certs/clientkey.pem" ⑧
  private_key_passwd="freebsdmallclient" ⑨
}
```

① Der Name des Netzwerks (SSID).

② Das als WPA2 bekannte RSN IEEE® 802.11i Protokoll wird verwendet.

③ Die `key_mgmt`-Zeile bezieht sich auf das verwendete Key-Management-Protokoll. In diesem Beispiel wird WPA gemeinsam mit der EAP-Authentifizierung verwendet.

④ Die für die Verbindung verwendete EAP-Methode.

⑤ Das `identity`-Feld enthält den von EAP verwendeten Identifizierungsstring.

⑥ Das Feld `ca_cert` gibt den Pfad zum CA-Zertifikat an. Diese Datei wird zur Verifizierung des Server-Zertifikats benötigt.

⑦ Die `client_cert`-Zeile gibt den Pfad zum Client-Zertifikat an. Jeder Client hat ein eigenes, innerhalb des Netzwerks eindeutiges, Zertifikat.

⑧ Das Feld `private_key` gibt den Pfad zum privaten Schlüssel des Client-Zertifikat an.

⑨ Das Feld `private_key_passwd` enthält die Passphrase für den privaten Schlüssel.

Danach fügen Sie die folgende Zeile in /etc/rc.conf ein:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Nun können Sie das drahtlose Gerät aktivieren:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

Alternativ kann das drahtlose Gerät manuell mit wpa_supplicant(8) und ifconfig(8) aktiviert werden.

#### WPA mit EAP-TTLS

Bei EAP-TLS müssen sowohl der Authentifizierungsserver als auch die Clients jeweils ein eigenes Zertifikat aufweisen. Bei EAP-TTLS ist das Client-Zertifikat optional. EAP-TTLS geht dabei vor wie ein Webserver, der einen sicheren SSL-Tunnel erzeugen kann, ohne dass der Besucher dabei über ein clientseitiges Zertifikat verfügen muss. EAP-TTLS verwendet einen verschlüsselten TLS-Tunnel zum sicheren Transport der Authentifizierungsdaten.

Die erforderliche Konfiguration erfolgt in /etc/wpa_supplicant.conf:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=TTLS ①
  identity="test" ②
  password="test" ③
  ca_cert="/etc/certs/cacert.pem" ④
  phase2="auth=MD5" ⑤
}
```

① Die für die Verbindung verwendete EAP-Methode.

② Das `identity`-Feld enthält den Identifizierungsstring für die EAP-Authentifizierung innerhalb des verschlüsselten TLS-Tunnels.

③ Das `password`-Feld enthält die Passphrase für die EAP-Authentifizierung.

④ Das Feld `ca_cert` gibt den Pfad zum CA-Zertifikat an. Diese Datei wird zur Verifizierung des Server-Zertifikats benötigt.

⑤ Die innerhalb des verschlüsselten TLS-Tunnels verwendete Authentifizierungsmethode. In Fall von PEAP ist dies `auth=MSCHAPV2`.

Folgende Zeilen müssen in /etc/rc.conf aufgenommen werden:

```
wlans_ath0="wlan0"
ifconfig_wlan0="WPA DHCP"
```

Nun kann das drahtlose Gerät aktiviert werden:

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
      AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
      bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
      wme burst roaming MANUAL
```

#### WPA mit EAP-PEAP



PEAPv0/EAP-MSCHAPv2 ist die gängigste PEAP-Methode. In diesem Kapitel wird der Begriff PEAP stellvertretend für diese Methode verwendet.

Protected EAP (PEAP) wurde als Alternative zu EAP-TTLS entwickelt und ist nach EAP-TLS der meist genutzte EAP-Standard. In einem Netzwerk mit verschiedenen Betriebssystemen sollte PEAP das am besten unterstützte Standard nach EAP-TLS sein.

PEAP arbeitet ähnlich wie EAP-TTLS. Es verwendet ein serverseitiges Zertifikat, um einen verschlüsselten TLS-Tunnel, über den die sichere Authentifizierung zwischen den Clients und dem Authentifizierungsserver erfolgt. In Sachen Sicherheit unterscheiden sich EAP-TTLS und PEAP

allerdings: PEAP überträgt den Benutzernamen im Klartext und verschlüsselt nur das Passwort, während EAP-TTLS sowohl den Benutzernamen, als auch das Passwort über den TLS-Tunnel überträgt.

Um EAP-PEAP zu konfigurieren, fügen Sie die folgenden Zeilen in /etc/wpa_supplicant.conf ein:

```
network={
  ssid="freebsdap"
  proto=RSN
  key_mgmt=WPA-EAP
  eap=PEAP ①
  identity="test" ②
  password="test" ③
  ca_cert="/etc/certs/cacert.pem" ④
  phase1="peaplabel=0"  ⑤
  phase2="auth=MSCHAPV2" ⑥
}
```

① Die für die Verbindung verwendete EAP-Methode.

② Das `identity`-Feld enthält den Identifizierungsstring für die innerhalb des verschlüsselten TLS-Tunnels erfolgende EAP-Authentifizierung.

③ Das Feld `password` enthält die Passphrase für die EAP-Authentifizierung.

④ Das Feld `ca_cert` gibt den Pfad zum CA-Zertifikat an. Diese Datei wird zur Verifizierung des Server-Zertifikats benötigt.

⑤ Dieses Feld enthält die Parameter für die erste Phase der Authentifizierung, den TLS-Tunnel. Je nachdem, welcher Authentifizierungsserver benutzt wird, kann ein spezifisches Label für die Authentifizierung verwendet werden. Meistens lautet das Label "client EAP encryption", dass durch `peaplabel=0` gesetzt wird. Weitere Informationen finden Sie in wpa_supplicant.conf(5).

⑥ Das innerhalb des verschlüsselten TLS-Tunnels verwendete Authentifizierungsprotokoll. In unserem Beispiel handelt es sich dabei um `auth=MSCHAPV2`.

Danach fügen Sie die folgende Zeile in /etc/rc.conf ein:

```
ifconfig_ath0="WPA DHCP"
```

Nun kann das drahtlose Gerät aktiviert werden.

```
# service netif start
Starting wpa_supplicant.
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 7
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 15
DHCPREQUEST on wlan0 to 255.255.255.255 port 67 interval 21
DHCPACK from 192.168.0.20
bound to 192.168.0.254 -- renewal in 300 seconds.
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
      ether 00:11:95:d5:43:62
```

```
        inet 192.168.0.254 netmask 0xffffff00 broadcast 192.168.0.255
        media: IEEE 802.11 Wireless Ethernet DS/11Mbps mode 11g
        status: associated
        ssid freebsdap channel 1 (2412 MHz 11g) bssid 00:11:95:c3:0d:ac
        country US ecm authmode WPA2/802.11i privacy ON deftxkey UNDEF
        AES-CCM 3:128-bit txpower 21.5 bmiss 7 scanvalid 450 bgscan
        bgscanintvl 300 bgscanidle 250 roam:rssi 7 roam:rate 5 protmode CTS
        wme burst roaming MANUAL
```

#### WEP

Wired Equivalent Privacy (WEP) ist Teil des ursprünglichen 802.11-Standards. Es enthält keinen Authentifzierungsmechanismus und verfügt lediglich über eine schwache Zugriffskontrolle, die sehr leicht umgangen werden kann.

WEP kann über ifconfig(8) aktiviert werden:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 inet 192.168.1.100 netmask 255.255.255.0 \
            ssid my_net wepmode on weptxkey 3 wepkey 3:0x3456789012
```

- `weptxkey` definiert den WEP-Schlüssel, der für die Datenübertragung verwendet wird. Dieses Beispiel verwendet den dritten Schlüssel. Der gleiche Schlüssel muss auch am Access Point eingestellt sein. Kennen Sie den vom Access Point verwendeten Schlüssel nicht, sollten Sie zuerst den Wert `1` (den ersten Schlüssel) für diese Variable verwenden.
- `wepkey` legt den zu verwendenden WEP-Schlüssel in der Form *Nummer:Schlüssel* fest. Schlüssel `1` wird standardmäßig verwendet. Die "Nummer" muss nur angegeben werden, wenn ein anderer als der erste Schlüssel verwendet werden soll.



Ersetzen Sie `0x3456789012` durch den am Access Point konfigurierten Schlüssel.

Weitere Informationen finden Sie in ifconfig(8).

Das Programm wpa_supplicant(8) eignet sich ebenfalls dazu, WEP für drahtlose Geräte zu aktivieren. Obige Konfiguration lässt sich dabei durch die Aufnahme der folgenden Zeilen in /etc/wpa_supplicant.conf realisieren:

```
network={
  ssid="my_net"
  key_mgmt=NONE
  wep_key3=3456789012
  wep_tx_keyidx=3
}
```

Danach müssen Sie das Programm noch aufrufen:

```
# wpa_supplicant -i wlan0 -c /etc/wpa_supplicant.conf
Trying to associate with 00:13:46:49:41:76 (SSID='dlinkap' freq=2437 MHz)
Associated with 00:13:46:49:41:76
```

### 54.3.5. Ad-hoc-Modus

Der IBSS-Modus, der auch als Ad-hoc-Modus bezeichnet wird, ist für Punkt-zu-Punkt-Verbindungen vorgesehen. Um beispielsweise eine Ad-hoc-Verbindung zwischen den Rechnern A und B aufzubauen, werden lediglich zwei IP-Adressen und eine SSID benötigt.

Auf Rechner A:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
      status: running
      ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst
```

Der adhoc-Parameter zeigt an, dass die Schnittstelle im IBSS-Modus läuft.

Rechner B sollte nun in der Lage sein, Rechner A zu finden:

```
# ifconfig wlan0 create wlandev ath0 wlanmode adhoc
# ifconfig wlan0 up scan
  SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
  freebsdap       02:11:95:c3:0d:ac    2   54M -64:-96  100 IS   WME
```

Der Wert I (Spalte CAPS) in dieser Ausgabe bestätigt, dass sich Rechner A im Ad-hoc-Modus befindet. Nun müssen Sie noch Rechner B eine andere IP-Adresse zuweisen:

```
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <adhoc>
      status: running
      ssid freebsdap channel 2 (2417 Mhz 11g) bssid 02:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst
```

Damit sind die Rechner A und B bereit und können untereinander Daten austauschen.

### 54.3.6. FreeBSD Host Access Points

FreeBSD kann als Access Point (AP) agieren. Dies verhindert, dass man sich einen Hardware AP kaufen oder ein Ad-hoc Netzwerk laufen lassen muss. Dies kann sinnvoll sein, falls der FreeBSD-Computer als Gateway zu einem anderen Netzwerk, wie dem Internet, fungiert.

#### Grundeinstellungen

Bevor Sie einen FreeBSD-Computer als AP konfigurieren, muss der Kernel mit der entsprechenden Netzwerkunterstützung für die drahtlose Karte, sowie die Sicherheitsprotokolle konfiguriert werden. Weitere Informationen finden Sie im Basiskonfiguration.



Die Verwendung der NDIS Treiber für Windows® erlauben zur Zeit keinen AP-Modus. Nur die nativen FreeBSD-Wireless-Treiber unterstützen den AP-Modus.

Nachdem die Netzwerkunterstützung geladen ist, überprüfen Sie, ob das Wireless-Gerät den hostbasierenden Access-Point Modus, der auch als hostap-Modus bekannt ist, unterstützt:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 list caps
drivercaps=6f85edc1<STA,FF,TURBOP,IBSS,HOSTAP,AHDEMO,TXPMGT,SHSLOT,SHPREAMBLE,MONITOR,
MBSS,WPA1,WPA2,BURST,WME,WDS,BGSCAN,TXFRAG>
cryptocaps=1f<WEP,TKIP,AES,AES_CCM,TKIPMIC>
```

Diese Ausgabe zeigt die Eigenschaften der Karte. Das Wort HOSTAP bestätigt, dass diese Wireless-Karte als AP agieren kann. Die verschiedenen unterstützten Algorithmen werden ebenfalls angezeigt: WEP, TKIP und AES. Diese Informationen zeigen an, welche Sicherheitsprotokolle auf dem AP nutzbar sind.

Das Wireless-Gerät kann nur während der Erzeugung des Pseudo-Geräts in den hostap-Modus gesetzt werden. Zuvor erstellte Pseudo-Geräte müssen also vorher zerstört werden:

```
# ifconfig wlan0 destroy
```

Danach muss das Gerät erneut erstellt werden, bevor die restlichen Netzwerkparameter konfiguriert werden können:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g
channel 1
```

Benutzen Sie danach erneut ifconfig(8), um den Status der wlan0-Schnittstelle abzufragen:

```
# ifconfig wlan0
```

```
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
      status: running
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 scanvalid 60
      protmode CTS wme burst dtimperiod 1 -dfs
```

Die hostap-Parameter geben die Schnittstelle an, die im hostbasierenden Access Point Modus läuft.

Die Konfiguration der Schnittstelle kann durch Hinzufügen der folgenden Zeilen in die Datei /etc/rc.conf automatisch während des Bootvorganges erfolgen:

```
wlans_ath0="wlan0"
create_args_wlan0="wlanmode hostap"
ifconfig_wlan0="inet 192.168.0.1 netmask 255.255.255.0 ssid freebsdap mode 11g channel
1"
```

**Hostbasierender Access Point ohne Authentifizierung oder Verschlüsselung**

Obwohl es nicht empfohlen wird, einen AP ohne jegliche Authentifizierung oder Verschlüsselung laufen zu lassen, ist es eine einfache Art zu testen, ob der AP funktioniert. Diese Konfiguration ist auch wichtig für die Fehlersuche bei Client-Problemen.

Nachdem der AP konfiguriert wurde, ist es möglich von einem anderen drahtlosen Computer eine Suche nach dem AP zu starten:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID/MESH ID    BSSID              CHAN RATE   S:N     INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M -66:-96  100 ES   WME
```

Der Client-Rechner hat den AP gefunden und kann nun eine Verbindung aufbauen:

```
# ifconfig wlan0 inet 192.168.0.2 netmask 255.255.255.0 ssid freebsdap
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:d5:43:62
      inet 192.168.0.2 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet OFDM/54Mbps mode 11g
      status: associated
      ssid freebsdap channel 1 (2412 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy OFF txpower 21.5 bmiss 7
      scanvalid 60 bgscan bgscanintvl 300 bgscanidle 250 roam:rssi 7
      roam:rate 5 protmode CTS wme burst
```

### WPA2-hostbasierter Access Point

Dieser Abschnitt beschäftigt sich mit der Konfiguration eines FreeBSD Access Point mit dem WPA2-Sicherheitsprotokoll. Weitere Einzelheiten zu WPA und der Konfiguration von Clients mit WPA finden Sie im WPA.

Der hostapd(8)-Dienst wird genutzt, um die Client-Authentifizierung und das Schlüsselmanagement auf dem AP mit aktiviertem WPA2 zu nutzen.

Die folgende Konfiguration wird auf dem FreeBSD-Computer ausgeführt, der als AP agiert. Nachdem der AP korrekt arbeitet, sollte hostapd(8) automatisch beim Booten durch folgende Zeile in /etc/rc.conf aktiviert werden:

```
hostapd_enable="YES"
```

Bevor Sie versuchen hostapd(8) zu konfigurieren, konfigurieren Sie zunächst die Grundeinstellungen, wie im Grundeinstellungen beschrieben.

#### WPA2-PSK

WPA2-PSK ist für kleine Netzwerke gedacht, in denen die Verwendung eines Authentifizierungs-Backend-Server nicht möglich oder nicht erwünscht ist.

Die Konfiguration wird in /etc/hostapd.conf durchgeführt:

```
interface=wlan0                 ①
debug=1                         ②
ctrl_interface=/var/run/hostapd ③
ctrl_interface_group=wheel      ④
ssid=freebsdap                  ⑤
wpa=2                           ⑥
wpa_passphrase=freebsdmall      ⑦
wpa_key_mgmt=WPA-PSK            ⑧
wpa_pairwise=CCMP               ⑨
```

① Die Wireless-Schnittstelle, die für den Access Point verwendet wird an.

② Der debuglevel von hostapd(8) während der Ausführung. Ein Wert von 1 ist der kleinste zulässige Wert.

③ Der Pfadname des Verzeichnisses, der von hostapd(8) genutzt wird, um die Domain-Socket-Dateien zu speichern, die für die Kommunikation mit externen Programmen, wie z.B. hostapd_cli(8), benutzt werden. In diesem Beispiel wird der Standardwert verwendet.

④ Die Gruppe die Zugriff auf die Schnittstellendateien hat.

⑤ Der Name des drahtlosen Netzwerks (SSID).

⑥ Aktiviert WPA und gibt an welches WPA-Authentifizierungprotokoll benötigt wird. Ein Wert von 2 konfiguriert den AP mit WPA2. Setzen Sie den Wert nur auf 1, wenn Sie das veraltete WPA benötigen.

⑦ Das ASCII-Passwort für die WPA-Authentifizierung.

⑧ Das verwendete Schlüsselmanagement-Protokoll. Dieses Beispiel nutzt WPA-PSK.

⑨ Die zulässigen Verschlüsselungsverfahren des Access-Points. In diesem Beispiel wird nur CCMP (AES) akzeptiert. CCMP ist eine Alternative zu TKIP und sollte wenn möglich eingesetzt werden. TKIP sollte nur da eingesetzt werden, wo kein CCMP möglich ist.

Als nächstes wird hostapd gestartet:

```
# service hostapd forcestart
```

```
# ifconfig wlan0
wlan0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether 04:f0:21:16:8e:10
    inet6 fe80::6f0:21ff:fe16:8e10%wlan0 prefixlen 64 scopeid 0x9
    nd6 options=21<PERFORMNUD,AUTO_LINKLOCAL>
    media: IEEE 802.11 Wireless Ethernet autoselect mode 11na <hostap>
    status: running
    ssid No5ignal channel 36 (5180 MHz 11a ht/40+) bssid 04:f0:21:16:8e:10
    country US ecm authmode WPA2/802.11i privacy MIXED deftxkey 2
    AES-CCM 2:128-bit AES-CCM 3:128-bit txpower 17 mcastrate 6 mgmtrate 6
    scanvalid 60 ampdulimit 64k ampdudensity 8 shortgi wme burst
    dtimperiod 1 -dfs
    groups: wlan
```

Sobald der AP läuft, können sich die Clients mit ihm verbinden. Weitere Informationen finden Sie im WPA. Es ist möglich zu sehen, welche Stationen mit dem AP verbunden sind. Geben Sie dazu `ifconfig wlan0 list sta` ein.

#### WEP-hostbasierter Access Point

Es ist nicht empfehlenswert, einen AP mit WEP zu konfigurieren, da es keine Authentifikationsmechanismen gibt und WEP leicht zu knacken ist. Einige ältere drahtlose Karten unterstützen nur WEP als Sicherheitsprotokoll. Diese Karten können nur mit einem AP ohne Authentifikation oder Verschlüsselung genutzt werden.

Das Wireless-Gerät kann nun in den hostap-Modus versetzt werden und mit der korrekten SSID und IP-Adresse konfiguriert werden:

```
# ifconfig wlan0 create wlandev ath0 wlanmode hostap
# ifconfig wlan0 inet 192.168.0.1 netmask 255.255.255.0 \
  ssid freebsdap wepmode on weptxkey 3 wepkey 3:0x3456789012 mode 11g
```

- Der `weptxkey` zeigt an, welcher WEP-Schlüssel bei der Übertragung benutzt wird. In diesem Beispiel wird der dritte Schlüssel benutzt, da die Nummerierung bei `1` beginnt. Dieser Parameter muss angegeben werden, damit die Daten verschlüsselt werden.

- Der `wepkey` gibt den gewählten WEP-Schlüssel an. Er sollte im folgenden Format *index:key* vorliegen. Wenn kein Index vorhanden ist, wird der Schlüssel `1` benutzt. Ansonsten muss der Index manuell festgelegt werden.

Benutzen Sie ifconfig(8) um den Status der wlan0-Schnittstelle erneut anzuzeigen:

```
# ifconfig wlan0
  wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
      ether 00:11:95:c3:0d:ac
      inet 192.168.0.1 netmask 0xffffff00 broadcast 192.168.0.255
      media: IEEE 802.11 Wireless Ethernet autoselect mode 11g <hostap>
      status: running
      ssid freebsdap channel 4 (2427 Mhz 11g) bssid 00:11:95:c3:0d:ac
      country US ecm authmode OPEN privacy ON deftxkey 3 wepkey 3:40-bit
      txpower 21.5 scanvalid 60 protmode CTS wme burst dtimperiod 1 -dfs
```

Es ist möglich, von einem anderen drahtlosen Computer eine Suche nach dem AP zu starten:

```
# ifconfig wlan0 create wlandev ath0
# ifconfig wlan0 up scan
SSID            BSSID              CHAN RATE  S:N   INT CAPS
freebsdap       00:11:95:c3:0d:ac    1   54M 22:1   100 EPS
```

Der Client-Rechner hat den AP gefunden und kann nun eine Verbindung aufbauen. Weitere Informationen finden Sie im WEP.

### 54.3.7. Benutzung von drahtgebundenen und drahtlosen Verbindungen

Eine Verbindung per Kabel bietet eine bessere Leistung und eine höhere Zuverlässigkeit, während die Wireless-Verbindung eine höhere Flexibilität und Mobilität bietet. Benutzer von Laptops wollen normalerweise beides nutzen und zwischen beiden Verbindungen hin und her schalten.

Unter FreeBSD ist es möglich zwei oder mehr Netzwerkschnittstellen in einem "failover"-Mode zu kombinieren. Diese Konfiguration nutzt die beste verfügbare Verbindung aus einer Gruppe von Netzwerkverbindungen. Sobald sich der Linkstatus ändert, wechselt das Betriebssystem automatisch auf eine andere Verbindung.

Link-Aggregation und Failover werden im Link-Aggregation und Failover behandelt. Ein Beispiel für die Verwendung von kabelgebundenen und drahtlosen Verbindungen gibt es im Failover Modus zwischen Ethernet- und drahtlosen Schnittstellen.

### 54.3.8. Problembehandlung

Dieser Abschnitt beschreibt eine Reihe von Maßnahmen zur Behebung von alltäglichen Problemen mit Drahtlosnetzwerken.

- Wird der Access Point bei der Suche nicht gefunden, überprüfen Sie, dass die Konfiguration des drahtlosen Geräts nicht die Anzahl der Kanäle beschränkt.

- Wenn sich das Gerät nicht mit dem Access Point verbinden kann, überprüfen Sie, ob die Konfiguration der Station auch der des Access Points entspricht. Dazu gehören auch die Authentifzierungsmethode und die Sicherheitsprotokolle. Halten Sie die Konfiguration so einfach wie möglich. Wenn Sie ein Sicherheitsprotokoll wie WPA oder WEP verwenden, können Sie testweise den Access Point auf *offene Authentifizierung* und *keine Sicherheit* einstellen.

  Für die Fehlersuche steht wpa_supplicant(8) zur Verfügung. Starten Sie das Programm manuell mit der Option `-dd` und durchsuchen Sie anschließend die Systemprotokolle nach eventuellen Fehlermeldungen.

- Sobald sich das Gerät mit dem Access Point verbinden kann, prüfen Sie die Netzwerkkonfiguration mit einfachen Werkzeugen wie ping(8).
- Zusätzlich gibt es auch zahlreiche Low-Level-Debugging-Werkzeuge. Die Ausgabe von Debugging-Informationen des 802.11 Protocol Support Layers lassen sich mit dem Programm wlandebug(8) aktivieren. Um beispielsweise während der Suche nach Access Points und des Aufbaus von 802.11-Verbindungen (Handshake) auftretende Systemmeldungen auf die Konsole auszugeben, verwenden Sie den folgenden Befehl:

  ```
  # wlandebug -i wlan0 +scan+auth+debug+assoc
    net.wlan.0.debug: 0 => 0xc80000<assoc,auth,scan>
  ```

  Der 802.11-Layer liefert umfangreiche Statistiken, die mit dem Werkzeug `wlanstats`, das sich in /usr/src/tools/tools/net80211 befindet, abgerufen werden können. Diese Statistiken sollten alle Fehler identifizieren, die im 802.11-Layer auftreten. Beachten Sie aber, dass einige Fehler bereits im darunterliegenden Gerätetreiber auftreten und daher in diesen Statistiken nicht enthalten sind. Wie Sie Probleme des Gerätetreibers identifizieren, entnehmen Sie bitte der Dokumentation des Gerätetreibers.

Wenn die oben genannten Informationen nicht helfen das Problem zu klären, erstellen Sie einen Problembericht, der die Ausgabe der weiter oben genannten Werkzeuge beinhaltet.

## 54.4. USB Tethering

Viele Mobiltelefone bieten die Möglichkeit, ihre Datenverbindung über USB (oft "Tethering" genannt) zu teilen. Diese Funktion verwendet entweder das RNDIS-, CDC- oder ein Apple® iPhone®/iPad®-Protokoll.

- Android™-Geräte benutzen in der Regel den urndis(4)-Treiber.
- Apple®-Geräte benutzen den ipheth(4)-Treiber.
- Ältere Geräte benutzen oft den cdce(4)-Treiber.

Bevor Sie ein Gerät anschließen, laden Sie den entsprechenden Treiber in den Kernel:

```
# kldload if_urndis
# kldload if_cdce
# kldload if_ipheth
```

Sobald das Gerät angeschlossen ist, steht es unter ue*0* wie ein normales Netzwerkgerät zur Verfügung. Stellen Sie sicher, dass die Option "USB Tethering" auf dem Gerät aktiviert ist.

Um diese Änderungen dauerhaft zu speichern und den Treiber beim Booten als Modul zu laden, müssen die entsprechenden Zeilen in /boot/loader.conf konfiguriert werden:

```
if_urndis_load="YES"
if_cdce_load="YES"
if_ipteth_load="YES"
```

# 54.5. Bluetooth

Bluetooth ermöglicht die Bildung von persönlichen Netzwerken über drahtlose Verbindungen bei einer maximalen Reichweite von 10 Metern und operiert im unlizensierten 2,4-GHz-Band. Solche Netzwerke werden normalerweise spontan gebildet, wenn sich mobile Geräte, wie Mobiltelefone, Handhelds oder Notebooks miteinander verbinden. Im Gegensatz zu Wireless LAN ermöglicht Bluetooth auch höherwertige Dienste, wie FTP-ähnliche Dateiserver, Filepushing, Sprachübertragung, Emulation von seriellen Verbindungen und mehr.

Dieses Kapitel beschreibt die Verwendung von USB-Bluetooth-Adaptern in FreeBSD. Weiterhin werden verschiedene Bluetooth-Protokolle und Programme vorgestellt.

## 54.5.1. Die Bluetooth-Unterstützung aktivieren

Der Bluetooth-Stack von FreeBSD verwendet das netgraph(4)-Framework. Viele Bluetooth-USB-Adapter werden durch den ng_ubt(4)-Treiber unterstützt. Auf dem Chip BCM2033 von Broadcom basierende Bluetooth-Geräte werden von den Treibern ubtbcmfw(4) sowie ng_ubt(4) unterstützt. Die Bluetooth-PC-Card 3CRWB60-A von 3Com verwendet den ng_bt3c(4)-Treiber. Serielle sowie auf UART basierende Bluetooth-Geräte werden von sio(4), ng_h4(4) sowie hcseriald(8) unterstützt.

Bevor ein Gerät angeschlossen wird, muss der entsprechende Treiber in den Kernel geladen werden. Hier verwendet das Gerät den ng_ubt(4)-Treiber:

```
# kldload ng_ubt
```

Ist das Bluetooth-Gerät beim Systemstart angeschlossen, kann das entsprechende Modul bei Booten geladen werden, indem der entsprechende Treiber in /boot/loader.conf hinzugefügt wird:

```
ng_ubt_load="YES"
```

Sobald der Treiber geladen ist, schließen Sie den USB-Adapter an. Eine Meldung ähnlich der folgenden wird auf der Konsole und in /var/log/messages erscheinen:

```
ubt0: vendor 0x0a12 product 0x0001, rev 1.10/5.25, addr 2
ubt0: Interface 0 endpoints: interrupt=0x81, bulk-in=0x82, bulk-out=0x2
```

```
ubt0: Interface 1 (alt.config 5) endpoints: isoc-in=0x83, isoc-out=0x3,
      wMaxPacketSize=49, nframes=6, buffer size=294
```

Verwenden Sie das Startskript zum Starten und Beenden des Bluetooth-Stacks. Es ist empfehlenswert, den Bluetooth-Stack zu beenden, bevor Sie den Adapter entfernen. Das Starten des Bluetooth-Stacks kann das Starten von hcsecd(8) erfordern. Wenn Sie den Bluetooth-Stack starten, erhalten Sie eine Meldung ähnlich der folgenden:

```
# service bluetooth start ubt0
BD_ADDR: 00:02:72:00:d4:1a
Features: 0xff 0xff 0xf 00 00 00 00 00
<3-Slot> <5-Slot> <Encryption> <Slot offset>
<Timing accuracy> <Switch> <Hold mode> <Sniff mode>
<Park mode> <RSSI> <Channel quality> <SCO link>
<HV2 packets> <HV3 packets> <u-law log> <A-law log> <CVSD>
<Paging scheme> <Power control> <Transparent SCO data>
Max. ACL packet size: 192 bytes
Number of ACL packets: 8
Max. SCO packet size: 64 bytes
Number of SCO packets: 8
```

### 54.5.2. Suche nach anderen Bluetooth-Geräten

Das Host Controller Interface (HCI) bietet eine einheitliche Methode für den Zugriff auf Bluetooth-Basisband-Funktionen. In FreeBSD wird ein netgraph HCI-Knoten für jedes Bluetooth-Gerät erstellt. Weitere Einzelheiten finden Sie in ng_hci(4).

Eine der wichtigsten Aufgaben ist das Auffinden von sich in Reichweite befindenden Bluetooth-Geräten. Diese Funktion wird als *inquiry* bezeichnet. Inquiry sowie andere mit HCI in Verbindung stehende Funktionen werden von hccontrol(8) zur Verfügung gestellt. Das folgende Beispiel zeigt, wie man herausfindet, welche Bluetooth-Geräte sich in Reichweite befinden. Eine solche Abfrage dauert nur wenige Sekunden. Beachten Sie, dass ein Gerät nur dann antwortet, wenn es sich im Modus *discoverable* befindet.

```
% hccontrol -n ubt0hci inquiry
Inquiry result, num_responses=1
Inquiry result #0
       BD_ADDR: 00:80:37:29:19:a4
       Page Scan Rep. Mode: 0x1
       Page Scan Period Mode: 00
       Page Scan Mode: 00
       Class: 52:02:04
       Clock offset: 0x78ef
Inquiry complete. Status: No error [00]
```

BD_ADDR stellt, ähnlich der MAC-Adresse einer Netzwerkkarte, die eindeutige Adresse eines Bluetooth-Gerätes dar. Diese Adresse ist für die Kommunikation mit dem Gerät nötig. Es ist aber

auch möglich, BD_ADDR einen Klartextnamen zuzuweisen. /etc/bluetooth/hosts enthält Informationen über die bekannten Bluetooth-Rechner. Das folgende Beispiel zeigt, wie man den Klartextnamen eines entfernten Geräts in Erfahrung bringen kann:

```
% hccontrol -n ubt0hci remote_name_request 00:80:37:29:19:a4
BD_ADDR: 00:80:37:29:19:a4
Name: Pav's T39
```

Wenn Sie ein entferntes Bluetooth-Gerät abfragen, wird dieses den Rechner unter dem Namen "your.host.name (ubt0)" finden. Dieser Name kann aber jederzeit geändert werden.

Entfernten Geräten können Aliase in /etc/bluetooth/hosts zugewiesen werden. Weitere Informationen zu /etc/bluetooth/hosts finden Sie in bluetooth.hosts(5).

Bluetooth ermöglicht Punkt-zu-Punkt-Verbindungen an denen nur zwei Bluetooth-Geräte beteiligt sind, aber auch Punkt-zu-Multipunkt-Verbindungen, bei denen eine Verbindung von mehreren Bluetooth-Geräten gemeinsam genutzt wird. Das folgende Beispiel zeigt, wie man eine Verbindung zu einem entferntem Gerät aufbauen kann:

```
% hccontrol -n ubt0hci create_connection BT_ADDR
```

`create_connection` aktzeptiert `BT_ADDR` oder auch einen Alias aus /etc/bluetooth/hosts.

Das folgende Beispiel zeigt, wie man die aktiven Basisbandverbindungen des lokalen Gerätes anzeigen kann:

```
% hccontrol -n ubt0hci read_connection_list
Remote BD_ADDR    Handle Type Mode Role Encrypt Pending Queue State
00:80:37:29:19:a4     41  ACL    0 MAST    NONE       0     0 OPEN
```

Ein *connection handle* ist für die Beendigung einer Basisbandverbindung nützlich. Im Normalfall werden inaktive Verbindungen aber automatisch vom Bluetooth-Stack getrennt.

```
# hccontrol -n ubt0hci disconnect 41
Connection handle: 41
Reason: Connection terminated by local host [0x16]
```

Rufen Sie `hccontrol help` auf, wenn Sie eine komplette Liste aller verfügbaren HCI-Befehle benötigen. Die meisten dieser Befehle müssen nicht als `root` ausgeführt werden.

### 54.5.3. Erstmaliger Verbindungsaufbau zwischen zwei Bluetooth-Geräten (Pairing)

In der Voreinstellung nutzt Bluetooth keine Authentifizierung, daher kann sich jedes Bluetoothgerät mit jedem anderen Gerät verbinden. Ein Bluetoothgerät, wie beispielsweise ein

Mobiltelefon, kann jedoch für einen bestimmten Dienst, etwa eine Einwählverbindung, eine Authentifizierung anfordern. Bluetooth verwendet zu diesem Zweck *PIN-Codes*. Ein PIN-Code ist ein maximal 16 Zeichen langer ASCII-String. Damit eine Verbindung zustande kommt, muss auf beiden Geräten der gleiche PIN-Code verwendet werden. Nachdem der Code eingegeben wurde, erzeugen beide Geräte einen *link key*, der auf den Geräten gespeichert wird. Beim nächsten Verbindungsaufbau wird der zuvor erzeugte Link Key verwendet. Diesen Vorgang bezeichnet man als Pairing. Geht der Link Key auf einem Gerät verloren, muss das Pairing wiederholt werden.

Der hcsecd(8)-Daemon verarbeitet Bluetooth-Authentifzierungsanforderungen und wird über die Datei /etc/bluetooth/hcsecd.conf konfiguriert. Der folgende Ausschnitt dieser Datei zeigt die Konfiguration für ein Mobiltelefon, das den PIN-Code "1234" verwendet:

```
device {
        bdaddr  00:80:37:29:19:a4;
        name    "Pav's T39";
        key     nokey;
        pin     "1234";
      }
```

Von der Länge abgesehen, unterliegen PIN-Codes keinen Einschränkungen. Einige Geräte, beispielsweise Bluetooth-Headsets, haben einen festen PIN-Code eingebaut. Die Option `-d` sorgt dafür, dass der hcsecd(8)-Daemon im Vordergrund läuft. Dadurch kann der Ablauf einfach verfolgt werden. Stellen Sie das entfernte Gerät auf receive pairing und initiieren Sie die Bluetoothverbindung auf dem entfernten Gerät. Sie erhalten die Meldung, dass Pairing akzeptiert wurde und der PIN-Code benötigt wird. Geben Sie den gleichen PIN-Code ein, den Sie in hcsecd.conf festgelegt haben. Der Computer und das entfernte Gerät sind nun miteinander verbunden. Alternativ können Sie das Pairing auch auf dem entfernten Gerät initiieren.

hcsecd(8) kann durch das Einfügen der folgenden Zeile in /etc/rc.conf beim Systemstart automatisch aktiviert werden:

```
hcsecd_enable="YES"
```

Es folgt nun eine beispielhafte Ausgabe des hcsecd(8)-Daemons:

```
hcsecd[16484]: Got Link_Key_Request event from 'ubt0hci', remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39',
link key doesn't exist
hcsecd[16484]: Sending Link_Key_Negative_Reply to 'ubt0hci' for remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Got PIN_Code_Request event from 'ubt0hci', remote bdaddr
0:80:37:29:19:a4
hcsecd[16484]: Found matching entry, remote bdaddr 0:80:37:29:19:a4, name 'Pav's T39',
PIN code exists
hcsecd[16484]: Sending PIN_Code_Reply to 'ubt0hci' for remote bdaddr 0:80:37:29:19:a4
```

### 54.5.4. Einwahlverbindungen und Netzwerkverbindungen mit PPP-Profilen einrichten

Ein Dial-Up Networking-Profil (DUN) kann dazu benutzt werden, ein Mobiltelefon als drahtloses Modem zu nutzen, um sich über einen Einwahlprovider mit dem Internet zu verbinden. Es kann auch dazu genutzt werden, einen Computer so zu konfigurieren, dass dieser Datenabfragen empfängt.

Der Zugriff auf ein Netzwerk über ein PPP-Profil kann einen Zugriff auf das LAN für ein oder mehrere Bluetooth-Geräte bieten. Eine PC-zu-PC-Verbindung unter Verwendung einer PPP-Verbindung über eine serielle Verbindung ist ebenfalls möglich.

Diese Profile werden unter FreeBSD durch ppp(8) sowie rfcomm_pppd(8) implementiert - einem Wrapper, der Bluetooth-Verbindungen unter PPP nutzbar macht. Bevor ein Profil verwendet werden kann, muss ein neuer PPP-Abschnitt in /etc/ppp/ppp.conf erzeugt werden. Beispielkonfigurationen zu diesem Thema finden Sie in rfcomm_pppd(8).

Dieses Beispiel verwendet rfcomm_pppd(8), um eine Verbindung zu einem entfernten Gerät mit der `BD_ADDR 00:80:37:29:19:a4` auf dem RFCOMM-Kanal `DUN` aufzubauen:

```
# rfcomm_pppd -a 00:80:37:29:19:a4 -c -C dun -l rfcomm-dialup
```

Die aktuelle Kanalnummer des entfernten Geräts erhalten Sie über das SDP-Protokoll. Es ist auch möglich, manuell einen RFCOMM-Kanal festzulegen. In diesem Fall führt rfcomm_pppd(8) keine SDP-Abfrage durch. Verwenden Sie sdpcontrol(8), um die RFCOMM-Kanäle des entfernten Geräts herauszufinden.

Der sdpd(8)-Server muss laufen, damit ein Netzzugriff mit dem PPPLAN-Profil möglich ist. Außerdem muss für den LAN-Client ein neuer Eintrag in /etc/ppp/ppp.conf erzeugt werden. Beispielkonfigurationen zu diesem Thema finden Sie in rfcomm_pppd(8). Danach starten Sie den RFCOMMPPP-Server über eine gültige RFCOMM-Kanalnummer. Der RFCOMMPPP-Server bindet dadurch den Bluetooth-LAN-Dienst an den lokalen SDP-Daemon. Das folgende Beispiel zeigt, wie man den RFCOMMPPP-Server startet.

```
# rfcomm_pppd -s -C 7 -l rfcomm-server
```

### 54.5.5. Bluetooth-Protokolle

Dieser Abschnitt gibt einen Überblick über die verschiedenen Bluetooth-Protokolle, ihre Funktionen sowie weitere Programme.

#### Das Logical Link Control and Adaptation Protocol (L2CAP)

Das Logical Link Control and Adaptation Protocol (L2CAP) bietet höherwertigen Protokollen verbindungsorientierte und verbindungslose Datendienste an. L2CAP erlaubt höherwertigen Protokollen und Programmen den Versand und Empfang von L2CAP-Datenpaketen mit einer Länge von bis zu 64 Kilobytes.

L2CAP arbeitet _kanal_basiert. Ein Kanal ist eine logische Verbindung innerhalb einer Basisbandverbindung. Jeder Kanal ist dabei an ein einziges Protokoll gebunden. Mehrere Geräte können an das gleiche Protokoll gebunden sein, es ist aber nicht möglich, einen Kanal an mehrere Protokolle zu binden. Jedes über einen Kanal ankommende L2CAP-Paket wird an das entsprechende höherwertige Protokoll weitergeleitet. Mehrere Kanäle können sich die gleiche Basisbandverbindung teilen.

Unter FreeBSD wird eine netgraph-Gerätedatei vom Typ *l2cap* für jedes einzelne Bluetooth-Gerät erzeugt. Diese Gerätedatei ist normalerweise mit der Bluetooth-HCI-Gerätedatei (downstream) sowie der Bluetooth-Socket-Gerätedatei (upstream) verbunden. Der Standardname für die L2CAP-Gerätedatei lautet "devicel2cap". Weitere Details finden Sie in ng_l2cap(4).

Ein nützlicher Befehl zum Anpingen von anderen Geräten ist l2ping(8). Einige Bluetooth-Geräte senden allerdings nicht alle erhaltenen Daten zurück. Die Ausgabe `0 bytes` im folgenden Beispiel ist also kein Fehler:

```
# l2ping -a 00:80:37:29:19:a4
0 bytes from 0:80:37:29:19:a4 seq_no=0 time=48.633 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=1 time=37.551 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=2 time=28.324 ms result=0
0 bytes from 0:80:37:29:19:a4 seq_no=3 time=46.150 ms result=0
```

Das Programm l2control(8) liefert Informationen über L2CAP-Dateien. Das folgende Beispiel zeigt, wie man die Liste der logischen Verbindungen (Kanäle) sowie die Liste der Basisbandverbindungen abfragen kann:

```
% l2control -a 00:02:72:00:d4:1a read_channel_list
L2CAP channels:
Remote BD_ADDR     SCID/ DCID   PSM  IMTU/ OMTU State
00:07:e0:00:0b:ca    66/   64     3   132/  672 OPEN
% l2control -a 00:02:72:00:d4:1a read_connection_list
L2CAP connections:
Remote BD_ADDR    Handle Flags Pending State
00:07:e0:00:0b:ca     41 O           0 OPEN
```

btsockstat(1) ist ein weiteres Diagnoseprogramm. Es funktioniert ähnlich wie netstat(1), arbeitet aber mit Bluetooth-Datenstrukturen. Das folgende Beispiel zeigt die gleiche Liste der logischen Verbindungen wie l2control(8) im vorherigen Beispiel.

```
% btsockstat
Active L2CAP sockets
PCB      Recv-Q Send-Q Local address/PSM       Foreign address   CID   State
c2afe900      0      0 00:02:72:00:d4:1a/3     00:07:e0:00:0b:ca 66    OPEN
Active RFCOMM sessions
L2PCB    PCB      Flag MTU   Out-Q DLCs State
c2afe900 c2b53380 1    127   0     Yes  OPEN
Active RFCOMM sockets
```

```
PCB      Recv-Q Send-Q Local address     Foreign address   Chan DLCI State
c2e8bc80      0    250 00:02:72:00:d4:1a 00:07:e0:00:0b:ca 3    6    OPEN
```

#### Radio Frequency Communication (RFCOMM)

Das RFCOMM-Protokoll emuliert serielle Verbindungen über das L2CAP-Protokoll. Bei RFCOMM handelt es sich um ein einfaches Transportprotokoll, das um Funktionen zur Emulation der 9poligen Schaltkreise von mit RS-232 (EIATIA-232-E) kompatiblen seriellen Ports ergänzt wurde. Es erlaubt bis zu 60 simultane Verbindungen (RFCOMM-Kanäle) zwischen zwei Bluetooth-Geräten.

Eine RFCOMM-Kommunikation besteht aus zwei Anwendungen (den Kommunikationsendpunkten), die über das Kommunikationssegment miteinander verbunden sind. RFCOMM unterstützt Anwendungen, die auf serielle Ports angewiesen sind. Das Kommunikationssegment entspricht der direkten Bluetooth-Verbindung zwischen den beiden Geräten.

RFCOMM kümmert sich um die direkte Verbindung von zwei Geräten, oder um die Verbindung zwischen einem Gerät und einem Modem über eine Netzwerkverbindung. RFCOMM unterstützt auch andere Konfigurationen. Ein Beispiel dafür sind Module, die drahtlose Bluetooth-Geräte mit einer verkabelten Schnittstelle verbinden können.

Unter FreeBSD ist das RFCOMM-Protokoll im Bluetooth Socket-Layer implementiert.

#### Das Service Discovery Protocol (SDP)

Das Service Discovery Protocol (SDP) erlaubt es Clientanwendungen, von Serveranwendungen angebotene Dienste sowie deren Eigenschaften abzufragen. Zu diesen Eigenschaften gehören die Art oder die Klasse der angebotenen Dienste sowie der Mechanismus oder das Protokoll, die zur Nutzung des Dienstes notwendig sind.

SDP ermöglicht Verbindungen zwischen einem SDP-Server und einem SDP-Client. Der Server enthält eine Liste mit den Eigenschaften der vom Server angebotenen Dienste. Jeder Eintrag beschreibt jeweils einen einzigen Serverdienst. Ein Client kann diese Informationen durch eine SDP-Anforderung vom SDP-Server beziehen. Wenn der Client oder eine Anwendung des Clients einen Dienst nutzen will, muss eine separate Verbindung mit dem Dienstanbieter aufgebaut werden. SDP bietet einen Mechanismus zum Auffinden von Diensten und deren Eigenschaften an, es bietet aber keine Mechanismen zur Verwendung dieser Dienste.

Normalerweise sucht ein SDP-Client nur nach Diensten, die bestimmte geforderte Eigenschaften erfüllen. Es ist aber auch möglich, anhand der Dienstbeschreibungen eine allgemeine Suche nach den von einem SDP-Server angebotenen Diensten durchzuführen. Diesen Vorgang bezeichnet man als Browsing.

Der Bluetooth-SDP-Server sdpd(8) und der Kommandozeilenclient sdpcontrol(8) sind bereits in der Standardinstallation von FreeBSD enthalten. Das folgende Beispiel zeigt, wie eine SDP-Abfrage durchgeführt wird:

```
% sdpcontrol -a 00:01:03:fc:6e:ec browse
Record Handle: 00000000
```

```
Service Class ID List:
        Service Discovery Server (0x1000)
Protocol Descriptor List:
        L2CAP (0x0100)
                Protocol specific parameter #1: u/int/uuid16 1
                Protocol specific parameter #2: u/int/uuid16 1

Record Handle: 0x00000001
Service Class ID List:
        Browse Group Descriptor (0x1001)

Record Handle: 0x00000002
Service Class ID List:
        LAN Access Using PPP (0x1102)
Protocol Descriptor List:
        L2CAP (0x0100)
        RFCOMM (0x0003)
                Protocol specific parameter #1: u/int8/bool 1
Bluetooth Profile Descriptor List:
        LAN Access Using PPP (0x1102) ver. 1.0
```

Beachten Sie, dass jeder Dienst eine Liste seiner Eigenschaften, wie etwa den RFCOMM-Kanal, zurückgibt. Je nachdem, welche Dienste der Benutzer benötigt, sollten einige dieser Eigenschaften notiert werden. Einige Bluetooth-Implementationen unterstützen kein Service Browsing und geben daher eine leere Liste zurück. Ist dies der Fall, ist es dennoch möglich, nach einem bestimmten Dienst zu suchen. Das folgende Beispiel demonstriert die Suche nach dem OBEX Object Push (OPUSH) Dienst:

```
% sdpcontrol -a 00:01:03:fc:6e:ec search OPUSH
```

Unter FreeBSD ist es die Aufgabe des sdpd(8)-Servers, Bluetooth-Clients verschiedene Dienste anzubieten. Sie können diesen Server durch das Einfügen der folgenden Zeile in /etc/rc.conf aktivieren:

```
sdpd_enable="YES"
```

Nun kann der sdpd(8)-Daemon durch folgende Eingabe gestartet werden:

```
# service sdpd start
```

Der lokale Server, der den entfernten Clients Bluetooth-Dienste anbieten soll, bindet diese Dienste an den lokalen SDP-Daemon. Ein Beispiel für eine solche Anwendung ist rfcomm_pppd(8). Einmal gestartet, wird der Bluetooth-LAN-Dienst an den lokalen SDP-Daemon gebunden.

Die Liste der vorhandenen Dienste, die am lokalen SDP-Server registriert sind, lässt sich durch eine SDP-Abfrage über einen lokalen Kontrollkanal abfragen:

```
# sdpcontrol -l browse
```

**OBEX Object-Push (OPUSH)**

OBEX ist ein häufig verwendetes Protokoll für den Dateitransfer zwischen Mobilgeräten. Sein Hauptzweck ist die Kommunikation über die Infrarotschnittstelle. Es dient daher zum Datentransfer zwischen Notebooks oder PDAs sowie zum Austausch von Visitenkarten oder Kalendereinträgen zwischen Mobiltelefonen und anderen Geräten mit PIM-Funktionen.

Server und Client von OBEX werden durch obexapp bereitgestellt, das als Paket oder Port comms/obexapp installiert werden kann.

Mit dem OBEX-Client werden Objekte zum OBEX-Server geschickt oder angefordert. Ein Objekt kann etwa eine Visitenkarte oder ein Termin sein. Der OBEX-Client fordert über SDP die Nummer des RFCOMM-Kanals vom entfernten Gerät an. Dies kann auch durch die Verwendung des Servicenamens anstelle der RFCOMM-Kanalnummer erfolgen. Folgende Dienste werden unterstützt: `IrMC`, `FTRN` und `OPUSH`. Es ist möglich, den RFCOMM-Kanal als Nummer anzugeben. Es folgt ein Beispiel für eine OBEX-Sitzung, bei der ein Informationsobjekt vom Mobiltelefon angefordert und ein neues Objekt (hier eine Visitenkarte) an das Telefonbuch des Mobiltelefons geschickt wird:

```
% obexapp -a 00:80:37:29:19:a4 -C IrMC
obex> get telecom/devinfo.txt
Success, response: OK, Success (0x20)
obex> put new.vcf
Success, response: OK, Success (0x20)
obex> di
Success, response: OK, Success (0x20)
```

Um OBEX-Push-Dienste anbieten zu können, muss der sdpd-Server gestartet sein. Ein Wurzelverzeichnis, in dem alle ankommenden Objekte gespeichert werden, muss zusätzlich angelegt werden. In der Voreinstellung ist dies /var/spool/obex. Starten Sie den OBEX-Server mit einer gültigen Kanalnummer. Der OBEX-Server registriert nun den OBEX-Push-Dienst mit dem lokalen SDP-Daemon. Das folgende Beispiel zeigt, wie der OBEX-Server gestartet wird:

```
# obexapp -s -C 10
```

**Das Serial-Port Profil (SPP)**

Das Serial Port Profile (SSP) ermöglicht es Bluetooth-Geräten eine serielle Kabelverbindung zu emulieren. Anwendungen sind dadurch in der Lage, über eine virtuelle serielle Verbindung Bluetooth als Ersatz für eine Kabelverbindung zu nutzen.

rfcomm_sppd(1) implementiert unter FreeBSD SSP und ein Pseudo-tty, das als virtuelle serielle Verbindung verwendet wird. Das folgende Beispiel zeigt, wie man eine Verbindung mit einem entfernten Serial-Port-Dienst herstellt. Ein RFCOMM-Kanal muss dabei nicht angegeben werden, da

rfcomm_sppd(1) den Kanal über SDP abfragen kann. Um dies zu umgehen, geben Sie einen RFCOMM-Kanal auf der Kommandozeile an.

```
# rfcomm_sppd -a 00:07:E0:00:0B:CA -t
rfcomm_sppd[94692]: Starting on /dev/pts/6...
/dev/pts/6
```

Sobald die Verbindung hergestellt ist, kann pseudo-tty als serieller Port verwenden werden.

```
# cu -l /dev/pts/6
```

Das pseudo-tty wird auf der Standardausgabe ausgegeben und kann von Wrapper-Skripten gelesen werden:

```
PTS=`rfcomm_sppd -a 00:07:E0:00:0B:CA -t`
cu -l $PTS
```

### 54.5.6. Problembehandlung

Wenn FreeBSD eine neue Verbindung akzeptiert, versucht es, die Rolle zu tauschen, um zum Master zu werden. Einige ältere Geräte, die dies nicht unterstützen, können keine Verbindung aufbauen. Da der Rollentausch ausgeführt wird sobald eine neue Verbindung aufgebaut wird, ist es nicht möglich, das entfernte Gerät zu fragen ob es den Rollentausch unterstützt. Es gibt jedoch eine HCI-Option, die dieses Verhalten deaktiviert:

```
# hccontrol -n ubt0hci write_node_role_switch 0
```

Verwenden Sie hcidump, das als Paket Port comms/hcidump installiert werden kann, um Bluetooth-Pakete anzuzeigen. Dieses Programm hat Ähnlichkeiten mit tcpdump(1) und kann zur Anzeige der Bluetooth-Pakete in einem Terminal, oder zur Speicherung von Paketen in einer Datei (Dump) verwendet werden.

## 54.6. LAN-Kopplung mit einer Bridge

Manchmal ist es nützlich, ein Netzwerk, wie ein Ethernetsegment, in separate Netzwerke aufzuteilen, ohne gleich IP-Subnetze zu erzeugen, die über einen Router miteinander verbunden sind. Ein Gerät, das zwei Netze auf diese Weise verbindet, wird als "Bridge" bezeichnet.

Eine Bridge arbeitet, indem sie die MAC-Adressen der Geräte in ihren Netzwerksegmenten lernt. Der Verkehr wird nur dann zwischen zwei Segmenten weitergeleitet, wenn sich Sender und Empfänger in verschiedenen Netzwerksegmenten befinden. Jedes FreeBSD-System mit zwei Netzwerkkarten kann als Bridge fungieren.

Bridging kann in den folgenden Situationen sinnvoll sein:

**Verbinden von Netzwerken**

Die Hauptaufgabe einer Bridge ist die Verbindung von zwei oder mehreren Netzwerksegmenten. Es gibt viele Gründe, eine hostbasierte Bridge einzusetzen, anstelle von Netzwerkkomponenten, wie beispielsweise Kabelverbindungen oder Firewalls. Eine Bridge kann außerdem ein drahtloses Gerät mit einem Kabelnetzwerk verbinden. Diese Fähigkeit der Bridge wird als HostAP-Modus bezeichnet. Die Bridge agiert in diesem Fall als Access Point für das drahtlose Gerät.

**Filtering / Traffic Shaping Firewall**

Eine Bridge kann eingesetzt werden, wenn Firewallfunktionen benötigt werden, ohne dabei Routing oder Network Adress Translation (NAT) zu verwenden.

Ein Beispiel dafür wäre ein kleines Unternehmen, das über DSL oder ISDN an einen ISP angebunden ist. Es verfügt über 13 erreichbare IP-Adressen und das Netzwerk besteht aus 10 Rechnern. In dieser Situation ist der Einsatz von Subnetzen sowie einer routerbasierten Firewall aufgrund der IP-Adressierung schwierig. Eine Bridge-basierte Firewall kann hingegen ohne Probleme konfiguriert werden.

**Netzwerküberwachung**

Eine Bridge kann zwei Netzwerksegmente miteinander verbinden und danach alle Ethernet-Rahmen überprüfen, die zwischen den beiden Netzwerksegmenten ausgetauscht werden. Dazu verwendet man entweder bpf(4) und tcpdump(1) auf dem Netzgerät der Bridge oder schickt Kopien aller Rahmen an ein zusätzliches Netzgerät, das als Span Port bekannt ist.

**Layer 2 VPN**

Zwei Ethernetnetzwerke können über einen IP-Link miteinander verbunden werden, indem die beiden Netzwerke über einen EtherIP-Tunnel gekoppelt werden, oder eine tap(4)-basierte Lösung wie OpenVPN eingesetzt wird.

**Layer 2 Redundanz**

Die Systeme eines Netzwerks können über das Spanning Tree Protocol (STP) redundant miteinander verbunden sein, um redundante Pfade zu blockieren.

Dieser Abschnitt beschreibt, wie ein FreeBSD-System mit Hilfe von if_bridge(4) als Bridge konfiguriert wird. Ein netgraph-Bridge-Treiber ist ebenfalls verfügbar und wird in ng_bridge(4) beschrieben.



Paketfilter können mit allen Firewallpaketen verwendet werden, die das pfil(9) -Framework benutzen. Eine Bridge kann auch als Traffic Shaper verwendet werden, wenn Sie altq(4) oder dummynet(4) einsetzen.

## 54.6.1. Die Bridge aktivieren

In FreeBSD handelt es sich bei if_bridge(4) um ein Kernelmodul, das von ifconfig(8) automatisch geladen wird, wenn eine Bridge-Schnittstelle erzeugt wird. Es ist auch möglich, die Unterstützung für den Treiber in den Kernel zu kompilieren, indem die Zeile `device if_bridge` in die Kernelkonfigurationsdatei hinzugefügt wird.

Eine Bridge wird durch das Klonen von Schnittstellen erzeugt. Um eine Bridge zu erzeugen, verwenden Sie:

```
# ifconfig bridge create
bridge0
# ifconfig bridge0
bridge0: flags=8802<BROADCAST,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
        id 00:00:00:00:00:00 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:00:00:00:00:00 priority 0 ifcost 0 port 0
```

Wenn eine Bridge erzeugt wird, erhält sie automatisch eine zufällig generierte Ethernet-Adresse. Die Parameter `maxaddr` sowie `timeout` legen fest, wie viele MAC-Adressen die Bridge in ihrer Forward-Tabelle halten kann und wie viele Sekunden jeder Eintrag erhalten bleiben soll, nachdem er zuletzt verwendet wurde. Die restlichen Parameter sind für die Konfiguration von STP notwendig.

Im nächsten Schritt werden die Schnittstellen, die die Bridge verbinden soll, zugewiesen. Damit die Bridge Datenpakete weiterleiten kann, müssen sowohl die Bridge als auch die Schnittstellen der zu verbindenden Netzwerksegmente aktiviert sein:

```
# ifconfig bridge0 addm fxp0 addm fxp1 up
# ifconfig fxp0 up
# ifconfig fxp1 up
```

Jetzt ist die Bridge in der Lage, Ethernet-Rahmen zwischen den Schnittstellen fxp0 und fxp1 weiterzuleiten. Um diese Konfiguration beim Systemstart automatisch zu aktivieren, müssen die folgenden Zeilen in /etc/rc.conf hinzugefügt werden:

```
cloned_interfaces="bridge0"
ifconfig_bridge0="addm fxp0 addm fxp1 up"
ifconfig_fxp0="up"
ifconfig_fxp1="up"
```

Wenn die Bridge eine IP-Adresse benötigt, muss diese der Schnittstelle der Bridge zugewiesen werden und nicht der Schnittstelle der gekoppelten Netzwerksegmente. Die IP-Adresse kann manuell gesetzt, oder über DHCP bezogen werden. Dieses Beispiel verwendet eine statische IP-Adresse:

```
# ifconfig bridge0 inet 192.168.0.1/24
```

Es ist auch möglich der Bridge-Schnittstelle eine IPv6-Adresse zuzuweisen. Um die Änderungen dauerhaft zu speichern, fügen Sie die Adressinformationen in /etc/rc.conf ein.



Nachdem ein Paketfilter aktiviert wurde, können Datenpakete, die von den Schnittstellen der gekoppelten Netzwerksegmente gesendet und empfangen werden, über die Bridge weitergeleitet oder nach bestimmten Regeln gefiltert oder auch komplett geblockt werden. Ist die Richtung des Paketflusses wichtig, ist es am besten, eine Firewall auf den Schnittstellen der einzelnen Netzwerksegmente einzurichten und nicht auf der Bridge selbst.

Eine Bridge verfügt über verschiedene Optionen zur Weiterleitung von Nicht-IP- und IP-Paketen, sowie Paketfilterung auf Layer 2 mittels ipfw(8). Weitere Informationen finden Sie in if_bridge(4).

### 54.6.2. Spanning Tree aktivieren

Damit ein Ethernet-Netzwerk richtig funktioniert, kann nur ein aktiver Pfad zwischen zwei Geräten existieren. Das STP-Protokoll erkennt Schleifen in einer Netzwerktopologie und setzt redundante Pfade in einen blockierten Zustand. Sollte eine der aktiven Verbindungen ausfallen, berechnet STP einen anderen Baum und ermöglicht es dann einem blockierten Pfad, alle Netzwerkverbindungen wiederherzustellen.

Das Rapid Spanning Tree Protocol (RSTP oder 802.1w), ist abwärtskompatibel zum veralteten STP. RSTP arbeitet schneller und tauscht Informationen mit benachbarten Switchen aus, um Pakete korrekt weiterzuleiten und eine Schleifenbildung zu verhindern. FreeBSD unterstützt die Betriebsmodi RSTP und STP, wobei RSTP als Standardmodus voreingestellt ist.

STP kann auf den Schnittstellen der durch die Bridge verbundenen Netzwerksegmente mittels ifconfig(8) aktiviert werden. Für eine Bridge, die die Schnittstellen fxp0 und fxp1 verbindet, aktivieren Sie STP wie folgt:

```
# ifconfig bridge0 stp fxp0 stp fxp1
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether d6:cf:d5:a0:94:6d
        id 00:01:02:4b:d4:50 priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 0 port 0
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 3 priority 128 path cost 200000 proto rstp
                role designated state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

Diese Bridge hat die Spanning-Tree-ID `00:01:02:4b:d4:50` und die Priorität `32768`. Da diese ID mit der `Root-ID` identisch ist, handelt es sich um die Root-Bridge dieses Netzwerks.

Auf einer anderen Bridge des Netzwerks ist STP ebenfalls aktiviert:

```
bridge0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        ether 96:3d:4b:f1:79:7a
```

```
        id 00:13:d4:9a:06:7a priority 32768 hellotime 2 fwddelay 15
        maxage 20 holdcnt 6 proto rstp maxaddr 100 timeout 1200
        root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4
        member: fxp0 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 4 priority 128 path cost 200000 proto rstp
                role root state forwarding
        member: fxp1 flags=1c7<LEARNING,DISCOVER,STP,AUTOEDGE,PTP,AUTOPTP>
                port 5 priority 128 path cost 200000 proto rstp
                role designated state forwarding
```

Die Zeile `root id 00:01:02:4b:d4:50 priority 32768 ifcost 400000 port 4` zeigt an, dass die Root-Bridge die ID `00:01:02:4b:d4:50` hat. Die Pfadkosten hin zur Root-Bridge betragen `400000`, wobei der Pfad zur Root-Bridge über `port 4` geht, der wiederum der Schnittstelle fxp0 entspricht.

### 54.6.3. Parameter der Bridge-Schnittstelle

Einige Parameter von `ifconfig` dienen ausschließlich der Konfiguration von Bridge-Schnittstellen. Dieser Abschnitt fasst die Verwendung dieser Parameter zusammen. Die vollständige Liste der verfügbaren Parameter wird in ifconfig(8) beschrieben.

**private**

Eine private Schnittstelle leitet keine Daten an einen Port weiter, bei dem es sich ebenfalls um eine private Schnittstelle handelt. Der Datenverkehr wird dabei komplett blockiert, auch Ethernet-Rahmen und ARP-Pakete werden nicht weitergeleitet. Wollen Sie hingegen nur spezifische Datenpakete blockieren, sollten Sie eine Firewall einsetzen.

**span**

Ein Span Port übertragt eine Kopie jedes Ethernet-Rahmens, der an der Bridge ankommt. Auf einer Bridge können beliebig viele Span Ports festgelegt werden. Wird eine Schnittstelle als Span Port konfiguriert, kann sie nicht mehr als normaler Bridge-Port verwendet werden. Eine derartige Konfiguration ist beispielsweise sinnvoll, um den Datenverkehr, der in einem Netzwerk über die Bridge läuft, auf einen Rechner zu übertragen, der mit einem Span Port der Bridge verbunden ist. Um beispielsweise eine Kopie aller Ethernet-Rahmen über die Schnittstelle fxp0 zu übertragen:

```
# ifconfig bridge0 span fxp4
```

**sticky**

Wenn die Schnittstelle eines über eine Bridge verbundenen Netzwerksegments als sticky gekennzeichnet wird, werden alle dynamisch gelernten Adressen als statische Adressen behandelt, sobald sie in den Forward-Cache der Bridge aufgenommen wurden. Sticky-Einträge werden niemals aus dem Cache entfernt oder ersetzt. Selbst dann nicht, wenn die Adresse von einer anderen Schnittstelle verwendet wird. Sie können dadurch die Vorteile statischer Adresseinträge nutzen, ohne die Forward-Tabelle vor dem Einsatz der Bridge mit statischen Einträgen füllen zu müssen. Clients, die sich in einem bestimmten von der Bridge verwalteten Segmente befinden, können dabei nicht in ein anderes Segment wechseln.

Ein Beispiel für den Einsatz von Sticky-Adressen ist die Kombination einer Bridge mit mehreren VLANs, um einen Router zu konfigurieren, der einzelne Kundennetzwerke voneinander trennt, ohne dabei IP-Adressbereiche zu verschwenden. Für das folgende Beispiel nehmen wir an, dass sich der Client `CustomerA` im VLAN `vlan100` und der Client `CustomerB` im VLAN `vlan101` befinden. Die Bridge hat die IP-Adresse `192.168.0.1`:

```
# ifconfig bridge0 addm vlan100 sticky vlan100 addm vlan101 sticky vlan101
# ifconfig bridge0 inet 192.168.0.1/24
```

In diesem Beispiel sehen beide Clients `192.168.0.1` als das Default-Gateway. Da der Brücken-Cache *sticky* ist, sind Sie nicht dazu in der Lage, die MAC-Adresse des anderen Kunden zu spoofen und dessen Datenverkehr abzufangen.

Sie können die Kommunikation zwischen den VLANs vollständig unterbinden, wenn Sie private Schnittstellen oder eine Firewall einsetzen:

```
# ifconfig bridge0 private vlan100 private vlan101
```

Die Kunden sind nun komplett voneinander isoliert und der komplette `/24`-Adressbereich kann zugewiesen werden, ohne dass Subnetze eingesetzt werden.

Die maximale mögliche Anzahl an eindeutigen MAC-Adressen hinter einer Schnittstelle kann festgelegt werden. Sobald das Limit erreicht ist, werden Pakete mit einer unbekannten Quell-Adresse solange verworfen, bis ein existierender Eintrag gelöscht wird oder abläuft.

Das folgende Beispiel setzt die maximale Anzahl von Netzgeräten für `CustomerA` für das VLAN `vlan100` auf 10.

```
# ifconfig bridge0 ifmaxaddr vlan100 10
```

Die Bridge unterstützt auch den Monitormodus. Dabei werden alle Pakete verworfen, nachdem sie von bpf(4) verarbeitet wurden. In diesem Modus erfolgt keine weitere Bearbeitung und auch keine Weiterleitung von Datenpaketen. Es ist daher möglich, die Eingabe von zwei oder mehr Netzwerkschnittstellen in einen einzigen gemeinsamen bpf(4)-Stream zu vereinen. Ein solcher Datenstrom ist beispielsweise nützlich, um den Datenverkehr für "network taps" zu rekonstruieren, die ihre RX/TX-Signale über verschiedene Schnittstellen senden. Um beispielsweise die Eingabe von vier Netzwerkschnittstellen in einzigen gemeinsamen Datenstrom zu vereinen:

```
# ifconfig bridge0 addm fxp0 addm fxp1 addm fxp2 addm fxp3 monitor up
# tcpdump -i bridge0
```

### 54.6.4. SNMP-Monitoring

Die Schnittstelle der Bridge sowie die STP-Parameter können durch den im Basissystem enthaltenen bsnmpd(1) überwacht werden. Die exportierten Bridge-MIBs entsprechen den IETF-

Standards, daher kann ein beliebiger SNMP-Client oder ein beliebiges Monitoring-Werkzeug eingesetzt werden, um die benötigten Daten zu erhalten.

Um das Monitoring auf der Bridge zu aktivieren, kommentieren Sie diese Zeile in /etc/snmpd.config aus, indem Sie das Zeichen # entfernen:

```
begemotSnmpdModulePath."bridge" = "/usr/lib/snmp_bridge.so"
```

Weitere Konfigurationsparameter wie Community-Namen und Zugriffslisten müssen ebenfalls in dieser Datei angepasst werden. Weitere Informationen finden Sie in bsnmpd(1) und snmp_bridge(3). Nachdem die Änderungen gespeichert wurden, fügen Sie folgende Zeile in /etc/rc.conf hinzu:

```
bsnmpd_enable="YES"
```

Danach starten Sie bsnmpd(1):

```
# service bsnmpd start
```

Die folgenden Beispiele verwenden das Softwarepaket Net-SNMP (net-mgmt/net-snmp), um die Bridge vom Client aus abzufragen. Alternativ kann auch der Port net-mgmt/bsnmptools benutzt werden. Auf dem SNMP-Client müssen danach die folgenden Zeilen in $HOME/.snmp/snmp.conf hinzugefügt werden, um die MIB-Definitionen der Bridge in Net-SNMP zu importieren:

```
mibdirs +/usr/shared/snmp/mibs
mibs +BRIDGE-MIB:RSTP-MIB:BEGEMOT-MIB:BEGEMOT-BRIDGE-MIB
```

Um eine einzelne Bridge über den IETF BRIDGE-MIB (RFC4188) zu überwachen:

```
% snmpwalk -v 2c -c public bridge1.example.com mib-2.dot1dBridge
BRIDGE-MIB::dot1dBaseBridgeAddress.0 = STRING: 66:fb:9b:6e:5c:44
BRIDGE-MIB::dot1dBaseNumPorts.0 = INTEGER: 1 ports
BRIDGE-MIB::dot1dStpTimeSinceTopologyChange.0 = Timeticks: (189959) 0:31:39.59 centi-
seconds
BRIDGE-MIB::dot1dStpTopChanges.0 = Counter32: 2
BRIDGE-MIB::dot1dStpDesignatedRoot.0 = Hex-STRING: 80 00 00 01 02 4B D4 50
...
BRIDGE-MIB::dot1dStpPortState.3 = INTEGER: forwarding(5)
BRIDGE-MIB::dot1dStpPortEnable.3 = INTEGER: enabled(1)
BRIDGE-MIB::dot1dStpPortPathCost.3 = INTEGER: 200000
BRIDGE-MIB::dot1dStpPortDesignatedRoot.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedCost.3 = INTEGER: 0
BRIDGE-MIB::dot1dStpPortDesignatedBridge.3 = Hex-STRING: 80 00 00 01 02 4B D4 50
BRIDGE-MIB::dot1dStpPortDesignatedPort.3 = Hex-STRING: 03 80
BRIDGE-MIB::dot1dStpPortForwardTransitions.3 = Counter32: 1
```

```
RSTP-MIB::dot1dStpVersion.0 = INTEGER: rstp(2)
```

Der Wert der Variable `dot1dStpTopChanges.0` ist hier 2, die STP-Topologie der Bridge wurde also bereits zweimal geändert. Unter einer Änderung versteht man die Anpassung eines oder mehrerer Links und die Kalkulation eines neuen Baums. Der Wert der Variable `dot1dStpTimeSinceTopologyChange.0` gibt an, wann dies zuletzt geschah.

Um mehrere Bridge-Schnittstellen zu überwachen, kann der private BEGEMOT-BRIDGE-MIB eingesetzt werden:

```
% snmpwalk -v 2c -c public bridge1.example.com
enterprises.fokus.begemot.begemotBridge
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge0" = STRING: bridge0
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseName."bridge2" = STRING: bridge2
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge0" = STRING: e:ce:3b:5a:9e:13
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseAddress."bridge2" = STRING: 12:5e:4d:74:d:fc
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge0" = INTEGER: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeBaseNumPorts."bridge2" = INTEGER: 1
...
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge0" = Timeticks:
(116927) 0:19:29.27 centi-seconds
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTimeSinceTopologyChange."bridge2" = Timeticks:
(82773) 0:13:47.73 centi-seconds
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge0" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpTopChanges."bridge2" = Counter32: 1
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge0" = Hex-STRING: 80 00 00 40
95 30 5E 31
BEGEMOT-BRIDGE-MIB::begemotBridgeStpDesignatedRoot."bridge2" = Hex-STRING: 80 00 00 50
8B B8 C6 A9
```

Um die über den `mib-2.dot1dBridge`-Subtree überwachte Bridge-Schnittstelle zu ändern:

```
% snmpset -v 2c -c private bridge1.example.com
BEGEMOT-BRIDGE-MIB::begemotBridgeDefaultBridgeIf.0 s bridge2
```

## 54.7. Link-Aggregation und Failover

Die von FreeBSD unterstützte lagg(4)-Schnittstelle erlaubt die Gruppierung von mehreren Netzwerkadaptern als eine virtuelle Schnittstelle, mit dem Ziel, Ausfallsicherheit (Failover) und Link Aggregation bereitzustellen. Bei Failover kann der Verkehr auch dann weiter fließen, wenn nur eine Schnittstelle verfügbar ist. Link Aggregation funktioniert am besten mit Switches, die LCAP unterstützen, da dieses Protokoll den Datenverkehr bidirektional verteilt, während es auch auf den Ausfall einzelner Verbindungen reagiert.

Die von der lagg-Schnittstelle unterstützten Protokolle bestimmen, welche Ports für den ausgehenden Datenverkehr benutzt werden, und ob ein bestimmter Port eingehenden Datenverkehr akzeptiert. Die folgenden Protokolle werden von lagg(4) unterstützt:

**Failover (Ausfallsicherheit)**

Dieser Modus sendet und empfängt Datenverkehr nur auf dem Masterport. Sollte der Masterport nicht zur Verfügung stehen, wird der nächste aktive Port verwendet. Der zuerst hinzugefügte Adapter der virtuellen Schnittstelle wird zum Masterport, jeder weitere Adapter dient als Gerät zur Ausfallsicherheit. Wenn ein Failover auf einem Nicht-Master Port stattfindet, wird der ursprüngliche Port wieder zum Master-Port, sobald er wieder verfügbar ist.

**fec / loadbalance (Lastverteilung)**

Cisco® Fast EtherChannel® (FEC) findet sich auf älteren Cisco® Switches. Es bietet eine statische Konfiguration und handelt weder Aggregation mit der Gegenstelle aus, noch werden Frames zur Überwachung der Verbindung ausgetauscht. Wenn der Switch LACP unterstützt, sollte diese Option auch verwendet werden.

**lacp**

Das IEEE® 802.3ad Link-Aggregation Control Protokoll (LACP). Mit LACP wird eine Menge von aggregierbaren Verbindungen mit der Gegenstelle in einer oder mehreren Link Aggregated Groups (LAG) ausgehandelt. Jede LAG besteht aus Ports der gleichen Geschwindigkeit, eingestellt auf Voll-Duplex-Betrieb. Der Verkehr wird über die Ports in der LAG mit der größten Gesamtgeschwindigkeit balanciert. Typischerweise gibt es nur eine LAG, die alle Ports enthält. Im Falle von Änderungen in der physischen Anbindung wird LACP schnell zu einer neuen Konfiguration konvergieren.

LACP balanciert ausgehenden Verkehr über die aktiven Ports basierend auf der gehashten Protokollheaderinformation und akzeptiert eingehenden Verkehr auf jedem aktiven Port. Der Hash beinhaltet die Ethernet-Quell- und Zieladresse, und, soweit verfügbar, den VLAN-Tag, sowie die IPv4 oder IPv6 Quell- und Zieladresse.

**roundrobin**

Dieser Modus verteilt ausgehenden Verkehr mittels einer Round-Robin-Zuteilung über alle aktiven Ports und akzeptiert eingehenden Verkehr auf jedem aktiven Port. Da dieser Modus die Reihenfolge von Ethernet-Rahmen verletzt, sollte er mit Vorsicht eingesetzt werden.

### 54.7.1. Beispiele

Dieser Abschnitt zeigt, wie man einen Cisco® Switch und ein FreeBSD-System für LACP Load Balancing konfiguriert. Weiterhin wird gezeigt, wie man zwei Ethernet-Schnittstellen, sowie eine Ethernet- und eine Drahtlos-Schnittstelle für den Failover-Modus konfigurieren kann.

*Beispiel 48. LACP Aggregation mit einem Cisco® Switch*

Dieses Beispiel verbindet zwei fxp(4) Ethernet-Schnittstellen einer FreeBSD-Maschine zu den ersten zwei Ethernet-Ports auf einem Cisco® Switch als eine einzelne, lastverteilte und ausfallsichere Verbindung. Weitere Adapter können hinzugefügt werden, um den Durchsatz zu erhöhen und die Ausfallsicherheit zu steigern. Ersetzen Sie die Namen der Cisco®-Ports, Ethernet-Geräte, channel-group Nummern und IP-Adressen im Beispiel durch Namen, die mit Ihrer lokalen Konfiguration übereinstimmen.

Da die Reihenfolge der Frames bei Ethernet zwingend eingehalten werden muss, fließt auch

jeglicher Verkehr zwischen zwei Stationen über den gleichen physischen Kanal, was die maximale Geschwindigkeit der Verbindung auf die eines einzelnen Adapters beschränkt. Der Übertragungsalgorithmus versucht, so viele Informationen wie möglich zu verwenden, um die verschiedenen Verkehrsflüsse zu unterscheiden und balanciert diese über die verfügbaren Adapter.

Fügen Sie auf dem Cisco®-Switch die Adapter *FastEthernet0/1* und *FastEthernet0/2* zu der channel-group *1* hinzu:

```
interface FastEthernet0/1
 channel-group 1 mode active
 channel-protocol lacp
!
interface FastEthernet0/2
 channel-group 1 mode active
 channel-protocol lacp
```

Erstellen Sie auf der FreeBSD Maschine die lagg(4)-Schnittstelle unter Verwendung von *fxp0* und *fxp1* und starten Sie die Schnittstelle mit der IP-Adresse *10.0.0.3/24*:

```
# ifconfig fxp0 up
# ifconfig fxp1 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto lacp laggport fxp0 laggport fxp1 10.0.0.3/24
```

Überprüfen Sie den Status der virtuellen Schnittstelle:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        inet 10.0.0.3 netmask 0xffffff00 broadcast 10.0.0.255
        media: Ethernet autoselect
        status: active
        laggproto lacp
        laggport: fxp1 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
        laggport: fxp0 flags=1c<ACTIVE,COLLECTING,DISTRIBUTING>
```

Ports, die als *ACTIVE* markiert sind, sind Teil der aktiven Aggregations-Gruppe, die mit dem Switch ausgehandelt wurde. Der Verkehr wird über diese Gruppe übertragen und empfangen. Benutzen Sie ifconfig(8) mit `-v`, um sich die LAG-Bezeichner anzeigen zu lassen.

Um den Status der Ports auf dem Switch anzuzeigen, benutzen Sie `show lacp neighbor`:

```
switch# show lacp neighbor
Flags:  S - Device is requesting Slow LACPDUs
```

```
        F - Device is requesting Fast LACPDUs
        A - Device is in Active mode       P - Device is in Passive mode

Channel group 1 neighbors

Partner's information:

                  LACP port                        Oper    Port     Port
Port      Flags   Priority  Dev ID         Age     Key     Number   State
Fa0/1     SA      32768     0005.5d71.8db8  29s    0x146   0x3      0x3D
Fa0/2     SA      32768     0005.5d71.8db8  29s    0x146   0x4      0x3D
```

Benutzen Sie `show lacp neighbor detail`, um weitere Informationen zu erhalten.

Damit diese Konfiguration auch nach einem Neustart erhalten bleibt, fügen Sie auf dem FreeBSD-System folgende Einträge in /etc/rc.conf hinzu:

```
ifconfig_fxp0="up"
ifconfig_fxp1="up"
cloned_interfaces="lagg0
ifconfig_lagg0="laggproto lacp laggport fxp0 laggport fxp1 10.0.0.3/24"
```

*Beispiel 49. Ausfallsicherer Modus*

Der ausfallsichere Modus kann verwendet werden, um zu einer zweiten Schnittstelle zu wechseln, sollte die Verbindung mit der Master-Schnittstelle ausfallen. Um den ausfallsicheren Modus zu konfigurieren, aktivieren Sie zunächst die zugrunde liegenden physikalischen Schnittstellen. Erstellen Sie dann die lagg(4)-Schnittstelle mit *fxp0* als Master-Schnittstelle und *fxp1* als sekundäre Schnittstelle. Der virtuellen Schnittstelle wird die IP-Adresse *10.0.0.15/24* zugewiesen:

```
# ifconfig fxp0 up
# ifconfig fxp1 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport fxp0 laggport fxp1 10.0.0.15/24
```

Die virtuelle Schnittstelle sollte in etwa so aussehen:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether 00:05:5d:71:8d:b8
        inet 10.0.0.15 netmask 0xffffff00 broadcast 10.0.0.255
        media: Ethernet autoselect
        status: active
        laggproto failover
```

```
        laggport: fxp1 flags=0<>
        laggport: fxp0 flags=5<MASTER,ACTIVE>
```

Der Verkehr wird auf *fxp0* übertragen und empfangen. Wenn die Verbindung auf *fxp0* abbricht, wird *fxp1* die Verbindung übernehmen. Sobald die Verbindung auf der Master-Schnittstelle wiederhergestellt ist, wird diese wieder als aktive Schnittstelle genutzt.

Damit diese Konfiguration auch nach einem Neustart erhalten bleibt, fügen Sie folgende Einträge in /etc/rc.conf hinzu:

```
ifconfig_fxp0="up"
ifconfig_fxp1="up"
cloned_interfaces="lagg0
ifconfig_lagg0="laggproto failover laggport fxp0 laggport fxp1 10.0.0.15/24"
```

*Beispiel 50. Failover Modus zwischen Ethernet- und drahtlosen Schnittstellen*

Für Laptop-Benutzer ist es normalerweise wünschenswert, "wireless" als sekundäre Schnittstelle einzurichten, die verwendet wird, wenn die Ethernet-Verbindung nicht verfügbar ist. Mit lagg(4) ist es möglich, ein Failover mit einer IP-Adresse zu konfigurieren, welches die Ethernet-Verbindung aus Performance- und Sicherheitsgründen bevorzugt, während es gleichzeitig möglich bleibt, Daten über die drahtlose Verbindung zu übertragen.

Dies wird erreicht, indem die MAC-Adresse der Ethernet-Schnittstelle mit der MAC Adresse der drahtlosen Schnittstelle überschrieben wird.

Theoretisch kann die Ethernet- oder die drahtlose MAC-Adresse so geändert werden, dass sie mit der jeweils anderen Adresse übereinstimmt. Bei einigen drahtlosen Schnittstellen fehlt jedoch die Unterstützung für das Überschreiben der MAC-Adresse. Daher wird empfohlen, die MAC-Adresse der Ethernet-Schnittstelle für diesen Zweck zu überschreiben.

Wenn der Treiber für die drahtlose Schnittstelle nicht im `GENERIC`-Kernel oder in einem angepassten Kernel enthalten ist, kann unter FreeBSD 12.1 mit `_driver__load="YES"` die entsprechende .ko-Datei in /boot/loader.conf geladen werden. Dann muss das System neu gestartet werden. Ein anderer, besserer Weg ist es, den Treiber über /etc/rc.conf zu laden, indem Sie ihn zu `kld_list` (siehe rc.conf(5)) hinzufügen und dann das System neu starten. Dies ist notwendig, da sonst der Treiber zum Zeitpunkt der Konfiguration der lagg(4)-Schnittstelle noch nicht geladen ist.

In diesem Beispiel ist die Ethernet-Schnittstelle *re0* der Master und die drahtlose Schnittstelle *wlan0* der Failover. Die Schnittstelle *wlan0* wurde aus der physischen Schnittstelle *ath0* erstellt, und die Ethernet-Schnittstelle wird mit der MAC-Adresse der drahtlosen Schnittstelle konfiguriert. Im ersten Schritt wird die MAC-Adresse der drahtlosen Schnittstelle ermittelt:

```
# ifconfig wlan0
wlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
    ether b8:ee:65:5b:32:59
    groups: wlan
    ssid Bbox-A3BD2403 channel 6 (2437 MHz 11g ht/20) bssid 00:37:b7:56:4b:60
    regdomain ETSI country FR indoor ecm authmode WPA2/802.11i privacy ON
    deftxkey UNDEF AES-CCM 2:128-bit txpower 30 bmiss 7 scanvalid 60
    protmode CTS ampdulimit 64k ampdudensity 8 shortgi -stbctx stbcrx
    -ldpc wme burst roaming MANUAL
    media: IEEE 802.11 Wireless Ethernet MCS mode 11ng
    status: associated
    nd6 options=29<PERFORMNUD,IFDISABLED,AUTO_LINKLOCAL>
```

Ersetzen Sie *ath0* durch den Namen der drahtlosen Schnittstelle. Die `ether`-Zeile wird die MAC-Adresse der angegebenen Schnittstelle enthalten. Ändern Sie nun die MAC-Adresse der zugrunde liegenden Ethernet-Schnittstelle:

```
# ifconfig re0 ether b8:ee:65:5b:32:59
```

Starten Sie die drahtlose Schnittstelle, aber ohne eine IP-Adresse zu setzen. Ersetzen Sie *FR* durch den entsprechenden Ländercode:

```
# ifconfig wlan0 create wlandev iwn0 country FR ssid my_router up
```

Stellen Sie sicher, dass die *re0*-Schnittstelle aktiv ist. Erstellen Sie die lagg(4)-Schnittstelle mit *re0* als Master und *wlan0* als Failover:

```
# ifconfig re0 up
# ifconfig lagg0 create
# ifconfig lagg0 up laggproto failover laggport re0 laggport wlan0
```

Die virtuelle Schnittstelle sollte in etwa so aussehen:

```
# ifconfig lagg0
lagg0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> metric 0 mtu 1500
        options=8<VLAN_MTU>
        ether b8:ee:65:5b:32:59
        laggproto failover lagghash l2,l3,l4
        laggport: re0 flags=5<MASTER,ACTIVE>
        laggport: wlan0 flags=0<>
        groups: lagg
        media: Ethernet autoselect
        status: active
```

Starten Sie dann den DHCP-Client, um eine IP-Adresse zu erhalten:

```
# dhclient lagg0
```

Damit diese Konfiguration auch nach einem Neustart erhalten bleibt, fügen Sie folgende Einträge in /etc/rc.conf hinzu:

```
ifconfig_re0="ether b8:ee:65:5b:32:59"
wlans_ath0="wlan0"
ifconfig_wlan0="WPA"
create_args_wlan0="country FR"
cloned_interfaces="lagg0"
ifconfig_lagg0="up laggproto failover laggport re0 laggport wlan0 DHCP"
```

## 54.8. Plattenloser Betrieb mit PXE

Das Intel® Preboot eXecution Environment (PXE) erlaubt es dem Betriebssystem über das Netzwerk zu starten. Zum Beispiel kann ein FreeBSD-System, ohne lokale Festplatte, über das Netzwerk gestartet und betrieben werden. Die Dateisysteme werden dabei über einen NFS-Server eingehangen. PXE-Unterstützung steht in der Regel im BIOS zur Verfügung. Um PXE beim Systemstart zu verwenden, müssen Sie im BIOS des Rechners die Option `Über das Netzwerk starten` aktivieren. Alternativ können Sie während der PC-Initialisierung auch eine Funktionstaste drücken.

Um die notwendigen Dateien für ein Betriebssystem für den Start über das Netzwerk bereitzustellen, benötigt ein PXE-Setup einen richtig konfigurierten DHCP-, TFTP- und NFS-Server, wobei:

- Die initialen Parameter, wie IP-Adresse, Dateiname und Speicherort der ausführbaren Bootdateien, Servername sowie Root-Pfad vom DHCP-Server bezogen werden.
- Der Loader für das Betriebssystem über TFTP gestartet wird.
- Die Dateisysteme über NFS geladen werden.

Sobald das Gastsystem über PXE startet, erhält es vom DHCP-Server Informationen, wo der initiale Bootloader per TFTP zu bekommen ist. Nachdem das Gastsystem diese Informationen erhalten hat, lädt es den Bootloader über TFTP herunter und führt diesen anschließend aus. In FreeBSD ist /boot/pxeboot der Bootloader. Nachdem /boot/pxeboot ausgeführt und der FreeBSD-Kernel geladen wurde, wird mit dem Rest der FreeBSD-Bootsequenz, wie in FreeBSDs Bootvorgang beschrieben, fortgefahren.

Dieser Abschnitt beschreibt, wie Sie diese Dienste auf einem FreeBSD-System so konfigurieren, sodass andere Systeme FreeBSD über PXE starten können. Weitere Informationen finden Sie in diskless(8).



Wie beschrieben, ist das System, welches diese Dienste bereitstellt, unsicher. Daher sollte es in einem geschützten Bereich des Netzwerks aufgestellt und von anderen Hosts als nicht vertrauenswürdig eingestuft werden.

### 54.8.1. Konfiguration der PXE-Umgebung

Die in diesem Abschnitt dargestellten Schritte konfigurieren die in FreeBSD enthaltenen NFS- und TFTP-Server. Der folgende Abschnitt beschreibt die Installation und Konfiguration des DHCP-Servers. In diesem Beispiel verwenden wir /b/tftpboot/FreeBSD/install, welches die Dateien für PXE-Benutzer enthält. Es ist wichtig, dass dieses Verzeichnis existiert und das der gleiche Verzeichnisname ebenfalls in /etc/inetd.conf und /usr/local/etc/dhcpd.conf gesetzt wird.

1. Erstellen Sie das Root-Verzeichnis, welches eine FreeBSD-Installation enthält und über NFS eingehangen werden kann:

```
# export NFSROOTDIR=/b/tftpboot/FreeBSD/install
# mkdir -p ${NFSROOTDIR}
```

2. Aktivieren Sie den NFS-Server, indem Sie folgende Zeile in /etc/rc.conf hinzufügen:

```
nfs_server_enable="YES"
```

   Exportieren Sie das Root-Verzeichnis über NFS, indem Sie folgende Zeile in /etc/exports hinzufügen:

```
/b -ro -alldirs -maproot=root
```

3. Starten Sie den NFS-Server:

```
# service nfsd start
```

4. Aktivieren Sie inetd(8), indem Sie folgende Zeile in /etc/rc.conf hinzufügen:

```
inetd_enable="YES"
```

5. Kommentieren Sie die folgende Zeile in /etc/inetd.conf aus, indem Sie sicherstellen, dass die Zeile nicht mit einem #-Zeichen beginnt:

```
tftp dgram udp wait root /usr/libexec/tftp tftp -l -s /b/tftpboot
```



Einige PXE-Versionen benötigen die TCP-Version von TFTP. In diesem Fall können Sie die zweite `tftp`-Zeile, welche `stream tcp` enthält, auskommentieren.

6. Starten Sie inetd(8):

```
# service inetd start
```

7. Installieren Sie das Basissystem nach ${NFSROOTDIR}, indem Sie die offiziellen Archive entpacken, oder ein neues Basissystem und einen FreeBSD-Kernel erstellen. Detaillierte Anweisungen hierzu finden Sie im “FreeBSD aus den Quellen aktualisieren”. Vergessen Sie jedoch nicht `DESTDIR=${NFSROOTDIR}` hinzuzufügen, wenn Sie die Kommandos `make installkernel` und `make installworld` ausführen.

8. Testen Sie den TFTP-Server und vergewissern Sie sich, dass Sie den Bootloader herunterladen können, der über PXE bereitgestellt wird:

```
# tftp localhost
tftp> get FreeBSD/install/boot/pxeboot
Received 264951 bytes in 0.1 seconds
```

9. Bearbeiten Sie ${NFSROOTDIR}/etc/fstab und erstellen Sie einen Eintrag, um das Root-Dateisystem über NFS einzuhängen:

```
# Device                                        Mountpoint    FSType   Options
Dump Pass$
        myhost.example.com:/b/tftpboot/FreeBSD/install       /         nfs      ro
0    0
```

   Ersetzen Sie *myhost.example.com* durch den Hostnamen oder die IP-Adresse des NFS-Servers. In diesem Beispiel wird das Root-Dateisystem schreibgeschützt eingehangen, um ein potenzielles Löschen des Inhalts durch die NFS-Clients zu verhindern.

10. Setzen Sie das root-Passwort in der PXE-Umgebung für Client-Maschinen, die über PXE starten:

```
# chroot ${NFSROOTDIR}
# passwd
```

11. Falls erforderlich, aktivieren Sie ssh(1) root-Logins für Client-Maschinen, die über PXE starten, indem Sie die Option `PermitRootLogin` in ${NFSROOTDIR}/etc/ssh/sshd_config aktivieren. Dies ist in sshd_config(5) dokumentiert.

12. Führen Sie alle weiteren Anpassungen der PXE-Umgebung in ${NFSROOTDIR} durch, wie zum Beispiel die Installation weiterer Pakete, oder dass Bearbeiten der Passwortdatei mit vipw(8).

Booten Sie von einem NFS-Root-Volume, so erkennt /etc/rc dies und startet daraufhin das /etc/rc.initdiskless Skript. Lesen Sie die Kommentare in diesem Skript um zu verstehen, was dort vor sich geht. Weil das NFS-Root-Verzeichnis schreibgeschützt ist, wir aber Schreibzugriff für /etc und /var benötigen, müssen wir diese Verzeichnisse über Speicher-Dateisysteme (memory backed file system) einbinden.

```
# chroot ${NFSROOTDIR}
# mkdir -p conf/base
# tar -c -v -f conf/base/etc.cpio.gz --format cpio --gzip etc
```

```
# tar -c -v -f conf/base/var.cpio.gz --format cpio --gzip var
```

Wenn das System bootet, werden Speicher-Dateisysteme für /etc und /var erstellt und eingehangen. Anschließend wird der Inhalt der cpio.gz-Dateien in diese Dateisysteme kopiert. Standardmäßig haben diese Dateisysteme eine maximale Kapazität von 5 Megabyte. Wenn die Archive nicht passen, was für gewöhnlich bei /var der Fall ist, erhöhen Sie die Kapazität indem Sie die Anzahl der benötigten 512 Byte Sektoren (5 Megabyte sind 10240 Sektoren) in ${NFSROOTDIR}/conf/base/etc/md_size und ${NFSROOTDIR}/conf/base/var/md_size für die Dateisysteme /etc und /var eintragen.

### 54.8.2. Konfiguration des DHCP-Servers

Der DHCP-Server muss nicht auf derselben Maschine laufen wie der TFTP- und NFS-Server, aber er muss über das Netzwerk erreichbar sein.

DHCP ist nicht Bestandteil des FreeBSD Basissystems, kann jedoch über den Port net/isc-dhcp44-server oder als Paket nachinstalliert werden.

Einmal installiert, bearbeiten Sie die Konfigurationsdatei /usr/local/etc/dhcpd.conf. Konfigurieren Sie die `next-server`, `filename` und `root-path` Einstellungen, wie in diesem Beispiel zu sehen ist:

```
subnet 192.168.0.0 netmask 255.255.255.0 {
range 192.168.0.2 192.168.0.3;
option subnet-mask 255.255.255.0;
option routers 192.168.0.1;
option broadcast-address 192.168.0.255;
option domain-name-servers 192.168.35.35, 192.168.35.36;
option domain-name "example.com";

# IP address of TFTP server
next-server 192.168.0.1;

# path of boot loader obtained via tftp
filename "FreeBSD/install/boot/pxeboot";

# pxeboot boot loader will try to NFS mount this directory for root FS
option root-path "192.168.0.1:/b/tftpboot/FreeBSD/install/";
}
```

Die Anweisung `next-server` wird benutzt, um die IP-Adresse des TFTP-Servers anzugeben.

Die Anweisung `filename` definiert den Pfad zu /boot/pxeboot. Da hier der relative Dateiname verwendet wird, bedeutet das, dass /b/tftpboot nicht im Pfad enthalten ist.

Die Option `root-path` bestimmt den Pfad zum NFS root-Dateisystem.

Sobald die Änderungen gespeichert werden, aktivieren Sie DHCP beim Systemstart, indem Sie die folgende Zeile in /etc/rc.conf hinzufügen:

```
dhcpd_enable="YES"
```

Starten Sie anschließend den DHCP-Dienst:

```
# service isc-dhcpd start
```

### 54.8.3. Fehlersuche bei PXE Problemen

Sobald alle Dienste konfiguriert und gestartet wurden, sollten PXE-Clients in der Lage sein, FreeBSD automatisch über das Netzwerk zu starten. Wenn ein bestimmter Client beim hochfahren keine Verbindung herstellen kann, sehen Sie im BIOS nach, ob die Option für den Start über das Netzwerk konfiguriert ist.

Dieser Abschnitt gibt einige Tipps zu Fehlerbehebung und zeigt, wie Sie Konfigurationsprobleme eingrezen können für den Fall, dass PXE-Clients nicht in der Lage sind über das Netzwerk zu starten.

1. Benutzen Sie den net/wireshark Port um Fehler im Netzwerkverkehr während des Bootvorgangs von PXE zu finden. Der Bootvorgang wird im folgenden Diagramm schematisch dargestellt.

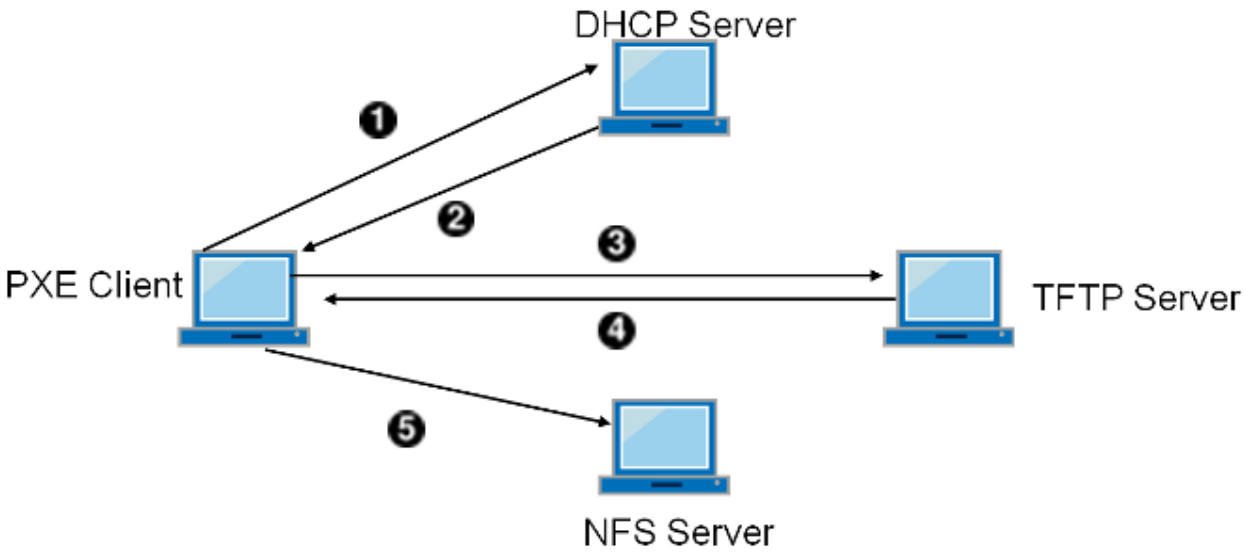



*Abbildung 60. PXE-Bootvorgang mit NFS Root Mount*

2. Schauen Sie in /var/log/xferlog auf dem TFTP-Server und vergewissern Sie sich, dass die pxeboot -Datei von der richtigen Adresse heruntergeladen wurde. Um die obige Konfiguration von /usr/local/etc/dhcpd.conf zu testen, geben Sie folgendes ein:

```
# tftp 192.168.0.1
tftp> get FreeBSD/install/boot/pxeboot
Received 264951 bytes in 0.1 seconds
```

Weitere Informationen finden Sie in tftpd(8) und tftp(1). Die BUGS-Sektionen dieser Seiten

dokumentieren einige Einschränkungen von TFTP.

3. Achten Sie darauf, dass Sie das Root-Dateisystem über NFS einhängen können. Auch hier können Sie Ihre Einstellungen aus /usr/local/etc/dhcpd.conf wie folgt testen:

```
# mount -t nfs 192.168.0.1:/b/tftpboot/FreeBSD/install /mnt
```

## 54.9. IPv6

IPv6 ist die neueste Version des bekannten IP-Protokolls, das auch als IPv4 bezeichnet wird. IPv6 bietet gegenüber IPv4 mehrere Vorteile sowie viele neue Funktionen:

- IPv6 hat einen 128 Bit großen Adressraum, der 340.282.366.920.938.463.463.374.607.431.768.211.456 Adressen erlaubt. Dies behebt das Problem der immer knapper werdenden IPv4-Adressen und einer eventuellen Erschöpfung des IPv4-Adressraums.
- Router speichern nur noch Netzwerk-Aggregationsadressen in ihren Routingtabellen. Dadurch reduziert sich die durchschnittliche Größe einer Routingtabelle auf 8192 Einträge. Dies ist mit den Problemen bei der Skalierbarkeit von IPv4 verbunden, da jeder zugeordnete Block von IPv4-Adressen erfordert, dass Routing-Informationen zwischen vielen Routern im Internet ausgetauscht werden müssen. Die Routing-Tabellen wurden mit der Zeit so groß, dass ein effizientes Routing jetzt kaum noch möglich ist.
- Die automatische Konfiguration von Adressen, die im RFC2462 beschrieben wird.
- Verpflichtende Multicast-Adressen.
- Integriertes IPsec (IP-Security).
- Eine vereinfachte Headerstruktur.
- Unterstützung für mobile IP-Adressen.
- Die Umwandlung von IPv4- in IPv6-Adressen.

FreeBSD enthält die IPv6-Referenzimplementation von KAME und erfüllt damit bereits alle für die Nutzung von IPv6 nötigen Voraussetzungen. Dieser Abschnitt konzentriert sich auf die Konfiguration und den Betrieb von IPv6.

### 54.9.1. Hintergrundinformationen zu IPv6-Adressen

Es gibt verschiedene Arten von IPv6-Adressen:

**Unicast**

Ein Paket, das an eine Unicast-Adresse gesendet wird, kommt nur an der Schnittstelle an, die dieser Adresse zugeordnet ist.

**Anycast**

Anycast-Adressen unterscheiden sich in ihrer Syntax nicht von Unicast-Adressen, sie wählen allerdings aus mehreren Schnittstellen eine Schnittstelle aus. Ein für eine Anycast-Adresse

bestimmtes Paket kommt an der nächstgelegenen (entsprechend der Router-Metrik) Schnittstelle an. Anycast-Adressen werden nur von Routern verwendet.

**Mulitcast**

Multicast-Adressen bestimmen Gruppen, denen mehrere Schnittstellen angehören. Ein Paket, das an eine Multicast-Adresse geschickt wird, kommt an allen Schnittstellen an, die zur Multicast-Gruppe gehören. Die von IPv4 bekannte Broadcast-Adresse (normalerweise `xxx.xxx.xxx.255`) wird bei IPv6 durch Multicast-Adressen verwirklicht.

Die kanonische Form einer IPv6-Adresse lautet `x:x:x:x:x:x:x:x`, wobei jedes "x" für einen 16-Bit-Hexadezimalwert steht. Ein Beispiel für eine IPv6-Adresse wäre etwa `FEBC:A574:382B:23C1:AA49:4592:4EFE:9982`.

Eine IPv6-Adresse enthält oft Teilzeichenfolgen aus lauter Nullen. Eine solche Zeichenfolge kann zu "::" verkürzt werden. Bis zu drei führende Nullen eines Hexquads können ebenfalls weggelassen werden. `fe80::1` entspricht also der Adresse `fe80:0000:0000:0000:0000:0000:0000:0001`.

Eine weitere Möglichkeit ist die Darstellung der letzten 32 Bit in der bekannten IPv4-Notation. `2002::10.0.0.1` ist also eine andere Schreibweise für die (hexadezimale) kanonische Form `2002:0000:0000:0000:0000:0000:0a00:0001`, die wiederum der Adresse `2002::a00:1` entspricht.

Benutzen Sie ifconfig(8), um die IPv6-Adresse eines FreeBSD-Systems anzuzeigen:

```
# ifconfig
rl0: flags=8943<UP,BROADCAST,RUNNING,PROMISC,SIMPLEX,MULTICAST> mtu 1500
         inet 10.0.0.10 netmask 0xffffff00 broadcast 10.0.0.255
         inet6 fe80::200:21ff:fe03:8e1%rl0 prefixlen 64 scopeid 0x1
         ether 00:00:21:03:08:e1
         media: Ethernet autoselect (100baseTX )
         status: active
```

Bei `fe80::200:21ff:fe03:8e1%rl0` handelt es sich um eine automatisch konfigurierte link-local-Adresse. Sie wird im Rahmen der automatischen Konfiguration aus der MAC-Adresse erzeugt.

Einige IPv6-Adressen sind reserviert. Eine Zusammenfassung dieser Adressen finden Sie in Reservierte IPv6-Adressen:

*Tabelle 32. Reservierte IPv6-Adressen*

| IPv6-Adresse | Präfixlänge | Beschreibung | Anmerkungen |
|---|---|---|---|
| `::` | 128 Bit | nicht festgelegt | entspricht `0.0.0.0` bei IPv4. |
| `::1` | 128 Bit | Loopback-Adresse | entspricht `127.0.0.1` bei IPv4. |
| `::00:xx:xx:xx:xx` | 96 Bit | Eingebettete IPv4-Adresse | Die niedrigen 32 Bit sind die kompatiblen IPv4-Adressen. |

| IPv6-Adresse | Präfixlänge | Beschreibung | Anmerkungen |
|---|---|---|---|
| `::ff:xx:xx:xx:xx` | 96 Bit | Eine auf IPv6 abgebildete IPv4-Adresse. | Die niedrigen 32 Bit sind IPv4-Adressen für Hosts, die kein IPv6 unterstützen. |
| `fe80::/10` | 10 Bit | link-local | Entspricht 196.254.0.0/16 bei IPv4. |
| `fc00::/7` | 7 Bit | unique-local | Diese einzigartigen Adressen sind für die lokale Kommunikation bestimmt und werden nur innerhalb von abgegrenzten Standorten (Sites) weitergeleitet. |
| `ff00::` | 8 Bit | Multicast | |
| `2000::-3fff::` | 3 Bit | Globaler Unicast | Alle globalen Unicast-Adressen stammen aus diesem Pool. Die ersten 3 Bit lauten `001`. |

Weitere Informationen zum Aufbau von IPv6-Adressen finden Sie im RFC3513.

### 54.9.2. IPv6 konfigurieren

Um ein FreeBSD-System als IPv6-Client zu konfigurieren, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
ifconfig_rl0_ipv6="inet6 accept_rtadv"
rtsold_enable="YES"
```

Die erste Zeile ermöglicht der angegebenen Schnittstelle, Router-Advertisement-Nachrichten zu empfangen. Die zweite Zeile aktiviert den Router-Solicitation-Daemon rtsold(8).

Falls die Schnittstelle eine statisch zugewiesene IPv6-Adresse benötigt, fügen Sie einen Eintrag mit der statischen Adresse und dem zugehörigen Präfix für das Subnetz hinzu:

```
ifconfig_rl0_ipv6="inet6 2001:db8:4672:6565:2026:5043:2d42:5344 prefixlen 64"
```

Um einen Standardrouter festzulegen, fügen Sie die Adresse hinzu:

```
ipv6_defaultrouter="2001:db8:4672:6565::1"
```

### 54.9.3. Verbindung zu einem Provider aufbauen

Um sich mit anderen IPv6-Netzwerken zu verbinden, benötigen Sie einen Provider oder einen Tunnel, der IPv6 unterstützt:

- Fragen Sie einen Internetprovider, ob er IPv6 anbietet.
- Hurricane Electric bietet weltweit IPv6-Tunnelverbindungen an.



Die Verwendung des Ports /usr/ports/net/freenet6 für Einwahlverbindungen.

Dieser Abschnitt beschreibt, wie Sie die Anweisungen eines Tunnel-Providers dauerhaft in /etc/rc.conf einrichten.

Der erste Eintrag in /etc/rc.conf erzeugt die generische Tunnelschnittstelle gif0:

```
cloned_interfaces="gif0"
```

Als nächstes konfigurieren Sie die IPv4-Adressen der lokalen und entfernten Endpunkte. Ersetzen Sie *MY_IPv4_ADDR* und *REMOTE_IPv4_ADDR* durch die tatsächlichen IPv4-Adressen:

```
cloned_interfaces_gif0="MY_IPv4_ADDR REMOTE_IPv4_ADDR"
```

Um die zugewiesene IPv6-Adresse als Endpunkt für den IPv6-Tunnel zu verwenden, fügen Sie folgende Zeile für FreeBSD 9.*x* (und neuer) ein:

```
ifconfig_gif0_ipv6="inet6 MY_ASSIGNED_IPv6_TUNNEL_ENDPOINT_ADDR"
```

Legen Sie dann die Standardroute für das andere Ende des IPv6-Tunnels fest. Ersetzen Sie *MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR* mit der Adresse des Standard-Gateways des Providers:

```
ipv6_defaultrouter="MY_IPv6_REMOTE_TUNNEL_ENDPOINT_ADDR"
```

Wenn das FreeBSD-System IPv6-Verkehr zwischen dem Netzwerk und der Außenwelt routen muss, aktivieren Sie das Gateway mit dieser Zeile:

```
ipv6_gateway_enable="YES"
```

### 54.9.4. Bekanntmachung von Routen und automatische Rechnerkonfiguration

Dieser Abschnitt beschreibt die Einrichtung von rtadvd(8), das Sie bei der Bekanntmachung der IPv6-Standardroute unterstützt.

Um rtadvd(8) zu aktivieren, fügen Sie folgende Zeile in /etc/rc.conf ein:

```
rtadvd_enable="YES"
```

Es ist wichtig, die Schnittstelle anzugeben, über die IPv6-Routen bekanntgemacht werden sollen. Soll rtadvd(8) rl0 verwenden, ist folgender Eintrag nötig:

```
rtadvd_interfaces="rl0"
```

Danach erzeugen Sie die Konfigurationsdatei /etc/rtadvd.conf. Dazu ein Beispiel:

```
rl0:\
    :addrs#1:addr="2001:db8:1f11:246::":prefixlen#64:tc=ether:
```

Ersetzen Sie dabei fxp0 durch die zu verwendende Schnittstelle, und `2001:db8:1f11:246::` durch das entsprechend zugewiesene Präfix.

Bei einem `/64`-Subnetz müssen keine weiteren Anpassungen vorgenommen werden. Anderenfalls muss `prefixlen#` auf den korrekten Wert gesetzt werden.

### 54.9.5. IPv6 und Abbildung von IPv6-Adressen

Wenn IPv6 auf einem Server aktiviert ist, kann es für die Kommunikation erforderlich sein, IPv4-Adressen auf IPv6-Adressen abzubilden. Diese Kompatibilität erlaubt es, das IPv4-Adressen als IPv6-Adressen dargestellt werden. Die Kommunikation von IPv6-Anwendungen mit IPv4 und umgekehrt kann jedoch ein Sicherheitsrisiko darstellen.

Diese Option dient nur der Kompatibilität und wird in den meisten Fällen nicht erforderlich sein. Die Option ermöglicht es IPv6-Anwendungen zusammen mit IPv4 in einer Dual-Stack-Umgebung zu funktionieren. Dies ist besonders nützlich für Anwendungen von Drittanbietern, die evtl. keine IPv6-Umgebungen unterstützen. Um diese Funktion zu aktivieren, fügen Sie folgendes in /etc/rc.conf hinzu:

```
ipv6_ip4mapping="YES"
```

Für einige Administratoren können die Informationen im RFC 3493 (Sektion 3.6 und 3.7) und RFC 4038 (Sektion 4.2) hilfreich sein.

## 54.10. Common Address Redundancy Protocol (CARP)

Das Common Address Redundancy Protocol (CARP) erlaubt es, mehreren Rechnern die gleiche IP-Adresse und Virtual Host ID (VHID) zuzuweisen und *Hochverfügbarkeit* bereitzustellen. Das bedeutet, dass ein oder mehrere Rechner ausfallen können und die anderen Rechner transparent einspringen, ohne dass die Benutzer etwas von einem Ausfall mitbekommen.

Neben der gemeinsamen IP-Adresse, haben die jeweiligen Rechner auch eine eindeutige IP-Adresse zur Verwaltung und Konfiguration. Alle Maschinen, die sich eine IP-Adresse teilen, verwenden die gleiche VHID. Die VHID für jede einzelne IP-Adresse muss, entsprechend der Broadcast-Domäne der Netzwerkschnittstelle, eindeutig sein.

Hochverfügbarkeit mit CARP ist in FreeBSD enthalten, jedoch unterscheidet sich die Konfiguration von der eingesetzten FreeBSD-Version. Dieser Abschnitt enthält die gleichen Konfigurationsdateien für verschiedene Versionen von FreeBSD.

Dieses Beispiel konfiguriert eine Failover-Unterstützung mit drei Servern (mit jeweils eigener, eindeutiger IP-Adresse), die alle den gleichen Web-Inhalt anbieten. Es werden zwei verschiedene Master namens `hosta.example.org` und `hostb.example.org` benutzt, mit einem gemeinsamen Backup namens `hostc.example.org`.

Die Lastverteilung dieser Maschinen wird dabei über Round RobinDNS konfiguriert. Mit Ausnahme des Hostnamens und der IP-Management-Adresse sind Master- und Backup-Maschinen identisch konfiguriert. Die Server müssen die gleiche Konfiguration und die gleichen Dienste aktiviert haben. Tritt ein Failover auf, können Anfragen an den Dienst mit der gemeinsam genutzten IP-Adresse nur dann richtig beantwortet werden, wenn der Backup-Server Zugriff auf denselben Inhalt hat. Die Backup-Maschine verfügt über zwei zusätzliche CARP-Schnittstellen, eine für jede IP-Adresse des Master-Content-Servers. Sobald ein Fehler auftritt, übernimmt der Backup-Server die IP-Adresse des ausgefallenen Master-Servers.

### 54.10.1. CARP mit FreeBSD 10 (und neuer) benutzen

Unterstützung für CARP erhalten Sie durch das Laden des Kernelmoduls carp.ko in /boot/loader.conf:

```
carp_load="YES"
```

So laden Sie das Modul ohne Neustart:

```
# kldload carp
```

Benutzer, die einen angepassten Kernel verwenden möchten, müssen die folgende Zeile in die Konfigurationsdatei aufnehmen. Anschließend muss der Kernel, wie in Konfiguration des FreeBSD-Kernels beschrieben, neu gebaut werden:

```
device  carp
```

Hostname, IP-Management-Adresse, Subnetzmaske, gemeinsame IP-Adresse und VHID werden durch Einträge in /etc/rc.conf gesetzt. Dieses Beispiel ist für `hosta.example.org`:

```
hostname="hosta.example.org"
ifconfig_em0="inet 192.168.1.3 netmask 255.255.255.0"
```

```
ifconfig_em0_alias0="inet vhid 1 pass testpass alias 192.168.1.50/32"
```

Die nächsten Einträge sind für `hostb.example.org`. Da der Rechner einen zweiten Master darstellt, verwendet er eine andere gemeinsame IP-Adresse und VHID. Die mittels `pass` angegebenen Passwörter müssen jedoch identisch sein, da CARP nur mit Systemen kommuniziert, die über das richtige Passwort verfügen.

```
hostname="hostb.example.org"
ifconfig_em0="inet 192.168.1.4 netmask 255.255.255.0"
ifconfig_em0_alias0="inet vhid 2 pass testpass alias 192.168.1.51/32"
```

Die dritte Maschine, `hostc.example.org` ist so konfiguriert, das sie aktiviert wird, wenn einer der beiden Masterserver ausfällt. Diese Maschine ist mit zwei CARPVHIDs konfiguriert, eine für jede virtuelle IP-Adresse der beiden Master-Server. Die CARP advertising skew, `advskew` wird gesetzt, um sicherzustellen, dass sich der Backup-Server später ankündigt wie der Master-Server, da `advskew` die Rangfolge steuert für den Fall, dass mehrere Backup-Server zur Verfügung stehen.

```
hostname="hostc.example.org"
ifconfig_em0="inet 192.168.1.5 netmask 255.255.255.0"
ifconfig_em0_alias0="inet vhid 1 advskew 100 pass testpass alias 192.168.1.50/32"
ifconfig_em0_alias1="inet vhid 2 advskew 100 pass testpass alias 192.168.1.51/32"
```

Durch die beiden konfigurierten CARPVHIDs ist `hostc.example.org` in der Lage festzustellen, wenn einer der Master-Server nicht mehr reagiert. Wenn der Master-Server sich später ankündigt als der Backup-Server, übernimmt der Backup-Server die gemeinsame IP-Adresse, bis der Master-Server erneut verfügbar ist.



Auch wenn der ursprüngliche Master-Server wieder verfügbar wird, gibt `hostc.example.org` die virtuelle IP-Adresse nicht automatisch wieder frei. Dazu muss Preemption aktiviert werden. Preemption ist standardmäßig deaktiviert und wird über die sysctl(8)-Variable `net.inet.carp.preempt` gesteuert. Der Administrator kann den Backup-Server zwingen, die IP-Adresse an den Master zurückzugeben:

```
# ifconfig em0 vhid 1 state backup
```

Sobald die Konfiguration abgeschlossen ist, muss das Netzwerk oder die Maschine neu gestartet werden. Hochverfügbarkeit ist nun aktiviert.

Die Funktionalität von CARP kann, wie in der Manualpage carp(4) beschrieben, über verschiedene sysctl(8) Parameter kontrolliert werden. Mit dem Einsatz von devd(8) können weitere Aktionen zu CARP-Ereignissen ausgelöst werden.

### 54.10.2. CARP mit FreeBSD 9 (und älter) benutzen

Die Konfiguration für diese Versionen von FreeBSD ist ähnlich wie im vorhergehenden Abschnitt beschrieben, mit der Ausnahme, dass zuerst ein CARP-Gerät in der Konfiguration erstellt und bezeichnet werden muss.

Unterstützung für CARP erhalten Sie durch das Laden des Kernelmoduls carp.ko in /boot/loader.conf:

```
if_carp_load="YES"
```

So laden Sie das Modul ohne Neustart:

```
# kldload carp
```

Benutzer, die einen angepassten Kernel verwenden möchten, müssen die folgende Zeile in die Konfigurationsdatei aufnehmen. Anschließend muss der Kernel, wie in Konfiguration des FreeBSD-Kernels beschrieben, neu gebaut werden:

```
device    carp
```

Als nächstes erstellen Sie auf jedem Rechner eine CARP-Schnittstelle:

```
# ifconfig carp0 create
```

Konfigurieren Sie Hostnamen, IP-Management-Adresse, die gemeinsam genutzte IP-Adresse und die VHID, indem Sie die erforderlichen Zeilen in /etc/rc.conf hinzufügen. Da anstelle eines Alias eine virtuelles CARP-Gerät verwendet wird, wird die tatsächliche Subnetzmaske `/24` anstatt `/32` benutzt. Hier sind die Einträge für `hosta.example.org`:

```
hostname="hosta.example.org"
ifconfig_fxp0="inet 192.168.1.3 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 1 pass testpass 192.168.1.50/24"
```

Beispiel für `hostb.example.org`:

```
hostname="hostb.example.org"
ifconfig_fxp0="inet 192.168.1.4 netmask 255.255.255.0"
cloned_interfaces="carp0"
ifconfig_carp0="vhid 2 pass testpass 192.168.1.51/24"
```

Die dritte Maschine, `hostc.example.org` ist so konfiguriert, das sie aktiviert wird, wenn einer der

beiden Masterserver ausfällt:

```
hostname="hostc.example.org"
ifconfig_fxp0="inet 192.168.1.5 netmask 255.255.255.0"
cloned_interfaces="carp0 carp1"
ifconfig_carp0="vhid 1 advskew 100 pass testpass 192.168.1.50/24"
ifconfig_carp1="vhid 2 advskew 100 pass testpass 192.168.1.51/24"
```



Preemption ist im GENERIC-Kernel deaktiviert. Haben Sie jedoch Preemption in einem angepassten Kernel aktiviert, dass `hostc.example.org` die virtuelle IP-Adresse nicht wieder an den Master-Server zurückgibt. Der Administrator kann jedoch den Backup-Server dazu zwingen, die übernommene IP-Adresse wieder an den Master-Server zurückzugeben:

```
# ifconfig carp0 down && ifconfig carp0 up
```

Dieser Befehl muss auf dem carp-Gerät ausgeführt werden, dass dem betroffenen System zugeordnet ist.

Sobald die Konfiguration abgeschlossen ist, muss das Netzwerk oder die Maschine neu gestartet werden. Hochverfügbarkeit ist nun aktiviert.

## 54.11. VLANs

VLANs sind eine Möglichkeit ein Netzwerk virtuell in viele Subnetze zu unterteilen. Man spricht hier auch von Segmentierung. Jedes Subnetz hat seine eigene Broadcast-Domäne und ist von anderen VLANs isoliert.

Unter FreeBSD müssen VLANs vom Treiber der Netzwerkkarte unterstützt werden. vlan(4) enthält eine Liste von Treibern mit integrierter VLAN-Unterstützung.

Für die Konfiguration eines VLAN werden zwei Informationen benötigt: die verwendete Netzwerkschnittstelle und das VLAN-Tag.

Das folgende Kommando konfiguriert ein VLAN mit der Netzwerkschnittstelle `em0` und dem VLAN-Tag `5`:

```
# ifconfig em0.5 create vlan 5 vlandev em0 inet 192.168.20.20/24
```



In diesem Beispiel fällt auf, dass der Name der Schnittstelle den Treibernamen und das VLAN-Tag enthält, getrennt durch einen Punkt. Diese Methode hat sich bewährt, da sie die Konfiguration von Systemen mit mehreren VLANs deutlich erleichtert.

Um VLANs beim Booten zu konfigurieren, muss /etc/rc.conf angepasst werden. Für das obige

Beispiel müssten folgende Zeilen in die Konfiguration aufgenommen werden:

```
vlans_em0="5"
ifconfig_em0_5="inet 192.168.20.20/24"
```

Das gleiche Schema kann benutzt werden, um weitere VLANs hinzuzufügen.

Es ist sinnvoll, einer Schnittstelle einen symbolischen Namen zuzuweisen, so dass bei einem Wechsel der zugehörigen Hardware nur wenige Konfigurationsvariablen aktualisiert werden müssen. Nehmen wir beispielsweise an, dass Überwachungskameras im VLAN1 auf `em0` betrieben werden. Wenn später die Karte `em0` durch eine Karte ersetzt wird, die den ixgb(4) Treiber verwendet, müssen nicht alle Referenzen auf `em0.1` durch `ixgb0.1` ersetzt werden.

Der folgende Befehl konfiguriert VLAN `5` auf der Netzwerkkarte `em0`. Die Schnittstelle bekommt den Namen `cameras` und eine IP-Adresse `192.168.20.20` mit einem `24`-Bit Präfix.

```
# ifconfig em0.5 create vlan 5 vlandev em0 name cameras inet 192.168.20.20/24
```

Dieser Befehl konfiguriert die Schnittstelle mit dem Namen `video`:

```
# ifconfig video.5 create vlan 5 vlandev video name cameras inet 192.168.20.20/24
```

Um die Änderungen beim Booten anzuwenden, fügen Sie folgenden Zeilen in /etc/rc.conf ein:

```
vlans_video="cameras"
create_args_cameras="vlan 5"
ifconfig_cameras="inet 192.168.20.20/24"
```

params: path: "/books/handbook/partv/" :sectnums!: :leveloffset: +1

# Anhang A: Bezugsquellen für FreeBSD

# Kapitel 55. CD and DVD Sets

Die FreeBSD-CDs und -DVDs werden von verschiedenen Online-Händlern angeboten:

- FreeBSD Mall, Inc.
  2420 Sand Creek Rd C-1 #347
  Brentwood, CA
  94513
  USA
  Phone: +1 925 240-6652
  Fax: +1 925 674-0821
  Email: <info@freebsdmall.com>
  WWW: https://www.freebsdmall.com
- Getlinux
  78 Rue de la Croix Rochopt
  Épinay-sous-Sénart
  91860
  France
  Email: <contact@getlinux.fr>
  WWW: http://www.getlinux.fr/
- Dr. Hinner EDV
  Kochelseestr. 11
  D-81371 München
  Germany
  Phone: (0177) 428 419 0
  Email: <infow@hinner.de>
  WWW: http://www.hinner.de/linux/freebsd.html
- Linux Center
  Galernaya Street, 55
  Saint-Petersburg
  190000
  Russia
  Phone: +7-812-309-06-86
  Email: <info@linuxcenter.ru>
  WWW: http://linuxcenter.ru/shop/freebsd

# Kapitel 56. FTP-Server

Die offiziellen Quellen von FreeBSD sind mit anonymous FTP über ein weltweites Netz von Spiegeln erhältlich. Die Seite ftp://ftp.FreeBSD.org/pub/FreeBSD/ ist über HTTP und FTP erreichbar. Sie besteht aus mehreren Servern, die von den Cluster-Administratoren des Projekts über GeoDNS betrieben wird, um Benutzer auf den nächsten verfügbaren Spiegel umzuleiten.

Sie können FreeBSD auch über anonymous FTP von den folgenden Spiegeln beziehen. Wenn Sie FreeBSD über anonymous FTP beziehen wollen, wählen Sie bitte einen Spiegel in Ihrer Nähe. Die unter "Haupt-Spiegel" aufgeführten Spiegel stellen normalerweise das komplette FreeBSD-Archiv (alle momentan erhältlichen Versionen für jede unterstützte Architektur) zur Verfügung. Wahrscheinlich geht es aber schneller, wenn Sie einen Spiegel in Ihrer Nähe benutzen. Die Länder-Spiegel stellen die neusten Versionen für die beliebtesten Architekturen bereit, sie stellen aber unter Umständen nicht das komplette FreeBSD-Archiv bereit. Auf alle Server kann mit anonymous FTP zugegriffen werden, einige Server bieten auch andere Zugriffsmethoden an. Die zur Verfügung stehenden Zugriffsmethoden sind bei jedem Server in Klammern angegeben.

Central Servers, Primary Mirror Sites, Armenia, Australia, {mirrors-austria}, Brazil, Czech Republic, Denmark, Estonia, Finland, France, Germany, Greece, Hong Kong, Ireland, Japan, Korea, Latvia, Lithuania, Netherlands, New Zealand, Norway, Poland, Russia, Saudi Arabia, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, Ukraine, United Kingdom, United States of America.

(aktualisiert am: UTC)

**Central Servers**

ftp://ftp.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.FreeBSD.org/pub/FreeBSD/ / http://ftp.FreeBSD.org/pub/FreeBSD/)

**Primary Mirror Sites**

Bei Problemen wenden Sie sich bitte an den Betreuer <mirror-admin@FreeBSD.org> dieser Domain.

- ftp://ftp1.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.FreeBSD.org/pub/FreeBSD/ / http://ftp4.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp10.FreeBSD.org/pub/FreeBSD/ / http://ftp10.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp11.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp14.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.FreeBSD.org/pub/FreeBSD/)

**Armenia**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@am.FreeBSD.org> dieser Domain.

- ftp://ftp1.am.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.am.FreeBSD.org/pub/FreeBSD/ / rsync)

**Australia**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@au.FreeBSD.org> dieser Domain.

- ftp://ftp.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.au.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.au.FreeBSD.org/pub/FreeBSD/ (ftp)

**{mirrors-austria}**

Bei Problemen wenden Sie sich bitte an den Betreuer <{mirrors-austria-email}> dieser Domain.

- {mirrors-austria-ftp} (ftp / ftpv6 / {mirrors-austria-ftp-http} / {mirrors-austria-ftp-httpv6})

**Brazil**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@br.FreeBSD.org> dieser Domain.

- ftp://ftp2.br.FreeBSD.org/FreeBSD/ (ftp / http://ftp2.br.FreeBSD.org/)
- ftp://ftp3.br.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp4.br.FreeBSD.org/pub/FreeBSD/ (ftp)

**Czech Republic**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@cz.FreeBSD.org> dieser Domain.

- ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / http://ftp.cz.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.cz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.cz.FreeBSD.org/pub/FreeBSD/)

**Denmark**

Bei Problemen wenden Sie sich bitte an den Betreuer <staff@dotsrc.org> dieser Domain.

- ftp://ftp.dk.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp.dk.FreeBSD.org/pub/FreeBSD/ / http://ftp.dk.FreeBSD.org/pub/FreeBSD/)

**Estonia**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@ee.FreeBSD.org> dieser Domain.

- ftp://ftp.ee.FreeBSD.org/pub/FreeBSD/ (ftp)

#### Finland

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@fi.FreeBSD.org> dieser Domain.

- ftp://ftp.fi.FreeBSD.org/pub/FreeBSD/ (ftp)

#### France

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@fr.FreeBSD.org> dieser Domain.

- ftp://ftp.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.fr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp1.fr.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp3.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.fr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp7.fr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.fr.FreeBSD.org/pub/FreeBSD/ (ftp)

#### Germany

Bei Problemen wenden Sie sich bitte an den Betreuer <de-bsd-hubs@de.FreeBSD.org> dieser Domain.

- ftp://ftp.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp1.de.FreeBSD.org/freebsd/ (ftp / http://www1.de.FreeBSD.org/freebsd/ / rsync://rsync3.de.FreeBSD.org/freebsd/)
- ftp://ftp2.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.de.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp4.de.FreeBSD.org/FreeBSD/ (ftp / http://ftp4.de.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.de.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.de.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp7.de.FreeBSD.org/pub/FreeBSD/)

#### Greece

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@gr.FreeBSD.org> dieser Domain.

- ftp://ftp.gr.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.gr.FreeBSD.org/pub/FreeBSD/ (ftp)

#### Hong Kong

ftp://ftp.hk.FreeBSD.org/pub/FreeBSD/ (ftp)

#### Ireland

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@ie.FreeBSD.org> dieser Domain.

- ftp://ftp3.ie.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

**Japan**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@jp.FreeBSD.org> dieser Domain.

- ftp://ftp.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp7.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.jp.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp9.jp.FreeBSD.org/pub/FreeBSD/ (ftp)

**Korea**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@kr.FreeBSD.org> dieser Domain.

- ftp://ftp.kr.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)
- ftp://ftp2.kr.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.kr.FreeBSD.org/pub/FreeBSD/)

**Latvia**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@lv.FreeBSD.org> dieser Domain.

- ftp://ftp.lv.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lv.FreeBSD.org/pub/FreeBSD/)

**Lithuania**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@lt.FreeBSD.org> dieser Domain.

- ftp://ftp.lt.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.lt.FreeBSD.org/pub/FreeBSD/)

**Netherlands**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@nl.FreeBSD.org> dieser Domain.

- ftp://ftp.nl.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nl.FreeBSD.org/os/FreeBSD/ / rsync)
- ftp://ftp2.nl.FreeBSD.org/pub/FreeBSD/ (ftp)

**New Zealand**

- ftp://ftp.nz.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.nz.FreeBSD.org/pub/FreeBSD/)

**Norway**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@no.FreeBSD.org> dieser Domain.

- ftp://ftp.no.FreeBSD.org/pub/FreeBSD/ (ftp / rsync)

**Poland**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@pl.FreeBSD.org> dieser Domain.

- ftp://ftp.pl.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp2.pl.FreeBSD.org

**Russia**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@ru.FreeBSD.org> dieser Domain.

- ftp://ftp.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ru.FreeBSD.org/FreeBSD/ / rsync)
- ftp://ftp2.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp2.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp5.ru.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp5.ru.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp6.ru.FreeBSD.org/pub/FreeBSD/ (ftp)

**Saudi Arabia**

Bei Problemen wenden Sie sich bitte an den Betreuer <ftpadmin@isu.net.sa> dieser Domain.

- ftp://ftp.isu.net.sa/pub/ftp.freebsd.org (ftp)

**Slovenia**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@si.FreeBSD.org> dieser Domain.

- ftp://ftp.si.FreeBSD.org/pub/FreeBSD/ (ftp)

**South Africa**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@za.FreeBSD.org> dieser Domain.

- {mirrors-south-africa-ftp} (ftp)
- ftp://ftp2.za.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.za.FreeBSD.org/pub/FreeBSD/ (ftp)

**Spain**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@es.FreeBSD.org> dieser Domain.

- ftp://ftp.es.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.es.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp3.es.FreeBSD.org/pub/FreeBSD/ (ftp)

**Sweden**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@se.FreeBSD.org> dieser Domain.

- ftp://ftp.se.FreeBSD.org/pub/FreeBSD/ (ftp)

- ftp://ftp2.se.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.se.FreeBSD.org/)
- ftp://ftp3.se.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / http://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/ / rsync://ftp4.se.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.se.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.se.FreeBSD.org/pub/FreeBSD/)

**Switzerland**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@ch.FreeBSD.org> dieser Domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ch.FreeBSD.org/pub/FreeBSD/)

**Taiwan**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@tw.FreeBSD.org> dieser Domain.

- ftp://ftp.ch.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ (ftp / ftp://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / http://ftp2.tw.FreeBSD.org/pub/FreeBSD/ / rsync / rsyncv6)
- ftp://ftp4.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.tw.FreeBSD.org/ / rsync)
- ftp://ftp7.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.tw.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp11.tw.FreeBSD.org/FreeBSD/)
- ftp://ftp12.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp14.tw.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp15.tw.FreeBSD.org/pub/FreeBSD/ (ftp)

**Ukraine**

- ftp://ftp.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp.ua.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp6.ua.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp6.ua.FreeBSD.org/pub/FreeBSD / rsync://ftp6.ua.FreeBSD.org/FreeBSD/)
- ftp://ftp7.ua.FreeBSD.org/pub/FreeBSD/ (ftp)

**United Kingdom**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@uk.FreeBSD.org> dieser Domain.

- ftp://ftp.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.uk.FreeBSD.org/pub/FreeBSD/ (ftp / rsync://ftp2.uk.FreeBSD.org/ftp.freebsd.org/pub/FreeBSD/)
- ftp://ftp3.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.uk.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp5.uk.FreeBSD.org/pub/FreeBSD/ (ftp)

**United States of America**

Bei Problemen wenden Sie sich bitte an den Betreuer <hostmaster@us.FreeBSD.org> dieser Domain.

- ftp://ftp1.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp2.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp3.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp4.us.FreeBSD.org/pub/FreeBSD/ (ftp / ftpv6 / http://ftp4.us.FreeBSD.org/pub/FreeBSD/ / http://ftp4.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp5.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp6.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp8.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp10.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp11.us.FreeBSD.org/pub/FreeBSD/ (ftp)
- ftp://ftp13.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp13.us.FreeBSD.org/pub/FreeBSD/ / rsync)
- ftp://ftp14.us.FreeBSD.org/pub/FreeBSD/ (ftp / http://ftp14.us.FreeBSD.org/pub/FreeBSD/)
- ftp://ftp15.us.FreeBSD.org/pub/FreeBSD/ (ftp)

# Kapitel 57. Benutzen von Subversion

## 57.1. Einführung

Seit Juli 2012 nutzt FreeBSD ausschließlich Subversion als Versionskontrollsystem zur Speicherung des gesamten FreeBSD Quellcodes, der Dokumentation und der Ports-Sammlung.



Subversion ist hauptsächlich ein Werkzeug für Entwickler. Die meisten Benutzer bevorzugen `freebsd-update` (FreeBSD-Update) um das FreeBSD Basissystem zu aktualisieren, und `portsnap` (Benutzen der Ports-Sammlung) um die FreeBSD Ports-Sammlung aktuell zu halten.

Dieser Abschnitt zeigt, wie Subversion unter FreeBSD installiert wird und wie Sie damit eine lokale Kopie des FreeBSD Repositories erstellen. Weitere Informationen über die Benutzung von Subversion sind ebenfalls enthalten.

## 57.2. SSL Root-Zertifikate

Die Installation von security/ca_root_nss erlaubt es Subversion die Identität des HTTPS-Repository-Servers zu überprüfen. Die SSL Root-Zertifikate können aus der Ports-Sammlung installiert werden:

```
# cd /usr/ports/security/ca_root_nss
# make install clean
```

Alternativ kann das Paket installiert werden:

```
# pkg install ca_root_nss
```

## 57.3. Svnlite

Mit `svnlite` enthält FreeBSD bereits eine vereinfachte Version von Subversion. Der Port oder das Paket ist nur erforderlich, wenn die Python oder Perl API benötigt wird, oder eine neuere Version von Subversion gewünscht ist.

Der einzige Unterschied zum normalen Subversion ist, dass der Name des Kommandos `svnlite` lautet.

## 57.4. Installation

Falls `svnlite` nicht verfügbar ist, oder die komplette Version von Subversion benötigt wird, muss das Programm installiert werden.

Subversion kann aus der Ports-Sammlung installiert werden:

```
# cd /usr/ports/devel/subversion
# make install clean
```

Subversion kann auch als Paket installiert werden:

```
# pkg install subversion
```

## 57.5. Subversion benutzen

Der `svn` Befehl wird verwendet, um eine Kopie der Quellen in ein lokales Verzeichnis zu holen. Die Dateien in diesem Verzeichnis werden *lokale Arbeitskopie* genannt.



Verschieben oder löschen Sie das Zielverzeichnis bevor Sie `checkout` benutzen.

In ein bestehendes nicht-`svn` Verzeichnis auszuchecken kann zu Konflikten zwischen den vorhandenen Dateien und denen aus dem Respository führen.

In Subversion werden URLs in der Form von *protocol://hostname/path* verwendet, um ein Repository zu kennzeichnen. Die erste Komponente des Pfades ist das FreeBSD Repository auf welches zugegriffen wird. Es gibt drei verschiedene Repositories. `base` für den Quellcode des FreeBSD Basissystems, `ports` für die Ports-Sammlung und `doc` für die Dokumentation. Als Beispiel spezifiziert die URL `svn://svn.FreeBSD.org/ports/head/` den Hauptzweig des Port-Repositories auf dem Mirror `svn.FreeBSD.org`, über das `svn`-Protokoll.

Das Auschecken aus einem bestimmten Repository kann wie folgt durchgeführt werden:

```
# svn checkout https://svn.FreeBSD.org/repository/branch lcwdir
```

wobei:

- *repository* eines der Projekt-Repositories ist: `base`, `ports` oder `doc`.
- *branch* vom verwendeten Repository abhängt. `ports` und `doc` werden meist im `head` Zweig aktualisiert, während `base` die neueste Version von -CURRENT unter `head` und die jeweilige neueste Version des -STABLE Zweiges unter `stable/9` (9.*x*) und `stable/10` (10.*x*) verwaltet wird.
- *lcwdir* das Zielverzeichnis ist, in dem die Inhalte des angegebenen Zweiges platziert werden sollen. Dies ist üblicherweise /usr/ports für `ports`, /usr/src für `base`, und /usr/doc für `doc`.

Dieses Beispiel checkt die Ports-Sammlung aus dem Repositroy über das HTTPS-Protokoll aus, und speichert die Arbeitskopie unter /usr/ports. Wenn /usr/ports bereits vorhanden ist, aber nicht von `svn` erstellt wurde, denken Sie vor dem Auschecken daran, das Verzeichnis umzubenennen oder zu löschen.

```
# svn checkout https://svn.FreeBSD.org/ports/head /usr/ports
```

Dies kann eine Weile dauern, da beim ersten Auschecken der komplette Zweig vom entfernten Repository heruntergeladen werden muss. Bitte haben Sie Geduld.

Nach dem ersten Auschecken können Sie Ihre lokale Arbeitskopie wie folgt aktualisieren:

```
# svn update lcwdir
```

Um /usr/ports aus dem oben erstellten Beispiel zu aktualisieren, benutzen Sie:

```
# svn update /usr/ports
```

Das Update ist viel schneller als ein Auschecken, da nur die Dateien übertragen werden müssen, die sich auch geändert haben.

Eine alternative Möglichkeit zur Aktualisierung Ihrer Arbeitskopie nach dem Auschecken ist es, das bestehende Makefile in den Verzeichnissen /usr/ports, /usr/src, und /usr/doc zu nutzen. Setzen Sie dazu `SVN_UPDATE` und benutzen Sie das `update` Ziel. Zum Beispiel, um /usr/src zu aktualisieren:

```
# cd /usr/src
# make update SVN_UPDATE=yes
```

## 57.6. Subversion Mirror Sites

Das FreeBSD Subversion Repository ist:

```
svn.FreeBSD.org
```

Dies ist ein öffentlich zugängliches Netzwerk aus Spiegeln, das GeoDNS verwendet, um einen entsprechenden Backend-Server auszuwählen. Um das FreeBSD Subversion Repository über einen Browser anzuzeigen, verwenden Sie http://svnweb.FreeBSD.org/.

HTTPS ist das bevorzugte Protokoll, jedoch muss das Paket security/ca_root_nss installiert werden, um Zertifikate automatisch zu validieren.

## 57.7. Weiterführende Informationen

Weitere Informationen über die Verwendung von Subversion finden Sie im "Subversion Buch" mit dem Namen Versionskontrolle mit Subversion, oder in der Subversion Dokumentation.

# Kapitel 58. Benutzen von rsync

Die folgenden Server stellen FreeBSD über das rsync-Protokoll zur Verfügung. Das Programm rsync überträgt lediglich geänderte Dateien und ist sehr nützlich, wenn Sie einen FreeBSD FTP-Spiegel betreiben. rsync ist für viele Betriebssysteme verfügbar. Für FreeBSD sehen Sie sich den Port oder das Paket net/rsync an.

**Großbritannien**

rsync://rsync.mirrorservice.org/

Verfügbare Sammlungen:

- ftp.freebsd.org: Kompletter Spiegel des FreeBSD FTP-Servers.

**Niederlande**

rsync://ftp.nl.FreeBSD.org/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD FTP-Servers.

**Russland**

rsync://ftp.mtu.ru/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD FTP-Servers.
- FreeBSD-Archive: Ein Spiegel des FreeBSD Archive-FTP-Servers.

**Schweden**

rsync://ftp4.se.freebsd.org/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD FTP-Servers.

**Taiwan**

rsync://ftp.tw.FreeBSD.org/

rsync://ftp2.tw.FreeBSD.org/

rsync://ftp6.tw.FreeBSD.org/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD FTP-Servers.

**Tschechische Republik**

rsync://ftp.cz.FreeBSD.org/

Verfügbare Sammlungen:

- ftp: Unvollständiger Spiegel des FreeBSD FTP-Servers.
- FreeBSD: Vollständiger Spiegel des FreeBSD FTP-Servers.

**USA**

rsync://ftp-master.FreeBSD.org/

Dieser Server darf nur von primären Spiegeln benutzt werden.

Verfügbare Sammlungen:

- FreeBSD: Das Hauptarchiv des FreeBSD FTP-Servers.
- acl: Die primäre ACL-Liste.

rsync://ftp13.FreeBSD.org/

Verfügbare Sammlungen:

- FreeBSD: Kompletter Spiegel des FreeBSD FTP-Servers.

# Anhang B: Bibliografie

Während die Manualpages eine definitive Referenz über bestimmte Teile des FreeBSD-Betriebssystems bieten, so können sie jedoch selten veranschaulichen, wie man die einzelnen Teile zusammenfügt, um ein vollständig laufendes Betriebssystem herzustellen. Daher gibt es keinen Ersatz für ein gutes Buch oder Benutzerhandbuch über die Administration von UNIX®-Systemen.

In der Regel handelt es sich im folgenden Kapitel um englische Ausgaben der genannten Werke. Übersetzungen oder Ausgaben in anderen Sprachen sind mit entsprechenden Hinweisen versehen.

## B.1. Bücher speziell für FreeBSD

*Internationale Bücher:*

- Using FreeBSD, herausgegeben von Drmaster, 1997 (in traditionellem Chinesisch). ISBN 9-578-39435-7.
- FreeBSD Unleashed (in vereinfachtem Chinesisch), herausgegeben von China Press. ISBN 7-111-10201-0.
- FreeBSD From Scratch Second Edition (in vereinfachtem Chinesisch), herausgegeben von China Press. ISBN 7-111-10286-X.
- FreeBSD Handbook Second Edition (in vereinfachtem Chinesisch), herausgegeben von Posts & Telecom Press. ISBN 7-115-10541-3.
- FreeBSD & Windows (in vereinfachtem Chinesisch), herausgegeben von China Railway Publishing House. ISBN 7-113-03845-X.
- FreeBSD Internet Services HOWTO (in vereinfachtem Chinesisch), herausgegeben von China Railway Publishing House. ISBN 7-113-03423-3.
- FreeBSD (in japanischer Sprache), herausgegeben von CUTT. ISBN 4-906391-22-2 C3055 P2400E.
- Complete Introduction to FreeBSD (in Japanese), published by Shoeisha Co., Ltd. ISBN 4-88135-473-6 P3600E.
- Personal UNIX Starter Kit FreeBSD (in japanischer Sprache), herausgegeben von ASCII. ISBN 4-7561-1733-3 P3000E.
- FreeBSD Handbook (japanische Übersetzung), herausgegeben von ASCII. ISBN 4-7561-1580-2 P3800E.
- FreeBSD mit Methode (in deutscher Sprache), herausgegeben von Computer und Literatur Verlag /Vertrieb Hanser, 1998. ISBN 3-932311-31-0.
- FreeBSD de Luxe (in German), published by Verlag Modere Industrie, 2003. ISBN 3-8266-1343-0.
- FreeBSD Install and Utilization Manual (in japanischer Sprache), herausgegeben von Mainichi Communications Inc., 1998. ISBN 4-8399-0112-0.
- Onno W Purbo, Dodi Maryanto, Syahrial Hubbany, Widjil Widodo *Building Internet Server with FreeBSD* (in indonesischer Sprache), herausgegeben von Elex Media Komputindo.
- Absolute BSD: The Ultimate Guide to FreeBSD (in traditionellem Chinesisch), herausgegeben von GrandTech Press, 2003. ISBN 986-7944-92-5.

- The FreeBSD 6.0 Book (in traditionellem Chinesisch, herausgegeben von Drmaster, 2006. ISBN 9-575-27878-X.

*Englischsprachige Bücher:*

- Absolute FreeBSD, 2nd Edition: The Complete Guide to FreeBSD, herausgegeben von No Starch Press, 2007. ISBN: 978-1-59327-151-0
- The Complete FreeBSD, herausgegeben von O'Reilly, 2003. ISBN: 0596005164
- The FreeBSD Corporate Networker's Guide, herausgegeben von Addison-Wesley, 2002. ISBN: 0201704811
- FreeBSD: An Open-Source Operating System for Your Personal Computer, herausgegeben von The Bit Tree Press, 2001. ISBN: 0971204500
- Teach Yourself FreeBSD in 24 Hours, herausgegeben von Sams, 2002. ISBN: 0672324245
- FreeBSD6 Unleashed, herausgegeben von Sams, 2006. ISBN: 0672328755
- FreeBSD: The Complete Reference, herausgegeben von McGrawHill, 2003. ISBN: 0072224096

## B.2. Handbücher

- Die Ohio State University hat ein UNIX Introductory Course veröffentlicht, welcher auch online im HTML- und PostScriptformat verfügbar ist.

  Eine italienische Übersetzung ist Teil des FreeBSD Italian Documentation Projects.

- Edinburgh University hat einen Online Guide für Anfänger in Sachen UNIX geschrieben.

## B.3. Administrations-Anleitungen

- Jpman Project, Japan FreeBSD Users Group. FreeBSD System Administrator's Manual (japanische Übersetzung). Mainichi Communications Inc., 1998. ISBN4-8399-0109-0 P3300E.
- Dreyfus, Emmanuel. Cahiers de l'Admin: BSD 2nd Ed. (in French), Eyrolles, 2004. ISBN 2-212-11463-X.

## B.4. Programmierhandbücher

- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Reference Manual*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-078-3
- Computer Systems Research Group, UC Berkeley. *4.4BSD Programmer's Supplementary Documents*. O'Reilly & Associates, Inc., 1994. ISBN 1-56592-079-1
- Harbison, Samuel P. and Steele, Guy L. Jr. *C: A Reference Manual*. 4th Ed. Prentice Hall, 1995. ISBN 0-13-326224-3
- Kernighan, Brian and Dennis M. Ritchie. *The C Programming Language*. 2nd Ed., PTR Prentice Hall, 1988. ISBN 0-13-110362-9
- Lehey, Greg. *Porting UNIX Software*. O'Reilly & Associates, Inc., 1995. ISBN 1-56592-126-7

- Plauger, P. J. *The Standard C Library*. Prentice Hall, 1992. ISBN 0-13-131509-9
- Spinellis, Diomidis. Code Reading: The Open Source Perspective. Addison-Wesley, 2003. ISBN 0-201-79940-5
- Spinellis, Diomidis. Code Quality: The Open Source Perspective. Addison-Wesley, 2006. ISBN 0-321-16607-8
- Stevens, W. Richard and Stephen A. Rago. *Advanced Programming in the UNIX Environment*. 2nd Ed. Reading, Mass. : Addison-Wesley, 2005. ISBN 0-201-43307-9
- Stevens, W. Richard. *UNIX Network Programming*. 2nd Ed, PTR Prentice Hall, 1998. ISBN 0-13-490012-X


## B.5. Betriebssystem-Interna


- Andleigh, Prabhat K. *UNIX System Architecture*. Prentice-Hall, Inc., 1990. ISBN 0-13-949843-5
- Jolitz, William. "Porting UNIX to the 386". *Dr. Dobb's Journal*. January 1991-July 1992.
- Leffler, Samuel J., Marshall Kirk McKusick, Michael J Karels and John Quarterman *The Design and Implementation of the 4.3BSD UNIX Operating System*. Reading, Mass. : Addison-Wesley, 1989. ISBN 0-201-06196-1

  Kapitel 2 dieses Buchs ist Teil des FreeBSD Documentation Projects und online erhältlich.

- Leffler, Samuel J., Marshall Kirk McKusick, *The Design and Implementation of the 4.3BSD UNIX Operating System: Answer Book*. Reading, Mass. : Addison-Wesley, 1991. ISBN 0-201-54629-9
- McKusick, Marshall Kirk, Keith Bostic, Michael J Karels, and John Quarterman. *The Design and Implementation of the 4.4BSD Operating System*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-54979-4
- Marshall Kirk McKusick, George V. Neville-Neil. *The Design and Implementation of the FreeBSD Operating System*. Boston, Mass. : Addison-Wesley, 2004. ISBN 0-201-70245-2
- Marshall Kirk McKusick, George V. Neville-Neil, Robert N. M. Watson *The Design and Implementation of the FreeBSD Operating System, 2nd Ed.*. Westford, Mass: Pearson Education, Ind., 2014. ISBN 0-321-96897-2
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 1: The Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63346-9
- Schimmel, Curt. *Unix Systems for Modern Architectures*. Reading, Mass. : Addison-Wesley, 1994. ISBN 0-201-63338-8
- Stevens, W. Richard. *TCP/IP Illustrated, Volume 3: TCP for Transactions, HTTP, NNTP and the UNIX Domain Protocols*. Reading, Mass. : Addison-Wesley, 1996. ISBN 0-201-63495-3
- Vahalia, Uresh. *UNIX Internals — The New Frontiers*. Prentice Hall, 1996. ISBN 0-13-101908-2
- Wright, Gary R. and W. Richard Stevens. *TCP/IP Illustrated, Volume 2: The Implementation*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63354-X

## B.6. Sicherheits-Anleitung


- Cheswick, William R. and Steven M. Bellovin. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-63357-4
- Garfinkel, Simson. *PGP Pretty Good Privacy* O'Reilly & Associates, Inc., 1995. ISBN 1-56592-098-8


## B.7. Hardware-Anleitung


- Anderson, Don and Tom Shanley. *Pentium Processor System Architecture*. 2nd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40992-5
- Ferraro, Richard F. *Programmer's Guide to the EGA, VGA, and Super VGA Cards*. 3rd ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-62490-7
- Die Intel Corporation veröffentlicht Dokumentationen Ihrer CPUs, Chipsets und Standards auf ihrer developer web site, normalerweise als PDF-Dateien.
- Shanley, Tom. *80486 System Architecture*. 3rd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40994-1
- Shanley, Tom. *ISA System Architecture*. 3rd Ed. Reading, Mass. : Addison-Wesley, 1995. ISBN 0-201-40996-8
- Shanley, Tom. *PCI System Architecture*. 4th Ed. Reading, Mass. : Addison-Wesley, 1999. ISBN 0-201-30974-2
- Van Gilluwe, Frank. *The Undocumented PC*, 2nd Ed. Reading, Mass: Addison-Wesley Pub. Co., 1996. ISBN 0-201-47950-8
- Messmer, Hans-Peter. *The Indispensable PC Hardware Book*, 4th Ed. Reading, Mass: Addison-Wesley Pub. Co., 2002. ISBN 0-201-59616-4


## B.8. UNIX® Geschichte


- Lion, John *Lion's Commentary on UNIX, 6th Ed. With Source Code*. ITP Media Group, 1996. ISBN 1573980137
- Raymond, Eric S. *The New Hacker's Dictionary, 3rd edition*. MIT Press, 1996. ISBN 0-262-68092-0. Auch bekannt als das Jargon File
- Salus, Peter H. *A quarter century of UNIX*. Addison-Wesley Publishing Company, Inc., 1994. ISBN 0-201-54777-5
- Simon Garfinkel, Daniel Weise, Steven Strassmann. *The UNIX-HATERS Handbook*. IDG Books Worldwide, Inc., 1994. ISBN 1-56884-203-1. Online verfügbar.
- Don Libes, Sandy Ressler *Life with UNIX* - special edition. Prentice-Hall, Inc., 1989. ISBN 0-13-536657-7
- *The BSD family tree*. https://svnweb.freebsd.org/base/head/shared/misc/bsd-family-tree=view=co oder unter /usr/shared/misc/bsd-family-tree auf einem FreeBSD-System.
- *Networked Computer Science Technical Reports Library*.
- *Old BSD releases from the Computer Systems Research group (CSRG).*

http://www.mckusick.com/csrg/: Das Paket mit 4 CD-ROMs enthält alle BSD-Versionen von 1BSD bis 4.4BSD und 4.4BSD-Lite2 (nicht aber 2.11BSD). Die letzte CD beinhaltet auch die finalen Sourcen inklusive den SCCS Dateien.

## B.9. Zeitschriften, Magazine und Journale

- Admin Magazin (in deutscher Sprache), herausgegeben von Medialinx AG. ISSN: 2190-1066
- BSD Magazine, herausgegeben von Software Press Sp. z o.o. SK. ISSN: 1898-9144
- BSD Now - Video Podcast, herausgegeben von Jupiter Broadcasting LLC
- BSD Talk Podcast, von Will Backman
- FreeBSD Journal, herausgegeben von S&W Publishing, gefördert durch The FreeBSD Foundation. ISBN: 978-0-615-88479-0

# Anhang C: Ressourcen im Internet

Gedruckte Medien können mit der schnellen Entwicklung von FreeBSD nicht Schritt halten. Elektronische Medien sind häufig die einzige Möglichkeit, über aktuelle Entwicklungen informiert zu sein. Da FreeBSD ein Projekt von Freiwilligen ist, gibt die Benutzergemeinde selbst auch technische Unterstützung. Die Benutzergemeinde erreichen Sie am besten über E-Mail, Internetforen oder Usenet-News.

Die wichtigsten Wege, auf denen Sie die FreeBSD-Benutzergemeinde erreichen können, sind unten dargestellt. Schicken Sie weitere Ressourcen, die hier fehlen, an die Mailingliste des FreeBSD documentation project, damit diese hier aufgenommen werden können.

## C.1. Webseiten

- Die FreeBSD Foren dienen als webbasiertes Diskussionsforum für Fragen und technische Diskussionen zu FreeBSD.
- Der BSDConferences YouTube-Kanal beinhaltet eine Sammlung von qualitativ hochwertigen Videos von BSD Konferenzen aus der ganzen Welt. Dies ist eine ausgezeichnete Art und Weise, den Entwicklern beim Präsentieren von neuen Arbeiten an FreeBSD zuzuschauen.

## C.2. Mailinglisten

Die Mailinglisten sind der direkteste Weg, um Fragen an das gesamte FreeBSD Publikum zu stellen oder eine technische Diskussion zu beginnen. Es existiert eine große Vielfalt von Listen mit einer Reihe von verschiedenen FreeBSD Themen. Wenn Sie Fragen an die richtige Mailingliste richten können Sie viel eher mit einer passenden Antwort darauf rechnen.

Die Chartas der verschiedenen Listen sind unten wiedergegeben. *Bevor Sie sich einer Mailingliste anschließen oder E-Mails an eine Liste senden, lesen Sie bitte die Charta der Liste.* Die meisten Mitglieder der Mailinglisten erhalten jeden Tag Hunderte E-Mails zum Thema FreeBSD. Die Chartas und Regeln, die den Gebrauch der Listen beschreiben, garantieren die hohe Qualität der Listen. Die Listen würden ihren hohen Wert für das Projekt verlieren, wenn wir weniger Regeln aufstellen würden.



*Um zu testen, ob Sie eine Nachricht an eine FreeBSD-Liste senden können, verwenden Sie bitte die Liste FreeBSD test.* Schicken Sie derartige Nachrichten bitte nicht an eine der anderen Listen.

Wenn Sie Sich nicht sicher sind, auf welcher Liste Sie Ihre Frage stellen sollen, sollten Sie den Artikel How to get best results from the FreeBSD-questions mailing list lesen.

Bevor Sie eine Nachricht an eine Mailingliste senden, sollten Sie die korrekte Nutzung der Mailinglisten erlernen. Dazu gehört auch das Vermeiden von sich häufig wiederholenden Diskussionen (lesen Sie deshalb zuerst die Mailing List Frequently Asked Questions).

Alle Mailinglisten werden archiviert und können auf dem FreeBSD World Wide Web Server durchsucht werden. Das nach Schlüsselwörtern durchsuchbare Archiv bietet die hervorragende

Möglichkeit, Antworten auf häufig gestellte Fragen zu finden. Nutzen Sie bitte diese Möglichkeit, bevor Sie Fragen auf einer Liste stellen. Beachten Sie auch, dass das zur Folge hat, dass die Nachrichten an die FreeBSD Mailinglisten für die Ewigkeit erhalten bleiben. Wenn Sie am Schutz Ihrer Privatsphäre interessiert sind, ziehen Sie die Verwendung einer Wegwerf-E-Mail-Adresse in Betracht und schreiben Sie nur solche Nachrichten, die für die Öffentlichkeit bestimmt sind.

### C.2.1. Beschreibung der Mailinglisten

*Allgemeine Listen:* Jeder kann die folgenden allgemeinen Listen abonnieren (und ist dazu aufgefordert):

| Mailingliste | Zweck |
|---|---|
| freebsd-advocacy | Verbreitung von FreeBSD |
| FreeBSD announcements | Wichtige Ereignisse und Meilensteine des Projekts (moderiert) |
| freebsd-arch | Architektur und Design von FreeBSD |
| freebsd-bugbusters | Diskussionen über die Pflege der FreeBSD Fehlerberichte-Datenbank und die dazu benutzten Werkzeuge |
| freebsd-bugs | Fehlerberichte |
| freebsd-chat | Nicht technische Themen, welche die FreeBSD-Gemeinschaft betreffen |
| freebsd-chromium | Diskussionen zum Einsatz von Chromium unter FreeBSD |
| FreeBSD-CURRENT | Gebrauch von FreeBSD-CURRENT |
| freebsd-isp | Für Internet-Service-Provider, die FreeBSD benutzen |
| freebsd-jobs | Anstellung und Beratung im FreeBSD-Umfeld |
| freebsd-quarterly-calls | Aufrufe für vierteljährliche Statusberichte (moderiert) |
| freebsd-questions | Benutzerfragen und technische Unterstützung |
| FreeBSD security notifications | Ankündigungen zum Thema Sicherheit (moderiert) |
| FreeBSD-STABLE; | Gebrauch von FreeBSD-STABLE |
| FreeBSD test | Schicken Sie Testnachrichten an diese Liste anstelle der wirklichen Listen |
| freebsd-women | FreeBSD Befürwortung von Frauen |

*Technische Listen:* Auf den folgenden Listen werden technische Diskussionen geführt. Bevor Sie eine der Listen abonnieren oder Nachrichten an sie schicken, lesen Sie sich die Charta der Liste durch, da der Inhalt und Zweck dieser Listen genau festgelegt ist.

| Mailingliste | Zweck |
| --- | --- |
| FreeBSD ACPI | Entwicklung von ACPI |
| freebsd-amd64 | Portierung von FreeBSD auf AMD64-Systeme (moderiert) |
| freebsd-apache | Diskussion über Ports, die mit Apache zusammenhängen. |
| freebsd-arm | Portierung von FreeBSD auf ARM®-Prozessoren |
| freebsd-atm | Benutzung von ATM-Netzen mit FreeBSD |
| freebsd-bluetooth | Bluetooth® unter FreeBSD verwenden |
| freebsd-cloud | FreeBSD auf Cloud-Plattformen (EC2, GCE, Azure, etc.) |
| freebsd-cluster | Benutzung von FreeBSD in einem Cluster |
| freebsd-database | Diskussion über Datenbanken und Datenbankprogrammierung unter FreeBSD |
| freebsd-desktop | FreeBSD als Desktop verwenden und verbessern |
| dev-ci | Build- und Testberichte von den Continuous Integration Servern |
| dev-reviews | Benachrichtigungen über das Review-System von FreeBSD |
| freebsd-doc | Erstellen der FreeBSD-Dokumentation |
| freebsd-drivers | Gerätetreiber für FreeBSD schreiben |
| freebsd-dtrace | Entwicklung und Benutzung von DTrace unter FreeBSD |
| freebsd-eclipse | Für FreeBSD-Anwender, welche die Eclipse IDE, deren Werkzeuge, Anwendungen und Ports einsetzen |
| freebsd-elastic | Diskussion zu ElasticSearch unter FreeBSD |
| freebsd-embedded | FreeBSD in eingebetteten Anwendungen einsetzen |
| freebsd-emulation | Emulation anderer Systeme wie Linux®, MS-DOS® oder Windows® |
| freebsd-enlightenment | Portierung von Enlightenment und Enlightenment-Applikationen |
| freebsd-eol | Support für FreeBSD-bezogene Software, die vom FreeBSD Project offiziell nicht mehr unterstützt wird. |
| freebsd-erlang | FreeBSD-spezifische Erlang-Diskussionen |

| Mailingliste | Zweck |
| --- | --- |
| freebsd-firewire | Technische Diskussion über FreeBSD FireWire® (iLink, IEEE 1394) |
| freebsd-fortran | Fortran unter FreeBSD |
| freebsd-fs | Dateisysteme |
| freebsd-games | Unterstützung für Spiele unter FreeBSD |
| freebsd-gecko | Angelegenheiten zur Gecko Rendering Engine |
| freebsd-geom | Diskussion über GEOM |
| freebsd-git | Diskussionen zur Verwendung von git im FreeBSD Project |
| freebsd-gnome | Portierung von GNOME und GNOME-Anwendungen |
| freebsd-hackers | Allgemeine technische Diskussionen |
| freebsd-haskell | FreeBSD-spezifische Haskell-Themen und Diskussionen |
| freebsd-hardware | Allgemeine Diskussion über Hardware, auf der FreeBSD läuft |
| freebsd-i18n | Internationalisierung von FreeBSD |
| freebsd-infiniband | Infiniband unter FreeBSD |
| freebsd-ipfw | Technische Diskussion über die Neubearbeitung der IP-Firewall Quellen |
| freebsd-isdn | Für Entwickler des ISDN-Systems |
| freebsd-java | Für Java™ Entwickler und Leute, die JDK™s nach FreeBSD portieren |
| freebsd-kde | Portierung von KDE und KDE-Anwendungen |
| freebsd-lfs | Portierung von LFS nach FreeBSD |
| freebsd-mips | Portierung von FreeBSD zu MIPS® |
| freebsd-mono | Mono und C# Anwendungen auf FreeBSD |
| FreeBSD multimedia | Multimedia Anwendungen |
| freebsd-new-bus | Technische Diskussionen über die Architektur von Bussen |
| freebsd-net | Diskussion über Netzwerke und den TCP/IP Quellcode |
| freebsd-numerics | Diskussionen über die Implementierung hochwertiger Funktionen in libm |
| freebsd-ocaml | FreeBSD-spezifische Diskussionen zu OCaml |
| freebsd-office | Office-Anwendungen für FreeBSD |

| Mailingliste | Zweck |
| --- | --- |
| freebsd-performance | Fragen zur Optimierung der Leistung stark ausgelasteter Systeme |
| freebsd-perl | Pflege der portierten Perl-Anwendungen. |
| freebsd-pf | Diskussionen und Fragen zu packet filter als Firewallsystem. |
| freebsd-pkg | Diskussionen über die Verwaltung von Binärpaketen und entsprechenden Werkzeugen |
| freebsd-pkg-fallout | Protokolle von fehlgeschlagenen Paketbauvorgängen |
| freebsd-pkgbase | Paketierung des FreeBSD-Basissystems |
| freebsd-platforms | Portierungen von FreeBSD auf nicht-Intel® Architekturen |
| freebsd-ports | Diskussion über die Ports-Sammlung |
| freebsd-ports-announce | Wichtige Neuigkeiten und Anweisungen zur Ports-Sammlung (moderiert) |
| freebsd-ports-bugs | Diskussion über Fehler und PRs der Ports |
| freebsd-ppc | Portierung von FreeBSD auf den PowerPC® |
| freebsd-proliant | Technische Diskussionen zum Einsatz von FreeBSD auf HP ProLiant-Serverplattformen |
| freebsd-python | FreeBSD-spezifische Diskussionen zu Python |
| freebsd-rc | Diskussion über das rc.d-System sowie dessen Weiterentwicklung |
| freebsd-realtime | Entwicklung von Echtzeiterweiterungen für FreeBSD |
| freebsd-riscv | Portierung von FreeBSD auf RISC-V®-Systeme |
| freebsd-ruby | FreeBSD-spezifische Diskussionen zu Ruby |
| freebsd-scsi | Diskussion über das SCSI-Subsystem |
| FreeBSD security | Sicherheitsthemen |
| freebsd-snapshots | Ankündigungen für FreeBSD Entwickler-Snapshots |
| freebsd-sparc64 | Portierung von FreeBSD auf SPARC® Systeme |
| freebsd-standards | Konformität von FreeBSD mit den C99- und POSIX®-Standards |
| freebsd-sysinstall | sysinstall(8) Entwicklung |
| freebsd-tcltk | FreeBSD spezifische Tcl/TK Diskussionen |
| freebsd-testing | Tests unter FreeBSD |

| Mailingliste | Zweck |
|---|---|
| freebsd-tex | Portierung von TeX und dessen Anwendungen nach FreeBSD |
| freebsd-threads | Leichtgewichtige Prozesse (Threads) in FreeBSD |
| freebsd-tilera | Diskussionen zur Portierung von FreeBSD auf die Tilera-CPU-Familie |
| freebsd-tokenring | Token-Ring Unterstützung in FreeBSD |
| freebsd-toolchain | Wartung der FreeBSD-Toolchain |
| freebsd-translators | Übersetzung von FreeBSD-Dokumenten und Programmen. |
| freebsd-transport | Diskussion über Transportprotokolle in FreeBSD |
| freebsd-usb | USB-Unterstützung in FreeBSD |
| freebsd-virtualization | Diskussion über verschiedene Virtualisierungsverfahren, die von FreeBSD unterstützt werden |
| freebsd-vuxml | Diskussion über die Infrastruktur von VuXML |
| freebsd-x11 | Wartung und Unterstützung von X11 auf FreeBSD |
| freebsd-xen | Diskussionen über die FreeBSD Portierung auf Xen™ - Implementierung und Verwendung |
| freebsd-xfce | Portierung und Wartung von XFCE |
| freebsd-zope | Zope für FreeBSD - Portierung und Wartung |

*Eingeschränkte Listen:* Die folgenden Listen wenden sich an Zielgruppen mit speziellen Anforderungen und sind nicht für die Öffentlichkeit gedacht. Bevor Sie eine dieser Listen abonnieren, sollten Sie einige der technischen Listen abonniert haben, um mit den Umgangsformen vertraut zu sein.

| Mailingliste | Zweck |
|---|---|
| freebsd-hubs | Betrieb von FreeBSD-Spiegeln |
| freebsd-user-groups | Koordination von Benutzergruppen |
| freebsd-wip-status | Status von in Arbeit befindlichen FreeBSD-Tätigkeiten |
| freebsd-wireless | Diskussionen zum 802.11-Stack sowie zur Entwicklung von Tools und Gerätetreibern |

*Zusammenfassungen:* Alle eben aufgezählten Listen sind auch in zusammengefasster Form (digest) erhältlich. In den Einstellungen Ihres Accounts legen Sie fest, in welcher Form Sie die Listen empfangen.

*SVN Listen:* Die folgenden Listen versenden die Log-Einträge zu Änderungen an verschiedenen

Teilen des Quellbaums. Diese Listen sollen *nur gelesen* werden, schicken Sie bitte keine Nachrichten an eine der Listen.

| Mailingliste | Teil des Quellbaums | Beschreibung |
|---|---|---|
| svn-doc-all | /usr/doc | Änderungen im doc Subversion Repository (mit Ausnahme von user, projects und translations) |
| svn-doc-head | /usr/doc | Änderungen im "head"-Zweig des doc Subversion Repository |
| svn-doc-projects | /usr/doc/projects | Änderungen im projects -Bereich des doc Subversion Repository |
| svn-doc-svnadmin | /usr/doc | Änderungen an den administrativen Skripten, Hooks und anderen Konfigurationsdateien des doc Subversion Repository |
| svn-ports-all | /usr/ports | Alle Änderungen des ports Subverison Repository |
| svn-ports-head | /usr/ports | Änderungen im "head"-Zweig des ports Subversion Repository |
| svn-ports-svnadmin | /usr/ports | Änderungen an den administrativen Skripten, Hooks und anderen Konfigurationsdateien des ports Subversion Repository |
| SVN commit messages for the entire src tree (except for "user" and "projects") | /usr/src | Änderungen im src Subversion Repository (außer für user und projects) |
| SVN commit messages for the src tree for head/-current | /usr/src | Änderungen im "head" Zweig des src Subversion Repository (der FreeBSD-CURRENT Zweig) |
| svn-src-projects | /usr/projects | Änderungen im projects Bereich des src Subversion Repository |
| svn-src-release | /usr/src | Änderungen im releases Bereich des src Subversion Repository |
| svn-src-releng | /usr/src | Änderungen im releng Zweig des src Subversion Repository (der security / release engineering Zweige) |

| Mailingliste | Teil des Quellbaums | Beschreibung |
|---|---|---|
| svn-src-stable | /usr/src | Änderungen an allen stable Zweigen des src Subversion Repository |
| svn-src-stable-6 | /usr/src | Änderungen im stable/6 Zweig des src Subversion Repository |
| svn-src-stable-7 | /usr/src | Änderungen im stable/7 Zweig des src Subversion Repository |
| svn-src-stable-8 | /usr/src | Änderungen im stable/8 Zweig des src Subversion Repository |
| SVN commit messages for only the 9-stable src tree | /usr/src | Änderungen im stable/9 Zweig des src Subversion Repository |
| svn-src-stable-10 | /usr/src | Änderungen im stable/10 Zweig des src Subversion Repository |
| svn-src-stable-11 | /usr/src | Änderungen im stable/11 Zweig des src Subversion Repository |
| svn-src-stable-12 | /usr/src | Änderungen im stable/12 Zweig des src Subversion Repository |
| svn-src-stable-other | /usr/src | Änderungen an älteren stable Zweigen des src Subversion Repository |
| svn-src-svnadmin | /usr/src | Änderungen an den administrativen Skripten, hooks, und anderen Daten zur Konfiguration des src Subversion Repository |
| svn-src-user | /usr/src | Änderungen am experimentellen user Bereich des src Subversion Repository |
| svn-src-vendor | /usr/src | Änderungen am Herstellerbereich des src Subversion Repository |

### C.2.2. Mailinglisten abonnieren

Um eine Liste zu abonnieren, besuchen die Webseite https://lists.freebsd.org und klicken dort auf die Liste, die Sie abonnieren wollen. Sie gelangen dann auf die Webseite der Liste, die weitere Anweisungen für diese Liste enthält.

Um eine Nachricht an eine Mailingliste zu schicken, schreiben Sie einfach eine E-Mail an Liste@FreeBSD.org. Die E-Mail wird dann an alle Mitglieder der Mailingliste verteilt.

Wenn Sie das Abonnement aufheben wollen, folgen Sie der URL, die am Ende jeder Mail der Liste

angegeben ist. Sie können das Abonnement auch mit einer E-Mail an Liste-unsubscribe@FreeBSD.org aufheben.

Verwenden Sie bitte die technischen Listen ausschließlich für technische Diskussionen. Wenn Sie nur an wichtigen Ankündigungen interessiert sind, abonnieren Sie die Mailingliste FreeBSD announcements, auf der nur wenige Nachrichten versendet werden.

### C.2.3. Chartas der Mailinglisten

*Alle* FreeBSD-Mailinglisten besitzen Grundregeln, die von jedem beachtet werden müssen. Für die ersten beiden Male, in denen ein Absender gegen diese Regeln verstößt, erhält er jeweils eine Warnung vom FreeBSD-Postmaster postmaster@FreeBSD.org. Ein dritter Verstoß gegen die Regeln führt dazu, dass der Absender in allen FreeBSD-Mailinglisten gesperrt wird und weitere Nachrichten von ihm nicht mehr angenommen werden. Wir bedauern sehr, dass wir solche Maßnahmen ergreifen müssen, aber heutzutage ist das Internet eine recht rauhe Umgebung, in der immer weniger Leute Rücksicht aufeinander nehmen.

Die Regeln:

- Das Thema einer Nachricht soll der Charta der Liste, an die sie gesendet wird, entsprechen. Wenn Sie eine Nachricht an eine technische Liste schicken, sollte die Nachricht auch technische Inhalte haben. Fortwährendes Geschwätz oder Streit mindern den Wert der Liste für alle Mitglieder und wird nicht toleriert. Benutzen Sie FreeBSD chat für allgemeine Diskussionen über FreeBSD.
- Eine Nachricht sollte an nicht mehr als zwei Mailinglisten gesendet werden. Schicken Sie eine Nachricht nur dann an zwei Listen, wenn das wirklich notwendig ist. Viele Leute haben mehrere Mailinglisten abonniert und Nachrichten sollten nur zu ungewöhnlichen Kombinationen der Listen, wie "-stable" und "-scsi", gesendet werden. Wenn Sie eine Nachricht erhalten, die im `Cc`-Feld mehrere Listen enthält, sollten Sie das Feld kürzen, bevor Sie eine Antwort darauf verschicken. *Unabhängig von dem ursprünglichen Verteiler sind Sie für Ihre eigenen Mehrfach-Sendungen selbst verantwortlich.*
- Persönliche Angriffe und Beschimpfungen sind in einer Diskussion nicht erlaubt. Dies gilt gleichermaßen für Benutzer wie Entwickler. Grobe Verletzungen der Netiquette, wie das Verschicken oder Zitieren von privater E-Mail ohne eine entsprechende Genehmigung, werden nicht gebilligt. Die Nachrichten werden aber nicht besonders auf Verletzungen der Netiquette untersucht. Es kann sein, dass eine Verletzung der Netiquette durchaus zu der Charta einer Liste passt, aber der Absender aufgrund der Verletzung eine Warnung erhält oder gesperrt wird.
- Werbung für Produkte oder Dienstleistungen, die nichts mit FreeBSD zu tun haben, sind verboten. Ist die Werbung als Spam verschickt worden, wird der Absender sofort gesperrt.

*Chartas einzelner Listen:*

**FreeBSD ACPI**

Die Entwicklung von ACPI und Energieverwaltungsfunktionen.

**FreeBSD announcements**

*Wichtige Ereignisse und Meilensteine*

Diese Liste ist für Personen, die nur an den wenigen Ankündigungen wichtiger Ereignisse interessiert sind. Die Ankündigungen betreffen Schnappschüsse und Releases, neue Merkmale von FreeBSD und die Suche nach freiwilligen Mitarbeitern. Auf der Liste herrscht wenig Verkehr und sie wird streng moderiert.

### freebsd-arch

*Architektur und Design von FreeBSD*

Auf dieser technischen Liste wird die FreeBSD-Architektur diskutiert. Beispiele für angemessene Themen sind:

- Wie das Bausystem zu verändern ist, damit verschiedene Läufe gleichzeitig möglich sind.
- Was am VFS geändert werden muss, damit Heidemann Schichten eingesetzt werden können.
- Wie die Schnittstelle der Gerätetreiber angepasst werden muss, damit derselbe Treiber auf verschiedenen Bussen und Architekturen eingesetzt werden kann.
- Wie ein Netzwerktreiber geschrieben wird.

### freebsd-bluetooth

*Bluetooth® unter FreeBSD*

Diese Liste diskutiert Probleme der Verwendung von Bluetooth® unter FreeBSD. Designprobleme, Implementierungsdetails, Patches, Fehler- und Statusberichte, Verbesserungsvorschläge sowie alle anderen mit Bluetooth® zusammenhängenden Themen werden hier behandelt.

### freebsd-bugbusters

*Bearbeitung der Fehlerberichte*

Auf dieser Liste wird die Bearbeitung der Fehlerberichte (PR, engl. problem report) koordiniert. Sie dient dem "Bugmeister" und allen Leuten, die ein Interesse an der Datenbank der Fehlerberichte haben, als Diskussionsforum. Auf dieser Liste werden keine spezifischen Fehler, Fehlerbehebungen oder PRs diskutiert.

### freebsd-bugs

*Fehlerberichte*

Auf dieser Liste werden Fehlerberichte gesammelt. Fehlerberichte sollten immer mit der Web-Schnittstelle erstellt werden.

### freebsd-chat

*Nicht technische Themen, welche die FreeBSD Gemeinschaft betreffen*

Auf dieser Liste werden nicht-technische soziale Themen diskutiert, die nicht auf die anderen Listen passen. Hier kann diskutiert werden, ob Jordan wie ein Frettchen aus einem Zeichentrickfilm aussieht oder nicht, ob grundsätzlich in Großbuchstaben geschrieben werden soll, wer zuviel Kaffee trinkt, wo das beste Bier gebraut wird und wer Bier in seinem Keller braut. Gelegentlich können auf den technischen Listen wichtige Ereignisse wie Feste, Hochzeiten oder Geburten angekündigt werden, aber nachfolgende Nachrichten sollten auf die Liste

FreeBSD chat gesendet werden.

### freebsd-chromium

*Diskussionen zum Einsatz von Chromium unter FreeBSD*

Auf dieser technischen Liste werden Fragen zur Entwicklung, zur Installation sowie zum Einsatz von Chromium unter FreeBSD diskutiert.

### freebsd-cloud

*FreeBSD auf verschiedenen Cloud-Plattformen betreiben*

Diese Liste diskutiert FreeBSD auf Amazon EC2, Google Compute Engine, Microsoft Azure und weiteren Cloud-Plattformen.

### freebsd-core

*FreeBSD Core Team*

Dies ist eine interne Mailingliste des FreeBSD Core Teams. Wenn in einer wichtigen Angelegenheit, die FreeBSD betrifft, entschieden werden muss oder die Angelegenheit einer genauen Prüfung unterzogen werden muss, können Nachrichten an diese Liste gesendet werden.

### FreeBSD-CURRENT

*Gebrauch von FreeBSD-CURRENT*

Diese Mailingliste ist für die Benutzer von FreeBSD-CURRENT eingerichtet. Auf ihr finden sich Ankündigungen über Besonderheiten von -CURRENT, von denen Benutzer betroffen sind. Sie enthält weiterhin Anweisungen, wie man ein System auf -CURRENT hält. Jeder, der ein -CURRENT System besitzt, muss diese Liste lesen. Die Liste ist nur für technische Inhalte bestimmt.

### freebsd-desktop

*FreeBSD als Desktop verwenden und verbessern*

Dies ist ein Forum für Diskussionen um FreeBSD auf dem Desktop. Es wird primär von Desktop-Portierern und Nutzern verwendet, um Probleme und Verbesserungen zu FreeBSDs Einsatz auf dem Desktop zu besprechen.

### freebsd-doc

Auf dieser Mailingliste werden Themen diskutiert, die im Zusammenhang mit der Erstellung der FreeBSD Dokumentation stehen. "The FreeBSD Documentation Project" besteht aus den Mitgliedern dieser Liste. Diese Liste steht jedem offen, Sie sind herzlich eingeladen teilzunehmen und mitzuhelfen.

### freebsd-drivers

*Gerätetreiber für FreeBSD schreiben*

Ein Forum für technische Diskussionen über das Schreiben von Gerätetreibern für FreeBSD. Daher werden hier vor allem Fragen behandelt, die sich um das Schreiben von Treibern, welche

die APIs des Kernels nutzen, drehen.

### freebsd-dtrace

*Entwicklung und Benutzung von DTrace unter FreeBSD*

DTrace ist Bestandteil von FreeBSD und stellt Laufzeitinformationen vom Kernel und Anwendungsprogrammen zur Verfügung. Diese Liste ist für Diskussionen von Entwicklern und Benutzern.

### freebsd-eclipse

*Für FreeBSD-Anwender, welche die Eclipse IDE deren Werkzeuge, Anwendungen und Ports einsetzen*

Das Ziel dieser Liste ist es, Unterstützung für all jene zu bieten, die mit der Installation, Verwendung, Entwicklung und Wartung der Eclipse-IDE sowie deren Werkzeugen und Anwendungen unter FreeBSD zu tun haben. Außerdem wird Hilfe bei der Portierung der IDE und deren Plugins auf FreeBSD geboten.

Zusätzlich soll diese Liste einen Informationsaustausch zwischen der Eclipse- und der FreeBSD-Gemeinde ermöglichen, von dem beide Seiten profitieren können.

Obwohl sich diese Liste auf die Anforderungen von Anwendern konzentriert, möchte sie auch Entwickler unterstützen, die an der Entwicklung von FreeBSD-spezifischen Anwendungen unter Nutzung des Eclipse-Frameworks arbeiten.

### freebsd-embedded

*FreeBSD in eingebetteten Anwendungen einsetzen*

Diese Liste diskutiert Themen im Zusammenhang mit dem Einsatz von ungewöhnlich kleinen und eingebetteten FreeBSD-Installationen. Auf dieser Liste werden ausschließlich technische Diskussionen geführt. Unter eingebetteten Systemen versteht diese Liste Systeme, bei denen es sich nicht um Desktopsysteme handelt, und die in der Regel nur einem einzigen Zweck dienen (im Gegensatz zu Desktopsystemen, die für die Bewältigung verschiedenster Aufgaben geeignet sind). In die Gruppe der eingebetteten Systeme gehören beispielsweise Telefone, Netzwerkgeräte wie Router, Switche oder PBX-Systeme, PDAs, Verkaufsautomaten und andere mehr.

### freebsd-emulation

*Emulation anderer Systeme wie Linux®, MS-DOS® oder Windows®*

Hier werden technische Diskussionen zum Einsatz von Programmen, die für andere Betriebssysteme geschrieben wurden, geführt.

### freebsd-enlightenment

*Enlightenment* Desktop-Umgebung für FreeBSD-Systeme. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

### freebsd-eol

*Support für FreeBSD-bezogene Software, die vom FreeBSD Project offiziell nicht mehr unterstützt wird.*

Diese Liste ist für all jene interessant, die Unterstützung für vom FreeBSD Project offiziell nicht mehr (in Form von Security Advisories oder Patches) unterstützte Programme benötigen oder anbieten wollen.

**freebsd-firewire**

*FireWire® (iLink, IEEE 1394)*

Auf dieser Liste wird das Design und die Implementierung eines FireWire®-Subsystems (auch IEEE 1394 oder iLink) für FreeBSD diskutiert. Relevante Themen sind die Standards, Busse und ihre Protokolle, sowie Adapter, Karten und Chipsätze. Des Weiteren die Architektur und der Quellcode, die nötig sind, diese Geräte zu unterstützen.

**freebsd-fortran**

*Fortran unter FreeBSD*

Diese Liste ist für Diskussionen rund um Fortran-Ports unter FreeBSD: Compiler, Bibliotheken, wissenschaftliche und technische Anwendungen von Laptops bis hin zu HPC-Clustern.

**freebsd-fs**

*Dateisysteme*

Diskussionen über FreeBSD-Dateisysteme. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

**freebsd-games**

*Spiele unter FreeBSD*

Eine Liste für technische Diskussionen im Zusammenhang mit Spielen unter FreeBSD. Die Liste ist für Personen, die an Portierungen arbeiten und alternative Lösungen erörtern. Personen, die an technischen Diskussionen interessiert sind, sind ebenfalls willkommen.

**freebsd-gecko**

Angelegenheiten zur *Gecko Rendering Engine*

Dies ist ein Forum über Gecko-Anwendungen, die FreeBSD verwenden.

Die Diskussion dreht sich um die Portierung von Gecko-Anwendungen, deren Installation, die Entwicklung sowie deren Unterstützung innerhalb von FreeBSD.

**freebsd-geom**

*GEOM*

Diskussion über GEOM und verwandte Implementierungen. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

**freebsd-git**

*Verwendung von git im FreeBSD Project*

Diskussionen über die Verwendung von git in der FreeBSD Infrastruktur. Personen, die einen Spiegel aufsetzen wollen, oder allgemeine Fragen zu git unter FreeBSD haben, können hier

Fragen stellen.

**freebsd-gnome**

*GNOME*

Diskussionen über die grafische Benutzeroberfläche GNOME. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

**freebsd-infiniband**

*Infiniband unter FreeBSD*

Technische Liste mit Diskussionen über Infiniband, OFED und OpenSM unter FreeBSD.

**freebsd-ipfw**

*IP Firewall*

Diskussionen über eine Neubearbeitung des IP-Firewall Quelltexts in FreeBSD. Dies ist eine technische Liste, in der nur technische Inhalte erwartet werden.

**freebsd-isdn**

*ISDN Subsystem*

Mailingliste für die Entwickler des ISDN Subsystems von FreeBSD.

**freebsd-java**

*Java™ Entwicklung*

Mailingliste, auf der die Entwicklung von Java™ Anwendungen für FreeBSD sowie die Portierung und Pflege von JDK™s diskutiert wird.

**freebsd-jobs**

*Stellenangebote und Stellengesuche*

In diesem Forum können Sie Stellenangebote und Stellengesuche, die mit FreeBSD zu tun haben, aufgeben. Diese Mailingliste ist *nicht* der Ort, um über allgemeine Beschäftigungsprobleme zu diskutieren; dazu gibt es anderswo geeignete Foren.

Beachten Sie bitte, dass diese Liste, wie die anderen `FreeBSD.org`-Listen, weltweit gelesen wird. Geben Sie daher bitte den Arbeitsort genau an. Geben Sie bitte auch an, ob Telearbeit möglich ist und ob Hilfen für einen Umzug angeboten werden.

Benutzen Sie in der E-Mail bitte nur offene Formate - vorzugsweise nur das Textformat. Andere Formate, wie PDF oder HTML, werden von den Lesern akzeptiert. Nicht offene Formate wie Microsoft® Word (.doc) werden vom Server der Liste abgelehnt.

**freebsd-hackers**

*Technische Diskussionen*

Dies ist ein Forum für technische Diskussionen über FreeBSD. Leute, die aktiv an FreeBSD arbeiten, können hier Probleme und deren Lösungen diskutieren. Interessierte, die den

Diskussionen folgen wollen, steht die Liste ebenfalls offen. Auf dieser Liste finden nur technische Diskussionen statt.

**freebsd-hardware**

*Allgemeine Diskussionen über Hardware*

Allgemeine Diskussionen über die Hardware, auf der FreeBSD läuft: Probleme und Ratschläge welche Hardware man kaufen sollte und welche nicht.

**freebsd-hubs**

*FreeBSD-Spiegel*

Ankündigungen und Diskussionsforum für Leute, die FreeBSD-Spiegel betreiben.

**freebsd-isp**

*Themen für Internet Service Provider*

Diese Liste ist für Internet Service Provider (ISP), die FreeBSD benutzen. Auf dieser Liste finden nur technische Diskussionen statt.

**freebsd-mono**

*Mono und C# Anwendungen auf FreeBSD*

Diese Liste beinhaltet Diskussionen über das Mono Entwicklungsframework auf FreeBSD. Dies ist eine technische Mailingliste. Es ist für Personen gedacht, die aktiv an der Portierung von Mono oder C# Anwendungen auf FreeBSD sind, um Probleme oder alternative Lösungen zu beratschlagen. Personen die der technischen Diskussion folgen möchten sind ebenso willkommen.

**freebsd-ocaml**

*FreeBSD-spezifische Diskussionen zu OCaml*

Diskussionen im Zusammenhang mit der OCaml-Unterstützung auf FreeBSD. Dies ist eine technische Mailingliste für Benutzer, die an OCaml-Ports, Bibliotheken und Frameworks von Drittanbietern arbeiten. Auch Benutzer, die an der technischen Diskussion interessiert sind, sind willkommen.

**freebsd-office**

*Office-Anwendungen für FreeBSD*

Diskussionen über Office-Anwendungen, ihre Installation, Entwicklung und Unterstützung innerhalb von FreeBSD

**freebsd-kde**

*KDE*

Diskussionen über KDE auf FreeBSD-Systemen. Dies ist eine technische Liste, in der nur technische Inhalte diskutiert werden.

### freebsd-announce

*Projekt-Infrastruktur Ankündigungen*

Diese Liste für Leute gedacht, die an Veränderungen im Zusammenhang der FreeBSD-Projekt Infrastruktur interessiert sind.

Diese moderierte Liste wird ausschließlich für Ankündigungen verwendet. Sie können keine Anfragen an diese Liste stellen und erhalten somit auch keine Antworten.

### freebsd-performance

*Diskussionsforum mit dem Ziel, die Leistung von FreeBSD zu verbessern.*

Auf dieser Liste diskutieren Hacker, Systemadministratoren und andere Interessierte die Leistung von FreeBSD. Zulässige Themen sind beispielsweise Systeme unter hoher Last, Systeme mit Leistungsproblemen oder Systeme, die Leistungsgrenzen von FreeBSD überwinden. Jeder, der mithelfen will, die Leistung von FreeBSD zu verbessern, sollte diese Liste abonnieren. Die Liste ist technisch anspruchsvoll und geeignet für erfahrene FreeBSD-Benutzer, Hacker oder Administratoren, die FreeBSD schnell, robust und skalierbar halten wollen. Auf der Liste werden Beiträge gesammelt oder Fragen nach ungelösten Problemen beantwortet. Sie ist kein Ersatz für das gründliche Studium der Dokumentation.

### freebsd-pf

*Diskussionen und Fragen zu packet filter als Firewallsystem.*

FreeBSD-spezifische Diskussionen zur Benutzung von packet filter (pf) als Firewallsystem. Sowohl technische Diskussionen als auch Anwenderfragen sind auf dieser Liste willkommen. Fragen zum ALTQ QoS Framework können ebenfalls gestellt werden.

### freebsd-pkg

*Diskussionen über die Verwaltung von Binärpaketen und entsprechenden Werkzeugen*

Diskussionen über die Verwendung von Binärpaketen, Werkzeuge zur Paketverwaltung, Entwicklung und Unterstützung innerhalb von FreeBSD, Verwaltung der Paket-Repositories und die Verwaltung von Paketen von Drittherstellern.

Beachten Sie, dass diese Liste nicht geeignet ist, um Probleme über nicht gebaute Pakete zu melden. Diese Probleme werden im allgemeinen als Problem des Ports betrachtet.

### freebsd-pkg-fallout

*Protokolle von fehlgeschlagenen Paketbauvorgängen*

Alle Fehlerprotokolle aus dem Paketcluster.

### freebsd-pkgbase

*Paketierung des FreeBSD-Basissystems*

Diskussion über die Implementierung und Probleme im Bezug auf die Paketierung des FreeBSD-Basissystems.

### freebsd-platforms

*Portierung auf nicht-Intel® Plattformen*

Plattformübergreifende Themen und Vorschläge für die Portierung auf nicht-Intel® Plattformen. Auf dieser Liste finden nur technische Diskussionen statt.

### freebsd-ports

*Diskussion über die Ports-Sammlung*

Diskussionen über die FreeBSD-Ports-Sammlung und die Infrastruktur der Sammlung. Die Liste dient auch der allgemeinen Koordination der Dinge, welche die Ports-Sammlung betreffen. Auf dieser Liste finden nur technische Diskussionen statt.

### freebsd-ports-bugs

*Diskussion über Fehler in den Ports*

Diskussion über Fehler in der Ports-Sammlung (/usr/ports), neue Ports oder Änderungen an bestehenden Ports. Auf dieser Liste finden nur technische Diskussionen statt.

### freebsd-proliant

*Technische Diskussionen zum Einsatz von FreeBSD auf HP ProLiant-Serverplattformen*

Diese Mailingliste bietet technische Diskussionen zum Einsatz von FreeBSD auf der ProLiant-Serverplattform von HP, darunter Fragen zu ProLiant-spezifischen Treibern, Konfigurationswerkzeugen sowie BIOS-Aktualisierungen. Daher ist sie die erste Anlaufstelle, um die Module hpasmd, hpasmcli, sowie hpacucli zu diskutieren.

### freebsd-python

*Python unter FreeBSD*

Diese technische Liste dient der Verbesserung der Python-Unterstützung unter FreeBSD. Sie wird von Personen gelesen, die an der Portierung von Python, von Python-Modulen Dritter und von Zope nach FreeBSD arbeiten. Personen, die diese technischen Diskussion verfolgen wollen, sind ebenfalls willkommen.

### freebsd-questions

*Benutzerfragen*

Auf dieser Mailingliste können Fragen zu FreeBSD gestellt werden. Fragen Sie bitte nicht nach Anleitungen, wenn Sie nicht sicher sind, dass Ihre Frage wirklich technischer Natur ist.

### freebsd-ruby

*Ruby unter FreeBSD*

Diese technische Liste dient der Verbesserung der Ruby-Unterstützung unter FreeBSD. Sie wird von Personen gelesen, die an der Portierung von Ruby, von Bibliotheken Dritter und Frameworks arbeiten. Personen, die diese technischen Diskussionen verfolgen wollen, sind ebenfalls willkommen.

**freebsd-scsi**

*SCSI Subsystem*

Diese Mailingliste ist für die Entwickler des SCSI Subsystems von FreeBSD. Auf dieser Liste finden nur technische Diskussionen statt.

**FreeBSD security**

*Sicherheitsthemen*

Sicherheitsthemen, die FreeBSD betreffen, wie DES, Kerberos, bekannte Sicherheitslöcher und Fehlerbehebungen. Stellen Sie bitte auf dieser Liste keine allgemeinen Fragen zum Thema Sicherheit. Willkommen sind allerdings Beiträge zur FAQ, das heißt eine Frage mit der passenden Antwort. Auf dieser Liste finden nur technische Diskussionen statt.

**FreeBSD security notifications**

*Ankündigungen zum Thema Sicherheit*

Ankündigungen über Sicherheitsprobleme von FreeBSD und deren Behebungen. Diese Liste ist kein Diskussionsforum, benutzen Sie FreeBSD security, um Sicherheitsthemen zu diskutieren.

**freebsd-snapshots**

*Ankündigungen für FreeBSD Entwickler-Snapshots*

Diese Liste informiert über die Verfügbarkeit von neuen FreeBSD-Snapshots aus den Zweigen head/ und stable/.

**FreeBSD-STABLE;**

*Gebrauch von FreeBSD-STABLE.*

Diese Mailingliste ist für die Benutzer von FreeBSD-STABLE eingerichtet. -STABLE ist der Zweig, in dem die Entwicklung nach einen RELEASE stattfindet, einschließlich Fehlerkorrekturen und neuer Funktionen. Die ABI wird wegen Binärkompatibilitäten stabil gehalten. Auf der Liste finden sich Ankündigungen über Besonderheiten von -STABLE, von denen Benutzer betroffen sind. Sie enthält weiterhin Anweisungen, wie man ein System auf -STABLE hält. Jeder, der ein -STABLE System besitzt, muss diese Liste lesen. Die Liste ist nur für technische Inhalte bestimmt.

**freebsd-standards**

*Konformität von FreeBSD mit den C99- und POSIX® Standards*

Dieses Forum ist für technische Diskussionen über die Konformität von FreeBSD mit den C99- und POSIX®-Standards.

**freebsd-teaching**

*Unterrichten mit FreeBSD*

Mailingliste, die das Unterrichten mit FreeBSD diskutiert.

**freebsd-testing**

*Tests unter FreeBSD*

Technische Liste, auf der Tests unter FreeBSD diskutiert werden, einschließlich ATF/Kyua, der Test/Build-Infrastruktur, und Portierungen von anderen Betriebssystemen (NetBSD, ...) nach FreeBSD.

**freebsd-tex**

*Portierung von TeX und dessen Anwendungen nach FreeBSD*

Technische Liste für Diskussionen im Zusammenhang mit TeX und dessen Anwendungen unter FreeBSD. Diese Liste ist für Menschen, die an der Portierung von TeX nach FreeBSD arbeiten. Es werden aber auch Probleme und Lösungen erörtert. Personen, die an technischen Diskussionen interessiert sind, sind ebenfalls willkommen.

**freebsd-toolchain**

*Wartung der FreeBSD-Toolchain*

Auf dieser Mailingliste werden alle Themen rund um die FreeBSD-Toolchain diskutiert. Dazu gehören der Status von Clang und GCC, aber auch Fragen zu Programmen wie Assemblern, Linkern und Debuggern.

**freebsd-translators**

*Übersetzung von FreeBSD-Dokumenten und Programmen*

Auf dieser Liste können Übersetzer von FreeBSD-Dokumenten über die Methoden und Werkzeuge für die Übersetzung diskutieren. Neue Benutzer werden gebeten sich vorzustellen und die Sprache zu erwähnen, an dessen Übersetzung sie interessiert sind.

**freebsd-transport**

*Diskussion über Transportprotokolle in FreeBSD*

Diese Liste behandelt die Probleme und das Design von FreeBSDs Netzwerkstack, darunter auch TCP, SCTP und UDP. Andere Netzwerkthemen sollten auf der FreeBSD networking diskutiert werden.

**freebsd-usb**

*USB-Unterstützung in FreeBSD.*

Auf dieser Liste finden nur technische Diskussionen statt.

**freebsd-user-groups**

*Koordination von Benutzergruppen*

Diese Liste ist für Koordinatoren lokaler Benutzergruppen und einem ausgesuchten Mitglied des Core Teams eingerichtet worden. Der Inhalt sollte Inhalte von Treffen und die Koordination von Projekten mehrerer Benutzergruppen beschränkt sein.

**freebsd-virtualization**

*Diskussion über verschiedene Virtualisierungsverfahren, die von FreeBSD unterstützt werden*

Eine Liste, auf der die verschiedenen Virtualisierungsverfahren, die von FreeBSD unterstützt werden, diskutiert werden. Auf der einen Seite liegt der Fokus auf der Implementierung der

zugrundeliegenden Funktionalitäten, ebenso wie das Hinzufügen neuer Eigenschaften. Auf der anderen Seite haben die Benutzer ein Forum, um Fragen bei Problemen zu stellen oder um ihre Anwendungsfälle zu besprechen.

### freebsd-wip-status

*Status von in Arbeit befindlichen FreeBSD-Tätigkeiten*

Diese Mailingliste kann dazu verwendet werden, eigene Kreationen und deren Fortschritt von FreeBSD-verwandten Tätigkeiten anzukündigen. Die Nachrichten werden moderiert. Es wird empfohlen, die Nachricht "An:" eine mehr themenverwandte FreeBSD-Liste zu senden und diese Liste nur in Blindkopie zu setzen. Auf diese Weise kann ihre in Arbeit befindliche Tätigkeit auch auf der themenverwandten Liste diskutiert werden, da auf dieser Liste keine Diskussionen erlaubt sind.

Sehen Sie sich das Archiv der Liste für passende Nachrichten an.

Redaktionelle Auszüge der Nachrichten an diese Liste werden eventuell alle paar Monate auf die FreeBSD Webseite als Teil der Statusberichte gestellt. Weitere Beispiele und zurückliegende Berichte können Sie auch dort finden.

### freebsd-wireless

*Diskussionen zum 802.11-Stack sowie zur Entwicklung von Tools und Gerätetreibern*

Die Mailingliste freebsd-wireless diskutiert Themen rund um den 802.11-Stack (sys/net80211). Besprochen werden die Entwicklung von Tools und Gerätetreibern sowie auftretende Probleme, neue Funktionen sowie die Wartung der existierenden Werkzeuge und Treiber.

### freebsd-xen

*Diskussionen über die FreeBSD Portierung auf Xen™ - Implementierung und Verwendung*

Eine Liste, welche die FreeBSD Portierung auf Xen™ behandelt. Das erwartete Nachrichtenaufkommen ist klein genug, so dass es als Forum für sowohl technische Diskussionen über die Implementierung und Entwurfsdetails, als auch administrative Verteilaspekte ausgelegt ist.

### freebsd-xfce

*XFCE*

Eine Liste, auf der Fragen zum Einsatz von XFCE unter FreeBSD diskutiert werden. Es handelt sich um eine technische Mailingliste, die sich primär an Entwickler richtet, die aktiv an der Portierung von XFCE nach FreeBSD arbeiten. Aber auch Nutzer, die einfach nur die technischen Diskussionen verfolgen wollen, sind willkommen. Diskutiert werden vor allem bei der Portierung auftretende Probleme und mögliche Lösungswege.

### freebsd-zope

*Zope*

Ein Forum für Diskussionen darüber, wie man die Zope-Umgebung auf FreeBSD portieren kann. Dies ist eine technische Mailingliste. Sie ist für Leute gedacht, die aktiv an der Portierung von

Zope auf FreeBSD arbeiten, um aufkommende Probleme oder alternative Lösungsansätze zu besprechen. Personen, die der technischen Diskussion folgen möchten, sind ebenfalls willkommen.

### C.2.4. Filter der Mailinglisten

Um die Verbreitung von Spam, Viren und anderen nicht erwünschten E-Mails zu verhindern, werden auf den FreeBSD-Mailinglisten Filter eingesetzt. Dieser Abschnitt beschreibt nur einen Teil der zum Schutz der Listen eingesetzten Filter.

Auf den Mailinglisten sind nur die unten aufgeführten Anhänge erlaubt. Anhänge mit einem anderen MIME-Typ werden entfernt, bevor eine E-Mail an eine Liste verteilt wird.

- application/octet-stream
- application/pdf
- application/pgp-signature
- application/x-pkcs7-signature
- message/rfc822
- multipart/alternative
- multipart/related
- multipart/signed
- text/html
- text/plain
- text/x-diff
- text/x-patch



Einige Mailinglisten erlauben vielleicht Anhänge mit anderem MIME-Typ. Für die meisten Mailinglisten sollte die obige Aufzählung aber richtig sein.

Wenn eine E-Mail sowohl aus einer HTML-Version wie auch aus einer Text-Version besteht, wird die HTML-Version entfernt. Wenn eine E-Mail nur im HTML-Format versendet wurde, wird sie in reinen Text umgewandelt.

## C.3. Usenet-News

Neben den Gruppen, die sich ausschließlich mit BSD beschäftigen, gibt es viele weitere in denen über FreeBSD diskutiert wird, oder die für FreeBSD-Benutzer wichtig sind.

### C.3.1. BSD spezifische Gruppen

- comp.unix.bsd.freebsd.announce
- comp.unix.bsd.freebsd.misc
- de.comp.os.unix.bsd (deutsch)

- fr.comp.os.bsd (französisch)

### C.3.2. Weitere UNIX Gruppen

- comp.unix
- comp.unix.questions
- comp.unix.admin
- comp.unix.programmer
- comp.unix.shell
- comp.unix.misc
- comp.unix.bsd

### C.3.3. X Window System

- comp.windows.x

## C.4. Offizielle Spiegel

Central Servers, Armenia, Australia, {mirrors-austria}, Czech Republic, Denmark, Finland, France, Germany, Hong Kong, Ireland, Japan, Latvia, Lithuania, Netherlands, Norway, Russia, Slovenia, South Africa, Spain, Sweden, Switzerland, Taiwan, United Kingdom, United States of America.

(aktualisiert am: UTC)

**Central Servers**

- https://www.FreeBSD.org/

**Armenia**

- http://www1.am.FreeBSD.org/ (IPv6)

**Australia**

- http://www.au.FreeBSD.org/
- http://www2.au.FreeBSD.org/

**{mirrors-austria}**

- http://www1.am.FreeBSD.org/ (IPv6)

**Czech Republic**

- http://www.cz.FreeBSD.org/ (IPv6)

**Denmark**

- http://www.dk.FreeBSD.org/ (IPv6)

**Finland**

- http://www.fi.FreeBSD.org/

**France**

- http://www1.fr.FreeBSD.org/

**Germany**

- http://www.de.FreeBSD.org/

**Hong Kong**

- http://www.hk.FreeBSD.org/

**Ireland**

- http://www.ie.FreeBSD.org/

**Japan**

- http://www.jp.FreeBSD.org/www.FreeBSD.org/ (IPv6)

**Latvia**

- http://www.lv.FreeBSD.org/

**Lithuania**

- http://www.lt.FreeBSD.org/

**Netherlands**

- http://www.nl.FreeBSD.org/

**Norway**

- http://www.no.FreeBSD.org/

**Russia**

- http://www.ru.FreeBSD.org/ (IPv6)

**Slovenia**

- http://www.si.FreeBSD.org/

**South Africa**

- http://www.za.FreeBSD.org/

**Spain**

- http://www.es.FreeBSD.org/
- http://www2.es.FreeBSD.org/

**Sweden**

- http://www.se.FreeBSD.org/

**Switzerland**

- http://www.ch.FreeBSD.org/ (IPv6)
- http://www2.ch.FreeBSD.org/ (IPv6)

**Taiwan**

- http://www.tw.FreeBSD.org/
- http://www2.tw.FreeBSD.org/
- http://www4.tw.FreeBSD.org/
- http://www5.tw.FreeBSD.org/ (IPv6)

**United Kingdom**

- http://www1.uk.FreeBSD.org
- http://www3.uk.FreeBSD.org/

**United States of America**

- http://www5.us.FreeBSD.org/ (IPv6)

# Anhang D: OpenPGP-Schlüssel

Verwenden Sie die nachstehenden Schlüssel, wenn Sie eine Signatur überprüfen oder eine verschlüsselte E-Mail an einen Ansprechpartner oder einen Entwickler schicken wollen. Eine vollständige Liste der FreeBSD OpenPGP-Schlüssel finden Sie im Artikel PGP Keys. Den vollständigen Schlüsselring der Entwickler von FreeBSD finden Sie unter pgpkeys.txt.

## D.1. Ansprechpartner

### D.1.1. Security Officer Team <security-officer@FreeBSD.org>

```
pub   rsa4096/D9AD2A18057474CB 2022-12-11 [C] [expires: 2030-01-24]
      Key fingerprint = 0BE3 3275 D74C 953C 79F8  1107 D9AD 2A18 0574 74CB
uid                           FreeBSD Security Officer <security-officer@freebsd.org>
sub   rsa4096/6E58DE901F001AEF 2022-12-11 [S] [expires: 2028-01-24]
sub   rsa4096/46DB26D62F6039B7 2022-12-11 [E] [expires: 2028-01-24]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGOVdeUBEADHF5VGg1iPbACB+7lomX6aDytUf0k2k2Yc/Kp6lfYv7JKU+1nr
TcNF7Gt1YkajPSeWRKNZw/X94g4w5TEOHbJ6QQWx9g+N7RjEq75actQ/r2N5zY4S
ujfFTepbvgR55mLTxlxGKFBmNrfNbpHRyh4GwFRgPlxf5Jy9SB+0m54yFS4QlSd0
pIzO0CLkjHUFy/8S93oSK2zUkgok5gLWruBXom+8VC3OtBElkWswPkE1pKZvMQCv
VyM+7BS+MCFXSdZczDZZoEzpQJGhUYFsdg0KqlLv6z1rP+HsgUYKTkRpcrumDQV0
MMuCE4ECU6nFDDTnbR8Wn3LF5oTt0GtwS0nWf+nZ1SFTDURcSPR4Lp/PKjuDAkOS
P8BaruCNx1ItHSwcnXw0gS4+h8FjtWNZpsawtzjjgApcl+m9KP6dkBcbN+i1DHm6
NG6YQVtVWyN8aOKmoC/FEm1CWh1bv+ri9XOkF2EqT/ktbjbT1hFoFGBkS9/35y1G
3KKyWtwKcyF4OXcArl6sQwGgiYnZEG3sUMaGrwQovRtMf7le3cAYsMkXyiAnEufa
deuabYLD8qp9L/eNo+9aZmhJqQg4EQb+ePH7bGPNDZ+M5oGUwReX857FoWaPhs4L
dAKQ1YwASxdKKh8wnaamjIeZSGP5TCjurH7pADAIaB3/D+ZNl2a7od+C1wARAQAB
tDdGcmVlQlNEIFNlY3VyaXR5IE9mZmljZXIgPHNlY3VyaXR5LW9mZmljZXJAZnJl
ZWJzZC5vcmc+iQJSBBMBCgA8AhsBBAsJCAcEFQoJCAUWAgMBAAIeBQIXgBYhBAvj
MnXXTJU8efgRB9mtKhgFdHTLBQJjlXeQBQkF3u+rAAoJENmtKhgFdHTLOVoQALS3
cj7rqYkHiV4zDYrgPEp9O1kAyGI8VdfGAMkDVTqr+wP4v/o7LIUrgwZl5qxesVFB
VknFr0Wp5g9h0iAjasoI5sDd6tH2SmumhBHXFVdftzDQhrugxH6fWRhHs0SaFYCk
Qt5nFbcpUfWgtQ35XTbsL8iENdYpjKXsSFQrJneGSwxIjWYTFn6ps/AI3gwR8+Bn
OffEFdYugJ049O6Vu6YBFJHrnMO7NbF4v95dVYuLtpMIaXWM+V9KITmhaBzFz5fM
Q7UOzcLlbxOYKNIWcp8QQk429mayKW5VUeUExUD1ZzBHn+P6ZG7QTMDu/RmBqiHo
ewCMVz4n9uXT5BiOngE4CvS0WQwHzK+k9MLpG2u/Bo9+LT0Ceh9Ou1rfU5+0tRwl
GyOFFjf3INS7I7gkcAwxQ7dzDItN/UQPZpg8y9mABU2x4enz0AvTnb61d/1dnTEr
tdNgU433he0ZnD1HurZCjBEWC656wv6iMdWcD8gjhMbmEpPmjvXcYlTO6zhEygSM
DiwdQCWK2W4++YJerA6ULBi3niNWBpofOFH8XylV56ruhjtHCo7+/3carcMoPOJv
lVZ1zCKxLro3TRBT15JTFBGqblRyTopFK3PuxW//GTnZOtpQEOV6yL4RAXcWeC1d
1hb5k/YxUmRF6XsDNEH4b08T8ZO8dV3dAV43Wh1oiQEzBBABCAAdFiEEuyjUCzYO
7pNq7RVv5fe8y6O93fgFAmObXVYACgkQ5fe8y6O93fiBlwf/W8y1XXJIx1ZA3n6u
f7aS70rbP9KFPr4U0dixwKE/gbtIQ9ckeNXrDDWz0v0NCz4qS+33IPiJg1WcY3vR
W90e7QgAueCo5TdZPImPbCs42vadpa5byMXS4Pw+xyT+d/yp2oLKYbj3En4bg1GM
```

```
w71DezIjvV+e01UR++u1t9yZ8LOWM5Kumz1zyQLZDZ8qIKt1bBfpa+E0cEqtNQWu
iGhQE3AHI8eWV+jBkg5y2zHRIevbWb1UPsj43lgkFtAGHk9rrM8Rmgr4AXr531iD
srBwauKZ/MElcF3MINuLH+gkPPaFHw/YIpLRLaZXZVsw3Xi1RNXI2n2ea29dvs/C
Lcf1vYkCMwQQAQgAHRYhBPwOh4rlr+eIAo1jVdOXkvSep+XCBQJjm14FAAoJENOX
kvSep+XC0DcP/1ZB7k9p1T+9QbbZZE1PJiHby3815ccH3XKexbNmmakHIn3L6Cet
F891Kqt9ssbhFRMNtyZ/k/8y8Hv5bKxVep5/HMyK+8aqfDFN0WMrqZh0/CiR6DJh
gnAmPNw/hAVHMHaYGII9kCrFfPFJ02FKoc81g9F08odb7TV+UlvRjkErhRxF+dGS
wQoO0RCbf0Z1cs7nd0Vb2z4IJh4XMxBjWc/uQ2Q9dH/0uRzwpAnR4YX+MG5YrX7Z
zBvDyR0r76iQwRSDKgioNgkr6R3rq1NZGdaj+8b0LzdOqtzKJ/eupDe3+H67e/EN
qymtreGjrubpiU9bKvYArisUqhE5KtguryvR6Qz9bj87nPg33DT3WWGVrwFRxBox
dbWzjQFv0wug8m4GAwVF7fPR5/eW7IHw8zvgn0vSPcZz7MZ4e6Y5jN4kA5/xWJYZ
Sps54qQWB+FA30unIXN68KqdIzONIbtaY3W4/JjJUCm4T+wEjKaH+wJX8w1DMjlg
mkTmGh/UrTyC1vXbPgk9Sy3cRTICR1T9z7W8UlmTtnKrUklrjlFR7SXzrEXzLGOX
Fm+NEHpHNXqzcm6c3QfzY/yQ9HSAQ/t7SUQ9caRePbDz3/msyPxtGFor9roQv6VN
wRXCyRgkH4Y5tPhJAQ8G/FxX+VXFb93QL0lfelb23/BBu6cUwW63SRn5iHUEExYI
AB0WIQQeB2Johg/5/ikUnJwDU9SVF1S13AUCZISO3wAKCRADU9SVF1S13NnqAP95
LA1Om9XSAKl76VtV+L3JPDdAwIdbNa0OsRT4Wm7U3wD/YoFrdHXVHHQFKwYeUUhj
XZcxnZLe9Ixo0/JP+RVFVw6JAlIEEwEKADwCGwEECwkIBwQVCgkIBRYCAwEAAh4F
AheAFiEEC+MydddMlTx5+BEH2a0qGAV0dMsFAmluhFoFCQ1lXf0ACgkQ2a0qGAV0
dMv8ixAAhYn3uxnjTCBZ5BJ6FMdwFXgGZMZVnL1qEoHA7HWB0ZoQlu29sFzBX97x
rkVZfWiU82V9/PYyHh4EY0hN9oOJMDDhSgi+DfdoEablIcIXR9uBBL/Snnak6uEF
gOD6VFH1T3o7iBmS/ixee2YFXq5QxAwEiANSAsxDm2rlyztPc5xM76fz5eEcgp1m
UDI7qFjoEkJVvRraJGVy3AUh4sr/94VAXPuCa7JnC6a8j9SD8XtzqIfcuqNBXXfA
m4ji4fN+KcDHtZPdkNLDBu1DEXQkOR/1S++NmXHYCEdQEuVOwLVsy6vQqgc+unju
Xh9sOrdiXUWylHh2hvgmwkjhMGOREiIRwNoA09m0OKZ2jo2dSlhNdVFRD5yhD92t
X9BarQbbjQtyCvoWnCJMFfw/vDnxfDdUdV3GxLhaOWvR8ItWQ7ts8tEZSjR/G76/
fHj1JRNuHRdhwype6ehieewJSMyzByhT0A3Njv3KLI4nwQj//MYzB5FJ9uOmKLcG
w7PB0yVjW4oRbG62+Qv6IQybCyrAL+12qgbUXbMsTqMwmsmdgtsqr/7OJNo3PBxX
2rUEio/mez8wuzNB0mJtlMynoMqXtbFX7AAXHk16fOGd6L6guW8+IhPedT3TvrjS
hqiQVvjOhM6rEYZ/YUfcIdkWJVpmE6+MRPVUufyMWeHaWqsKXoa5Ag0EY5V2yQEQ
AOqjzPpMUCGu8eElXnAd2PruC6hi+lc/yC90KqizxIuW6qLQBaAkTCWq7suYpDqo
ygn7YM3rL50S285WAECAXrcst/cVAqr0UH/e6p4iJCUIiXcfjd/wq20RnN/+VuvL
hjpCFLY5czfVS31D7Uh9MbC+zUTz8nVTiNCsAao0qSdfJDIzB4nSO+9xIsme/dLs
I5QlU5PdxOBV6HdEhCUXOoratJCbKAOlLxtPwyMKxmv4oZ7Mqlt10peKjhpBb97q
cIzJhHxujQZD0OmzIA6xoQ2eSCGdxCEDsZ09kr3Esw1AwKnQ51xmWpFWNFk6627M
1bo8+hzOz81CrTZhYrgE+1JXv6V1L2A9lMsimdE1BHNycDS+dBOpIB9qxXCwAab4
ykfvNxox/ZPDUrTy7v7mDI5uDNTNCYYsKCj1UidycOKzSziB9Oa2uvmMJ5XstgNB
f7Z8Cky1dtVd4ol6bU9L5nos9tbYeSXFi4bmcWB7AJiVCMq6N+LBbUKWGLglB4TU
1qhttpqv31X9V6ges5gARY/RuRTKsVyhwsn7SDcqmNKRy0im2AYakwEp7hT07ula
hOSLxjP+5hCf+nSJlwbxJ8ozwjjbzeN2yLlJSI00klkIFBNUDt3wzFRW/n6qlf+/
lepgzekfNrYMtfPB8AT07Z2A3U4xlgiV346dZymbY/EjABEBAAGJBHIEGAEKACYW
IQQL4zJ110yVPHn4EQfZrSoYBXR0ywUCY5V2yQIbAgUJAgIpAAJACRDZrSoYBXR0
y8F0IAQZAQoAHRYhBLYVJ36BCH33XIGDO25Y3pAfABrvBQJjlXbJAAoJEG5Y3pAf
ABrvuBsQAOlQFPXhx6whO4yw5ZizIS02YHhSVMVYKS2T9jPIKi1qxnEiEw9eKH0b
W0Oj0TEhZPyM2NJID7DRWK5r8+KsMu8jwm1fUmIrefAx6fCVfCWRECT1MlbL3jhh
6AcX/nK2e3Bn8vgExhzczO3JlvD6wPCc0FkpiY7yDB9ihu1+gbE5Hg6dvftttRXD
rbEdAifbNp9KYxDigxdlOb0S14hjCBysLWH5Su/khcIlkeuqZcI8TmDldnUb2OqT
CVpFhaNwsPSrHBzmb0s2sXo4FL03pLsOdwhi31W6kjk4KvW5FKrOpoEwUMKVNMf5
0DHdvonUoUHRSIc/cV5NqUWHwvc0T5031qk0CCRRa/+/iij/p2RG7c1mx7ZECj+j
ZfmvjSqT+WHJ1BFlNJMWyK4fdVRZWyCaoAecdbukwzDwUCUqHJFIWeFtbut7SOPx
cwg7sbnKNApAKdi491dvH75s/U/OwRYO/2P+ymHlqtyix2jq0ReSVYcQPXswQ8i2
```

```
ifX41F+xTSl4RWCBBeXB1Nxk3+HsV4Jnnp1zAJZ0KlKW/oJxbNFdI1TImkpr2p8i
oFf+aiePLvDkgeaG8vABgjoihPXWHVAMR8Z+GvBY/A6OdexpibkTvC/zDr0/Exs4
lsylZKDwvvFbctcpHVXBeCBQLX5vfLrsTkaCLWF/SV9OdMykvYKU7ZAP+gKEwhp+
HPFuOHZbOBhqFUdkfeCkdzX/QGdzTuz349roRhgz2vRfN7MtbtuzA6NWWhEWt5Dc
UgX/Y5I3Q4Z2bt3JiXQ6WJMgMMOXAr+XxtxyRRykc1HV3DQ/cq8OWYubNnIbgebP
NIFr2OIWKsR9yDaucZzpmLfzaMZUAu5hWmU9fIw5SIKGnQABBnNMhilfD+CkETp6
baTvjTK4rpaobjJdeCTrsWgfXRNC8x3hDvcrjPD70MylOGVQdx8GYChWJnCKXsLT
GX7KwdfxkjclTyzWvdcCemp0eLhamLGb9y1dtWdNIDcVCvZJy0lipHVUdFYYxb4i
LZJANL631tlPM6AA8sO1/L4mqEGnAIHVrUQd+2QkSiOl9mKlpgaR/fJz683BR5Qe
n9ywX0JPtBupqPW3t9VbO/uNxUqlHCeAhPi9NLOpujpYLfgW5QAfS3u0nkp5nrbk
CoQUua2q0Oj7J0mFmtWtcE1c9+THmFJVb8j2G9yQw3ADe3Qp9ALazP5nVDVri8NZ
BhHK1/KuBmRYZtcyfqXUnKoiiWAlm5rHaRiztW7e3wqm2oJu/RkEAagybutEuBWh
2Ej2+gDxjEKKtIKGu54lif4kqTwwjKTcN1ekGihwwgCMUkBSBeNXk1ClkzLFHwES
JCcFwdEgpVYQTKFsu0emYISyco3IpUajGzfUiQRyBBgBCgAmAhsCFiEEC+MydddM
lTx5+BEH2a0qGAV0dMsFAmWFy28FCQPyfaYCQMF0IAQZAQoAHRYhBLYVJ36BCH33
XIGDO25Y3pAfABrvBQJjlXbJAAoJEG5Y3pAfABrvuBsQAOlQFPXhx6whO4yw5Ziz
IS02YHhSVMVYKS2T9jPIKi1qxnEiEw9eKH0bW0Oj0TEhZPyM2NJID7DRWK5r8+Ks
Mu8jwm1fUmIrefAx6fCVfCWRECT1MlbL3jhh6AcX/nK2e3Bn8vgExhzczO3JlvD6
wPCc0FkpiY7yDB9ihu1+gbE5Hg6dvftttRXDrbEdAifbNp9KYxDigxdlOb0S14hj
CBysLWH5Su/khcIlkeuqZcI8TmDldnUb2OqTCVpFhaNwsPSrHBzmb0s2sXo4FL03
pLsOdwhi31W6kjk4KvW5FKrOpoEwUMKVNMf50DHdvonUoUHRSIc/cV5NqUWHwvc0
T5031qk0CCRRa/+/iij/p2RG7c1mx7ZECj+jZfmvjSqT+WHJ1BFlNJMWyK4fdVRZ
WyCaoAecdbukwzDwUCUqHJFIWeFtbut7SOPxcwg7sbnKNApAKdi491dvH75s/U/O
wRYO/2P+ymHlqtyix2jq0ReSVYcQPXswQ8i2ifX41F+xTSl4RWCBBeXB1Nxk3+Hs
V4Jnnp1zAJZ0KlKW/oJxbNFdI1TImkpr2p8ioFf+aiePLvDkgeaG8vABgjoihPXW
HVAMR8Z+GvBY/A6OdexpibkTvC/zDr0/Exs4lsylZKDwvvFbctcpHVXBeCBQLX5v
fLrsTkaCLWF/SV9OdMykvYKUCRDZrSoYBXR0yyOqEACitDvbkbfjaton6izr4T8Q
U2yvhJHkf4B6KeVDbKY1J4784OxXp2bJgPeF53SYBe8gm3YHjp8ULh4A/19U4hsw
yE8ymcm5nIs8OLyBdxkuBZJGEnzxH3woiyYqWH7991kzhEjUkuMgKLuTI1HiO0oL
MuPQNhUHOnWafSVPC0XO/tIL12OmoUuc7ligY9Z9AceFjTZOuHamixHAAc6hpxdI
W+yhC/qTpc2VK0niWeuQfq3453iRTf9MnR5Beztl3ZYRWcx7UiFuKGwZwBibNnNm
Us6GyQcJ5UTa1oeJcLqHi0Lf/r0jXo3wgJq7EZjjVyU+GI2ZVoDOaS6c4/OvLm62
XoeSlnn/dQxUcjUki+x8lb69IxSF1xAgsC/oNtFZYd5rHdlnqIBUYK0lLtSCXBkz
VeivSiQa0hL5on8LDu1nw2bXyW61yt/YxVb4FanMxAqdYVBhOfU0RaPNifH01rbb
4TwC9bTZN1LQ1KI/Swb/SruUE0RyT28fhYRtsReS2PnUODghJSFDJbwFbBZf6RKI
16q1xqKRRvxIWPm+lMOi1NLOKR9P+OKy9HmChMw0UJUcVl1cJ2xtRl3wi5t6AA6H
oNv/TrLeYVgMR9wYmKlpvjTQ5jTdrbHD1XP5jGsp8QsJMGja1m/7cryReCpcVxvI
meReaOdgz+zDmQqq3O5zuIkEcgQYAQoAJgIbAhYhBAvjMnXXTJU8efgRB9mtKhgF
dHTLBQJnhEEJBQkF0/JAAkDBdCAEGQEKAB0WIQS2FSd+gQh991yBgztuWN6QHwAa
7wUCY5V2yQAKCRBuWN6QHwAa77gbEADpUBT14cesITuMsOWYsyEtNmB4UlTFWCkt
k/YzyCotasZxIhMPXih9G1tDo9ExIWT8jNjSSA+w0Viua/PirDLvI8JtX1JiK3nw
MenwlXwlkRAk9TJWy944YegHF/5ytntwZ/L4BMYc3MztyZbw+sDwnNBZKYmO8gwf
YobtfoGxOR4Onb37bbUVw62xHQIn2zafSmMQ4oMXZTm9EteIYwgcrC1h+Urv5IXC
JZHrqmXCPE5g5XZ1G9jqkwlaRYWjcLD0qxwc5m9LNrF6OBS9N6S7DncIYt9VupI5
OCr1uRSqzqaBMFDClTTH+dAx3b6J1KFB0UiHP3FeTalFh8L3NE+dN9apNAgkUWv/
v4oo/6dkRu3NZse2RAo/o2X5r40qk/lhydQRZTSTFsiuH3VUWVsgmqAHnHW7pMMw
8FAlKhyRSFnhbW7re0jj8XMIO7G5yjQKQCnYuPdXbx++bP1PzsEWDv9j/sph5arc
osdo6tEXklWHED17MEPIton1+NRfsU0peEVggQXlwdTcZN/h7FeCZ56dcwCWdCpS
lv6CcWzRXSNUyJpKa9qfIqBX/monjy7w5IHmhvLwAYI6IoT11h1QDEfGfhrwWPwO
jnXsaYm5E7wv8w69PxMbOJbMpWSg8L7xW3LXKR1VwXggUC1+b3y67E5Ggi1hf0lf
TnTMpL2ClAkQ2a0qGAV0dMsNxRAAsuW1aLh+hgydW+iH6DmdQRMESsB1kE02kO14
```

```
62TAQaziIAvNoxw5h48xvyEnrDA8d+9IDMyxdrLmAbndUlSveMa9+EPiGHwr6VTy
FL8nA5F7DcFi4mjEyGKe18JcaAlYUtvHgWH6EjiX2iSXpsrJFEhtfFNolZ5sp9LF
I6hOBihSJxZK4sbMR7Q6IkDuAVpTFLiejBRlsXpFvTGL6040CtxbL5cqkVMYP38r
FMTuc3pGGJA4wb5EC1dGjUi6XjbYH7kuCAFyXqV9eQQP61x7K9W8qnXW+weCIMKf
SX7AcCtH1jXBAM6lqpPrh6amc+/rbg2eNA7DmgJnEY4apIcDB/b4khRMga2ozeGW
WyIvOaVvR2R7ALQ+Rgut85cM+4+Vl2PHmOzW/yYdHVb5REQItFR5COb/mGUqYhkC
tiV3nXo/K0uOQKu5SBbNzLuNvuwdn+Eimxjl18VnrGG7sjtUa0MLmtr62GiEVrhr
Dqa/biHp8LdWkAQjLZ4aTRh2XZiggaVFZHmkw3ILPyKKM21UXdM0YRk3TGVK8ODQ
y58ebPS4v9yYT9gUA9UDkDYeGcF2qjoDPVNvcG6H8jCSsPRl1KZwtqqITCOSAIAP
I4Nu97kO6nbOyQpYlwjd1MhvVXnP66mHSvmqaxbNGX1mF9B/yERkBkooNZrKuJSv
BTC2J1qJBHIEGAEKACYCGwIWIQQL4zJ110yVPHn4EQfZrSoYBXR0ywUCaW6EcgUJ
CaGkeQJAwXQgBBkBCgAdFiEEthUnfoEIffdcgYM7bljekB8AGu8FAmOVdskACgkQ
bljekB8AGu+4GxAA6VAU9eHHrCE7jLDlmLMhLTZgeFJUxVgpLZP2M8gqLWrGcSIT
D14ofRtbQ6PRMSFk/IzY0kgPsNFYrmvz4qwy7yPCbV9SYit58DHp8JV8JZEQJPUy
VsveOGHoBxf+crZ7cGfy+ATGHNzM7cmW8PrA8JzQWSmJjvIMH2KG7X6BsTkeDp29
+221FcOtsR0CJ9s2n0pjEOKDF2U5vRLXiGMIHKwtYflK7+SFwiWR66plwjxOYOV2
dRvY6pMJWkWFo3Cw9KscHOZvSzaxejgUvTekuw53CGLfVbqSOTgq9bkUqs6mgTBQ
wpU0x/nQMd2+idShQdFIhz9xXk2pRYfC9zRPnTfWqTQIJFFr/7+KKP+nZEbtzWbH
tkQKP6Nl+a+NKpP5YcnUEWU0kxbIrh91VFlbIJqgB5x1u6TDMPBQJSockUhZ4W1u
63tI4/FzCDuxuco0CkAp2Lj3V28fvmz9T87BFg7/Y/7KYeWq3KLHaOrRF5JVhxA9
ezBDyLaJ9fjUX7FNKXhFYIEF5cHU3GTf4exXgmeenXMAlnQqUpb+gnFs0V0jVMia
SmvanyKgV/5qJ48u8OSB5oby8AGCOiKE9dYdUAxHxn4a8Fj8Do517GmJuRO8L/MO
vT8TGziWzKVkoPC+8Vty1ykdVcF4IFAtfm98uuxORoItYX9JX050zKS9gpQJENmt
KhgFdHTLozcQAJCdYMPo1SPgxxRI40BCD4ekuA6ZMzdTuuso9XTTLf3moh+eu0Y+
Pv6ISOFI93g09V34H3XnuaOnA16DyApO6mCPZ58DunjRRYHv4wxbeff6vw9twPQb
bcdNgI2ZXSKjUef4FygkpUhy6075m6zoUfcsR/22Ggzd8Qw3chfoqiBZs7SlGwEH
sxkB+dJvgCMwYaLGqS9lNbHSEFnETb0ihFSHKDrK6pYrFnfsuArJN5owuTN+dzim
8g0agB3Pzs4A6+9Hpt9sN78VEY+ZzFhQD8knEgqSNbFIaNxsG5/L8/RcXuNyPvvz
5MIkhfacOX37YFfAhCORfN3PaVWYQ7oDnv8Nh69i9SkxXKUcwX0/iYaim8pwwM6a
HXhTlqBgKR5b5Hcx6Cwx1GTRUpdbJYThmz7LQgzjBIhmoGbxR3vyJjSyPGBZa3Ih
jAETYzMt7j0qGPiJUx7kcHeK7apAH1xB4DnfuPLSC1GjWptb31A22wjWy/75SJk8
0FoyYKPj5RqiQF+5CcSLRwTJEEi+9ggM8mUJiFNUOtfZgATgmCyEKG4fL1N4iyrj
fBha11/EO3LcH4t0rj5i8SkT+w8A0bgpOliLhnKpFQMy+cRewkzNIS2ynGlHqnKA
biuBEr1ji0iHNHczTMuqSYwraVHis+098D+1me5+9SaDp8yEIKgVuTUAuQINBGOV
dwcBEADKRVXM2aPWTzbIFu/dVFuaOsJ653dZyDvkuK3N+6tkxM/ChhwrsaOTcH32
drei59OASOeKW2qYNe2+2tIT5Gt+xxZC9vJ+FcjHcdArAJt0BH1LHSuscCldkrnG
HlXK1dbqXDeD3Vw+7LtSO/wOAllxSRtOO8ydGfPZSWYkBIgKu4IbxKIpwtLJvtGu
mqP6HKgoFgijNcgjzssyV5Xka79vUkZ2M4jEfqtlNMfcYPV5ehjMFJhcGyXPZzkV
byx+321Ov3bvBIKAGK4JRThWRv00qJUhlkUjImr89CUg/4vqxDeGiE+FnwXvi/Vs
tQJqzcYgDvsypsx0LtyUsQ53K/svoXHfltIKsNdBaCPY1i3C+5EfumtIjo7iUl0q
Katpdw2n4j8bQhZq+AhkIcwRzOnuP7/zTXdxdF6VWyekUKYe5ZH738/Ktpb1JR4X
sydKdcV5oX3/Qh43VTIZgJs+fopgQF5NpqLbTXym9KRRHp9cWNyv1DwOIPqD7Eo+
PeTLLyIPXiDAnAt/89EhFDFh/7Lcsicz6n6cnxsGXk3CI/b4NqPkZxr5nC1WAPxB
k6D5PdOZL/3Cu96MvG2cjgdMzIO0/T2loC1DtLKo6qNDMN5JxckoN6EV5XIpT0L3
z3V9Cb8C6XPQ1pa9ftRpAlydPJVroCXPegiULqSkW9SGwtKFFwARAQABiQI8BBgB
CgAmFiEEC+MydddMlTx5+BEH2a0qGAV0dMsFAmOVdwcCGwwFCQICKQAACgkQ2a0q
GAV0dMsMLQ/8CA5hHB3RZchUU6JlAY23itrs3GSIZJIC+fJXOt1D5XuYidhQb5mE
vtK0Eud7FRLHHJ+Sn5DsE68ipZb1m5dCM7oqn2sMQC3H1qX2SR80zEpwx34F7zjs
hTEK7qkQwBwK+zzQtAp209LBBdmRc2nfDPXy6rZF1MeJoBAjRieAlaoO+SmPLlM7
/wZkcBH+fOxU+f4gCHneTIxQdlRQ1/RgAXL002x8QmSMZ8Zd3LNeUsU9JQKH2drn
lOpFmFmUlxXmhDP+MpjycLNppldAJGyYIS1wp/TMd51+jU2WfR2q267vkc1t/x4N
```

```
w2XdCfb5KPD2nz9JdTVrYMl++Aqr04I808V056q0V83HMyu34XAtruItKaz7V13M
sou0sKaC4pGmQixRYVJjsfpkuU47YBchWQTcZzu4gMJM33E3AvbD6FmhZokH+quR
CGMxvlkKZ3I/NJsZdx1jj+qBY/4WSUWLHinjJLycy06PO6xXTOgrDKy8BE0envml
jNG7TxYpda11QAjTo7MtG8U1fDiRH8sZ2Jso75wbwdJHYmlRA3VYcYEqsywF+JoF
2uPf1ejP7szK/p9/wNSXZOFen1ZvEuXBSF8aFGrd3AeMRmk/Tlg96NTT6+nkvVaN
BLCZUZfNUhuvH0vkLZNn19u0+vGBuAGt8zH2cVLFbhS7U7kG0QvnyaSJAjwEGAEK
ACYCGwwWIQQL4zJ110yVPHn4EQfZrSoYBXR0ywUCZYXLqgUJA/J9aAAKCRDZrSoY
BXR0y6DtEACbExPX5JXm+2MzxLj/Rid48fDxnVpu5/vFDXa9uJEuoDl1VHz2a1mu
UqNtYXCMGW+D2Q6h0lUkJ0XIioUpxu4KZWDzRmePhVmy6/5gUWGq45FRS1Lgj3sU
iI8VdSkhiZ1rPWE1QCmQM++RqQjC+ow+m4gjkAkPOsKjnVr2SGUfGL7q68UeuAAQ
dK427GhvHDTlF9XMAH3S8LYX+uHUvtpt5ajRgx5D+XjS9ESMQuYF2BKdJb5pTB92
bjuVq3x+9z0w4PPW8V3roaxZkhMH+MVputN6IvP9qkL0ff+429KPre9TULlq1iBT
jWVloGQVBrKPrsk914GOGKihkd89kMyh2JyejO0r0mKaku0bXMXAlxdBxjXxEvHA
ILSlaRbW7UsF16fDvKC8o5lSDvzY1KWyzhnRSLGq7HsyxVfX9n6metyd+IclEoPC
62FVdXwOQyVKXm21+IS6XZNDDLYBuNwcCiBpSU6S+UZEP7kFN2FtpGoq8u6PtYpg
w9E0c0MbXY1TiOb55UzXHyWSCHy8eHawAQNdjr6V93Xz1etkfvAQJSRU+N9AhugC
EiSAaiURb7opJp2RlLxB5jlLPxS222l0z5CnO7UdliMJY0bzFfkJPOHFnF+mHITL
JTLogzNZrZUGYoN+FhxkhPswlp1wLJSB5UsKMe6n3TtCs9l53EmyAIkCPAQYAQoA
JgIbDBYhBAvjMnXXTJU8efgRB9mtKhgFdHTLBQJnhEEuBQkF0/ICAAoJENmtKhgF
dHTLJtUP/1dPcTrrUm1lOOCkbq3ZxirJOegxZal7YxmCagz6zg6V9zPCA39jLkyD
dsnzzL/fOGwGzgPUS+xGhlZUFFvUFj1ETLGShBUJY4X8USTvSaBg2CO7kQ50hIul
o1b4FJOToiIdChCOIxHGiws73iBHLMK1Q0gJ/u5xr3sD/Ea8e6jFDbywJbcYjMnx
l53mr0bN7H7rM3W/S002PJQlNMa7JWl6673wpGkW7JBYt0t9SPtWQf3sLgIR6n3N
d3kQYz8EDmpyRXqQMz+BheNX1UDfRrbLAEwg/nVejhYWI31iWziHD79u+8ArxiBb
nDPV0Y25AiGBUBOVvtTv2nLdzm49bAfdAvnFToAp4N+2Knzm2CxdfUROPmuzVAIf
xsR4/vE8VMr762llsnzakzA4xzEoTQ4xtKksxV+738lKZnwH8/umne/GpLPSzH/v
OohpLwYVxbO3M1b+D1lmFdflkoTVNmPcjGh986VlGkRHQdOxymy72uGNYnKllcfs
+sbNz77Km03ov5qF4mZO6IaWwZtIbSZZIttgEWKxl0iusOUULQmaAVtAxI+STQY5
Cul2wFt1RbgANSecC0ts9HiEeWnT/v0sUFG06vmRYjhnQT9YP+/N3u8xHcgNGHwN
ilPTnS87qVIsf8fiWHMZq2V53UjDm3dCIljocF5YGq2ySTEXjjkYiQI8BBgBCgAm
AhsMFiEEC+MydddMlTx5+BEH2a0qGAV0dMsFAmluhHIFCQmhpDsACgkQ2a0qGAV0
dMs5ag//V7mahuI9VikX1lG599GSjm1cXeDrZmGtamz9hYE7EzGlyxf5WQn83Sxu
2kNe9UbaLcIpKzZ9CWcb7NfC6odUIzJsO21qSKNuoP7FV9472JTNFMhaj1+o1P8b
v1nGOU4H9IH/y0Wqkuteef3t4le0iCtePMU3vXgfUFAmm/VF/tgmJQb2hLkXFNCQ
QwK//jjm9QHajaU2wEpvskarsRGfGBintBbUqUUxyPgirqAC8dFAWrjefmGePWEW
JfRLql39II+xbjEzADBB8yqnxLuaUQfVF7t2oAOfp09dP++bCua7FO1WfuFoa1Yw
um84/XszGgkxDRFkBxabpMCurPWgyhkGto4TqKdcWTMl4S1rTxt4Bm/6fqMn+04z
/AiXgU/iIbEKhvQUxsTxyNBLPFIRD7VmrqrOr3wkHgGHJgP0656ABq5J+T4mtFhX
uL0U9nGSm5lV9Hl1ana+Kcs4DqzgQO3QJk/SoulBDZ6at0pK2Y016JrvY51fYpx6
a/IOaXE2JRi8itVlpciDTjSOJs4eNlUx0C3b/LdzU5LZqHtBea1xWxlKPn0VFjYr
YNML6o5B6AL05AcySZj8kNZTWE2ZfOR2usLdiB8b+u2F7WGXZBiQYSnr75WsN94S
nYwO46v5SlzGqTwEN/UBfTkKPklVNgeOMn7fgjziEtMmtBlb7ws=
=ijsc
-----END PGP PUBLIC KEY BLOCK-----
```

### D.1.2. Core Team Secretary <core-secretary@FreeBSD.org>

```
pub   rsa4096/4D632518C3546B05 2024-02-17 [SC] [expires: 2028-02-08]
      Key fingerprint = 1A23 6A92 528D 00DD 7965  76FE 4D63 2518 C354 6B05
```

```
uid                           FreeBSD Core Team Secretary <core-
secretary@FreeBSD.org>
sub   rsa4096/CABFDE12CA516ED2 2024-02-17 [E] [expires: 2028-02-08]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQINBGXQ1o8BEAC+Rcg8cmVxuP17Vu+q5KgCx/XiulQuqKXAqqBlYCH2jqk6DINP
yFrREGBhzd/qNmlAYEahQ4Zgl0bUZNTrZVDyzicOvPP0jH+KSTQwRs7NOawEdlVO
cyHrwDCPEqf5ZzD4NhfTriEOw+j0pEH/onitUGvoQRtx15xWyaJQxDEBMTYMLewE
86D1bltwnTNczE3UZb7oQLJXkAX5hcLtou70XJGgZITvJkK+kp/xot2eFjnqRz/u
WeXnKhYAmC07EKwZ1uw047eHKwMMRBYqzApLwoQtfE430Kxf2q8de64x8zDbi6YM
1J4r8OAxOtHVyfJ0j7Q23DEZz0VVb4b1Tx5OG2Re/KSNvqI0awJO4TcRmOR88OyY
dzyXgnX6Sa7GVQY1FXvn7vtFuDAt7egZOzeomSHL9bdX07LTQ4UtM88EV9wm3q4q
smoatV9jsvPQ1zxCU3aQD/5eWTJH2/kz1LIuBL/Qi5XQpJn91lBtUWJrCgkHWPGu
f//rnnXmsG7DAChW+yZ7cFO8lfNa8sFhPqSxCYphWmJTrvadyQtDngB8JakWdnmK
pfGS6y5lel+181vw38ZZKt04AKM+nDY8051lBM7Q9Q6kTLI33UZeImndx5xYukVD
kV6aQ31HYfEark15c7iEz+OAcwFnM2ntXMt7kKGd40CqzusiPcQkPqPbAQARAQAB
tDhGcmVlQlNEIENvcmUgVGVhbSBTZWNyZXRhcnkgPGNvcmUtc2VjcmV0YXJ5QEZy
ZWVCU0Qub3JnPokCVwQTAQoAQQIbAwgLCQ0IDAcLAwUVCgkICwUWAwIBAAIeBQIX
gBYhBBojapJSjQDdeWV2/k1jJRjDVGsFBQJnp8PiBQkHeofTAAoJEE1jJRjDVGsF
GMIQALhj+mNpH8OmTFeihQ6t9P8un3llz6Wmqe/Q+ULWeqJvV/uC1J5T9fnoGhwF
MgECuguXYJtoYQ16KXnsSOs1tcqMOK9GtEtFJTGe2DtfLiBednwUvu9j3HmTlwLN
M+7rqkiC0HCg2qSjcMjxbVbA5BSgNkgfyTSO2YdjfaZ+ceiHwo/qa5xWE6i2dMR1
PLGMMHTeELdtdSH6VR9/3h0qt3qzwdMBRCVAQHbim7CqwjUH9jg+IOySXl81jNoB
xuVZR3pKshyY4Oxo1dK+W86Uiff/+c4jCAxokGWIR7C4MkZZWUQqV7920gkZFC/5
qzpT1A1/sFUg8HfFTr0vCoPSVFWn2+Tto3vr78DICVaAf02aYAFlyKK18BMCoHkS
hDDO+/JQZmvHOIgEYK+T2WN1c9gm9IDJzGZuH0X2C/vkREnJKkccJfB4pXuN10wt
fqyP9fn8h6+/t+sv3qNLm4d8fkmLXofulG3WB4i/F+Hip2rjPvvBCF7zl4xy6cJ2
xVY5HUOBTqmIwVhYwUpXaqNoWa/qJBLTuot3z7ciKmKX6Lq+Dze5dzhrPNl/CalC
HBf3miHRK3TZbYLooG3bcEWgxU2BnBi3vl5NpCoUOKkPYRlALXi2TyHmPxO7oT4e
mIzS6BgnXOqO+AXvbKKfayuSePdBqkNMK/SMC4Dylkf6Xj25iQIzBBABCgAdFiEE
EBpxaxYrAOVb7eoFrbv4YQo3ibcFAmbu7x4ACgkQrbv4YQo3ibcr5A//TIcbD/EW
YJz0zrUQdc9xG3UNfU6uHmQzAuUy5ginevyqv0TSso5qvSkOHvdxbi41rfMiB2RJ
V3q0n0PSvHFld89fOTZUMZXaPvozCiBYWScrt+KA/2pq3K8mUumHS+IFtpHLL2Tu
+gI8XCHUFxO9HUThM/rFlIyEdzoctgmqQ7IG4uZG+o2J3w09llhDAUe0vraJK1On
p9yFACnRrhqvl41JeUWcv9MH9JcwHUqtUo9WLTCIb+hkByTOyRfHBYpYw//bdXdf
6rkCwKVWpyMDbk61zq2VsS1kqbP9IH/A8CsBnA6mg+zPq2i7HIFw8Swj4OGJcIvG
a9ubUYJRjDhX/vBpNrtncANZ88FQmA+Maq0vu0LS5IIGyIKkvd1fKIsyvBDDa9kE
nfCW0XMkJA0Gf+kxdB+elXQHBwKO2sr5BiKnJT5lJq3Yu9fxxGBnf93yiN4E9bmF
gG7cZxpyb1Bp76TJhLcANyybOTjCtiNRrgqeaSx9/6hSPfPigGXIne0H2lmJO6oq
jUrsYmFiWvU9sc7AcPVw/eHG8FgW35TuwKX71z8w994iaahUPNcSVyXOUD+QNROv
HhGUXrc81t613rivh22N0NZpNubVatq43KV7+/bnPyWBIi1Awh3vIFsNNSQSrYxF
lUuQaAHQXTeZMZ/7npE4t86seMt0T7BGn765Ag0EZdDWjwEQAL3VwFifpnRCYzQ4
VZ1dAjp8w542iRprqeA+C3tvNbk2OvKpN3DIc49l2bgNZ/VI7/T58lEKrFsgLK4w
AWtv18OV7xuh0AuOblmq5MvcPrUelkPj5HA7M6Ng/rAubuHfwdP/IjIrzzY6+XDh
fP9N5KIv9VRJYT23BbvLPeit4J4tQEB/NpxL3L6zI75qq4R3T/EkHP6Y9Buup9Lr
isJlcxkSK+CyORCgTpEz6fWsXiTDgS7cTaQ969XCygBpj4wRQzwkBdokxo1wsiFt
4zLt07/PrPYjeHltTDkrF9XDNhLNJ5GliHnd01oHg1j5/n+9Osvh1maoFwuBxXTN
nTZ2P+7RKLBaVQVSkUa5KJbXoM3v7bMbXM5aLs2XjglrAzrZOV+Y7zOu5UYQdpXd
7x8XAEtEozRJzt7dosQIhKx9h4LlltFFulDUpf5VCvcUEoAzMbiX0aju9n5Rwsql
```

```
DHatU9Mm3W8OdFXj1foIXZD+VX2Jp9DgxPLoAJ0CWhPXJ8f+WFSJZCjWoPJEJKWY
0EOxiXyAuvAyniAZAC/eKZkfGckXmu7edRgYbRTTwPmZ/axa/k9aHsHLwmbxuZo/
xxQXzqU70EleHZ31A3mOqsC1epjN6dn4AFKgvVesP/K/fbvSsfsfiABS2A/68ne4
zJG73Tnk4L6J79vrXc2iMJbmdg3ZABEBAAGJAjwEGAEKACYCGwwWIQQaI2qSUo0A
3Xlldv5NYyUYw1RrBQUCZ6fEFQUJB3qIBgAKCRBNYyUYw1RrBQd9EACMVckxzy4w
aUGlERguJ+kslS8MkMjNqnfDPLRDVxbUxNvbbhw7/u9blM65DCGCeNLc2n4oiu5C
E3I095AKmvq/0dOa81mEEdZkC1CVc64bXWbEyz5AtSHUpgdxRso6C+YopndSiz1T
WcIagQRXfWaw3FBWPooA87gmibmSmCegCtqx+uyc5QxX2eFI8mRK7vlfnGpYKHs0
D1/yUSGQOwoNRJ5FYm+ynfDE3FzHEQL7lv1vpv1k3xNKfBziMMg4IMEBKNHV4VKN
qJpa8UCodeTGSWQdNnCeCWPsxz5oQjCcVH9Z3e8sMpWLHhRcBZzSwXUOws2GbRMH
xHwrfrpHJcrBhe64pgfGOvZLUJ9BDs+8egTHsqRFacipbtTR+hhVhuJEHDaQQWuM
8IRHj9HIuTAczET8JTDHTMIoo8DZdOtiW/YgqCDwYghkI77d12oNQqYoeJ2HiqbK
cGzwCpsR0A+p/iOAxJG13tsxqZV8TQ8iTokWG6ACtZ7sfeWEhxqMbUKUMgogZn0I
3n1kV+tUZC6BQrIYI7TiKg95wLZsIydeoIsQoNZwvyKAXfVmQ62YjIX8njZwN+07
8/ipUPJxCYa8zL8BZyDmoFJqa3y9z+11+vtiZ9t+aTwGvjpHDwyeCJco7go9cU3m
GRFZYciqIoG4n3tl8Pob15vFlVqk47rRqg==
=8TzT
-----END PGP PUBLIC KEY BLOCK-----
```

### D.1.3. Ports Management Team Secretary <portmgr-secretary@FreeBSD.org>

```
pub   ed25519/E3C401F60D709D59 2023-03-06 [SC] [expires: 2027-03-05]
      Key fingerprint = BED4 A1D3 6555 B681 2E9F  ABDA E3C4 01F6 0D70 9D59
uid                           FreeBSD Ports Management Team Secretary <portmgr-
secretary@FreeBSD.org>
sub   cv25519/2C92B55E27A641C3 2023-03-06 [E] [expires: 2027-03-05]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mDMEZAXJvxYJKwYBBAHaRw8BAQdASFAC20WL3R1T6uNyGMZbfJCxDkcP4C5vi3Op
tcZ2fbq0R0ZyZWVCU0QgUG9ydHMgTWFuYWdlbWVudCBUZWFtIFNlY3JldGFyeSA8
cG9ydG1nci1zZWNyZXRhcnlARnJlZUJTRC5vcmc+iJYEExYKAD4WIQS+1KHTZVW2
gS6fq9rjxAH2DXCdWQUCZAXJvwIbAwUJB4TOAAULCQgHAwUVCgkICwUWAwIBAAIe
BQIXgAAKCRDjxAH2DXCdWYN1AP43TjyfZtZ3DLYT++g0+SuPsoO/3yWVybA+UmFL
zb8MngEA+LLNUfvEwCuXS/soh+ww5bpfmi3UUmeGiQEAXug3iA+JATMEEAEKAB0W
IQT7N0XIbxXo7ayBMvzYKU7Du8TX1QUCZAXLkwAKCRDYKU7Du8TX1XHMB/9R1MX4
6zMgpKqPPt76GOI+eGEdBK6bY8aJZjQGdqTh9f6VtXVoTGIG7cvhc9X8tDBoB0PT
2KZWheF51AV1+NHU4HwLAQ1BMebrFvWSfkw4xg4fBGwDhz9/GN85No+Js772V5ey
8lRiL6meRVWxMlLyWcxGd8JjcC5yX/iAUQ3SBGCLqW7unWjjg7CTd+AMBwcqPGrv
ax8q6eFVguJcHJAjMnKf6HAy4cpK3s+uMoUBCGnszSN12B3ysKfyC4pNO/pix5tA
Q5v8aRqTeFPh5zmNhWo0KGPzplTPqRQSHDl7GDQC8Ru3MhzFkeWzHsexjZVwS6W2
DPcYpuuAsA0XOZIZiQIzBBABCgAdFiEEEEBpxaxYrAOVb7eoFrbv4YQo3ibcFAmQF
0u0ACgkQrbv4YQo3ibccwg/9F2Xuic3nhKxRbB3mJeDo6SYQETa/Gh1qQ34+8zlt
8UMazOx67gnYQfy+pXjro6eQ2up0a4eUYezcNOudqAQD21nRz3HA6EQVNcE/TzEA
xl5CJntTaLOt7S+EDXFW5BuQIvhhoMGgm8+WNVgA0EJ7tfL0OcYBSvr19fqwChEn
9c14cSk6mgHSsleP5NvskYN053pxHwy0LTSb8YBBv52th37t/CRFC1363rS5q+D7
JixFopd1O5pKpA5ipvE4gGgRjPtwjx0SjjepwK/3fuhEJQQyKzTIKlMfu2Dj/iR2
Li1Sfccau5LQXOj9fUITU3u1YG7yrm8VGzT7ao4d+KRwgMLjd2pLqiGIbbJwGBiP
```

```
FRmtilWQoeIlmSlFX4obAA517DOK0pW1mH8+eEn4EJd3SekT3yzFyKTASv0J48Z8
3F928xg+eZvHxVC0t1J+J5IG0gt3EEncuWKIPQGR7PiQbti6R3FQVTz6WfMWOebP
Qi0E9F/Aqakr6Vj2sKGrDq+ebpaF5G8Yw1YrUl2IDiPzkCegp3ZbI0wh11Xvzhi8
LXPQGK4jBQas4G8cegfitzmtdGRHYrbMv0R9I4mvaL+WlOuD2AvyVG28lguqVhnN
AZP+ohdquYyX2CNCVvbKWAtXo6Ur0vWG8BL8m6defAtEkIwVBALaOHQOSI3aNUz4
lwy4OARkBcm/EgorBgEEAZdVAQUBAQdAsefmSfxEOdOr02+K/6noYCuJ1FeAWVz6
jFYQ+9w6jggDAQgHiH4EGBYKACYWIQS+1KHTZVW2gS6fq9rjxAH2DXCdWQUCZAXJ
vwIbDAUJB4TOAAAKCRDjxAH2DXCdWRl4AP9h5ot212BK29S6ZcMBhHvmtF5PG1oD
c7LnZycSRmbFiwEAndCMpAGOhDW8iVgDd0wLQq/ZMPe+xccfG1b3zFH2EgE=
=iiAT
-----END PGP PUBLIC KEY BLOCK-----
```

### D.1.4. <doceng-secretary@FreeBSD.org>

```
pub   rsa2048/E1C03580AEB45E58 2019-10-31 [SC] [expires: 2022-10-30]
      Key fingerprint = F24D 7B32 B864 625E 5541  A0E4 E1C0 3580 AEB4 5E58
uid                           FreeBSD Doceng Team Secretary <doceng-
secretary@freebsd.org>
sub   rsa2048/9EA8D713509472FC 2019-10-31 [E] [expires: 2022-10-30]
```

```
-----BEGIN PGP PUBLIC KEY BLOCK-----

mQENBF27FFcBCADeoSsIgyQUY8vREwkTikwFFlNg31MVy5s/Nq1cNK1PRfRMnprS
yfB62KqbYuz16bmQKaA9zHN4FGfiTvR6tl66LVHm1s/5HPiLv8sP14GsruLro9zN
v72dO7a9i68bMw+jarPOnu9dGiDFEI0dACOkdCGEYKEUapQeNpmWRrQ46BeXyFwF
JcNx76bJJUkwk6fWC0W63D762e6lCEX6ndoaPjjLBnFvtx13heNGUc8RukBwe2mA
U5pSGHj47J05bdWiRSwZaXa8PcW+20zTWaP755w7zWe4h60GANY7OsT9nuOqsioJ
QonxTrJuZweKRV8fNQ1EfDws3HZr7/7iXvO3ABEBAAG0PEZyZWVCU0QgRG9jZW5n
IFRlYW0gU2VjcmV0YXJ5IDxkb2Nlbmctc2VjcmV0YXJ5QGZyZWVic2Qub3JnPokB
VAQTAQoAPhYhBPJNezK4ZGJeVUGg5OHANYCutF5YBQJduxRXAhsDBQkFo5qABQsJ
CAcDBRUKCQgLBRYDAgEAAh4BAheAAAoJEOHANYCutF5YB2IIALw+EPYmOz9qlqIn
oTFmk/5MrcdzC5iLEfxubbF6TopDWsWPiOh5mAuvfEmROSGf6ctvdYe9UtQV3VNY
KeeyskeFrIBOFo2KG/dFqKPAWef6IfhbW3HWDWo5uOBg01jHzQ/pB1n6SMKiXfsM
idL9wN+UQKxF3Y7S/bVrZTV0isRUolO9+8kQeSYT/NMojVM0H2fWrTP/TaNEW4fY
JBDAl5hsktzdl8sdbNqdC0GiX3xb4GvgVzGGQELagsxjfuXk6PfOyn6Wx2d+yRcI
FrKojmhihBp5VGFQkntBIXQkaW0xhW+WBGxwXdaAl0drQlZ3W+edgdOl705x73kf
Uw3Fh2a5AQ0EXbsUVwEIANEPAsltM4vFj2pi5xEuHEcZIrIX/ZJhoaBtZkqvkB+H
4pu3/eQHK5hg0Dw12ugffPMz8mi57iGNI9TXd8ZYMJxAdvEZSDHCKZTX9G+FcxWa
/AzKNiG25uSISzz7rMB/lV1gofCdGtpHFRFTiNxFcoacugTdlYDiscgJZMJSg/hC
GXBdEKXR5WRAgAGandcL8llCToOt1lZEOkd5vJM861w6evgDhAZ2HGhRuG8/NDxG
r4UtlnYGUCFof/Q4oPNbDJzmZXF+8OQyTNcEpVD3leEOWG1Uv5XWS2XKVHcHZZ++
ISo/B5Q6Oi3SJFCVV9f+g09YF+PgfP/mVMBgif2fT20AEQEAAYkBPAQYAQoAJhYh
BPJNezK4ZGJeVUGg5OHANYCutF5YBQJduxRXAhsMBQkFo5qAAAoJEOHANYCutF5Y
kecIAMTh2VHQqjXHTszQMsy3NjiTVVITI3z+pzY0u2EYmLytXQ2pZMzLHMcklmub
5po0X4EvL6bZiJcLMI2mSrOs0Gp8P3hyMI40IkqoLMp7VA2LFlPgIJ7K5W4oVwf8
khY6lw7qg2l69APm/MM3xAyiL4p6MU8tpvWg5AncZ6lxyy27rxVflzEtCrKQuG/a
oVaOlMjH3uxvOK6IIxlhvWD0nKs/e2h2HIAZ+ILE6ytS5ZEg2GXuigoQZdEnv71L
xyvE9JANwGZLkDxnS5pgN2ikfkQYlFpJEkrNTQleCOHIIIp8vgJngEaP51xOIbQM
```

```
CiG/y3cmKQ/ZfH7BBvlZVtZKQsI=
=MQKT
-----END PGP PUBLIC KEY BLOCK-----
```