Harald Maaßen

Linux Professional Institute
LPIC-2

LPIC-2

Sicher zur erfolgreichen Linux-Zertifizierung

- Vorbereitung auf die Prüfungen 201 und 202
- Kommentierte Testfragen für beide Prüfungen
- Aktuelle Lernziele der Version 4.5

Prüfungssimulator mit sofortiger Auswertung zum Download

Rheinwerk Computing

Liebe Leserin, lieber Leser,

dieses Buch möchte Sie gezielt auf die Prüfungen des Linux Professional Institute vorbereiten, die für das Zertifikat LPIC-2 abgelegt werden müssen. Harald Maaßen, der selbst Kurse zur Vorbereitung auf die LPI-Prüfungen gibt, hat nach dem bewährten Konzept seines Buches zu LPIC-1 dieses Buch zu LPIC-2 verfasst.

Aktuell zu den ab Februar 2017 geltenden Lernzielen stellt er alle prüfungsrelevanten Themen vor und erklärt, welche Kenntnisse Sie für die Prüfung haben müssen. Zu jeder Prüfung hat er eine Testprüfung mit je 120 Fragen verfasst. Die Fragen ähneln in der Art den in der Prüfung verwendeten. Alle Antwortmöglichkeiten werden im Lösungsteil näher erläutert, auch die falschen. So finden Sie schnell heraus, worauf Sie in der Prüfung besonders achten müssen.

Damit Sie die Prüfungssituation schon einmal testen können, wurde ein Prüfungssimulator erstellt, der alle Testfragen enthält, die Sie auch im Buch finden. Die Oberfläche ähnelt der in der Prüfung verwendeten. Auch das Arbeiten unter Zeitdruck können Sie testen: Je nach Anzahl der Fragen wird eine Zeitbegrenzung festgesetzt.

Falls Sie Fragen zu diesem Buch haben, Anregungen, Kritik oder Lob geben möchten, wenden Sie sich an mich.

Viel Erfolg für die Prüfungen!

Ihre Anne Scheibe
Lektorat Rheinwerk Computing

anne.scheibe@rheinwerk-verlag.de
www.rheinwerk-verlag.de
Rheinwerk Verlag · Rheinwerkallee 4 · 53227 Bonn

Hinweise zur Benutzung

Impressum

Dieses E-Book ist ein Verlagsprodukt, an dem viele mitgewirkt haben, insbesondere:

Lektorat Anne Scheibe
Korrektorat Claudia Lochbaum, Dresden
Herstellung E-Book Melanie Zinsler
Covergestaltung Barbara Thoben, Köln
Coverbild Corbis
Satz E-Book III-satz, Husby

Wir hoffen sehr, dass Ihnen dieses Buch gefallen hat. Bitte teilen Sie uns doch Ihre Meinung mit und lesen Sie weiter auf den Serviceseiten.

Bibliografische Information der Deutschen Nationalbibliothek:
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

ISBN 978-3-8362-4499-2 (E-Book)
ISBN 978-3-8362-5511-0 (E-Book zum Buch)
ISBN 978-3-8362-5510-3 (Bundle)

3., aktualisierte Auflage 2017

Inhalt

LPI 202

Übungsfragen zu LPI 117-202 503

Wie man Zertifizierungsprüfungen besteht

Herzlich willkommen!

Wahrscheinlich werden Sie sich jetzt wundern, warum hier gerade ein Karren vor ein Pferd gespannt wurde, weil die Prüfungen doch eigentlich erst zum Schluss abgelegt werden. Damit soll vermieden werden, dass Sie taktische Fehler bei der Prüfungsvorbereitung machen, die schon viele Prüfungskandidaten vor Ihnen begangen haben. Lesen Sie also bitte dieses Vorwort bis zum Ende, wo ich Ihre Aufmerksamkeit jetzt schon hierher gelenkt habe. Ihre Geduld wird sich auszahlen.

Dieses Buch bietet Ihnen die optimale Möglichkeit, Ihr Wissen über die Themen zu verbessern, mit denen sich die LPIC-2-Prüfungen des Linux Professional Institute (LPI) befassen. Sie sollten aber zusätzlich die Dokumentationen des Systems nutzen und einschlägige Quellen im Internet konsultieren. Es werden im Buch gelegentlich einige interessante Adressen genannt.

Das Zertifikat, das Sie mit dem Bestehen dieser Prüfungen erwerben, wird Ihnen erhebliche Vorteile auf dem Arbeitsmarkt bringen. Aber auch für Arbeitgeber ist es von Vorteil, die Fachkompetenz der eigenen Mitarbeiter schriftlich belegen zu können. Eine vollständige und aktuelle Auflistung der möglichen Zertifizierungen des LPI finden Sie auf deren Webseite:

https://www.lpi.org

Hinweise zum Buch

Für wen ist dieses Buch?

Dieses Buch richtet sich an all diejenigen, die zur Förderung ihrer beruflichen Laufbahn Fachwissen erlangen und dieses zertifizieren lassen wollen. Das Buch ist ausdrücklich nicht als Nachschlagewerk gedacht, sondern bereitet gezielt auf die LPIC-Examen 201 und 202 vor. Um die Level-2-Zertifizierung des Linux Professional Institute zu erwerben, müssen Sie beide Prüfungen bestehen. Sie finden die aktuellen Prüfungsthemen in diesem Wiki:

http://wiki.lpi.org

Voraussetzungen

Um die Zertifizierung *LPIC-2: Linux Engineer* aktiv verwenden zu können, benötigen Sie zuvor die Zertifizierung *LPIC-1: Linux Administrator*. Die Kenntnisse, die Sie wahr-

scheinlich bereits für die LPIC-1-Zertifizierung erworben haben, sind für die Arbeit mit diesem Buch von erheblichem Vorteil.

Wenn in diesem Buch Themen behandelt werden, die aus dem Bereich »Linux für Einsteiger« zu stammen scheinen, so hat das lediglich den Hintergrund, dass diese Themen für Sie prüfungsfähig aufgearbeitet werden sollen.

Damit die vorgestellten Themen auch praktisch angewendet werden können, benötigen Sie einen Computer, auf dem eine beliebige Linux-Distribution installiert ist. Da die Prüfungen des LPI unabhängig von einem bestimmten Hersteller bzw. einer bestimmten Distribution erstellt worden sind, sind Sie hier in Ihrer Auswahl grundsätzlich nicht eingeschränkt. Praktischer ist es allerdings, wenn Sie mehrere Linux-Distributionen in virtuellen Maschinen einsetzen. Sie können dann die distributionsspezifischen Unterschiede selbst sehen und gegebenenfalls testen. Sollten Sie sich für den Einsatz mehrerer Distributionen entscheiden, empfehle ich Ihnen die Auswahl eines Debian-basierten (z. B. Debian, Ubuntu, Mint) und eines Red Hat-basierten Systems (z. B. Scientific, CentOS, Fedora). So können Sie gleichzeitig die Vor- und Nachteile dieser beiden Welten einmal selbst (hoffentlich vorurteilsfrei) unter die Lupe nehmen.

Sie sollten nach Möglichkeit die meisten Themen, die in diesem Buch behandelt werden, praktisch anwenden. Es gibt hiervon wenige Ausnahmen, wie z. B. *Icinga*. Sie müssen lediglich grob wissen, worum es sich hierbei handelt, aber der Installationsaufwand wäre völlig unangemessen. In solchen Fällen reicht theoretisches Wissen.

Der Aufbau des Buchs

Das Buch ist in vier Abschnitte unterteilt. Für beide Prüfungen, die zum Erwerb des Zertifikates notwendig sind, gibt es jeweils eine Sektion, die zum Selbststudium der jeweiligen Prüfungsinhalte gedacht ist. Außerdem gibt es für beide Prüfungen einen Bereich mit realistischen Fragen, wie sie auch in der Prüfung gestellt werden könnten. Zum besseren Verständnis sind die Antworten zu den Fragen genau erläutert. Sie sollten nicht versuchen, irgendwelche Fragen einfach auswendig zu lernen, weil es sich bei den in diesem Buch verwendeten Übungen ohnehin nicht um Originalfragen handelt. Auch Fragen, die Sie im Internet finden, sollten Sie nicht auswendig lernen. Seien Sie sich im Klaren darüber, dass Sie spätestens in Ihrem Berufsleben Ihr wirkliches Wissen unter Beweis stellen müssen!

Die Kapitel in diesem Buch sind genauso angeordnet und benannt wie die sogenannten *Objectives* des LPI. Jedem Kapitel ist eine *Wichtung* (im Original als *Weight* bezeichnet) zugeordnet. Die Wichtung gibt einen klaren Hinweis auf die Anzahl der Fragen, die zu dem jeweiligen Thema gestellt werden. Die Wichtung entspricht nämlich der genauen Fragenanzahl in der Prüfung von insgesamt 60 möglichen Fragen.

Wie man mit diesem Buch arbeitet

In den LPI-Prüfungen werden Sie mit sehr vielen Fragen konfrontiert, die sich mit Kommandos und deren (unter Umständen selten verwendeten) Optionen beschäftigen. Es wurde beim Erstellen dieses Buchs sehr sorgfältig darauf geachtet, genau die Parameter und Optionen eines Kommandos niederzuschreiben, die für die Prüfungen relevant sind. Das ist aber leider keine Garantie dafür, dass keine anderen Details in der Prüfung abgefragt werden. Sie sollten sich also zusätzlich zu den dokumentierten Beispielen mit den *Manpages* der entsprechenden Kommandos beschäftigen. In der zweiten Hälfte des Buchs gilt das oben Beschriebene sinngemäß für Konfigurationsdateien und deren Inhalte.

Es ist übrigens für die meisten Leser schwierig, zunächst das ganze Buch zu lesen und danach beide Prüfungen auf einmal abzulegen. Sie sollten also lieber nach der ersten Buchhälfte die erste Prüfung ablegen und danach die zweite Hälfte des Buchs angehen.

Die Prüfungssimulation

Die für das Buch entwickelte Prüfungssimulation (zu finden unter »Materialien zum Buch« auf *www.rheinwerk-verlag.de/4353*) basiert auf XML und kann z. B. mit dem Webbrowser Firefox ausgeführt werden. Öffnen Sie zu diesem Zweck einfach die Datei *pruefungssimulator_starten.html*. Sie sollten dieses Programm aber erst dann verwenden, wenn Sie sich gründlich mit den Themen des Buchs beschäftigt haben. Sie können mit dem Programm Ihren Kenntnisstand überprüfen, aber die Aussagekraft des erzielten Ergebnisses sinkt natürlich umgekehrt proportional mit der Anzahl der Durchgänge durch die Prüfungssimulation. Wenn Sie merken, dass Sie mit einigen Themen noch Probleme haben, dann arbeiten Sie diese bitte auf.

Hinweise zur Prüfung

Onlineprüfung

Wenn Sie die ersten beiden Prüfungen zu LPIC-1 bereits abgelegt haben, sind Ihnen die Anmeldeverfahren wahrscheinlich noch bekannt. Seit Juli 2012 bietet Prometric übrigens keine LPIC-Prüfungen mehr an. Sie müssen sich also bei Pearson VUE registrieren, wenn Sie die Prüfung in einem Prüfungscenter ablegen wollen. Besuchen Sie dazu die folgende Website:

http://www.pearsonvue.com

Sie benötigen bei einer Neuanmeldung Ihre LPI-ID. Wenn das Konto für Sie eingerichtet worden ist, werden Sie per E-Mail darüber informiert. Es ist dann sofort mög-

lich, Prüfungen verschiedenster Hersteller bzw. Organisationen online zu buchen. Sie können den Zeitpunkt selbst bestimmen und ein Prüfungszentrum in Ihrer Nähe aus der Datenbank auswählen. Die Bezahlung erfolgt bequem per Kreditkarte oder mittels zuvor gekaufter Gutscheine (Voucher) und Sie werden sofort per E-Mail benachrichtigt, sobald der Termin für Sie reserviert wurde. Im Augenblick benötigt Pearson VUE 24 Stunden Vorlaufzeit für die Buchung einer Prüfung. Sie können also eine Prüfung frühestens für den nächsten Tag buchen. Die LPI-Prüfungen kosten derzeit 149 EUR zuzgl. Steuer.

Papierprüfung

Es gibt hin und wieder die Möglichkeit, LPI-Prüfungen auf Papier abzulegen. Das geschieht meist auf Messen oder Kongressen. Diese Prüfungen können normalerweise zu einem erheblich günstigeren Preis (nämlich derzeit 90 Euro) abgelegt werden als die im vorangehenden Abschnitt genannten Onlineprüfungen. Da die Papierprüfungen in Kalifornien ausgewertet werden, kann es allerdings etwa drei Wochen dauern, bis Sie über das Ergebnis Ihrer Prüfung informiert werden. In Deutschland werden Papierprüfungen zum Beispiel auf der CeBIT in Hannover und auf den Linux-Tagen in Chemnitz angeboten. Weitere Veranstaltungen, bei denen Sie Papierprüfungen ablegen können, finden Sie auf dieser Website:

http://lpievent.lpice.eu

Hier wird Ihnen auch gleich die Möglichkeit geboten, sich zu einer Prüfung anzumelden.

Punktevergabe

Die Punktevergabe bei den Prüfungen sieht im Moment folgendermaßen aus:

Zum Bestehen einer beliebigen LPIC-Prüfung sind 500 Punkte erforderlich. Sie bekommen 60 Fragen vorgelegt und können zwischen 200 und 800 Punkte erreichen. Da die Fragen nicht gleich gewertet werden, können Sie aus der erreichten Punktzahl nicht genau auf die Anzahl der richtig beantworteten Fragen schließen.

Die Prüfungen können jeweils Betafragen enthalten, die Ihre Punktzahl nicht beeinträchtigen. Da diese Fragen nicht gesondert markiert sind, müssen Sie sie ebenfalls beantworten. Auch wegen der eingestreuten Betafragen ist eine genaue Berechnung der erreichten Prozentpunkte nicht möglich.

Über die Inhalte der Prüfung müssen sie übrigens Stillschweigen bewahren. Es gibt eine Geheimhaltungserklärung, die Sie vor der Prüfung akzeptieren müssen.

Sprachen

In Deutschland steht die LPIC-2-Prüfung in den Sprachen Deutsch und Englisch zur Verfügung. Viele Prüflinge legen die Prüfung in englischer Sprache ab, um eventuellen Übersetzungsfehlern aus dem Weg zu gehen. Das ist bei einigen Prüfungen, die ich aus eigener Erfahrung kenne, auch absolut angebracht. Die Fragen in den LPIC-Prüfungen sind allerdings im Verhältnis zu den gängigen Herstellerprüfungen kurz gefasst und bieten deshalb wenig Stoff für Übersetzungsfehler. Wenn Ihr Englisch nicht erstklassig ist, sollten Sie die Prüfungen lieber in Ihrer Muttersprache ablegen.

Danksagung

Das Buch ist eigentlich längst fertig, aber ich hänge hier in einer Zeitschleife, um diese wenigen Zeilen der Danksagung zu schreiben. Am Ende kommt wahrscheinlich noch nicht einmal etwas besonders Geistreiches dabei heraus, aber die wenigen, die hier genannt werden, können sich sicher sein, dass ich tagelang auf ihre Namen gestarrt und überlegt habe, was ich ihnen sagen möchte.

Das Jahr 2016 war aus verschiedenen Gründen etwas anstrengend für mich und ich möchte die Gelegenheit ergreifen, mich bei den Menschen zu bedanken, die mich mit Rat und Tat unterstützt haben. Einige werden sich vielleicht wundern, auf dieser kurzen Liste zu stehen, aber es wird schon einen Grund geben:

Aida Rosenthal, Antoinette Hütten, Carolin Oberlaner, Carsten Stier,
Claudia Zentgraf, Dimitrios »Taki« Bogiatzoules, Dominik Maaßen,

Dominik Pauls, Fabian Thorns, Barbara und Gerd Krause,

Ines Schmiede, Kai Schell, Lisa Haßdenteufel, Markus Häbe,

Markus Papadopoulos, Martin Selle, Michael Kästner,

Michael Marquardt, Reiner Brandt, Torsten König und Willi Pauls

Mein besonderer Dank gilt Anne Scheibe. Sie ist die Lektorin dieses und etlicher anderer Fachbücher. Ich habe noch nie mit einer anderen Lektorin zusammengearbeitet, will mir das aber auch nicht vorstellen.

Last but not least ist da noch Claudia Lochbaum. Sie hat bei diesem Buch das Korrektorat durchgeführt. Ohne sie hätte der Feinschliff des Buches mit Sicherheit weniger Spaß gemacht und ich hoffe, dass sie bei zukünftigen Auflagen wieder dabei ist.

Und jetzt wünsche ich Ihnen viel Spaß mit interessanten Themen rund um Linux und viel Erfolg beim Bestehen der Prüfungen!

Harald Maaßen

LPI 201

200 Kapazitätsplanung

Proaktive Netzwerkadministration setzt voraus, dass Sie vorhersehen können, wohin sich der Bedarf Ihrer IT- Systeme entwickelt. Sollten Sie keine Kristallkugel besitzen, können Sie auf die in diesem Kapitel besprochenen Tools zurückgreifen.

200.1 Messen und Problembehandlung bei der Ressourcenverwendung

Wichtung: 6

Beschreibung: Kandidaten sollten dazu in der Lage sein, verwendete Hardwareressourcen und Netzwerkbandbreiten zu messen. Sie sollten auch Ressourcenengpässe identifizieren und beseitigen können.

Wichtigste Wissensgebiete:

- Erfassen der CPU-Verwendung
- Erfassen der Speicherauslastung
- Erfassen von Disk I/O
- Erfassen von Netzwerk I/O
- Erfassen von Firewall- und Routing-Durchsatz
- Abbilden der Client-Bandbreitenverwendung
- Vergleichen und Korrelieren von Systemsymptomen mit ähnlichen Problemen
- Abschätzen des Durchsatzes und Identifizieren von Engpässen in einem System, inklusive Netzwerk

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `iostat, iotop`
- `vmstat`
- `netstat, ss`
- `iptraf`
- `pstree, ps`
- `w`
- `lsof`
- `top, htop`
- `uptime`
- `sar`
- `swap`
- Prozesse, die durch I/Os blockiert sind
- empfangene und gesendete Blocks

Allgemeines

Wenn Sie Leistungsengpässe auf einem System diagnostizieren müssen, stehen Ihnen verschiedene Werkzeuge zur Verfügung. Mit einigen dieser Tools haben Sie sich schon im Zusammenhang mit dem LPIC-Level 1 auseinandergesetzt. Andere sind Ihnen vielleicht noch neu.

Im Falle eines Leistungseinbruchs muss zunächst festgestellt werden, welche Komponente eines Systems überlastet ist. Grundsätzlich kommen hierbei alle Elemente infrage. Deshalb sollten zumindest folgende Zustände geprüft werden:

- Prozessorauslastung
- Speicherauslastung (inklusive Swap)
- Datenträgerzugriffe (I/O) und freie Kapazitäten
- verwendete Netzwerkbandbreite und gleichzeitige Client-Zugriffe
- Anzahl und Zustand von Prozessen

Werkzeuge zur Diagnose von bestehenden Engpässen

sar, iostat

Um Ihren Werkzeugkasten zu füllen, sollten Sie zunächst das Paket *sysstat* (mittels `apt-get` oder `yum`) installieren. Die Programme `iostat` und `sar` stehen Ihnen sonst nicht zur Verfügung. Das Paket *sysstat* enthält übrigens mehr als nur die hier beschriebenen Tools. Es soll sich aber hier wieder nur auf prüfungsrelevante Funktionen beschränkt werden.

Um Informationen über ein System zu erhalten, lesen alle Tools des Pakets *sysstat* im Pseudoverzeichnis */proc*. Da hierbei keine Schreibzugriffe stattfinden, können alle diese Programme im Sicherheitskontext eines normalen Benutzers ausgeführt werden.

Ein Beispiel für die Verwendung von `sar` ist die Anzeige der Prozessoraktivität. Hier wurden fünf Messungen in einem zeitlichen Abstand von drei Sekunden durchgeführt:

```
harald@archangel:~$ sar 3 5
Linux 3.9.4 (archangel)         22.06.2013   _i686_  (4 CPU)
15:08:42      CPU  %user  %nice %system   %iowait    %idle
15:08:45      all   6,19   0,00    2,18      8,12    83,51
15:08:48      all  16,74   0,00    3,51     16,32    63,43
15:08:51      all  30,13   0,00    4,81      0,00    65,06
15:08:54      all  23,95   0,00    3,46      1,43    71,16
15:08:57      all   5,44   0,00    0,84      1,42    92,30
Durchschnitt: all  16,45   0,00    2,96      5,47    75,12
```

Die Ausgabe des Kommandos zeigt, dass von einem Benutzer Prozessorlast erzeugt wird. Die angezeigten Werte unter `%iowait` sind ein Hinweis darauf, dass hier Prozesse auf I/Os, also auf die Ein- bzw. Ausgabe eines Datenträgers gewartet haben. Sollten sich Ihnen die angezeigten Parameter nicht intuitiv erschließen, lesen Sie bitte die (ohnehin hochinteressante) Manpage von `sar`.

Sie können `sar` auch verwenden, um die Aktivität von Netzwerkschnittstellen zu untersuchen:

```
harald@archangel:~$ sar -n DEV 5 1
Linux 3.9.4 (archangel)   22.06.2013   _i686_  (4 CPU)

15:29:06       IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s
15:29:11        sit0      0,00      0,00      0,00      0,00
15:29:11          lo      0,00      0,00      0,00      0,00
15:29:11        eth0   1852,60   3466,40    143,82   4132,90

Durchschnitt:  IFACE   rxpck/s   txpck/s    rxkB/s    txkB/s
Durchschnitt:   sit0      0,00      0,00      0,00      0,00
Durchschnitt:     lo      0,00      0,00      0,00      0,00
Durchschnitt:   eth0   1852,60   3466,40    143,83   4132,90
```

Der Übersichtlichkeit halber wurde nur eine einzige Messung durchgeführt, und es wurden auch einige Spalten aus der Tabelle entfernt. Das Ergebnis zeigt, dass es Aktivitäten auf der Schnittstelle eth0 gegeben hat. Es wurden pro Sekunde 1852,60 Pakete empfangen und 3466,40 Pakete gesendet. Die empfangene Datenmenge betrug 143,82 kB pro Sekunde und die gesendete Datenmenge 4132,90 kB pro Sekunde.

Wenn Sie `sar` mit der Option `-b` verwenden, können Sie I/O-Transferraten messen:

```
harald@archangel:~$ sar -b 10 3
Linux 3.9.4 (archangel)       22.06.2013      _i686_  (4 CPU)

16:14:31          tps      rtps      wtps   bread/s    bwrtn/s
16:14:41       210,30    178,30     32,00   4478,40    2528,80
16:14:51       357,10    251,60    105,50  26902,40   34640,80
16:15:01       388,10    284,50    103,60  59079,20   59119,20
Durchschnitt:  318,50    238,13     80,37  30153,33   32096,27
```

Drei Abtastungen mit Abständen von je zehn Sekunden ergaben hier im Durchschnitt 318,5 Transfers pro Sekunde zu den physikalischen Geräten. Davon waren 238,13 Lese- und 80,37 Schreibzugriffe. Im Durchschnitt wurden pro Sekunde 30.153,33 Blöcke gelesen und 32.096,27 Blöcke geschrieben. Ein Block hat hierbei eine Größe von 512 Bytes.

Wenn Sie Leistungsdaten über einen längeren Zeitraum erheben wollen, können Sie *sysstat* als Daemon im Hintergrund laufen lassen. Bearbeiten Sie hierfür die Datei */etc/default/sysstat*. Setzen Sie `ENABLED="true"`, und starten Sie danach *sysstat* mit:

```
root@archangel:~# /etc/init.d/sysstat start
```

Sie können sich die gesammelten Daten nach frühestens zehn Minuten mit diesem Kommando anzeigen lassen:

```
root@archangel:~# sar -A
```

Ein anderes interessantes Diagnosewerkzeug aus dem Paket *sysstat* ist `iostat`. Sie können mit diesem Programm Statistiken bezüglich CPU und I/Os von Geräten und Partitionen generieren. Wenn Sie das Programm ohne Optionen starten, sind die Informationen über die CPU-Nutzung mit denen von `sar` vergleichbar.

```
root@archangel:~# iostat
Linux 3.9.4 (archangel)         22.06.2013      _i686_  (4 CPU)

avg-cpu:  %user   %nice %system %iowait  %steal   %idle
           2,43    0,01    0,17    0,27    0,00   97,12

Device:            tps    kB_read/s    kB_wrtn/s    kB_read    kB_wrtn
sda               3,62        14,76        30,31   31639060   64973608
sdb               0,21         0,84         8,62    1801374   18485424
sdc               0,01         0,40         0,00     862053        263
```

Wie Sie sehen, sind die hier angezeigten statistischen Informationen über I/Os auf Datenträger bezogen und nicht auf Partitionen.

Einige wichtige Schalter für `iostat` sind:

- `-c` zeigt nur die CPU-Verwendung.
- `-d` zeigt nur die Verwendung von Geräten (Datenträgern).
- `-h` leichtere Lesbarkeit (human readable).
- `-x` zeigt erweiterte Statistiken.

Im Übrigen können Sie `iostat` auch mehrere Messwerte in einem festgelegten Intervall ausgeben lassen. Die Syntax ist mit der Syntax von `sar` identisch, also etwa:

```
root@archangel:~# iostat 3 5
```

iotop

`iotop` muss zunächst installiert werden. Es handelt sich hierbei um ein eigenständiges Paket mit eben diesem Namen. Die Optik erinnert an das Programm `top`, mit dem Unterschied, dass `iotop` zum Monitoring von *I/Os* gedacht ist, während `top` vorwie-

gend CPU und Speicherressourcen überwacht. `iotop` ist also geeignet, Prozesse zu überwachen, die gerade auf Datenträger zugreifen. Die Ausgabe von `iotop` ist nicht sonderlich buchtauglich, aber Sie sollten das Programm ohnehin unbedingt selbst testen. Es gibt (genau wie bei `top`) einen interaktiven Modus und einen Stapelverarbeitungsmodus. Nützliche Optionen für `iotop` sind:

- `-o`, `--only` zeigt nur die Prozesse an, die gerade aktiv sind.
- `-b`, `--batch` aktiviert den nicht-interaktiven Modus, der gut für Aufzeichnungszwecke geeignet ist.
- `-n NUM`, `--iter=NUM` legt die Anzahl der Wiederholungen fest.
- `-d SEC`, `--delay=SEC` legt den Abstand zwischen Wiederholungen fest.
- `-u USER`, `--user=USER` zeigt nur Prozesse dieser User bzw. dieses Users an.

vmstat

Hier handelt es sich auch um ein vielseitiges Werkzeug zum Aufspüren von Engpässen, auch wenn der Name lediglich auf ein Statistiktool für virtuellen Speicher schließen lässt. Sie können mit `vmstat` folgende Komponenten analysieren:

- Prozesse
- Speicher (RAM und Swap)
- I/O (gesendete und empfangene Blöcke)
- System (z. B. Interrupts pro Sekunde und Kontextwechsel)
- CPU

Um die I/Os einer Partition zu untersuchen, können Sie z. B. folgendes Kommando verwenden:

```
root@archangel:~# vmstat -p /dev/sda1
sda1    reads   read sectors    writes  requested writes
      1742197       59828706   5794204         128000840
```

Die Ausgabe von `vmstat` ist bei der Verwendung der meisten anderen Schalter sehr umfangreich (insbesondere in der Breite), weshalb hier auf seitenfüllende Beispiele verzichtet werden soll. Am besten sehen Sie das, wenn Sie zwei Terminals öffnen (eins für die Manpage und das andere zum Testen der Kommandos) und sich einen Überblick über die verwendeten Ressourcen Ihres eigenen Computers verschaffen. Testen Sie bitte selbstständig die folgenden Kommandos:

- `root@archangel:~# vmstat -a`

  zeigt Informationen sowohl für aktiven als auch inaktiven Speicher
- `root@archangel:~# vmstat  2 5`

  zeigt in der ersten Zeile Durchschnittswerte seit dem letzten Systemstart und anschließend fünf Messwerte im Abstand von zwei Sekunden an

- `root@archangel:~# vmstat -d`
  gibt Statistiken zu Festplattenzugriffen (Disk) aus
- `root@archangel:~# vmstat -D`
  zeigt eine Zusammenfassung der Festplattenstatistiken (Disk)
- `root@archangel:~# vmstat -s`
  zeigt eine Tabelle mit unterschiedlichen Ereigniszählern und Speicherstatistiken
- `root@archangel:~# vmstat -f`
  gibt die Anzahl der Forks seit dem letzten Neustart aus

netstat

Bei `netstat` handelt es sich um ein Diagnoseprogramm, mit dem Sie sich u. a. Netzwerkverbindungen (Sockets) und Routing-Tabellen anzeigen lassen können. Im Zusammenhang mit Netzwerkengpässen betrachtet, können Sie `netstat` dazu verwenden, die Anzahl gleichzeitiger Verbindungen zu einem Computer oder zu einem bestimmten Dienst zu ermitteln. Im folgenden Beispiel wird `grep` verwendet, um die Ausgabe des Kommandos auf bestehende TCP-Verbindungen zu beschränken:

```
root@archangel:~# netstat -an |grep VERBUNDEN|grep tcp
tcp   0   0 192.168.50.1:48074    88.221.216.97:80        VERBUNDEN
tcp   0   0 192.168.50.1:5900     192.168.50.175:61911    VERBUNDEN
tcp   0   0 192.168.50.1:38577    88.221.216.40:80        VERBUNDEN
tcp6  0   0 192.168.50.1:11111    192.168.50.101:49390    VERBUNDEN
tcp6  0  52 2a01:198:5dd::1:22    2a01:198:5dd:0:d4:61910 VERBUNDEN
tcp6  0  16 192.168.50.1:11111    79.247.187.164:61013    VERBUNDEN
tcp6  0   0 2a01:198:5dd::1:445   2a01:198:5dd:0:d4:53720 VERBUNDEN
tcp6  0   0 192.168.50.1:80       192.168.50.175:61936    VERBUNDEN
tcp6  0   0 2a01:198:5dd::1:445   2a01:198:5dd:0:f4:53158 VERBUNDEN
```

Auf diesem Server ist offensichtlich nicht besonders viel los. Wenn Sie die Anzahl der Zugriffe auf einen bestimmten Dienst ermitteln müssen, bietet es sich an, einfach auf den eingehenden Port zu filtern. Bei einem Webserver könnte das etwa so aussehen:

```
root@archangel:~# netstat -an |grep VERBUNDEN|grep :80
```

In Topic 205 wird `netstat` noch einmal unter einem anderen Blickwinkel betrachtet. Sie finden dort auch einige für die Prüfung wichtige Optionen.

ss

Ein neueres Programm zur Abfrage von Informationen zu Sockets ist `ss`. Dieses Tool stellt seine Anfragen direkt an den Kernel und antwortet deshalb schneller als `netstat`.

Es gibt eine große Schnittmenge zwischen `netstat` und `ss`, was die verwendbaren Optionen anbelangt. Da beide Programme in der Praxis regelmäßige Verwendung finden, sollten Sie ein wenig damit experimentieren, falls nicht schon geschehen. Aufgrund der Ähnlichkeit der beiden Werkzeuge zueinander soll hier auf ein Beispiel verzichtet werden.

iptraf

Bei `iptraf` handelt es sich um einen interaktiv bedienbaren LAN-Monitor, der auf *ncurses* basiert. Die Bedienung ist dementsprechend, nach Programmaufruf ohne Optionen, menügeführt.

Sie haben mit diesem Programm die Möglichkeit, sich unterschiedlich detaillierte Statistiken bezüglich der Netzwerkschnittstellen anzusehen oder auch den aktuellen Netzwerkverkehr live zu beobachten. Mit ein wenig Übung und Erfahrung sind Sie mithilfe von `iptraf` schnell in der Lage, z. B. die Netzwerkauslastung eines hostbasierten Routers zu analysieren und ggf. die Quelle übermäßiger Netzwerklasten zu identifizieren. Es ist wenig sinnvoll, die Menüführung von `iptraf` in diesem Buch darzustellen. Die Bedienung ist aber selbsterklärend. Beachten Sie bitte, dass das Programm, wenn Sie es mit Optionen starten, nicht interaktiv bedienbar ist. Was das betrifft, sollten Sie die Manpage des Tools mit der entsprechenden Menüführung vergleichen.

ps

Mit dem Kommando `ps` können Sie die aktuell auf einem Computer laufenden Prozesse anzeigen. Es handelt sich hierbei allerdings nur um eine Momentaufnahme. Programme, die über eine kurze Lebensdauer verfügen (z. B. `df`), können Sie mit `ps` nicht anzeigen. Wenn Sie das Kommando `ps` ohne Optionen ausführen, erhalten Sie lediglich eine Liste der Prozesse, die auf dem aktuellen Terminal laufen. Das sind in der Regel nur die Shell und das `ps`-Kommando selbst. Wenn Jobs im Hintergrund laufen (oder stehen), werden diese auch angezeigt. Sie können bei der Übergabe von Optionen einen Strich voranstellen oder diesen weglassen. In Abhängigkeit von der verwendeten Programmversion werden Sie zu unterschiedlichen Ergebnissen kommen. Nach der UNIX-Syntax wird der Option ein Strich vorangestellt, während nach der BSD-Syntax kein Strich verwendet wird. GNU-Optionen sind wiederum die ausgeschriebenen langen Optionen. Diesen werden zwei Striche vorangestellt. Am besten lesen Sie im Zweifelsfall immer in der Manpage der gerade verwendeten Version von `ps` nach. In der Prüfung wird normalerweise die GNU-Syntax verwendet.

Optionen können, wie bei den meisten Programmen, nach Belieben miteinander kombiniert werden. Folgende Optionen sind sowohl für die Prüfung als auch für die Praxis wichtig:

- `-a` zeigt auch die Prozesse anderer Benutzer an. Voraussetzung ist, dass diese Prozesse mit einem Terminal verknüpft sind.
- `-u` zeigt auch die Startzeit, den Pfad zur ausführbaren Datei und den ausführenden Benutzer an.
- `-x` führt auch die Prozesse auf, die nicht mit einem Terminal verbunden sind (z. B. init oder cron).
- `-C prozess` sorgt für die Ausgabe aller Instanzen eines auf der Kommandozeile angegebenen Prozesses.
- `-U benutzer` zeigt die Prozesse eines bestimmten Benutzers an.

Es folgen einige typische Beispiele, die ähnlich auch in der Prüfung Verwendung finden können.

Um alle Instanzen des nfsd (NFS-Daemon), die auf einem System laufen, anzuzeigen, geben Sie folgendes Kommando ein:

```
root@ubuntu-server:~# ps -C nfsd
  PID TTY          TIME CMD
 9592 ?        00:00:00 nfsd
 9593 ?        00:00:00 nfsd
 9594 ?        00:00:00 nfsd
... weitere Zeilen wurden abgeschnitten...
```

Eine sehr beliebte Kombination ist `-aux`. Damit werden alle Prozesse sämtlicher Benutzer angezeigt. Zusätzlich werden Prozesse gelistet, die nicht mit einem Terminal verbunden sind. Die Ausgabe enthält umfangreiche Informationen zum Prozess:

```
root@ubuntu-server:~# ps aux
USER  PID %CPU %MEM   VSZ   RSS  TTY  STAT  START  TIME  COMMAND
root    1  0.0  0.3   2912  1848    ?    Ss  Oct16  0:04  /sbin/init
root    2  0.0  0.0      0     0    ?     S  Oct16  0:00  [migration/0]
root    3  0.0  0.0      0     0    ?    SN  Oct16  0:00  [ksoftirqd/0]
root    4  0.0  0.0      0     0    ?     S  Oct16  0:00  [watchdog/0]
... weitere Zeilen wurden abgeschnitten...
```

Wenn Sie in einer langen Prozessliste schnell nach einem laufenden Prozess suchen müssen, empfiehlt es sich, die Ausgabe des Kommandos an `grep` weiterzuleiten. Beispiel:

```
root@ubuntu-server:~# ps -A | grep apache
  398 ?        00:00:00 apache2
  404 ?        00:00:00 apache2
... weitere Zeilen wurden abgeschnitten...
```

pstree

Das Kommando `pstree` zeigt ebenfalls die aktuell auf einem Computer laufenden Prozesse an. Im Gegensatz zu `ps` zeigt `pstree` aber auch die Hierarchie der Prozesse in einem Baumdiagramm aus ASCII-Zeichen an. Ein Ausschnitt der Ausgabe des Kommandos sieht folgendermaßen aus:

```
root@ubuntu-server:~# pstree -pn
... weitere Zeilen wurden abgeschnitten ...
        ├─apache2(16940)─┬─apache2(16944)
        │                ├─apache2(16945)
        │                ├─apache2(16946)
        │                ├─apache2(16947)
        │                ├─apache2(24428)
        │                ├─apache2(24429)
        │                └─apache2(26313)
        ├─smbd(17676)─┬─smbd(17678)
        │             ├─smbd(19715)
        │             └─smbd(31266)
... weitere Zeilen wurden abgeschnitten...
```

Folgende Optionen sollten Sie kennen:

- `-a` zeigt zusätzlich die Optionen und Argumente an, die einem Prozess an der Kommandozeile übergeben wurden.
- `-G` hat eine Ausgabe im VT100-Modus zur Folge. Das führt bei den meisten Terminals zu einer optisch ansprechenderen Ausgabe.
- `-p` zeigt zusätzlich die PIDs (Prozess-IDs) an.
- `-n` sorgt für eine Sortierung nach PIDs. Normalerweise gibt `pstree` die Prozesse in alphabetischer Reihenfolge aus.

top

Mit `top` können Sie die auf einem Computer laufenden Prozesse in Echtzeit überwachen. Während der Ausführung werden die folgenden Informationen dynamisch im Terminal ausgegeben:

- allgemeine Informationen: Uhrzeit, Uptime, Anzahl der angemeldeten Benutzer und durchschnittliche Systemauslastung
- Tasks: Anzahl der Prozesse insgesamt, laufende Prozesse, schlafende Prozesse, gestoppte Prozesse und Zombieprozesse
- CPU(s): Benutzung durch Benutzerprozesse, Systemprozesse, den Idle-Prozess u. a.

- Mem: Speicher insgesamt, Speicher in Verwendung, freier Speicher und für Buffers verwendeter Speicher
- Swap: diverse Werte zur Verwendung der Swap-Partition(en)

Anschließend folgt eine Auflistung der laufenden Prozesse. Diese werden standardmäßig in der Reihenfolge der CPU-Verwendung ausgegeben. Die inaktiven Prozesse werden anschließend in der Reihenfolge der PIDs ausgegeben. Beachten Sie bitte, dass das Programm `top` selbst auch Systemressourcen belegt und deshalb das Messergebnis beeinflusst. Dies demonstriert auch das folgende Beispiel:

```
PID USER   PR  NI  VIRT  RES  SHR  S %CPU %MEM     TIME+  COMMAND
223 root   15   0  2056 1016  752  R  1.5  0.0   0:00.03  top
 1 root    16   0   680  104   72  S  0.0  0.0   0:03.71  init
 2 root    RT   0     0    0    0  S  0.0  0.0   0:04.70  migration/0
 3 root    34  19     0    0    0  S  0.0  0.0  11:55.58  ksoftirqd/0
 4 root    RT   0     0    0    0  S  0.0  0.0   0:16.45  migration/1
 5 root    34  19     0    0    0  S  0.0  0.0   3:09.48  ksoftirqd/1
 6 root    10  -5     0    0    0  S  0.0  0.0   0:04.55  events/0
```

Sie können `top` sowohl beim Aufruf als auch zur Laufzeit Optionen übergeben. Für die Prüfung müssen Sie die gängigsten Optionen für beide Situationen kennen.

Wichtige Interaktivoptionen:

- `k` (kill) tötet einen Prozess. Es muss sowohl die PID als auch das Signal angegeben werden.
- `n` (number of) Anzahl der Zeilen, die `top` ausgibt.
- `r` (renice) ändert den Nice-Wert eines Prozesses zur Laufzeit.
- `h` (help) gibt eine Hilfe aus.
- `q` (quit) beendet das Programm.

htop

Mit dem neueren Programm `htop` können Sie im Prinzip dieselben Aufgaben durchführen wie mit `top`. Es sind allerdings einige zum Teil nützliche Funktionen hinzugekommen:

- vertikales und horizontales Scrollen
- farbige Hervorhebungen
- Mausunterstützung
- Bei Interaktivoptionen können Prozesse direkt angewählt werden, anstatt die PID anzugeben.
- `htop` startet und reagiert schneller.

uptime und w

Mithilfe der Kommandos `uptime` und `w` können Sie sich einen schnellen Überblick darüber verschaffen, wie stark ein System innerhalb der letzten Minuten ausgelastet war und wer diese Auslastung verursacht hat. Hier ist ein Beispiel für die Verwendung des Befehls `w`:

```
[root@centos01 ~]# w
11:07:44 up 284 days,19:05,2 user,load average: 0.70, 0.11, 0.07
USER     TTY      FROM             LOGIN@   IDLE    JCPU   PCPU   WHAT
harald   tty1     -                28Mar09  20days  0.02s  0.02s  -bash
root     pts/0    192.168.50.200   10:34    0.00s   0.38s  0.00s  w
```

Als Erstes wird die aktuelle Systemzeit (11:07:44) ausgegeben. Das System läuft demnach bereits seit 284 Tagen, 19 Stunden und 5 Minuten. Es sind aktuell zwei Benutzer verbunden. Die folgenden drei Werte zeigen die durchschnittliche Systemlast der letzten 1, 5 und 15 Minuten an.

Es folgt eine Auflistung der am System angemeldeten Benutzer. Für jeden einzelnen Benutzer wird angezeigt, mit welchem Terminal er verbunden ist (`TTY`), von wo aus er eingeloggt ist (`FROM`) und wann er sich angemeldet hat (`LOGIN@`). Außerdem werden die Leerlaufzeit (`IDLE`), die seit der Anmeldung benötigte CPU-Zeit (`JCPU`) und die von aktuell noch laufenden Prozessen verwendete CPU-Zeit (`PCPU`) ausgegeben. In der letzten Spalte (`WHAT`) wird angezeigt, was der Benutzer gerade macht.

Der Befehl `uptime` zeigt lediglich die erste Zeile des Kommandos `w`:

```
[root@centos01 ~]# uptime
11:08:47 up 284 days,19:06,2 user,load average: 0.37, 0.11, 0.03
```

Man kann `uptime` gut verwenden, um Windows-Benutzern zu zeigen, wie selten man sein System neu starten muss.

lsof

Das Programm `lsof` zeigt kurz gesagt an, welche Dateien und Sockets auf einem System geöffnet sind und durch welches Programm. Wenn sich ein Dateisystem nicht aushängen lässt, weil darauf noch Dateien geöffnet sind, können Sie mithilfe von `lsof` schnell feststellen, wer oder was den `umount`-Vorgang blockiert. Um zu überprüfen, wer oder was das Verzeichnis */storage* verwendet, können Sie einfach so vorgehen:

```
root@archangel:~# lsof /storage
COMMAND  PID  USER   FD    TYPE   DEVICE  SIZE/OFF   NODE NAME
smbd     5875 root   39r   DIR    8,2     4096       2 /storage
smbd     5875 root   41r   DIR    8,2     4096       3145729 /storage/MP3
smbd     7787 root   cwd   DIR    8,2     4096       2 /storage
```

Sie sehen in der ersten Spalte, dass jemand über Samba auf das Verzeichnis */storage* zugreift. Sie können jetzt als Nächstes das Programm `smbstatus` verwenden und aufgrund der verwendeten PIDs (5875 und 7787) genau sehen, welcher Benutzer auf das Verzeichnis zugreift. Sie finden darüber genauere Informationen unter Topic 209.

Wenn Sie `lsof` ohne Optionen und Parameter verwenden, werden alle Dateien und Sockets angezeigt, die gerade benutzt werden. Das sind normalerweise mehrere Zehntausend, weshalb Sie hier immer filtern sollten.

200.2 Prognostizieren zukünftiger Ressourcenanforderungen

Wichtung: 2

Beschreibung: Die Kandidaten sollten dazu in der Lage sein, die Ressourcenbenutzung aufzuzeichnen, um spätere Ressourcenanforderungen vorhersagen zu können.

Wichtigste Wissensgebiete:

- Verwenden von Monitoring- und Messwerkzeugen zum Aufzeichnen der Infrastrukturverwendung
- Vorhersagen der Grenzen von Systemkapazitäten
- Beobachten des Wachstums und Bewerten der Kapazitätsauslastung
- Darstellen des Trends der Kapazitätsauslastung
- Kenntnis von Monitoring-Lösungen wie Icinga2, Nagios, collectd, MRTG und Cacti

Liste wichtiger Dateien, Termini und Anwendungen:

- Diagnose
- Vorhersehen des Wachstums
- Ressourcenerschöpfung

Allgemeines

Die Anforderungen an EDV-Systeme nehmen immer schneller zu. Das gilt für nahezu alle Parameter eines Systems. Hatte man 1995 noch Festplatten mit einer Kapazität von 1 GB, so sind wir heutzutage im Terrabytebereich angelangt. Wo man 1995 noch mit einem 16-MB-Arbeitsspeicher auskam, benötigt man jetzt 16 GB. Ähnlich sieht es aus, wenn man die Leistungsfähigkeit von Prozessoren und Grafikarten oder den Datendurchsatz von Netzwerkkomponenten betrachtet.

Wenn Sie über mehrere Jahre hinweg in einem Unternehmen arbeiten, werden Sie diese Trends selbst zu spüren bekommen. Typische Auswirkungen sind:

- Netzwerkbandbreiten reichen nicht mehr aus, um aktuelle Anwendungen zu handhaben.
- Serverfestplatten laufen voll.
- Backups dauern entsprechend lange.
- Benutzer beschweren sich über zu langsame Systeme.

Um solche Probleme zu vermeiden, müssen Sie die entsprechenden Parameter im Auge behalten und Trends vorhersehen. Erfreulicherweise gibt es hierzu hilfreiche Software.

Werkzeuge zur Aufzeichnung der Ressourcenverwendung

collectd

Sie haben auf einer der vorangehenden Seiten schon gesehen, dass man mittels *sysstat* zumindest in kleinem Rahmen Leistungsdaten eines Computers aufzeichnen kann. Es gibt aber auch Programmpakete, die auf solche Aufgaben spezialisiert sind. Als Erstes wäre hier *collectd* zu nennen. Man kann dem Namen schon entnehmen, dass es sich hierbei um einen Daemon handelt, der Informationen über Leistungsdaten eines Computers sammelt. Es gibt viele Plug-ins für *collectd*, die allerdings teilweise eine separate Konfiguration benötigen. Die Installation läuft unter Debian – wie Sie wahrscheinlich schon richtig vermutet haben – mit:

```
root@archangel:~# apt-get install collectd
```

Es wird dann je nach Debian-Version automatisch das Paket *rrdtool* mitinstalliert. Dieses Tool kann später zur Auswertung der aufgezeichneten Daten verwendet werden. Passen Sie *collectd* zunächst an Ihre eigenen Bedürfnisse an, indem Sie die Datei */etc/collectd/collectd.conf* bearbeiten. Sie können Plug-ins aktivieren, indem Sie jeweils das Kommentarzeichen am Anfang der Zeile entfernen. Starten Sie den Daemon anschließend neu:

```
root@archangel:~# /etc/init.d/collectd restart
```

Das Datensammlungsverzeichnis von *collectd* liegt, solange es nicht geändert worden ist, unterhalb von */var/lib/collectd*. Sie werden an den hier abgelegten Daten allerdings, so wie sie sind, wenig Freude haben, denn sie sind nicht menschenlesbar. Die Auswertung und grafische Aufbereitung erledigen *rrdtool* und Frontends wie z. B. *Cacti* oder *MRTG*.

Cacti

Cacti ist ein webbasiertes Werkzeug zur Auswertung von Leistungsdaten. Damit Sie es verwenden können benötigen Sie einen Webserver mit PHP und MySQL als

Grundlage. Die Installation geschieht dann unter Debian und seinen Derivaten wie gehabt mit:

```
root@archangel:~# apt-get install cacti
```

Bei Red Hat und seinen Derivaten führt `yum install cacti` zum gleichen Ziel. Nach Abschluss der Installation können Sie die Grundkonfiguration mithilfe eines Webbrowsers durchführen. Verwenden Sie hierbei einfach die lokale Adresse:

http://localhost/cacti

Die Konfiguration ist vollständig menügeführt, soll hier aber aus Platzgründen und wegen der geringen Bedeutung für die Prüfung nicht vorgeführt werden.

Der Standard-User ist *admin*, und Sie müssen bei der ersten Verwendung ein Passwort vergeben. Die Bedienung des Programms selbst ist zwar komplex, aber intuitiv erlernbar.

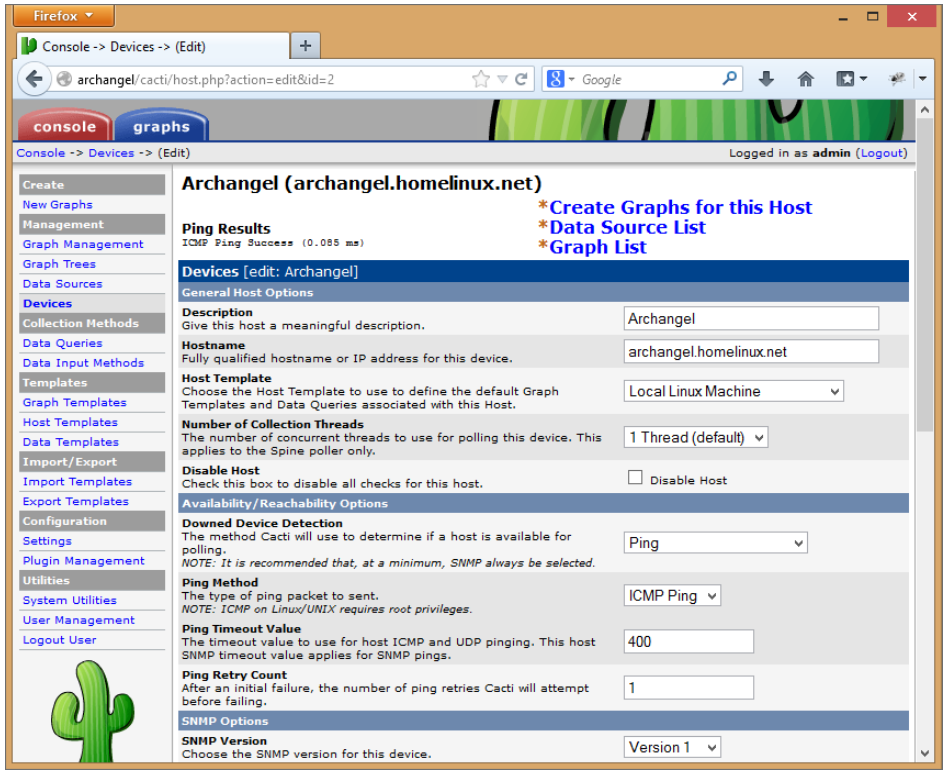

Abbildung 200.1 Cacti nach Abschluss der Konfiguration in Firefox

MRTG

MRTG ist ein Akronym für Multi Router Traffic Grapher. Wie der Name schon vermuten lässt, wurde das Programm ursprünglich zur Darstellung von Netzwerkdatenver-

kehr entwickelt. Inzwischen wird es hauptsächlich verwendet, um bestehende Messwerte auszuwerten und grafisch darzustellen. Sie können es z. B. einsetzen, um die von *collectd* gesammelten Daten anzuzeigen. Genau wie *Cacti* benötigt *MRTG* einen Webserver mit *PHP* und *MySQL* als Basis. Außerdem muss der *snmpd*-Daemon installiert sein. Die Installation und Grundkonfiguration funktioniert wie in folgendem Beispiel:

```
root@archangel:~# apt-get install snmpd
root@archangel:~# apt-get install mrtg
```

Nach der Installation erstellen Sie eine neue Konfigurationsdatei. Für diesen Zweck gibt es eigens ein Konfigurationswerkzeug:

```
root@archangel:~# cfgmaker public@192.168.50.1 > /etc/mrtg.cfg
```

Hierbei ist *public* der Name der SNMP-Domäne, gefolgt von der IP-Adresse des zu verwaltenden Systems. Erstellen Sie anschließend ein Webverzeichnis für *MRTG*:

```
root@archangel:~# mkdir /var/www/mrtg
```

Die Datei *index.html* für den Browserzugriff wird ebenfalls von einem eigens dafür vorgesehenen Perlscript generiert:

```
root@archangel:~# indexmaker /etc/mrtg.cfg > /var/www/mrtg/index.html
```

Jetzt können Sie *MRTG* ausführen. Davon ausgehend, dass Ihre Umgebungsvariable *LANG* mit *UTF-8* konfiguriert ist, verwenden Sie folgendes Kommando:

```
root@archangel:~# env LANG=C /usr/bin/mrtg /etc/mrtg.cfg
```

MRTG funktioniert nicht mit UTF-8. Der Zugriff auf MRTG erfolgt dann über die lokale URL:

http://localhost/mrtg

Nagios

Nagios ist eine umfangreiche Monitoring-Lösung, mit der ganze Unternehmen überwacht werden können. Im Falle eines Problems kann Nagios verschiedene Kommunikationskanäle verwenden, um eine Person zu alarmieren. Hierzu können SMS, E-Mail, Telefonanrufe oder Instant Messenger verwendet werden. *Nagios* verwendet ähnlich wie *Cacti* ein browserbasiertes Frontend, setzt aber nicht auf *collectd* auf. Entfernte Systeme werden u. a. mittels *SNMP* (*Simple Network Management Protocol*) abgefragt. Auch für *Nagios* ist ein Webserver mit *MySQL* und *PHP* als Grundlage nötig. Die Installation erfolgt aktuell mit dem Kommando:

```
root@archangel:~# apt-get install nagios3
```

Die Konfiguration und auch die Benutzung beginnen mit der Eingabe des URL:

http://localhost/nagios3

Icinga2

Bei *Icinga* handelt es sich um einen Fork von *Nagios*. Nachdem das Projekt große Fortschritte in seiner Entwicklung gemacht hatte, entschied man sich aber für eine komplette Neuprogrammierung, um ein eigenständiges, stabiles Produkt zu erhalten, nämlich Icinga2.

Die Installation von Icinga2 ist, wenn man den vollen Funktionsumfang nutzen will, recht aufwendig und würde den Rahmen dieses Kapitels bei Weitem sprengen.

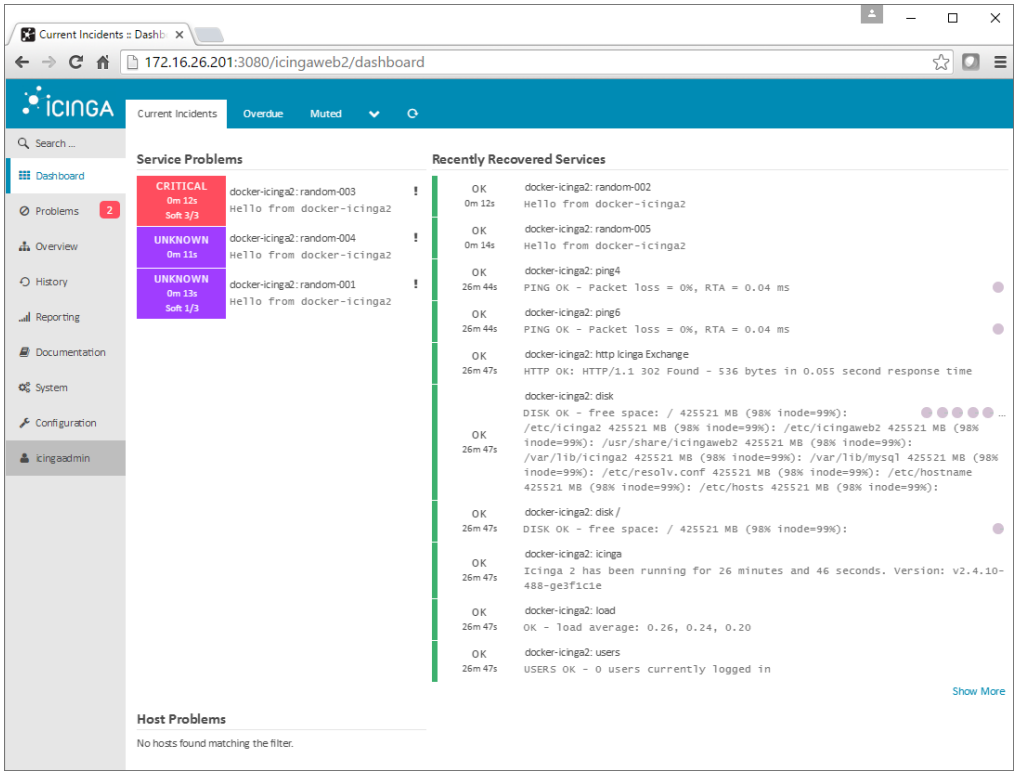

Abbildung 200.2 Das konfigurierbare Dashboard von Icinga2 gibt einen schnellen Überblick über gefundene Probleme.

Prüfungstipp

Verwenden Sie nicht zu viel Energie auf das Verständnis der Details von *collectd*, *Icinga2*, *Nagios*, *Cacti* und *MRTG*. Sie sollten diese Produkte nur kennen, müssen sie aber nicht konfigurieren können. Konzentrieren Sie sich mehr auf die Diagnosewerkzeuge!

201 Der Linux-Kernel

Das zweite Kapitel dieses Buchs widmet sich dem Kern von Linux: dem Kernel. Konfigurieren und kompilieren Sie ihn so, wie Sie ihn benötigen oder wie Sie ihn haben wollen.

201.1 Kernel-Komponenten

Wichtung: 2

Beschreibung: Sie sollten dazu in der Lage sein, Kernel-Komponenten zu verwenden, die für spezifische Hardware, Gerätetreiber, Systemressourcen und Anforderungen notwendig sind. Dieses Lernziel umfasst auch das Bereitstellen verschiedener Typen von Kernel-Images, das Identifizieren von stabilen und Entwicklungs-Kerneln sowie die Verwendung von Kernel-Modulen.

Wichtigste Wissensgebiete:

- Dokumentation für Kernel 2.6.x, 3.x und 4.x

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */usr/src/linux*
- */usr/src/linux/Documentation*
- *zImage*
- *bzImage*
- *xz compression*

Allgemeines

Eigentlich wollte Linus Torvalds lediglich eine Terminalemulation programmieren, als er die ersten Zeilen für ein Betriebssystem schrieb, das heute weltweit verwendet wird. Er benötigte diesen Terminalemulator, um damit auf den Universitätscomputer in Helsinki zugreifen zu können, weil der Terminalemulator des Betriebssystems, das er zu dieser Zeit verwendete (*MINIX* von Andrew S. Tanenbaum), für seine Begriffe einfach zu schlecht war. Er wollte durch dieses Programmierprojekt gleichzeitig Erfahrungen mit seinem damals neuen 386er-Computer sammeln. Deshalb schrieb er den Terminalemulator in reinem Assembler, und zwar nicht unter *MINIX*, sondern direkt auf die Hardware. Er konnte den Terminalemulator einfach von einer Diskette booten

oder eben *MINIX* von der Festplatte aus starten. Leider war der Terminalemulator auf diese Weise isoliert auf der Diskette. Er konnte nicht auf die Festplatte zugreifen und deshalb auch keine Programme und Dateien vom Universitätscomputer herunterladen und abspeichern. Deshalb wurde es nötig, Treiber für die Festplatte und auch einen Dateisystemtreiber zu programmieren. Den Dateisystemtreiber schrieb Linus Torvalds kompatibel zu *MINIX*, damit er zwischen den beiden Systemen Daten austauschen konnte. Ein paar Wochen später war der »Terminalemulator« multiuserfähig und verfügte bereits über vier gleichzeitig nutzbare Konsolen.

Die erste Linux-Version mit der Nummer 0.01 stellte Linus Torvalds am 17.09.1991 auf einem FTP-Server der Öffentlichkeit zur Verfügung. Zu diesem Zeitpunkt wusste natürlich noch niemand, was der Schwedisch sprechende Student aus Finnland da auf seinem 386er mit 4-MB-RAM ins Leben gerufen hatte.

Der Sinn eines Kernels liegt in der Abstraktion der Hardware eines Computers. Er läuft sozusagen direkt auf der Hardware und stellt der Software, die auf einem Computer installiert wird, Schnittstellen zur Verfügung, damit diese nicht direkt mit der Hardware kommunizieren muss. Die installierten Programme laufen in einem Systembereich, der als das User-Land bezeichnet wird. Weil ein Kernel einer Anwendung, die sich im User-Land befindet, an der Grenze immer die gleichen Schnittstellen (APIs) zur Verfügung stellt, wird eine Anwendungssoftware von der Hardware unabhängig. Dadurch sind diese Anwendungen leicht auf andere Hardwareplattformen portierbar. Ein schönes Beispiel hierfür ist der Samba-Server. Sie finden Samba-Server nicht nur auf PCs, sondern auch auf den kleinen NAS-Geräten, die es inzwischen in jedem Elektronikmarkt zu kaufen gibt. In solchen Geräten finden Sie teilweise aus Gründen der Energieeffizienz RIS-Prozessoren vor, die mit der x86-Architektur kaum etwas gemeinsam haben. Dennoch läuft hier der Samba-Server, und er wird bei einigen Geräten genauso konfiguriert wie auf einem PC.

Hinweis

Die Kernel-Versionen 3.x und 4.x unterscheiden sich von der Behandlung her nicht von den Kerneln der Versionen 2.6.x. Linus Torvalds hat lediglich mit der Zahl 3 das dritte Jahrzehnt von Linux eingeläutet. Die theoretische Version 2.6.40 wurde deshalb zur Version 3.0 erklärt. Über die Anhebung der Kernel-Version auf 4.0 wurde demokratisch (via google+) entschieden.

Zu den Kernel-Quellen gehörende Dateien und Verzeichnisse

Wenn Sie neue Kernel kompilieren oder bestehende Kernel anpassen wollen, gibt es ein Verzeichnis, das für diese Arbeiten vorgesehen ist. Das Verzeichnis ist */usr/src*.

Normalerweise sollten Sie die Kernel-Quellen an dieser Stelle entpacken. Hierbei entsteht dann ein Verzeichnisbaum, der typischerweise z. B. */usr/src/linux-linux-4.6.2.* heißt. In diesem Beispiel handelt es sich um die Kernel-Quellen der Version linux-4.6.2. Sie sollten einen Softlink */usr/src/linux* erstellen, der auf diesen Verzeichnisbaum zeigt. Führen Sie zu diesem Zweck einfach folgendes Kommando aus:

```
ln -s /usr/src/linux-linux-4.6.2 /usr/src/linux
```

Im Verzeichnis */usr/src/linux/Documentation* finden Sie geradezu eine Flut an Informationen zum Kernel und hinsichtlich der Vorgehensweisen zu allen erdenklichen Spezialfällen. Vielen Themengebieten ist sogar ein eigenes Unterverzeichnis gewidmet. So gibt es z. B. ein Verzeichnis *Laptop*, in dem wiederum Treiberbeschreibungen einzelner Hersteller abgelegt sind. In jedem Verzeichnis gibt es eine Datei mit der Bezeichnung *00-INDEX* (auch im Basisverzeichnis der Dokumentation), die Ihnen dabei hilft, den Überblick über die vielen Dokumente zu behalten.

Wenn Sie die Dokumente in ein bequemeres Format umwandeln wollen, können Sie das machen, indem Sie in das Verzeichnis */usr/src/linux* wechseln und dann die folgenden Kommandos eingeben:

```
root@arch-deb:/usr/src/linux# make htmldocs
root@arch-deb:/usr/src/linux# make pdfdocs
```

Es werden dann entsprechende HTML- und PDF-Dokumente generiert. Beachten Sie bitte, dass ggf. vorher zusätzliche Programme für die Konvertierung installiert werden müssen. Während der Erstellung der HTML-Dokumente wird `xmlto` benötigt, und für die PDF-Dokumentenerstellung muss `passivetex` auf dem System installiert sein.

Nach einem Kompiliervorgang, wie Sie ihn auf den nächsten Seiten noch im Detail kennenlernen werden, finden Sie den statischen Teil des Kernels in diesem Pfad:

/usr/src/linux/arch/x86/boot/bzImage

Das Präfix *bz* (big size) ist hierbei ein Hinweis darauf, dass es sich um einen großen Kernel (größer als 512k) handelt und hat nichts mit dem Kompressionsalgorithmus *bz* zu tun. Kleine Kernel mit der Bezeichnung *zImage* kommen heutzutage eigentlich kaum noch vor. Dieser Teil des Kernels wird, je nach Vorgehensweise, entweder automatisch oder von Hand in das Verzeichnis */boot* kopiert.

Um den Systemstart zu beschleunigen und um Speicherplatz auf Datenträgern zu sparen, werden Kernel komprimiert. Bei neuen Kernelversionen wird der besonders effiziente *xz-Algorithmus* verwendet. Das ist zwar für die Administration nicht sonderlich wichtig, aber Sie sollten es für die Prüfung wissen.

201.2 Einen Linux-Kernel kompilieren

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen Kernel richtig zu kompilieren und, je nach Erfordernissen, spezifische Eigenschaften des Kernels zu aktivieren oder zu deaktivieren. Dieses Lernziel beinhaltet das Kompilieren bzw. Rekompilieren des Kernels, das Integrieren von Updates und das Erkennen von Änderungen in einem neuen Kernel. Des Weiteren sollten die Prüflinge dazu in der Lage sein, ein *initrd*-System-Image zu erstellen und neue Kernel zu installieren.

Wichtigste Wissensgebiete:

- */usr/src/linux/*
- Kernel-Makefiles
- make-Ziele der Kernelversionen 2.6.x, 3.x und 4.x
- Anpassen der aktuellen Kernel-Konfiguration
- Kompilieren eines neuen Kernels, inklusive der passenden Kernel-Module
- Installation eines neuen Kernels, einschließlich Module
- Sicherstellen, dass der Bootmanager den neuen Kernel und alle weiteren notwendigen Dateien findet
- Modul-Konfigurationsdateien
- Verwendung von DKMS zum Kompilieren von Kernelmodulen
- Kenntnis von *dracut*

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `mkinitrd`
- `mkinitramfs`
- `make`
- `make`-Ziele (`all`, `config`, `xconfig`, `menuconfig`, `oldconfig`, `cloneconfig`, `prepare-all`, `mrproper`, `zImage`, `bzImage`, `modules`, `modules_install`, `rpm-pkg`, `binrpm-pkg`, `deb-pkg`)
- `gzip`
- `bzip2`
- Modulwerkzeuge
- */usr/src/linux/**
- */usr/src/linux/.config*
- */lib/modules/kernel-version/**
- `Depmod`
- *dkms*

Allgemeines

Manchmal kann es nötig oder zumindest sinnvoll sein, einen Kernel komplett neu zu erstellen. So ist es z. B. möglich, dass man eine Hardwarekomponente in seinem Computer verwendet, die vom installierten Kernel nicht unterstützt wird. Ein anderer Grund könnte eine optimale Anpassung des Kernels an die Hardware des Computers sein, um die Leistung des Systems vollständig auszunutzen. Das ist unter Linux ohne Weiteres möglich, weil der Quellcode für den Kernel nicht wie bei anderen Betriebssystemen geheim gehalten wird, sondern einfach aus dem Internet heruntergeladen werden kann. So können also alle notwendigen Änderungen an der Konfiguration des Kernels vorgenommen werden und anschließend kann eine Neukompilierung des Kernels durchgeführt werden. Den meisten Linux-Distributionen liegt auch der Quellcode des mitgelieferten Kernels bei. Man findet ihn in der Regel unter dem standardisierten Verzeichnis */usr/src*. Im Internet bekommen Sie aktuelle Kernel unter *http://www.kernel.org*.

Identifizieren von Kernel-Versionen

Die Version des aktuell laufenden Kernels finden Sie heraus, indem Sie folgendes Kommando ausführen:

```
harald@archangel:/> uname -r
2.6.38.3
```

Diese Versionsnummer findet sich aber auch noch an anderen Stellen wieder. Unter */lib/modules/*finden Sie in Abhängigkeit von der verwendeten Distribution vielleicht sogar eine ganze Historie der Kernel, die einmal auf Ihrem System installiert waren. Ein anderes Verzeichnis wäre */usr/src/<Kernel-Version>*, in dem sich optional der Quellcode eines Kernels befindet. Außerdem ist die Version auf der Website *http://www.kernel.org* zu finden.

Aber was verbirgt sich hinter dieser Buchstaben- und Zahlenkombination? Es existiert natürlich auch hier ein Schema, das bestimmten Konventionen unterliegt:

- Die erste Zahl, in diesem Fall also die 2, steht für das sogenannte *Major Release*. Das Major Release wird nur dann erhöht, wenn es sozusagen revolutionäre Erneuerungen im Kernel zu verzeichnen gibt. Aktuell ist das Major Release 4. Die Version 3.0 wurde im Mai 2011 lediglich eingeführt, weil Linus Torvalds der Ansicht war, dass die Bezeichnungen der Patch-Level zu große Werte aufwiesen. Außerdem kündigt die Zahl 3 das dritte Jahrzehnt der Existenz von Linux an. Der Wechsel zum *Major Release* 4 wurde demokratisch via *google+* abgestimmt.
- Die zweite Zahl (im vorangehenden Beispiel die 6) stellt die Version des *Minor Releases* dar. Das Minor Release änderte sich immer dann, wenn einem Kernel neue wesentliche Funktionen hinzugefügt wurden. Dass die Ziffer an zweiter

Stelle steht, sollte aber nicht zu einer Unterschätzung der Versionsänderungen führen. So bestehen bereits zwischen den Versionen 2.2 und 2.4 bahnbrechende Unterschiede. Es gibt (bei älteren Kerneln) zudem noch eine Unterscheidung zwischen Minor-Release-Versionen mit geraden und ungeraden Werten:

- Kernel mit geradzahliger Versionsnummer des *Minor Release* (2.2, 2.4, 2.6 usw.) werden als »*stable*« bezeichnet und gelten somit als sicher (stable = haltbar, dauerhaft).
- Versionen, die eine ungerade Ziffer im *Minor Release* tragen (2.1, 2.3, 2.5 usw.), bezeichnet man als Entwickler-Kernel. Der Verwendungszweck ist hier deutlich dem Namen zu entnehmen. Es wird aber natürlich gern gesehen, dass solche Kernel auf nicht produktiven Computern zu Testzwecken eingesetzt werden. Ab der Version 2.6 werden allerdings die Entwickler-Kernel nicht mehr auf diese Art unterschieden. Hierdurch soll der Aufwand, der beim Portieren von neuen Funktionen einer (tatsächlich nicht existierenden) 2.7er-Kernel-Version in die aktuellen Produktions-Kernel entsteht, entfallen. Stattdessen wird bei den 2.6er-Kerneln eine vierte Ziffer zur Versionsverwaltung verwendet (z. B. 2.6.38.3).

- Die dritte Zahl (im vorangehenden Beispiel die `38`) steht für das *Patch-Level* des Kernels. Bei Erhöhung des *Patch-Levels* ist allerdings nur mit kleineren Änderungen und Fehlerbeseitigungen des Vorgänger-Kernels zu rechnen.
- Alles was danach folgt (im Beispiel ist es `4-21.9-smp`), ist lediglich eine Bezeichnung, die man vor dem Kompilieren im Makefile des Kernels angeben kann. Hier sind der eigenen Fantasie für Benennungskonventionen keine Grenzen gesetzt.

Den Kernel konfigurieren

Was die Konfiguration des Kernels angeht, soll vorab bemerkt werden, dass sich die Arbeitsschritte in Abhängigkeit von der ausgewählten Kernel-Version zwischen 2.6, 3.x oder 4.x nicht unterscheiden.

Bevor es losgeht und man den Kernel kompilieren kann, muss erst noch eine Menge Konfigurationsarbeit geleistet werden. Hierbei geht es um die Erstellung oder Änderung einer Datei mit der Bezeichnung *.config*. Diese Datei wird später vom Compiler ausgewertet, um den Kernel in der Form zu erstellen, in der man ihn benötigt. Es ist glücklicherweise nicht erforderlich, diese Datei von Hand zu erstellen; es wird von den Kernel-Entwicklern sogar ausdrücklich davon abgeraten. Hier ist der Kopf der Datei:

```
# Automatically generated make config: don't edit
# Linux kernel version: 2.6.38.4-21.9
# Thu Aug  2 18:13:32 2007
CONFIG_X86=y
CONFIG_MMU=y
```

```
CONFIG_UID16=y
CONFIG_GENERIC_ISA_DMA=y
CONFIG_GENERIC_IOMAP=y
# Code maturity level options
CONFIG_EXPERIMENTAL=y
CONFIG_CLEAN_COMPILE=y
CONFIG_BROKEN_ON_SMP=y
```

Die Originaldatei ist insgesamt mehr als 3.000 Zeilen lang. Es ist auch für nicht ausgebildete Programmierer leicht zu erkennen, dass hier Unmengen von Kompileranweisungen festgelegt worden sind. Die meisten davon sind vom Typ Boolean, beinhalten also lediglich Ja-/Nein-Werte.

Sie können als Basis auch die *.config*-Datei des laufenden Kernels verwenden. Dadurch ersparen Sie sich eine Menge Arbeit. Für die Konfiguration stehen mehrere Frontends zur Verfügung. Da wären:

```
make config
make menuconfig
make xconfig
make gconfig
```

In allen vier Fällen erhalten Sie dasselbe Ergebnis, nämlich eine hoffentlich nach Ihren Vorstellungen konfigurierte *.config*-Datei. Es muss aber auch in jedem Fall das Paket *make* und ein *C-Compiler* auf dem Computer installiert sein. Der am meisten genutzte C-Compiler ist hier wohl der GNU-C-Compiler – kurz `gcc`. Das Programm `make` wird dann entsprechend das *Makefile* auswerten, das sich im selben Verzeichnis befindet wie die Datei *.config*. Im Header des *Makefile* findet man auch die Informationen über die Version des Kernels.

Konfigurationskommandos

make config

Die puristischste Methode zur Erstellung der Konfiguration ist `make config`. Man muss bei der Arbeit mit `make config` alle Kernel-Parameter angeben. Es gibt inzwischen aber mehr als tausend dieser Kernel-Parameter, sodass diese Methode durchaus als zeitraubend bezeichnet werden muss. In den 90er Jahren des letzten Jahrhunderts war es aufgrund deutlich weniger Parameter noch erträglich mit `make config` zu arbeiten.

```
archangel:/usr/src/linux # make config
  HOSTLD  scripts/kconfig/conf
scripts/kconfig/conf arch/i386/Kconfig
#
```

```
# using defaults found in .config
*
* Linux Kernel Configuration
*
* Code maturity level options
*
Prompt for development and/or incomplete code/drivers (EXPERIMENTAL) [Y/n/?] y
  Select only drivers expected to compile cleanly (CLEAN_COMPILE) [Y/n/?] y
```

Bei den meisten Abfragen läuft es auf die gleichen Antwortmöglichkeiten hinaus. Hierbei steht:

- `n` für `no`. Die entsprechende Komponente wird also nicht in den Kernel aufgenommen.
- `y` für `yes`. Die Komponente wird in den statischen Teil des Kernels kompiliert.
- `m` für `modular`. Die Komponente wird als Kernelmodul aufgenommen und kann somit später im laufenden Betrieb geladen oder entladen werden.

Es kann aber auch jederzeit ein Fragezeichen eingegeben werden, um Informationen zu der aktuellen Option zu bekommen. Auch werden dann Empfehlungen dazu gegeben, wann die jeweilige Komponente benötigt wird und wann nicht.

make menuconfig

Mit `make menuconfig` kann man schon erheblich komfortabler arbeiten. Es basiert auf der Bibliothek *ncurses*. Solche Programme erinnern von der Optik her an den wohl allseits bekannten Midnight Commander. Nun kann man sich leicht vorstellen, dass `make menuconfig` mit Cursortasten bedient wird und die Konfiguration im Verhältnis zu `make config` erheblich erleichtert wird. Im Prinzip sind schon alle Kernel-Parameter vorkonfiguriert und man navigiert nur noch zu den Unterpunkten innerhalb der angebotenen Optionen, die man konfigurieren möchte. Zum Schluss verlässt man das Programm einfach und bestätigt die Abfrage, ob die Kernel-Konfiguration gespeichert werden soll, mit »Yes«.

make xconfig

Hierbei handelt es sich um ein Konfigurationsfrontend mit einer echten Grafikoberfläche. Deshalb kann diese Methode auch nur verwendet werden, wenn man ein X Window System auf dem Computer installiert hat. Viele Administratoren bevorzugen heutzutage diese Methode. Ich gebe hier aber zu bedenken, dass die Gewöhnung an grafische Tools zur Verwaltung die Fernadministration mittels `ssh` erschwert.

make gconfig

Als letzte Methode sei hier der Vollständigkeit halber noch `make gconfig` genannt. Das ist ein weiteres grafisches Frontend, das ebenfalls ein X Window System voraussetzt. Zusätzlich muss auf dem System GTK+ vorhanden sein.

make oldconfig

Eigentlich ist `make oldconfig` keine Konfigurationsmethode, die sich mit den vorangehenden Methoden vergleichen lässt. Man benutzt sie, um die Konfiguration eines bestehenden Kernels in eine *.config*-Datei zu übernehmen. Auf diese Art und Weise kann man sich bei einem Kernel-Update unter Umständen eine Menge Arbeit sparen, weil man bei der Konfiguration nicht ganz von vorn beginnen muss. Das Programm fragt nur Einstellungen für Funktionen ab, die im vorherigen Kernel noch nicht vorhanden waren.

Der Vorgang des Kompilierens

Es kann also nun endlich losgehen. Nachdem alles vorbereitet ist, kann der Kernel kompiliert werden. Hierbei werden, wie schon erwähnt, die *.config*-Datei und das Makefile ausgewertet. Bitte vergessen Sie deshalb nicht, in den Header des Makefiles ein paar passende Werte zu schreiben. Durch eine vernünftige Versionierung können Sie den Kernel dann leichter von anderen Kerneln unterscheiden.

Für die Prüfung müssen Sie mehrere Befehle zur Kompilierung des Kernels kennen, wenn auch bei neueren Kernel-Versionen nur noch ein einziger Befehl erforderlich ist. Hier ist die Auflistung in der Ausführungsreihenfolge:

- `make dep`
- `make clean`
- `make bzImage` bzw. `make zImage`
- `make modules`
- `make modules_install`

Ab Kernel-Version 2.6 muss man lediglich die Kommandos `make` und `make modules_install` ausführen. Alles andere geschieht automatisch. In der Prüfung wird davon ausgegangen, dass man auch noch mit älteren Kerneln konfrontiert wird. Deshalb müssen Sie den Umgang mit diesen auch noch beherrschen. Zum Üben können Sie im Internet problemlos ältere Kernel-Versionen herunterladen. Zum Beispiel:

ftp://ftp.kernel.org/pub/linux/kernel/v1.0/

Diese URL enthält wirklich das, wonach sie aussieht – den allerersten Linux-Kernel, den Linus Torvalds selbst geschrieben hat.

Wenn Sie sich beim Experimentieren Frust ersparen wollen, kompilieren Sie lieber nicht Ihren neuen zukünftigen Produktiv-Kernel, sondern deklarieren Sie ihn gleich

als Übungsobjekt. Konfigurieren Sie diesen wie in den oberen Abschnitten beschrieben, und kompilieren Sie ihn (im Verzeichnis der Kernel-Sourcen) mit den folgenden Befehlen: `make dep` prüft Abhängigkeiten in den Quelldateien und erstellt in jedem Unterverzeichnis der Kernel-Sourcen die Datei *.depend*. `make clean` entfernt ggf. Kompilate von ehemaligen Kompiliervorgängen und alte *.depend*-Dateien. `make bzImage` erstellt den statischen Teil des Kernels unterhalb des aktuellen Verzeichnisses in *arch/i386/boot/bzImage*. Das *b* steht hierbei für »big« und impliziert, dass der Kernel ein größeres Format zulässt. Zur Kompression eines Kernelimages kommen die Mechanismen *gzip*, *bzip2*, *lzma*, *xz*, *lzo* oder *lz4* in Frage. Sie können während der Konfiguration des Kernels den Algorithmus auswählen.

Bei älteren Kernel-Versionen wurde `make zImage` durchgeführt, wobei das *z* ein Hinweis auf die vorhandene *zlb*-Kompression war. Die ersten Kernel waren übrigens unkomprimiert und wurden mit `make Image` erstellt. `make modules` kompiliert die Module und `make modules_install` installiert die Module unter */lib/modules/<Kernel-Version>*.

Es besteht auch die Möglichkeit, alle Schritte mit einem einzigen Befehl zu übergeben, wenn man dem System vertraut. Das sieht dann so aus:

```
make dep clean bzImage modules modules_install
```

Wenn alles gut gegangen ist, kann man jetzt den Kernel in das System einbauen. Die Module sind sowieso schon am richtigen Platz und Sie müssen nur noch das Kernel-Image in das Verzeichnis */boot* kopieren. Also z. B. so:

```
cp /usr/src/linux/arch/i386/boot/bzImage /boot/vmlinuz-2.4.18
```

Wenn als Bootloader GRUB verwendet wird, kann man mit dem Kommando `update-grub` einen Eintrag für den neuen Kernel in der Datei *menu.lst* bzw. *grub.cfg* erstellen lassen. Sie können das System anschließend neu starten und aus dem Bootmenü den neuen Kernel auswählen.

Beim nächsten Neustart des Systems steht der Kernel dann hoffentlich zur Verfügung. Wenn nicht, sind folgende Probleme denkbar:

- Sie haben Teile des Kernels modular gehalten, die beim Systemstart erforderlich sind. Hier hilft das Ausführen von `mkinitrd` (bzw. `mkinitramfs`, wenn Sie Debian verwenden). Dieser aktualisiert die initiale RAM-Disk unter */boot/initrd*. Diese wird in einer frühen Phase des Startvorgangs ausgeführt und lädt Module, die zum Fortsetzen des Bootprozesses notwendig sind.
- Eine andere Möglichkeit wäre, dass `make clean` nicht richtig gearbeitet hat. Insbesondere bei neueren Kernel-Versionen empfehle ich, die Methode `make mrproper` zu bevorzugen.

Lesen Sie spätestens jetzt die *README*-Datei der verwendeten Kernel-Version. Diese kleine Lektüre lohnt sich spätestens ab Version 2.6 sowieso. Es gibt viele neue Optionen für `make`, die einem die Arbeit erheblich erleichtern können oder einfach nur tolle Erfindungen der Kernel-Programmierer sind. Für die Prüfung sind diese Parameter zumindest noch nicht von Bedeutung.

Hinweis

Die nun folgende Beschreibung, wie ein Kernel gepatcht wird, ist nicht mehr direkt Bestandteil der Prüfung. Da das hierfür verwendete Kommando `patch` aber in anderen Zusammenhängen prüfungsrelevant ist, wird es an dieser Stelle dennoch ausgeführt.

Kernel patchen

Um neue Funktionen in die Kernel-Quellen zu integrieren, können Sie Patches verwenden. Üblicherweise verwendet man diese Patches, um zusätzliche Treiber einzubinden. In der Praxis sind zusätzliche Treiber allerdings lediglich Kernel-Module und die Hersteller dieser Erweiterungen für den Kernel liefern gleich ein Setup-Programm mit, das einem die Aufgabe des Kompilierens abnimmt. Ein anderer Grund für das Anwenden von Patches ist die Aktualisierung von Kernel-Quellen. Auf der Website *www.kernel.org* finden Sie immer komplette Versionen verschiedener Kernel, aber auch die Patch-Dateien, um bestehende Kernel-Quellen zu aktualisieren. Bei den heute verfügbaren Internetbandbreiten ist es allerdings vernünftiger, einfach einen neuen Kernel komplett herunterzuladen. Ein vollständiger Kernel umfasst lediglich ca. 70 MB.

Wenn Sie zur Übung doch einmal einen Patch anwenden möchten, sollten Sie bei den Patch-Versionen beachten, dass ein Patch immer nur die direkte Vorgängerversion eines Kernels aktualisieren kann. So aktualisiert etwa *patch-2.6.39-rc4* den Kernel *2.6.38*. Der Suffix *rc4* ist ein Hinweis darauf, dass es sich hierbei um den vierten Release Candidate handelt.

Arbeiten mit patch, gunzip und bzip2

Um den eigentlichen Patch-Vorgang durchzuführen, verwendet man das Programm `patch`. Da dieses Programm normalerweise von *stdin* liest, leiten Sie einfach die Ausgabe von `gunzip`, `bzip2` oder `cat` an `patch` um, je nachdem in welchem Format die Patch-Datei vorliegt. Im folgenden Beispiel wird davon ausgegangen, dass der Kernel mit `bzip2` komprimiert wurde. Das ist eigentlich (auch auf *www.kernel.org*) gängige Praxis, weil der Kompressionsgrad gegenüber `gzip` höher ist. Bevor Sie den Patch tatsächlich anwenden, sollten Sie vorher einen Test mit der Option `--dry-run` durchfüh-

ren, damit Sie bei einem eventuell auftretenden Fehler nicht Ihre Kernel-Quellen in einen unbrauchbaren Zustand versetzen:

```
root@arch-deb:/usr/src# bzip2 -dc patch-2.6.39-rc4.bz2 | patch -p1 --dry-run
patching file .mailmap
Hunk #1 FAILED at 20.
Hunk #2 FAILED at 70.
Hunk #3 FAILED at 78.
Hunk #4 FAILED at 98.
4 out of 4 hunks FAILED -- saving rejects to file .mailmap.rej
patching file CREDITS
```

In diesem Fall wäre der Patch-Vorgang tatsächlich fehlgeschlagen. Die Ursache für den Fehler war hier eine für den Kernel unpassende Patch-Version. Gut, dass es sich lediglich um einen Test mit `--dry-run` gehandelt hat.

Die Option `-d` sorgt bei `bzip2` übrigens für die Dekomprimierung, und `-c` veranlasst die Ausgabe nach *stdout*. Ohne Angabe der Option `-c` wäre die komprimierte Datei in eine nichtkomprimierte Datei konvertiert worden, was hier aber nicht das Ziel war.

Wenn der Patch mittels `gzip` gepackt wurde, müssen Sie stattdessen dieses Kommando verwenden:

```
root@arch-deb:/usr/src# gunzip -c patch-2.6.39-rc4.gz | patch -p1 --dry-run
```

Die Option `-c` leitet die Ausgabe von `gunzip` ebenfalls nach `stdout` um, damit das Programm `patch` das Ergebnis von `stdin` lesen kann.

Anpassen, Kompilieren und Installieren eines Kernels inklusive Kernel-Module

Eine Konfiguration mit Debian

Zunächst müssen die benötigten Verzeichnisse erstellt, Werkzeuge besorgt und die Kernel-Quellen aus dem Internet heruntergeladen werden. Die Werkzeuge installieren Sie am besten mittels `apt-get`:

```
root@hv01:/usr/src# apt-get install build-essential ncurses-dev
```

Den Download der Kernelquellen können Sie mithilfe des Programms `wget` durchführen.

```
root@hv01:/usr/src# wget https://cdn.kernel.org/pub/linux/kernel/v4.x/linux-
4.6.3.tar.xz
```

Jetzt befindet sich das Archiv schon genau da, wo Sie es benötigen. Da sich die Verzeichnisstruktur bei *http://www.kernel.org* eigentlich nie ändert, können Sie das Kommando intuitiv ändern, zumindest was die Kernel-Versionen betrifft.

Packen Sie das Archiv aus, indem Sie `tar` verwenden:

```
root@hv01:/usr/src# tar -xJvf linux-4.6.3.tar.xz
```

Als Nächstes sollten Sie einen Softlink mit der Bezeichnung *linux* erstellen, der auf das soeben entstandene Verzeichnis zeigt:

```
root@hv01:/usr/src# ln -s linux-4.6.3 linux
```

Sie können jetzt in das *linux*-Verzeichnis wechseln:

```
root@hv01:/usr/src# cd linux
```

Wenn Sie sich den Inhalt des Verzeichnisses ansehen, werden Sie feststellen, dass es zwar ein Makefile, aber keine *.config*-Datei gibt. Sie können das Makefile mit einem Editor Ihrer Wahl öffnen, damit Sie für die Versionierung im oberen Teil der Datei etwas Sinnvolles eintragen können. So können Sie später mehrere Kernel leicht voneinander unterscheiden und vermeiden vor allem Konflikte in den Modulverzeichnissen. Im Rohzustand sieht der Kopf der Datei so aus:

```
VERSION = 4
PATCHLEVEL = 6
SUBLEVEL = 3
EXTRAVERSION =
NAME = Mein-wunderbarer-Kernel
```

Um Arbeit bei der Grundkonfiguration zu sparen und auf Anhieb einen Kernel zu erzeugen, der zu Ihrem laufenden System passt, können Sie die Konfiguration Ihres bestehenden Kernels kopieren. Dadurch erhalten Sie eine gute Ausgangsbasis für die eigene Konfiguration:

```
root@hv01:/usr/src/linux# cp /boot/config-$(uname -r) .config
```

Alternativ können Sie dieses Kommando verwenden:

```
root@hv01:/usr/src/linux# make oldconfig
```

Es sollte jetzt ohne Probleme möglich sein, `make menuconfig` auszuführen. Ein kleiner Test wird es zeigen:

```
root@arch-deb:/usr/src/linux# make menuconfig
```

Wenn Sie `menuconfig` wieder verlassen, sollten Sie die Konfiguration speichern. Sie können `make menuconfig` jederzeit erneut ausführen, um die Konfiguration fortzusetzen. Wenn Sie später einen lauffähigen Kernel erhalten wollen, sollten Sie keine Optionen verändern, von denen Sie nicht wissen, was diese bewirken. Grundsätzlich

können Sie aber zu jeder einzelnen Option die Hilfe aufrufen und sich belesen, wenn Sie sich nicht sicher sind.

Sollten Sie zwischendurch Fehler bei der Konfiguration des Kernels gemacht haben, können Sie die Kernel-Quellen jederzeit mit `make clean` bzw. `make mrproper` wieder in den Ursprungszustand zurückversetzen. Eine *.config*-Datei lässt sich übrigens auch verwenden, wenn Sie gar keine Änderungen vornehmen, sondern einfach `make menuconfig` starten, wieder beenden und speichern.

Nachdem Sie die Konfiguration abgeschlossen haben, können Sie den Compiler starten. Es ist ganz interessant, das Kommando `time` voranzustellen, damit man sehen kann, wie lange der Vorgang gedauert hat.

```
root@hv01:/usr/src/linux# time make all
```

Das Kommando `make all` sorgt dafür, dass der eigentliche Kernel und die Module kompiliert werden. Zusätzlich wird das komprimierte Kernelimage (*bzImage*) erzeugt. Wenn dieser Vorgang abgeschlossen ist, können Sie sowohl den Kernel, als auch die Module installieren, indem Sie dieses Kommando verwenden:

```
root@hv01:/usr/src/linux# make install modules_install
```

Überprüfen Sie bitte, dass im Verzeichnis */lib/modules* ein dem neuen Kernel entsprechendes Unterverzeichnis existiert und im Verzeichnis */boot* dem neuen Kernel entsprechende Dateien angelegt wurden.

Jetzt wird es Debian-spezifisch. Um die initiale RAM-Disk unter Debian zu erstellen, können Sie das Programm `mkinitramfs` verwenden. Das Programm `mkinitrd` funktioniert hier nicht.

```
root@hv01:/usr/src/linux# mkinitramfs -o /boot/initrd.img-4.6.3
```

Eine moderne Alternative zu `mkinitramfs` oder `mkinitrd` ist `dracut`. Dieses Programm arbeitet vollautomatisch und benötigt im Normalfall weder Optionen noch Parameter um initiale RAM-Disks zu erstellen. Wenn Sie `dracut` erst jetzt (via `apt-get`) installieren, wird das Programm nach der Installation sogar unaufgefordert seine Arbeit verrichten. Ein Blick in die entsprechende Manpage ist natürlich dennoch zu empfehlen.

Zum Schluss muss dem Bootloader noch mitgeteilt werden, dass es ein neues Linux-Image gibt. Das geht am leichtesten mit `update-grub2`:

```
root@hv01:/lib/modules# update-grub2
Generating grub configuration file ...
Found linux image: /boot/vmlinuz-4.6.3
Found initrd image: /boot/initrd.img-4.6.3
```

```
Found linux image: /boot/vmlinuz-3.16.0-4-amd64
Found initrd image: /boot/initrd.img-3.16.0-4-amd64
done
```

Wenn Sie mit Ihrem Kernel zufrieden sind, können Sie ihn zwecks einfacherer Bereitstellung auf anderen Systemen paketieren. Führen Sie hierzu einfach das folgende Kommando aus:

```
root@arch-deb:/usr/src/linux# make deb-pkg
```

Sie können den Kernel dann mithilfe von `dpkg` einfach auf anderen Systemen installieren.

Eine Konfiguration mit Fedora

Die Konfiguration des Kernels und auch die Installation werden bei einem Fedora-System natürlich sehr ähnlich durchgeführt. Es gibt nur ein paar Details, die man unbedingt kennen muss. In diesem Abschnitt sollen also vor allem diese Unterschiede hervorgehoben werden, im Telegrammstil:

```
[root@arch-fc ~]# cd /usr/src
```

Auf einem Fedora-System gibt es von Haus aus kein `wget`, also:

```
[root@arch-fc src]# yum install wget
```

Nun folgt der Download der Kernel-Quellen:

```
[root@arch-fc src]# wget http://www.kernel.org/pub/linux/kernel/v4.6/
linux-4.6.3.tar.xz
```

Packen Sie das Archiv mit `tar` aus:

```
[root@arch-fc src]# tar -xvJf linux-4.6.3.tar.xz
```

Erstellen Sie den Softlink *linux*:

```
[root@arch-fc src]# ln -s linux-4.6.3 linux
```

Sie können jetzt in das *linux*-Verzeichnis wechseln:

```
[root@arch-fc src]# cd linux
```

Bei Fedora müssen Sie berücksichtigen, dass die *ncurses*-Entwicklungsumgebung *ncurses-devel* heißt. Ansonsten kommen auch hier `gcc` und `make` zum Einsatz. Besorgen Sie die Werkzeuge:

```
[root@arch-fc linux]# yum install ncurses-devel gcc make
```

Die nächsten Schritte sind wieder die gleichen wie bei Debian. Für die Erstkonfiguration machen Sie:

```
[root@arch-fc linux]# make oldconfig
[root@arch-fc linux]# make menuconfig
```

Wenn Sie sehen wollen, wie lange Ihr Computer braucht, um den Kernel zu übersetzen, können Sie dem `make`-Kommando auch hier den Befehl `time` voranstellen.

```
[root@arch-fc linux]# time make all
```

Anschließend installieren Sie wie gehabt den Kernel und die Module mithilfe dieses Kommandos:

```
[root@arch-fc linux]# make install modules_install
```

Auch bei Red Hat-Derivaten kommt `dracut` zum Erstellen der initialen RAM-Disk zum Einsatz:

```
[root@arch-fc linux]# yum install dracut
```

Anschließend können Sie die initiale RAM-Disk mit `dracut` erzeugen.

```
[root@arch-fc linux]# dracut
```

Führen Sie anschließend `grub-mkconfig` (bzw. `grub2-mkconfig`, wenn Sie GRUB 2 verwenden) aus, damit der neue Kernel beim Systemstart zur Verfügung steht. Starten Sie den Computer neu, um das Ergebnis zu überprüfen. Wenn Sie denselben Kernel auf mehreren Computern installieren wollen, können Sie auch hier bequem ein RPM-Paket erstellen. Verwenden Sie dazu das folgende Kommando:

```
[root@arch-fc linux]# make binrpm-pkg
```

DKMS

DKMS (*Dynamic Kernel Module Support*) kommt zum Einsatz, wenn ein Softwareprodukt oder eine Hardwarekomponente eigene Kernelmodule benötigt. Das ist z. B. bei einigen Grafikkarten von *Nvidia* und bei *Virtualbox* der Fall.

Ohne *DKMS* müssten die jeweils mitgelieferten Kernelmodule nach einem Upgrade des laufenden Kernels manuell aktualisiert und wieder eingebunden werden. *DKMS* prüft bei jedem Systemstart automatisch, ob eine neue Kernelversion installiert wurde und bindet die Module, die mittels *DKMS* installiert wurden, in den neuen Kernel bei Bedarf ein.

Das ist natürlich besonders bei Linux-Distributionen interessant, wo in Aktualisierungen häufig neue Kernelversionen enthalten sind (z. B. Ubuntu, dessen Derivate

oder Arch Linux). Damit *DKMS* funktioniert, sind die entsprechenden Headerfiles passend zum laufenden Kernel erforderlich und müssen heruntergeladen werden, wenn sie nicht bereits vorhanden sind.

Als praktisches Beispiel soll hier die Installation von *Virtualbox* unter *Manjaro* dienen. Sie werden bestimmt keine Schwierigkeiten haben, diese Konfiguration auf Ihr eigenes System zu übertragen, auch wenn *Manjaro* `pacman` zum Paketmanagement verwendet.

Zunächst wird mithilfe des Paketmanagers *Virtualbox* installiert:

```
[root@arch-book /]# pacman -S virtualbox
```

Sie können davon ausgehen, dass *Virtualbox* nach der Installation nicht lauffähig ist, weil die benötigten Kernelmodule fehlen. Also müssen diese ebenfalls installiert werden:

```
[root@arch-book /]# pacman -S virtualbox-dkms-manjaro
```

Was nun noch fehlt, sind die Headerfiles des laufenden Kernels:

```
[root@arch-book /]# pacman -S linux45-headers
```

Schon während deren Installation greift *DKMS* ein und installiert die für *Virtualbox* benötigten Module.

```
==> dkms install -m vboxguest -v 5.0.22_OSE -k 4.5.7-1-MANJARO
==> dkms install -m vboxhost -v 5.0.22_OSE -k 4.5.7-1-MANJARO
```

Sie sehen hier gleichzeitig die Befehlssyntax des Kommandos `dkms` und drei wesentliche Optionen, nämlich:

- `-m` für den Namen des zu ladenden Moduls
- `-v` für die Version des Moduls
- `-k` die Kernelversion, in die `dkms` die Module integriert

Wenn Sie die Module in einen anderen installierten Kernel integrieren wollen, können Sie `dkms` von Hand ausführen und die Parameter der entsprechenden Optionen anpassen.

Es muss nur noch ein Setupscript ausgeführt werden, damit die benötigten Module geladen werden:

```
[root@arch-book /]# /sbin/rcvboxdrv setup
Unloading modules:
Loading modules: vboxnetadp vboxnetflt vboxpci vboxdrv
```

Sie können *Virtualbox* jetzt ohne einen Neustart des Systems verwenden.

201.3 Kernel und Kernel-Module zur Laufzeit verwalten und kernel-bezogene Fehlerbehebung

Wichtung: 4

Beschreibung: Kandidaten sollen einen 2.6.x-, einen 3.x- oder 4.x-Kernel und seine ladbaren Module verwalten und untersuchen können. Kandidaten sollen gängige Probleme beim Systemstart und im laufenden Betrieb erkennen und beheben können. Kandidaten sollen Geräteerkennung und -verwaltung mit *udev* verstehen. Dieses Prüfungsziel beinhaltet die Fehlersuche in *udev*-Regeln.

Wichtigste Wissensgebiete:

- Einsatz von Kommandozeilentools zum Abfragen von Informationen über den zurzeit geladenen Kernel und seine Module
- Manuelles Laden und Entladen von Kernel-Modulen
- Erkennen, wann Module entladen werden können
- Erkennen, welche Parameter ein Modul akzeptiert
- Modulaliase (Einrichten des Systems so, dass Module auch über andere Namen als ihre Dateinamen geladen werden können)
- */proc*-Dateisystem
- Inhalt von */*, */boot* und */lib/modules*
- Werkzeuge und Dienstprogramme zur Analyse der vorhandenen Hardware
- *udev*-Regeln

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */lib/modules/kernel-version/modules.dep*
- Modul-Konfigurationsdateien in */etc*
- */proc/sys/kernel*
- `depmod`
- `insmod`
- `lsmod`
- `rmmod`
- `modinfo`
- `modprobe`
- `uname`
- `dmesg`
- `lspci`

- `lsdev`
- `uname`
- `lsusb`
- `sysctl`
- `udevmonitor`
- `udevadm monitor`
- */etc/udev*
- */etc/sysctl.conf*, */etc/sysctl.d*

Allgemeines

Der Kernel eines Betriebssystems hat – wie Sie bereits wissen – die Aufgabe, als Schnittstelle (API) zwischen Hard- und Software zu fungieren. Natürlich handelt es sich bei dem Kernel selbst um ein Softwareprodukt, das aber immerhin den Kernbestandteil des Betriebssystems darstellt. Bei Linux handelt es sich inzwischen um einen zum großen Teil modular aufgebauten Kernel. Die einzelnen Module dieses Kernels können zur Laufzeit des Systems geladen und auch wieder entladen werden. Man kann die Module eines Kernels mit Treibern anderer Betriebssysteme vergleichen. Der modulare Aufbau hilft dabei, unnötige Neustarts des ganzen Systems zu vermeiden. Im Gegensatz hierzu stehen monolithische Kernel. Diese können schwere Störungen verursachen, wenn eine Hardwarekomponente versagt oder aus dem System entfernt wird. Dafür sind sie allerdings schneller als modular aufgebaute Kernel.

Ausschließlich monolithisch aufgebaute Kernel sind verhältnismäßig unflexibel, weil beim Systemstart, unabhängig von der verwendeten Hardware, der gesamte Kernel geladen werden muss. Sollte der laufende Kernel irgendwelche Geräte nicht unterstützen, muss er gegen einen neuen Kernel ersetzt werden. Der erste Linux-Kernel war auch ein Monolith. Da Linus Torvalds diesen Kernel aber speziell für seinen 386er mit 4-MB-RAM geschrieben hatte, war das auch gut so.

Im Gegensatz zu den monolithischen Kerneln stehen die modular aufgebauten Kernel. Diese werden als Microkernel bezeichnet, weil der feste Bestandteil des Kernels sehr klein ist. Ein bekanntes Beispiel für ein solches System ist *MINIX*, das von Andrew Stuart Tanenbaum in Zusammenarbeit mit einigen seiner Studenten entwickelt wurde. Über die jeweiligen Vor- und Nachteile von Linux und *MINIX* gerieten Linus Torvalds und Andrew Stuart Tanenbaum in einen heftigen Streit. Die Flames, die in diesem Zusammenhang damals über das Usenet ausgetauscht worden sind, sind bis heute im Internet zu finden.

Zum Kernel gehörende Dateien und Verzeichnisse

In einem Verzeichnis unterhalb von */usr/src* befinden sich die Quellen für den Linux-Kernel. Es wird hier ein Verzeichnis angelegt, das die genaue Versionsnummer des Kernels enthält, z. B. */usr/src/linux-4.6.2*. Zusätzlich wird ein Softlink erstellt, sodass die zu konfigurierenden Kernel-Quelldateien unter dem Pfad */usr/src/linux* zu erreichen sind.

Der statische Teil des lauffähigen Kernels befindet sich im Verzeichnis */boot*. Im Normalfall findet man dort einen Softlink namens `vmlinuz`, der auf den tatsächlichen Namen des Kernels zeigt. Hier gibt es natürlich bei der Verwendung mehrerer Kernel entsprechende Abweichungen. Auch sind distributionsspezifische Unterschiede denkbar.

Die Module des Kernels, um die es zunächst gehen soll, befinden sich in Verzeichnissen unterhalb von */lib/modules*. An dieser Stelle wird pro installiertem Kernel ein Unterverzeichnis angelegt, das nach der Kernel-Versionsnummer benannt ist, z. B. */lib/modules/4.6.2*.

Hier befanden sich in Kernel-Versionen vor 2.4 die Unterverzeichnisse für verschiedene Kategorien von Modulen. Ab Version 2.4 gibt es zunächst ein Unterverzeichnis namens *kernel* und erst darauf folgend die Verzeichnisse mit den jeweiligen Kategorien. Hier einige Beispiele:

- */lib/modules/2.6.38.3/kernel/fs* für Dateisystemmodule
- */lib/modules/2.6.38.3/kernel/net* für Netzwerkkartenmodule
- */lib/modules/2.6.38.3/kernel/scsi* für SCSI-Adaptermodule
- */lib/modules/2.6.38.3/kernel/video* für Grafikadaptermodule

An dieser Stelle findet man auch die Datei *modules.dep*. Diese Datei enthält Informationen über die Abhängigkeiten der Kernel-Module untereinander. Die Moduldateien selbst sind durch den Compiler erstellte ausführbare Dateien mit der Dateierweiterung *o* wie Objekt. Bei neuen Kernel-Versionen (2.6) ist diese Erweiterung in *ko* wie Kernel-Objekt geändert worden.

Module zur Laufzeit beeinflussen und konfigurieren

Mit den folgenden Befehlen und Methoden können die Module des laufenden Kernels beeinflusst und überprüft werden. So ist es möglich, im laufenden Betrieb Gerätetreiber zu laden oder zu entladen. Das kann z. B. erforderlich werden, wenn eine Hardwarekomponente wegen fehlerhaften Verhaltens neu initialisiert werden muss. Im Folgenden werden die dafür benötigten Kommandos dargestellt:

uname

Der Befehl `uname` gibt Informationen zum laufenden System aus. Mit entsprechenden Optionen versehen, kann man hier auch die Versionsnummer des laufenden Kernels überprüfen.

Zum Beispiel:

```
harald@archangel:/> uname -r
2.6.38.3
```

oder:

```
harald@archangel:/> uname -a
Linux archangel 2.6.38.3 #1 Wed Apr 27 11:58:59 UTC 2011 i686 i686 i386 GNU/
Linux
```

Die von `uname` generierten Informationen kann man im Folgenden benutzen, um sich genauere Informationen über bestimmte Module anzusehen oder um die zum laufenden Kernel passenden Module zu laden bzw. zu entladen.

lsmod

Mit dem Befehl `lsmod` kann man den Status der Module eines laufenden Kernels anzeigen. Hierbei greift `lsmod` auf das Verzeichnis */proc/modules* zu und gibt das Ergebnis in einer übersichtlicheren Form aus.

Hier der Vergleich:

```
harald@archangel:~> cat /proc/modules
vfat 17792 0 - Live 0xe0c37000
fat 43804 1 vfat, Live 0xe0c6e000
usbserial 34024 0 - Live 0xe11e8000
8139too 30464 0 - Live 0xe10fe000
mii 9088 1 8139too, Live 0xe0fad000
parport_pc 44356 1 - Live 0xe0e81000
reiserfs 263024 1 - Live 0xe0e8e000
ext3 145032 1 - Live 0xe0dc2000
```

Nun die Ausgabe mit dem `lsmod`-Befehl:

```
harald@archangel:~> lsmod
Module                  Size  Used by
vfat                   17792  0
fat                    43804  1 vfat
usbserial              34024  0
8139too                30464  0
mii                     9088  1 8139too
```

```
parport                 40392  3 ppdev,parport_pc,lp
reiserfs               263024  1
ext3                   145032  1
```

Die Originalausgabe wurde aus Platzgründen um ca. 80 % gekürzt.

modinfo

Mit `modinfo` kann man ein Modul des Kernels genauer unter die Lupe nehmen:

```
archangel:/ # modinfo /lib/modules/2.6.38.3/kernel/drivers/usb/storage/
                      usb-storage.ko
filename:       /lib/modules/2.6.38.3/kernel/drivers/usb/storage/
                usb-storage.ko
author:         Matthew Dharm <mdharm-usb@one-eyed-alien.net>
description:    USB Mass Storage driver for Linux
license:        GPL
vermagic:       2.6.38.3 SMP 586 REGPARM gcc-3.3
supported:      yes
depends:        ide-core,usbcore,scsi_mod
alias:          usb:v03EEp6901d*dc*dsc*dp*ic*isc*ip*
alias:          usb:v03F0p0107d*dc*dsc*dp*ic*isc*ip*
```

In diesem Beispiel finden sich Details über das für Massenspeichergeräte, die an eine USB-Schnittstelle angeschlossen sind, zuständige Modul. Der Autor mitsamt seiner E-Mail-Adresse wird angezeigt. Zudem erhält man Informationen über den Verwendungszweck des Moduls und die Versionsnummer. Außerdem werden die Abhängigkeiten von anderen Modulen aufgezeigt. Das sind sehr wesentliche Informationen, die auch noch von anderen Programmen, die Sie gleich kennenlernen werden, ausgewertet und genutzt werden.

`modinfo` versteht einige Optionen, die überwiegend dazu gedacht sind, nicht benötigte Informationen auszublenden. Da wären:

- `-a` zeigt nur den Autor des Moduls an.
- `-d` zeigt die Beschreibung (Description).
- `-l` zeigt die Lizenz.
- `-p` zeigt zu übergebende Parameter, falls möglich.
- `-n` zeigt den Dateinamen des Moduls an.

insmod

Mit dem Kommando `insmod` lassen sich Module in den laufenden Kernel integrieren. Das Programm erwartet die Übergabe des Moduls mit kompletter Pfadangabe und eventuellen Optionen, falls das Modul diese benötigt.

Es werden automatisch Abhängigkeiten geprüft, aber nicht automatisch aufgelöst. Sollte ein zu ladendes Modul also von weiteren Modulen abhängen, gibt `insmod` lediglich eine Fehlermeldung aus. Das Laden des Moduls für USB-Massenspeichergeräte könnte dann etwa so aussehen:

```
archangel:~ # insmod /lib/modules/2.6.38.3/kernel/drivers/usb/storage/
usb-storage.ko
```

`insmod` gibt im Erfolgsfall keine Bestätigungsmeldung aus. Das fragliche Modul wird kommentarlos in den Arbeitsspeicher geladen und vom Kernel verwendet.

rmmod

Mit diesem Kommando kann man nicht mehr benötigte Module wieder aus dem Arbeitsspeicher entfernen. Hierbei ist allerdings keine Pfadangabe erforderlich, weil `rmmod` mit dem */proc/modules*-Verzeichnis arbeitet und dieses nicht explizit angegeben werden muss (und kann). Beispiel:

```
archangel:~ # rmmod usb-storage.ko
```

Dieses Kommando entfernt also bei Bedarf den nicht mehr benötigten USB-Massenspeichertreiber. Es werden, wie auch beim `insmod`-Kommando, keine Erfolgsmeldungen ausgegeben. Es gibt allerdings auch hier Fehlermeldungen, wenn man versucht, ein Modul zu entladen, das noch von einem anderen Modul oder von einem Programm benutzt wird:

```
archangel:~ # rmmod ide_core.ko
ERROR: Module ide_core is in use by ide_cd,ide_disk,piix
```

Ähnliches geschieht, wenn man versucht, einen Dateisystemtreiber zu entfernen, während er von einem Laufwerk gerade benötigt wird.

Optionen für `rmmod`:

- `-v` für den Verbose Mode
- `-f` erzwingt das Entladen eines Moduls, auch wenn eventuelle Abhängigkeiten nicht erfüllt sind. Das ist natürlich ziemlich gefährlich.

modprobe

Dieses Kommando ist eine optimierte Kombination von `insmod` und `rmmod` in einem einzigen Programm. Der Verwendungszweck ist grundsätzlich den Verwendungszwecken älterer Werkzeuge ähnlich, jedoch bietet `modprobe` einige zusätzliche Annehmlichkeiten. So kann `modprobe` z. B. nicht nur Abhängigkeiten zwischen Modulen erkennen, sondern hieraus resultierende Probleme auch beheben. Es löst also sol-

che Abhängigkeiten auf, indem es fehlende Module selbstständig findet und auch lädt. Pfadangaben wie bei `insmod` sind hier nicht nötig, weil `modprobe` den `uname -r`-Befehl nutzt, um selbst das Basisverzeichnis der Module für den momentan laufenden Kernel zu finden.

Sie laden also ein Modul einfach durch die Übergabe seines Namens bzw. von mehreren Namen, wenn Sie mehrere Module gleichzeitig laden wollen:

```
archangel:~ # modprobe ext3 reiserfs
```

Beim Entfernen von Modulen kann man ebenfalls mehrere Module durch Leerzeichen voneinander getrennt übergeben. Beispiel:

```
archangel:~ # modprobe -r ext3 reiserfs
```

depmod

Die Abhängigkeiten zwischen den Modulen werden in einer Datei mit der Bezeichnung *modules.dep* zentral festgehalten. Beispiel:

```
harald@archangel:~$  cat /lib/modules/2.6.38.3/modules.dep
/lib/modules/2.6.38.3/kernel/sound/pci/ac97/snd-ac97-codec.ko: /lib/modules/
2.6.38.3/kernel/sound/core/snd-pcm.ko /lib/modules/2.6.38.3/kernel/sound/core/
snd-timer.ko /lib/modules/2.6.38.3/kernel/sound/core/snd.ko /lib/modules/
2.6.38.3/kernel/sound/soundcore.ko /lib/modules/2.6.38.3/kernel/sound/core/
snd-page-alloc.ko/lib/modules/2.6.38.4-21.9-mp/kernel/sound/oss/ymfpci.ko:
/lib/modules/2.6.38.3/kernel/sound/oss/ac97_codec.ko /lib/modules/2.6.38.3/
kernel/sound/oss/uart401.ko /lib/modules/2.6.38.3/kernel/sound/oss/sound.ko
/lib/modules/2.6.38.3/kernel/sound/soundcore.ko
```

Erfreulicherweise ist es normalerweise nicht erforderlich, diese doch recht unübersichtliche Datei von Hand zu bearbeiten. Die hier vorliegende Datei ist übrigens erheblich gekürzt worden.

Ohne Optionen erstellt `depmod` eine neue *modules.dep*-Datei, indem es vorher die Informationen über Abhängigkeiten bei allen vorhandenen Modulen erfragt und sammelt. Das kann unter Umständen ein paar Sekunden dauern.

Nützliche Optionen für `depmod` sind:

- `-n` Trockenlauf mit Ausgabe nach *stdout*
- `-A` Schnelldurchgang; es wird vor der Erstellung einer neuen *modules.dep*-Datei geprüft, ob es überhaupt Module gibt, die neuer als die Module in der bestehenden *modules.dep*-Datei sind.

Modulkonfigurationsdateien

modules.dep

Über diese Datei wurde bereits im Zusammenhang mit `depmod` berichtet. Das folgende Beispiel soll anhand eines Auszugs aus dieser Datei noch einmal demonstrieren, wie die Einträge in der Datei *modules.dep* aufgebaut sein müssen:

```
/lib/modules/2.6.38.3/kernel/sound/pci/ac97/snd-ac97-codec.ko:/lib/modules/
2.6.38.3/kernel/sound/core/snd-pcm.ko
```

Dieser exemplarische Auszug ist eine Kopie der ersten Zeilen aus dem vorangehenden Beispiel. Hier hängt das Modul *snd-ac97-codec.ko* von dem Modul *snd-pcm.ko* ab. Der Doppelpunkt stellt dabei den Zusammenhang her.

Das Verzeichnis /proc/sys/kernel

Zur Laufzeit legt der Kernel seine Konfigurationsinformationen im */proc*-Dateisystem ab. Sie finden hier etliche Informationen, indem Sie die in diesem Verzeichnis enthaltenen Dateien z. B. mit `cat` einsehen. Die meisten Dateinamen sind hierbei selbsterklärend. Hier ein paar Beispiele:

```
harald@archangel:/proc/sys/kernel$ cat osrelease
2.6.32-30-generic-pae
harald@archangel:/proc/sys/kernel$ cat ostype
Linux
harald@archangel:/proc/sys/kernel$ cat hostname
archangel
harald@archangel:/proc/sys/kernel$ cat modprobe
/sbin/modprobe
```

Wenn Sie Änderungen an diesen Dateien vornehmen, sollten Sie bedenken, dass diese bei einem Neustart des Systems verloren gehen, weil das */proc*-Dateisystem sich nicht auf der Festplatte eines Systems befindet, sondern lediglich Informationen abbildet, die sich im Arbeitsspeicher befinden.

Tools zur Analyse

dmesg

Mit dem Programm `dmesg` können Sie den Kernel Ring Buffer auslesen und steuern. Dieser Puffer ist Bestandteil des */proc*-Dateisystems und befindet sich in */proc/kmsg*. Entsprechend wird der Inhalt des Puffers beim Herunterfahren des Systems gelöscht. Der Kernel Ring Buffer beinhaltet alle Meldungen des Kernels seit dem Systemstart. Wie Sie bereits wissen, finden Sie diese Meldungen auch in den Protokolldateien

/var/log/messages und */var/log/syslog*, aber zur schnellen Diagnose ist das Programm `dmesg` besser geeignet. Ein typischer Aufruf dieses Programms erfolgt in Kombination mit `tail`. Beispiel:

```
root@arch-deb-book:/# dmesg|tail
[ 2990.306604] sd 7:0:0:0: Attached scsi generic sg2 type 0
[ 2990.307881] sd 7:0:0:0: [sdb] 15633408 512-byte logical blocks:
(8.00 GB/7.45 GiB)
[ 2990.309118] sd 7:0:0:0: [sdb] Write Protect is off
[ 2990.309127] sd 7:0:0:0: [sdb] Mode Sense: 43 00 00 00
[ 2990.309133] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[ 2990.314477] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[ 2990.314486]  sdb: sdb1
[ 2990.319849] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[ 2990.319857] sd 7:0:0:0: [sdb] Attached SCSI removable disk
[ 2991.020668] FAT: utf8 is not a recommended IO charset for FAT filesystems,
filesystem will be case sensitive!
```

Die Werte in den eckigen Klammern zeigen an, nach wie vielen Sekunden seit dem letzten Neustart des Systems ein Ereignis eingetreten ist. Das erste hier gelistete Ereignis ist also nach 2990 Sekunden eingetreten. Die Stellen nach dem Punkt sind übrigens Mikrosekunden. Wenn Sie die obigen Zeilen analysieren, werden Sie feststellen, dass an den Computer ein 8 GB großes USB-Speichermedium angeschlossen worden ist. Das Gerät verwendet nun die Gerätedatei */dev/sdb*. Diese kleine Diagnose könnte z. B. aus folgenden Gründen ausgeführt worden sein: Zum einen wissen Sie jetzt, dass das Gerät physikalisch funktioniert. Der Kernel hätte bei einer Fehlfunktion wahlweise eine entsprechende Fehlermeldung oder schlicht und ergreifend gar nichts ausgegeben. Außerdem können Sie jetzt mit Sicherheit davon ausgehen, dass das Gerät auf */dev/sdb* abgebildet worden ist, und können nun den richtigen Datenträger zur Partitionierung bzw. Formatierung gefahrlos auswählen.

Wenn Ihnen Fehlermeldungen beim Systemstart entgangen sind, können Sie einfach den gesamten Kernel Ring Buffer auslesen, indem Sie die Ausgabe von `dmesg` an das Programm `less` weiterleiten. So bekommen Sie sozusagen Informationen von der nullten Sekunde an:

```
root@arch-deb-book:/# dmesg|less
[    0.000000] Initializing cgroup subsys cpuset
[    0.000000] Initializing cgroup subsys cpu
[    0.000000] Linux version 2.6.32-5-686 (Debian 2.6.32-41)
(ben@decadent.org.uk) (gcc version 4.3.5 (Debian 4.3.5-4) )
#1 SMP Mon Jan 16 16:04:25 UTC 2012
[    0.000000] KERNEL supported cpus:
[    0.000000]   Intel GenuineIntel
```

```
[    0.000000]   AMD AuthenticAMD
[    0.000000]   NSC Geode by NSC
[    0.000000]   Cyrix CyrixInstead
[    0.000000]   Centaur CentaurHauls
[    0.000000]   Transmeta GenuineTMx86
```

Sie können `dmesg` außerdem mit den folgenden Optionen verwenden:

- `-c` löscht den Pufferinhalt, nachdem er ausgegeben worden ist.
- `-r` zeigt den Puffer roh an. Hierbei werden auch die Loglevel-Präfixe mit ausgegeben.
- `-s <bufsize>` legt die Größe des Puffers für die Anzeige fest. `16292` ist die Standardgröße. Wenn ein System schon mehrere Jahre lang läuft, müssen Sie eventuell einen höheren Wert einstellen, wenn Sie den kompletten Inhalt des Kernel Ring Buffers einsehen wollen.
- `-n <level>` legt fest, welches Level von Meldungen ausgegeben werden soll. Werte von `1` bis `8` sind möglich. Level `1` zeigt z. B. nur Panikmeldungen an.

/sbin/lspci

Das Programm `lspci` wird verwendet, um Informationen über den PCI-Bus eines Systems und die daran angeschlossenen Geräte anzuzeigen. Wenn Sie keine Optionen verwenden, erhalten Sie eine einfache Aufstellung über Ihren PCI-Bus und die angeschlossenen Geräte. Hierbei greift das Programm auf die Textdatenbank */usr/share/misc/pci.ids* zu. Wichtige und nützliche Optionen sind:

- `-v` zeigt umfangreichere Informationen an (verbose).
- `-vv` zeigt noch umfangreichere Informationen an (very verbose).
- `-vvv` zeigt die umfangreichsten Informationen an (very, very verbose).
- `-x` gibt alle Informationen als Hexdump aus. Auch diese Option kann gesteigert werden, und zwar mit `-xxx` oder `-xxxx`.
- `-k` zeigt an, welche Kernel-Treiber und Module ein Gerät steuern.
- `-m` oder `-mm` erstellt die Ausgabe in einer für Maschinen leichter lesbaren Form.

lsusb

Mit `lsusb` können Sie einen USB-Bus und die daran angeschlossenen Geräte untersuchen. Die Ausgabe des einfachen Kommandos ohne Optionen kann z. B. so aussehen:

```
root@arch-deb-book:/# lsusb
Bus 002 Device 004: ID 0781:5566 SanDisk Corp.
Bus 002 Device 003: ID 046d:c52f Logitech, Inc. Wireless Mouse M305
Bus 002 Device 002: ID 8087:0020 Intel Corp. Integrated Rate Matching Hub
```

```
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 001 Device 003: ID 13d3:5130 IMC Networks
Bus 001 Device 002: ID 8087:0020 Intel Corp. Integrated Rate Matching Hub
Bus 001 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
```

Ähnlich wie `lspci` schlägt das Programm die ermittelten Hersteller-IDs in einer Datei nach, nämlich in */var/lib/usbutils/usb.ids*. Es gibt auch für dieses Tool einen Verbose Mode, wenn Sie die Option `-v` verwenden. Es bleibt aber im Gegensatz zu `lspci` bei einer einzigen Stufe von Verbose.

Es gibt auch die Möglichkeit, die Hersteller-ID und die Produkt-ID eines Gerätes anzugeben, über das Sie genauere Informationen benötigen. Verwenden Sie hierfür die Option `-d` und trennen Sie Hersteller- und Produkt-ID durch einen Doppelpunkt voneinander:

```
root@archangel:/# lsusb -d 0c4b:0300
Bus 002 Device 002: ID 0c4b:0300 Reiner SCT Kartensysteme GmbH cyberJack pinpa
d(a)
```

/usr/bin/lsdev

Mit `lsdev` können Sie sich einen schnellen Überblick über die in einem Computer verwendete Hardware verschaffen. Das Programm bezieht seine Informationen direkt aus dem */proc*-Dateisystem. Hier liest es die Dateien *interrupts*, *ioports* und *dma*. Die gesammelten Informationen werden anschließend in einer Übersicht dargestellt. Da die Ausgabe des Programms vom Format her nicht buchtauglich ist, sollten Sie es einfach einmal selbst ausführen. Optionen oder Parameter nimmt dieses Programm nicht entgegen.

/bin/uname

Das Programm `uname` kann diverse Systeminformationen ausgeben. In der Hauptsache wird es allerdings verwendet, um schnell die laufende Kernel-Version zu ermitteln. Verwenden Sie hier die Option `-r`:

```
root@archangel:~# uname -r
3.0.0-16-generic-pae
```

Wenn Sie die Header-Files passend zu Ihrem laufenden Kernel nachinstallieren wollen, können Sie dieses Kommando nahtlos in eine Befehlszeile mit `apt-get` integrieren. Das Ergebnis sähe so aus:

```
root@archangel:/# apt-get install linux-headers-`uname -r`
```

Dieses Kommando können Sie verwenden, bevor Sie einen Treiber (z. B. für eine Grafikkarte) modular in den laufenden Kernel integrieren müssen.

Wenn Sie alle Informationen, die `uname` bereitstellen kann, anzeigen wollen, verwenden Sie die Option `-a`. Es werden dann alle Informationen in einer einzigen Zeile ausgegeben:

```
root@archangel:/# uname -a
Linux archangel 3.0.0-16-generic-pae #28-Ubuntu SMP Fri Jan 27 19:24:01
UTC 2012 i686 i686 i386 GNU/Linux
```

strace

Das Programm `strace` kann verwendet werden, um die Ausführung eines Kommandos zu analysieren. Es werden alle Systemaufrufe, die während der Programmausführung durchgeführt werden, angezeigt. Zusätzlich zeigt `strace` Signale an, die das zu untersuchende Programm während der Ausführung empfängt. Im einfachsten Fall führen Sie `strace` direkt gefolgt von dem zu analysierenden Kommando aus. Wenn Sie z. B. die Systemaufrufe, die zum Erstellen einer Datei verwendet werden, anzeigen wollen, können Sie folgendes Kommando verwenden:

```
root@arch-deb-book:/# strace touch testfile
```

Die Ausgabe diese Kommandos ist allerdings nicht sehr aufschlussreich, wenn Sie nicht gerade ein systemnah arbeitender Programmierer (z. B. Linus Torvalds) sind. Etwas besser wird es, wenn Sie den Zähler ins Spiel bringen. Sie können dann sehen, wie viele Systemaufrufe welches Typs verwendet werden, wenn mittels `touch testfile` eine neue Datei erstellt wird:

```
root@arch-deb-book:/# strace -c touch testfile
% time     seconds  usecs/call     calls    errors syscall
------ ----------- ----------- --------- --------- ----------------
  -nan    0.000000           0         3           read
  -nan    0.000000           0         6           open
  -nan    0.000000           0         9           close
  -nan    0.000000           0         1           execve
  -nan    0.000000           0         5         5 access
  -nan    0.000000           0         3           brk
  -nan    0.000000           0         1           dup2
  -nan    0.000000           0         1           munmap
  -nan    0.000000           0         1           uname
  -nan    0.000000           0         5           mprotect
  -nan    0.000000           0         2           rt_sigaction
  -nan    0.000000           0         1           rt_sigprocmask
  -nan    0.000000           0         1           getrlimit
  -nan    0.000000           0        13           mmap2
  -nan    0.000000           0         5           fstat64
  -nan    0.000000           0         2         1 futex
```

```
  -nan    0.000000    0       1           set_thread_area
  -nan    0.000000    0       1           set_tid_address
  -nan    0.000000    0       1           set_robust_list
  -nan    0.000000    0       1           utimensat
------ ----------- ---- ------- --------- ----------------
100.00    0.000000           63         6 total
```

Es haben also insgesamt 63 Systemaufrufe stattgefunden. Während der täglichen Administration kommen Sie mit `strace` eher nicht in Berührung.

strings

Wenn Sie eine Binärdatei mit einem Texteditor öffnen, um den Inhalt dieser Datei zu betrachten, bekommen Sie normalerweise nichts Verwertbares zu Gesicht. In einem solchen Fall können Sie `strings` einsetzen. Dieses Werkzeug gibt die druckbaren Zeichen eines Programms am Bildschirm aus. Sie können ein Ihnen unbekanntes Programm also zunächst untersuchen, wenn Sie z. B. nicht mehr wissen, was sein Verwendungszweck ist. Die Ausgabe dieser Kommandos ist immer umfangreich, weshalb Sie es einfach selbst ausprobieren sollten. Versuchen Sie z. B.:

```
root@arch-deb-book:/# strings /bin/bash
```

ltrace

Wenn Sie feststellen müssen, welche Bibliotheken ein Programm während seiner Ausführung aufruft, können Sie `ltrace` verwenden. Dieses Programm kann, abgesehen von seiner namensgebenden Aufgabe, auch Signale anzeigen, die ein Programm empfängt und die Systemaufrufe darstellen, die das Programm ausführt. Dadurch weist `ltrace` eine enorme Ähnlichkeit mit `strace` auf.

Ähnlich wie `strace` ist dieses Programm eher für Programmierer interessant und für das Tagesgeschäft eines Administrators ein Exot.

lsof

Mit `lsof` (list open files) können Sie geöffnete Dateien anzeigen. Hierbei gilt es zu bedenken, dass unter Linux alles entweder eine Datei oder ein Prozess ist. Sie können mit `lsof` also wesentlich mehr anzeigen, als das, was der Name zunächst vermuten lässt. Dazu gehören z. B. Unix-Sockets, IP-Sockets und NFS-Dateien. Es ist wenig sinnvoll, `lsof` ungefiltert auszuführen, weil dann einfach nur eine unübersichtliche große Liste angezeigt würde. Mit `wc` können Sie die Anzahl der Zeilen anzeigen, die `lsof` ausgeben würde:

```
root@arch-deb-book:/# lsof| wc -l
5480
```

In diesem Fall wären es also 5.480 Zeilen geworden. Eine sinnvolle Filterung bzw. Kombination mit anderen Programmen erscheint also angebracht.

`lsof` kann z. B. sehr nützlich sein, wenn ein beschäftigter Datenträger sich mit `umount` nicht aushängen lässt, weil noch Dateien im Hintergrund geöffnet sind. Sie können die zugreifenden Prozesse ermitteln und diese dann mit `kill` beenden. Danach lässt sich der Datenträger aushängen. Im folgenden Beispiel lässt sich die USB-Festplatte nicht aushängen, die unter */media/disk-1* eingehängt ist:

```
root@archangel:~# umount /media/disk-1
umount: /media/disk-1: Das Gerät wird momentan noch benutzt
```

Als Erstes muss jetzt festgestellt werden, welche Prozesse auf das Laufwerk zugreifen. Dann kann auch eine Entscheidung darüber getroffen werden, ob es sich um einen kritischen Prozess handelt, der durch gewaltsames Beenden Datenverluste verursachen könnte:

```
root@archangel:~# lsof /media/disk-1
COMMAND   PID USER   FD   TYPE DEVICE SIZE    NODE          NAME
bash    13601 root  cwd    DIR   8,33 4096 2244609 /media/disk-1
```

Das sieht ungefährlich aus. Es wird auf keine Dateien zugegriffen. Lediglich eine Shell (`bash`) mit der Prozess-ID (PID) `13601` hält ein Verzeichnis (`DIR`) geöffnet. Es besteht also kein Anlass zur Sorge, wenn Sie diesen Zugriff gewaltsam unterbinden:

```
root@archangel:~# kill -s 9 13601
```

Bei einer wiederholten Abfrage mit `lsof` wird es jetzt keine Antwort mehr geben, und der Datenträger kann ausgehängt werden.

Die Anzeige von `lsof` kann übrigens sehr gut gefiltert werden. Für das vorangehende Beispiel hätte als Ausgabe die PID gereicht. Besonders dann, wenn viele Dateien geöffnet gewesen wären, hätten Sie leicht folgendes Konstrukt realisieren können:

```
root@archangel:~# kill -s 9 $(lsof -t /media/disk-1)
```

Es wären dann alle Prozesse, die auf */media/disk-1* zugegriffen hätten, auf einen Schlag beendet worden. Was die Überprüfung von Netzwerkdiensten anbelangt, kann `lsof` auch wertvolle Dienste leisten. Sie können nämlich mit der Option `-i` sehr leicht feststellen, welche Prozesse Verbindungen zum Netzwerk herstellen:

```
root@archangel:~# lsof -i :80
COMMAND   PID     USER   FD   TYPE DEVICE SIZE NODE NAME
apache2  7605     root   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 11047 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 11048 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
```

```
apache2 11049 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 11050 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 11051 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 13968 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 13973 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 13973 www-data   8u  IPv4  70619       TCP
. archangel.homelinux.net: www-> 24.215.7.162:49668 (ESTABLISHED)
apache2 13974 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 13975 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
apache2 13977 www-data   3u  IPv4  23571       TCP *:www (LISTEN)
```

An Port 80 lauscht also ein Apache-Webserver. Die meisten Abhörer (*Listener*) warten auf eingehende Verbindungen (`LISTEN`). Es gibt eine eingehende Verbindung, die aktiv von einem Computer mit der Adresse 24.215.7.162 genutzt wird (`ESTABLISHED`).

udev (userspace /dev)

Der Daemon *udev* ist für die Verwaltung der Gerätedateien unterhalb von */dev* auf einem Linux-System zuständig. *udev* löst das System *devfs* ab, wodurch einige Probleme, die es mit *devfs* gab, der Vergangenheit angehören. *devfs* war, im Gegensatz zu *udev*, ein fester Bestandteil des Kernels, woraus sich einige Nachteile ergaben:

- Weil das Verzeichnis */dev* unter *devfs* statisch war, war es auch unhandlich und groß.
- Durch den wachsenden Bedarf an Geräten gingen mit dem statischen *devfs* langsam die fortlaufenden Gerätenummern aus.
- Wenn USB-Geräte an einen anderen Port oder eine SCSI-Festplatte an einen anderen Zweig des SCSI-Baums angeschlossen wurden, änderten sich auch die zugehörigen Gerätedateien.
- Ein Programm konnte nicht ohne weiteres feststellen, ob eine zugehörige Hardwarekomponente im System eingesteckt war oder nicht, weil die Gerätedateien nach dem Entfernen einer Komponente blieben, wo sie waren.
- Übrigens ging *devfs* nicht konform mit der LSB (Linux Standard Base).

All diese Schwierigkeiten sind mit der Einführung von *udev* auf einen Schlag beseitigt worden. Ab Kernel-Version 2.6 ist *devfs* endgültig aus dem Kernel verschwunden.

udev konfigurieren

Weil *udev* im User-Land ausgeführt wird, ist es problemlos zu konfigurieren. Für die Konfiguration relevante Dateien befinden sich im Verzeichnis */etc/udev* und die Regeln für *udev* in */etc/udev/rules.d*. Theoretisch gesehen ist */etc/udev/udev.conf* die

Hauptkonfigurationsdatei von *udev*. In der Praxis findet man hier lediglich eine Angabe, die dazu dient, den Protokollierungsgrad von *udev* anzupassen:

```
udev_log="err"
```

Es ist möglich, hier das Basisverzeichnis, in dem sich die Gerätedateien befinden (*/dev*), zu ändern, z. B. mit:

```
udev_root=/geraete
```

Das hätte allerdings zur Folge, dass die Geräte nicht mehr gefunden werden. Außerdem kann hier der Standardpfad zu den Regeln geändert werden. Zu diesem Zweck gibt es die Option `udev_rules`. Aber auch hier stellt sich die Frage der Sinnhaftigkeit.

Interessanter ist da der Inhalt von */etc/udev/rules.d*. Hier befinden sich die Regeln für *udev*, die jeweils die Dateierweiterung *.rules* aufweisen. Die Regeln werden in lexikalischer Reihenfolge abgearbeitet. Ähnlich wie bei den *init*-Skripten werden hier numerische Präfixe verwendet, damit man die tatsächliche Verarbeitungsreihenfolge leichter steuern kann.

Ein typisches Beispiel für die Praxis ist die Zuordnung von Netzwerkkarten zu ihren Aliasnamen. Sie finden die Regeln für diese Zuordnungen normalerweise in der Datei *70-persistent-net.rules*. Wenn Sie bei einem Computer eine defekte Netzwerkkarte austauschen, werden Sie schnell feststellen, dass die neue Netzwerkkarte nicht denselben Aliasnamen verwendet wie die ursprüngliche. Das kann eine Menge Nacharbeiten zur Folge haben, wenn Sie jetzt z. B. die Firewall-Konfiguration an die neuen Gegebenheiten anpassen müssen (weil z. B. aus eth0 = eth1 geworden ist). Da ist es natürlich erheblich einfacher, die Regel in *udev* anzupassen. Sie sehen hier den relevanten Ausschnitt einer Regeldatei, nachdem eine Realtek-Netzwerkkarte gegen eine Intel-e1000-Karte ausgetauscht worden ist:

```
# PCI device 0x10ec:0x8139 (8139too)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}==
"00:a1:b0:f0:ad:dd", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL==
"eth*", NAME="eth0"

# PCI device 0x8086:0x1013 (e1000)
SUBSYSTEM=="net", ACTION=="add", DRIVERS=="?*", ATTR{address}==
"00:0d:60:66:86:fd", ATTR{dev_id}=="0x0", ATTR{type}=="1", KERNEL==
"eth*", NAME="eth1"
```

Sie müssen nicht die ganze Regel verstehen, um zu sehen, dass der Alias einer Netzwerkkarte an die MAC-Adresse gebunden wird. Die relevanten Teile der Regeln sind hervorgehoben. Um das Problem zu lösen, muss also lediglich die obere Regel gelöscht und in der zweiten Regel `eth1` in `eth0` geändert werden. Ein Neustart des Computers ist nicht erforderlich; es genügt die Eingabe von:

```
root@archangel:/etc/udev/rules.d# /etc/init.d/udev restart
```

Regeländerungen erfordern niemals einen Systemneustart. Es reicht aus, den *udev*-Daemon neu zu starten.

udev überwachen

Zur Überwachung von *udev* können Sie den `udevmonitor` verwenden. Der Name dieses Werkzeugs ist allerdings nur noch für die Prüfung von Belang. Wenn Sie nicht gerade eine sehr alte Linux-Distribution verwenden, werden Sie den `udevmonitor` vergeblich suchen. Er wurde nämlich in das wesentlich universeller einsetzbare `udevadm` integriert. Dieses Tool ist sehr interessant. Sie können damit live zusehen, wie *udev* arbeitet. Geben Sie einfach folgendes Kommando ein:

```
root@arch-deb-book:/# udevadm monitor
monitor will print the received events for:
UDEV - the event which udev sends out after rule processing
KERNEL - the kernel ueven
```

Wie das Programm schon selbst sagt, wird es Ereignisse anzeigen, die bei der Abarbeitung von *udev*-Regeln auftreten und Kernel-Meldungen, die mit *udev* im Zusammenhang stehen. Wenn Sie *udev* in Aktion sehen wollen, brauchen Sie jetzt nur ein USB- oder PCMCIA-Gerät an Ihren Computer zu stecken oder es zu entfernen. Wenn Sie ein Notebook verwenden, können Sie auch das WLAN ausschalten:

```
KERNEL[1304251240.465648] change   /devices/pci0000:00/0000:00:1c.0/
0000:02:00.0/ieee80211/phy0/rfkill1 (rfkill)
UDEV  [1304251240.466555] change   /devices/pci0000:00/0000:00:1c.0/
0000:02:00.0/ieee80211/phy0/rfkill1 (rfkill)
KERNEL[1304251240.517226] remove   /devices/pci0000:00/0000:00:1c.0/
0000:02:00.0/leds/iwl-phy0::assoc (leds)
KERNEL[1304251240.517302] remove   /devices/pci0000:00/0000:00:1c.0/
0000:02:00.0/leds/iwl-phy0::RX (leds)
```

sysctl

Das Werkzeug `sysctl` wird verwendet, wenn dem Kernel zur Laufzeit Parameter übergeben werden müssen. Alternativ können diese Parameter auch direkt in die entsprechende Datei des */proc*-Dateisystems geschrieben werden. Das ist allerdings in der Regel aufgrund langer Pfadnamen umständlicher. Um das IP-Forwarding für die Protokolle IPv4 und IPv6 einzuschalten, können Sie `sysctl` mit der Option `-w` (write) verwenden:

```
root@archangel:~# sysctl -w net.ipv6.conf.all.forwarding=1
net.ipv6.conf.all.forwarding = 1
root@archangel:~# sysctl -w net.ipv4.ip_forward=1
net.ipv4.ip_forward = 1
```

Die Deaktivierung geht genauso; allerdings wird dann anstatt der `1` jeweils eine `0` als Parameter übergeben.

Ohne das Programm `sysctl` zu verwenden, hätten Sie die `1` jeweils mit folgenden Kommandos an den richtigen Stellen im Dateisystem platziert:

```
echo 1 > /proc/sys/net/ipv4/ip_forward
echo 1 > /proc/sys/net/ipv6/conf/all/forwarding
```

Parameter, die direkt übergeben werden, sind flüchtig. Nach einem Neustart des Systems müssen diese erneut übergeben werden. Damit Parameter dauerhaft eingestellt werden, können Sie diese in die Konfigurationsdatei */etc/sysctl.conf* eintragen. Alternativ ist es möglich, im Verzeichnis */etc/sysctl.d* Dateien mit der Erweiterung *.conf* zu erstellen und hier entsprechende Einträge vorzunehmen. Die folgenden beiden Einträge in dieser Datei schalten schon während des Systemstarts das IP-Forwarding für die Protokolle IPv4 und IPv6 ein:

```
net.ipv4.ip_forward=1
net.ipv6.conf.all.forwarding=1
```

Analyse der Protokolldateien

Eine wichtige Anlaufstelle bei der Diagnose sind die Protokolldateien. Sie finden die Protokolle fast aller Programme, die Protokolldateien erzeugen, im Verzeichnis */var/log*. Besonders wichtig, sowohl für die Praxis als auch für die Prüfung, sind die Dateien */var/log/messages* und */var/log/syslog*. Beachten Sie jedoch, dass nicht alle Linux-Distributionen die Datei *syslog* verwenden, während die Datei *messages* zumindest zusätzlich auf jedem Linux-Derivat zu finden ist. Bei der Diagnose ist es sinnvoll, die Dateien jeweils mithilfe von `grep` zu filtern, sodass nur die Einträge angezeigt werden, die für eine aktuelle Analyse relevant sind. Wenn Sie zum Beispiel nur kernel-bezogene Meldungen überprüfen wollen, können Sie dieses Kommando verwenden:

```
root@arch-deb-book:/# grep kernel /var/log/messages
```

Sie erhalten dann auch noch einmal Einblick in die Meldungen, die Sie beim Systemstart nicht lesen konnten, weil diese Meldungen den Bildschirm zu schnell passiert haben.

Die meisten Daemons führen ebenfalls Buch in Protokolldateien. Einige legen eigene Protokolldateien oder gar ganze Protokollunterverzeichnisse (dann meist ebenfalls unterhalb von */var/log)* an. Wenn Sie herausfinden wollen, wohin ein neu installiertes Programm protokolliert, können Sie den Inhalt von */var/log* oder die jeweiligen Konfigurationsdateien einsehen. Bei einigen standardisierten Programmen gibt auch die Konfigurationsdatei des Syslog (*/etc/syslog.conf*) Aufschluss.

Bei Daemons, die den normalen Messagelog verwenden, bietet sich natürlich ebenfalls eine Filterung durch `grep` an, z. B. für den Daemon *dhcpd*:

```
root@archangel:~# grep dhcp /var/log/syslog
Mar 29 08:47:25 archangel dhcpd: DHCPREQUEST for 192.168.50.139
from 00:25:d3:e5:e2:54 via eth1
Mar 29 08:47:25 archangel dhcpd: DHCPACK on 192.168.50.139
to 00:25:d3:e5:e2:54 via eth1
Mar 29 09:31:19 archangel dhcpd: DHCPDISCOVER from e0:b9:a5:7f:0e:7b via eth1
```

Einige häufig vorkommende Protokolldateien und Verzeichnisse bekannter Daemons sind:

- */var/log/apache/access.log* – Apache Webserver
- */var/log/squid/access.log* – Squid-Proxy
- */var/log/samba/** – Samba-Server
- */var/log/auth.log* – diverse Authentifizierungsanbieter (z. B. PAM)
- */var/log/mail.** – sämtliche Mailserverdienste (z. B. SMTP, IMAP, POP)

In Abhängigkeit von den installierten Programmen gibt es natürlich noch erheblich mehr Protokolldateien.

Analyse am /proc-Dateisystem

Beim */proc*-Dateisystem handelt es sich um ein Pseudodateisystem, das sich, physikalisch gesehen, im Arbeitsspeicher eines Computers befindet. Dieses Pseudodateisystem dient als Schnittstelle zum Kernel. In den Unterverzeichnissen und Dateien von */proc* werden Prozesse und Kernel-Strukturen abgebildet.

Auf höchster Ebene finden Sie je ein Verzeichnis für jeden laufenden Prozess im Verzeichnisbaum von */proc*. Der Name des jeweiligen Verzeichnisses entspricht immer der Prozess-ID (PID) des jeweiligen Prozesses. Unterhalb dieser Verzeichnisse befinden sich jeweils mehrere Unterverzeichnisse und Dateien. Einige dieser Dateien sind nullterminiert, was zu unübersichtlichen Darstellungen führt, wenn Sie solche Dateien einfach mittels `cat` ausgeben. Das folgende Beispiel demonstriert die Ausgabe der Prozessumgebung für einen `getty`-Prozess, hier mit der PID 2352:

```
root@arch-deb-book:/# cat /proc/2352/environ
HOME=/init=/sbin/initTERM=linuxBOOT_IMAGE=/boot/vmlinuz-2.6.32-5-686PATH=
/sbin:/usr/sbin:/bin:/usr/binPWD=/rootmnt=/rootSHELL=/bin/shRUNLEVEL=
2PREVLEVEL=NCONSOLE=/dev/consoleINIT_VERSION=
sysvinit-2.88root@arch-deb-book:/#
```

Selbst der neue Prompt wurde mit in die Ausgabe des Kommandos gesetzt. Die Ausgabe des Kommandos wird übersichtlich und verwendbar, wenn die Nullterminierung in einen Zeilenvorschub übersetzt wird. Zu diesem Zweck wird die Ausgabe an das Programm `tr` weitergeleitet:

```
root@arch-deb-book:/# cat /proc/2352/environ|tr "\000" "\n"
HOME=/
init=/sbin/init
TERM=linux
BOOT_IMAGE=/boot/vmlinuz-2.6.32-5-686
PATH=/sbin:/usr/sbin:/bin:/usr/bin
PWD=/
rootmnt=/root
SHELL=/bin/sh
RUNLEVEL=2
PREVLEVEL=N
CONSOLE=/dev/console
INIT_VERSION=sysvinit-2.88
```

Es gibt viele Informationen, die Sie zu jedem einzelnen Prozess abfragen können, wie z. B. Prozesspriorität, Startzeit (nach Systemstart), Prozessstatus, Speicherbelegung usw. Aber die Aufführung all dieser Dateien würde den Rahmen des Kapitels sprengen. Einige wichtige Dateien direkt unterhalb von */proc* sind:

- *cpuinfo* enthält Informationen über den verwendeten Prozessor.
- *dma* zeigt die verwendeten DMA-Kanäle an.
- *interrupts* gibt die verwendeten Interrupts (IRQs) aus.
- *ioports* zeigt die aktuell registrierten Ein- und Ausgabeports an.
- *meminfo* gibt detailliert Auskunft über die Speicherverwendung des Systems.
- *modules* enthält eine Liste der geladenen Kernel-Module. Hier holt `lsmod` seine Informationen ab.

Sie sollten sich den Inhalt dieser Pseudodateien einfach einmal auf Ihrem System ansehen. Die Dateien sind nicht nullterminiert, weshalb Sie einfach, z. B. mit einem solchen Kommando, sondiert werden können:

```
root@arch-deb-book:/# cat /proc/cpuinfo
```

Wichtige Unterverzeichnisse von */proc* sind z. B. */proc/sys/kernel* und */proc/sys/net*. Diese Verzeichnisse sind Ihnen im Laufe dieses Buchs schon begegnet. Sie sollten sich aber im Klaren darüber sein, dass Änderungen an den Dateien in diesen Unterverzeichnissen (und natürlich innerhalb von */proc* im Allgemeinen) flüchtig sind und einen Systemneustart nicht überdauern. Wenn Sie dauerhaft Änderungen durchführen wollen, erstellen Sie bitte entsprechende Einträge in der Konfigurationsdatei */etc/sysctl.conf*.

202 Systemstart

Nachdem der Kernel in den Speicher geladen und ausgeführt worden ist, kann der Startvorgang fortgesetzt werden. Ein init-System übernimmt ab jetzt das Kommando und koordiniert das Starten und Beenden der Prozesse.

202.1 Anpassen des Systemstarts

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, das Verhalten von Systemdiensten in verschiedenen Runleveln abzufragen und zu ändern. Dazu wird ein gründliches Verständnis der Struktur von *systemd*, des *SysVinit*-Systems und des Bootprozesses vorausgesetzt. Dieses Lernziel beinhaltet des Weiteren den Umgang mit Runleveln.

Wichtigste Wissensgebiete:

- *systemd*
- *SysVinit*
- Spezifikationen der *Linux Standard Base (LSB)*

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */usr/lib/systemd/*
- */etc/systemd/*
- */run/systemd/*
- `systemctl`
- `systemd-delta`
- */etc/inittab*
- */etc/init.d/*
- */etc/rc.d/*
- `chkconfig`
- `update-rc.d`
- `init` und `telinit`

Allgemeines

Bei Linux ist es nicht nur möglich, das Betriebssystem zu starten und am Ende des Tages wieder herunterzufahren. Es gibt hier im Gegensatz zu anderen Betriebssystemen mehrere Abstufungen, die als Runlevel bezeichnet werden. Für den Wechsel von einem Runlevel in ein anderes ist, zumindest bei *SysVinit* basierten Systemen, der Prozess `init` zuständig. Es werden hier zunächst der Ablauf und die Konfiguration des in die Tage gekommenen *SysVinit* und später von *systemd* beschrieben.

Da *init* nach dem Laden des Kernels der erste Prozess ist, der startet, verwendet er die PID 1. Beim Herunterfahren des Betriebssystems ist *init* übrigens der letzte Prozess, der sich beendet. Im folgenden Beispiel, das die Ausgabe des `ps -A`-Kommandos wiedergibt, erkennt man deutlich, wo der *init*-Prozess angesiedelt ist:

```
archangel:~ # ps -A
  PID TTY          TIME CMD
    1 ?        00:00:00 init
    2 ?        00:00:00 migration/0
    3 ?        00:00:00 ksoftirqd/0
    4 ?        00:00:00 migration/1
    5 ?        00:00:00 ksoftirqd/1
    6 ?        00:00:00 events/0
```

Wenn Sie das Kommando `pstree` ausführen, repräsentiert `init` die Wurzel des Prozessbaums.

SysVinit

Runlevel und ihre Funktion

Runlevel sind sozusagen Zustände oder Funktionsstufen, in denen sich ein Linux-basiertes System befinden kann. Leider sind nicht alle Runlevel durch eine genaue Normierung festgelegt. Man muss also ggf. nachlesen, wie die Distribution, die man gerade verwendet, die Runlevel organisiert. Offiziell gestaltet sich das Ganze folgendermaßen:

- Runlevel 0 ist bei allen Distributionen gleich definiert. Wenn der Computer in dieses Runlevel eintritt, schaltet er sich aus – vorausgesetzt natürlich, das BIOS unterstützt eine automatische Abschaltung des Gerätes.
- Runlevel 1 entspricht Runlevel s oder auch Runlevel S. Dies gilt ebenfalls für alle Distributionen in gleicher Weise. Dieser Status wird auch als Single User Mode bezeichnet, weil Linux hier nur rudimentäre Systemfunktionen zur Verfügung stellt. In dieser Stufe gibt es noch keine Netzwerkfunktionalität und keine Multi-User-Unterstützung (daher die Bezeichnung Single User Mode). Runlevel 1 sollte nur zur Wartung und Administration verwendet werden.

- Runlevel 2 ist nicht in allen Distributionen gleich definiert. In der Regel kann man zumindest davon ausgehen, dass es in Runlevel 2 keine Unterstützung für eine grafische Anmeldung gibt. In jedem Fall gibt es in dieser Stufe eine Multi-User-Funktionalität und bei manchen Distributionen wird hier das Netzwerk gestartet.
- Runlevel 3 ist ebenfalls nicht bei allen Distributionen gleich konfiguriert. Multi-User-Funktionalität ist selbstverständlich vorhanden. Das Netzwerk läuft spätestens in diesem Status bei allen bekannten Distributionen. Unterschiede bestehen lediglich, distributionsspezifisch gesehen, darin, ob zu diesem Zeitpunkt eine grafische Anmeldung zur Verfügung steht oder nicht.
- Runlevel 4 wird von keiner bekannten Distribution genutzt.
- Runlevel 5 ist das höchste funktionale Runlevel und bietet in der Regel Netzwerk, Multi-User-Unterstützung und eine grafische Anmeldung, wenn eine solche installiert ist.
- Runlevel 6 ist dagegen distributionsübergreifend festgelegt. Wenn man dieses Runlevel startet, fährt `init` das System herunter und startet es anschließend neu.

Die Konfigurationsdatei inittab

Die Datei */etc/inittab* ist die Hauptkonfigurationsdatei für den `init`-Prozess, dessen Programmdatei sich übrigens im */sbin*-Verzeichnis befindet. Hier kann beispielsweise festgelegt werden, in welches Runlevel Linux standardmäßig bootet. So kann es gerade für Sie interessant sein, zu verhindern, dass der Rechner jedes Mal bis zur grafischen Anmeldung hochfährt, wenn Sie lediglich ein paar Kommandos testen wollen, für die man nur eine Konsole benötigt. Suchen Sie also in */etc/inittab* folgende Zeilen:

```
# The default runlevel is defined here
id:3:initdefault:
```

Im gezeigten Beispiel ist Runlevel 3 als Standardwert eingestellt. Wenn man diesen Wert ändert, ist er beim nächsten Neustart wirksam. Die Werte 0 oder 6 sind hier demzufolge nicht gerade empfehlenswert. Ein paar Zeilen weiter in der Datei *inittab* können Sie nachverfolgen, wie die Runlevel in Ihrer Distribution organisiert sind:

```
# runlevel 0  is  System halt
# runlevel 1  is  Single user mode
# runlevel 2  is  Local multiuser without remote network
# runlevel 3  is  Full multiuser with network
# runlevel 4  is  Not used
# runlevel 5  is  Full multiuser with network and xdm
# runlevel 6  is  System reboot
```

Die meisten Linux-Distributionen stellen von sich aus sechs Konsolen bereit, auf denen man gleichzeitig arbeiten kann. Es ist aber auch möglich, weitere Konsolen hinzuzufügen oder zu entfernen. Achten Sie aber darauf, nicht die Konsole zu belegen, die Ihr X Window System verwendet:

```
1:2345:respawn:/sbin/mingetty --noclear tty1
2:2345:respawn:/sbin/mingetty tty2
3:2345:respawn:/sbin/mingetty tty3
4:2345:respawn:/sbin/mingetty tty4
5:2345:respawn:/sbin/mingetty tty5
6:2345:respawn:/sbin/mingetty tty6
```

Die erste Ziffer jeder Zeile ist die ID der zu startenden Konsole. Sie muss mit der Nummer des verwendeten Terminals übereinstimmen. Nach dem ersten Doppelpunkt folgen die Runlevel, in denen die jeweiligen Konsolen verfügbar sein sollen. Runlevel 1 ist nicht vertreten, weil dieses an anderer Stelle konfiguriert wird und auch nur eine einzige Konsole zulässt. Die Anweisung `respawn` ist ein Kunstwort und könnte wohl am ehesten mit »wieder hervorbringen« übersetzt werden. Eine Konsole, die beendet wurde, wird hierdurch automatisch wieder neu gestartet. Das passiert vor allem dann, wenn `mingetty` eine Benutzeranmeldung an das Programm */bin/login* übergibt und der Benutzer dann ein falsches Passwort eingibt. Nach einer Benutzerabmeldung durch das Kommando `logout` muss ebenfalls eine neue Instanz von `mingetty` erzeugt werden.

`/sbin/mingetty` ist eines der wichtigsten Konsolenprogramme. TTY ist eine sehr alte Abkürzung, die für Teletypewriter steht.

Die letzte interessante und außerdem prüfungsrelevante Option innerhalb der Datei *inittab* ist:

```
# what to do when CTRL-ALT-DEL is pressed
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

Hier wird festgelegt, was passieren soll, wenn jemand den sogenannten Affengriff ausführt. In diesem Fall wird der Computer nach genau vier Sekunden (`-t 4`) neu gestartet (`-r`). Die genaue Verwendung des Befehls `shutdown` wird weiter unten erläutert. Vorher wird eine Konsolenmeldung an alle angemeldeten Benutzer gesendet. Da es sich um eine Konsolenmeldung handelt, werden natürlich keine Windows-Benutzer, die z. B. auf einen Samba-Server zugreifen, benachrichtigt. Sollten Sie den Computer immer allein verwenden, spricht nichts dagegen, die `shutdown`-Anweisung gegen `init 0` oder `init 6` zu tauschen. Damit eine solche Änderung sofort wirksam wird und nicht erst nach einem Neustart des Systems, sollten Sie anschließend `telinit q` ausführen. Merken Sie sich bitte auch für die Prüfung, dass `telinit q` den `init`-Prozess veranlasst, im laufenden Betrieb die Datei *inittab* neu einzulesen.

Verzeichnisse und Dateien des init-Prozesses

Abgesehen von der Datei */etc/inittab* gibt es noch einige andere Dateien und Verzeichnisse, die der `init`-Prozess ausliest und verwendet. Hier gibt es allerdings in Abhängigkeit von der verwendeten Distribution zum Teil erhebliche Unterschiede. Das macht sich vor allem in der Platzierung von Skripten bemerkbar, die beim Runlevel-Wechsel involviert sind. So arbeiten Red Hat-basierte Systeme meist mit einem monolithischen Initialisierungsskript, während die meisten anderen Distributionen kleine Einzelskripte verwenden. Das Basisverzeichnis ist aber bei den meisten Systemen leicht auffindbar, weil durch entsprechende Softlinks für eine Art Redundanz gesorgt wird. So heißt dieses Basisverzeichnis typischerweise */etc/init.d* bzw. */etc/rc.d*. Hier befinden sich bei den meisten Distributionen die einzelnen `init`-Skripte. Man kann diese natürlich auch im laufenden Betrieb von Hand ausführen. So kann man mit den folgenden Befehlszeilen z. B. `sendmail` beenden:

```
/etc/init.d/sendmail stop
```

oder:

```
/etc/rc.d/sendmail stop
```

Ähnlich sieht es aus, wenn man `sendmail` wieder starten möchte:

```
/etc/init.d/sendmail start
```

oder:

```
/etc/rc.d/sendmail start
```

Schon nimmt `sendmail` seine Arbeit wieder auf. Welche Übergabeparameter ein solches Skript versteht, kann man im Skript selbst nachsehen. Alle diese Skripte verstehen, so wie in unserem Beispiel, `start` und `stop`. Einige verstehen auch noch `status` oder `reload`. Mit `status` kann man nachprüfen, ob der zugehörige Daemon läuft oder nicht. Die `reload`-Anweisung dient dazu, einem Programm ein SIGHUP zu senden und es somit zu veranlassen, seine Konfigurationsdateien neu einzulesen. Das kann hilfreich sein, wenn man diese Dateien verändert hat und das Programm diese einlesen soll, ohne vorübergehend seinen Dienst einzustellen. Es lohnt sich durchaus, sich mit einem solchen Skript näher auseinanderzusetzen, auch wenn man es nicht lesen kann wie eine Geschichte. Man kann auch ohne Programmierkenntnisse ein paar Informationen finden, die zum Verständnis beitragen, wie die Skripte verarbeitet werden. Als Beispiel dienen hier ein paar Fragmente aus dem Startskript von `apache2`:

```
### BEGIN INIT INFO
# Provides:                    apache2 httpd2
# Required-Start:              $local_fs $remote_fs $network
```

```
# X-UnitedLinux-Should-Start:  $named $time postgresql sendmail mysql ypclient
dhcp radiusd
# Required-Stop:               $local_fs $remote_fs $network
# X-UnitedLinux-Should-Stop:
# Default-Start:               3 5
# Default-Stop:                0 1 2 6
# Short-Description:           Apache2 httpd
# Description:                 Start the httpd daemon Apache 2
### END INIT INFO
```

Die wichtigsten Informationen, die Sie diesem Startskript entnehmen können, sind u. a. die Angaben darüber, in welchem Runlevel Apache überhaupt laufen soll. `Default-Start` sagt, dass Apache in Runlevel 3 und 5 laufen soll. In Runlevel 0, 1, 2 und 6 läuft er laut `Default-Stop` standardmäßig nicht. Runlevel 4 wird gar nicht erst erwähnt, aber das ist wohl auch verständlich. Im oberen Teil des Skripts erkennt man (hinter `Required-Start`), welche anderen Dienste Apache voraussetzt. Natürlich hängen diese ihrerseits von weiteren Programmen ab. In diesem Zusammenhang interessiert es darüber hinaus, welche Parameter das Startskript von Apache versteht. Dazu muss man wissen, dass der erste Parameter eines Shell-Skripts immer in der Variablen `$1` hinterlegt wird. Also sucht man zunächst nach der Variablen `$1` innerhalb des Skripts:

```
action="$1"
```

Offensichtlich übergibt das Skript die Eingabevariable an eine andere Variable namens `action`. Also muss nun nach dieser Variablen und eventuell nach einer Auswahlanweisung gesucht werden. Die Suche wird die folgenden beiden Fundstellen ergeben:

```
case "$action" in
    stop|try-restart|*status*|probe)
```

und:

```
case "$action" in
    start*)
```

Das ist das Ergebnis. An */etc/rc.d/apache2* kann man die Parameter `stop`, `start`, `try-restart`, `status` und `probe` anhängen. Dieses Beispiel war im Übrigen schon vergleichsweise kompliziert, weil die meisten anderen Skripte ohne Umweg mit der Variablen `$1` arbeiten.

Nachdem Sie jetzt wissen, wie diese Skripte manuell bedient werden, kann man dieses Vorgehen auch auf die Arbeitsweise von `init` übertragen.

Bei den meisten Distributionen liegen unterhalb des Verzeichnisses */etc/init.d* bzw. */etc/rc.d* Unterverzeichnisse, die den jeweiligen Runleveln zugeordnet sind. Diese heißen */etc/init.d/rc1.d*, */etc/init.d/rc2.d* usw. Bei anderen Distributionen liegen diese Verzeichnisse durchaus auch direkt unterhalb von */etc*. In diesen Verzeichnissen befinden sich Softlinks, die auf die jeweiligen Startskripte zeigen, die in diesem Runlevel gestartet werden sollen. Diese Softlinks sehen in Runlevel 3 beispielsweise so aus:

```
K04apache2
S13named
K05sendmail
S18apache2
```

In Wirklichkeit liegen hier etwa 60 Softlinks, aber vier Softlinks sollen zur Veranschaulichung genügen. Beim Eintreten in ein Runlevel führt `init` alle Skripte aus, deren Softlink ein *S*, wie »start«, vorangestellt wurde. Durch die folgende zweistellige Nummerierung weiß `init`, in welcher Reihenfolge er die Skripte abarbeiten muss. Beim Verlassen des Runlevels arbeitet er alle Skripte ab, deren Softlink ein *K*, wie »kill«, vorangestellt wurde. Auch hier gibt die zweistellige Nummerierung einen Hinweis auf die Verarbeitungsreihenfolge.

Die letzte Frage, die sich noch stellt, ist, wieso ein und dasselbe Skript, das durch zwei unterschiedliche Softlinks aktiviert worden ist, einmal einen Daemon startet und diesen ein anderes Mal beendet:

```
K04apache2
S18apache2
```

Beide Softlinks zeigen nämlich auf */etc/rc.d/apache2*. Doch die Lösung ist ganz einfach: Jedes Skript kann durch die Abfrage einer Variablen, nämlich `$0`, feststellen, ob es durch Aufruf seines eigenen Dateinamens oder durch den Aufruf eines Links gestartet worden ist und dementsprechend mit Verzweigungen innerhalb des Programms reagieren.

chkconfig und update-rc.d

Damit Sie die Softlinks zum Starten bzw. Beenden der Startskripts nicht in mühseliger Kleinarbeit von Hand erstellen müssen, können Sie in Abhängigkeit von der verwendeten Distribution die Tools `chkconfig` oder `update-rc.d` verwenden. Beide sind dazu in der Lage, für ein angegebenes Skript zu ermitteln, in welchem Runlevel es gestartet oder beendet werden muss und welche Abhängigkeiten hierbei zu erfüllen sind. Auf Red Hat-artigen Systemen finden Sie normalerweise `chkconfig` vor, während bei Debian-Derivaten eher `update-rc.d` verwendet wird.

Die wichtigsten Optionen für `chkconfig` sind:

- `--add` fügt dem Management von `chkconfig` einen neuen Service hinzu; Softlinks für relevante Runlevel werden ggf. erzeugt.
- `--del` entfernt einen Service aus dem Management von `chkconfig`; vorhandene Softlinks in den Runlevel-Verzeichnissen werden gelöscht.
- `--on` prüft für einen verwalteten Service, in welchen Runleveln dieser laufen muss, und generiert die entsprechenden Softlinks.
- `--off` entfernt die Softlinks für einen verwalteten Service.
- `--list` zeigt alle Dienste an, die `chkconfig` verwaltet und in welchen Runleveln diese laufen.

Die Handhabung des Programms ist recht einfach. Hier sehen Sie einen Ausschnitt der Ausgabe des Kommandos `chkconfig--list`:

```
rsyslog       0:Aus   1:Aus   2:Ein   3:Ein   4:Ein   5:Ein   6:Aus
saslauthd     0:Aus   1:Aus   2:Aus   3:Aus   4:Aus   5:Aus   6:Aus
sendmail      0:Aus   1:Aus   2:Aus   3:Aus   4:Aus   5:Aus   6:Aus
smolt         0:Aus   1:Aus   2:Ein   3:Ein   4:Ein   5:Ein   6:Aus
```

Es wird für jeden Dienst angezeigt, in welchem Runlevel er automatisch gestartet wird und in welchem nicht.

Bei `update-rc.d` handelt es sich um ein Skript, das in Perl geschrieben wurde. Die Bedienung unterscheidet sich erheblich von der `chkconfig`-Syntax. Das Programm setzt voraus, dass es für einen Dienst ein Skript im Verzeichnis */etc/init.d* gibt. Wenn Sie Start- und Stopplinks für einen neu installierten Dienst automatisch erzeugen lassen wollen, verwenden Sie einfach die Option `defaults`:

```
root@arch-deb:/etc/rc2.d# update-rc.d ssh defaults
```

In der Praxis werden für die meisten Dienste allerdings schon während der Installation die entsprechenden Softlinks erstellt. Aus Sicherheitsgründen ist aber z. B. `ssh` bei vielen Distributionen vom automatischen Start ausgenommen.

Wenn Sie einen Dienst nicht deinstallieren, sondern lediglich einen automatischen Start verhindern wollen, können Sie die Option `disable` verwenden. Es werden dann automatisch alle Startlinks in Stopplinks umgewandelt, indem einfach das Präfix *S* des Links in ein *K* umbenannt wird. Diesen Vorgang können Sie einfach durch die Option `enable` wieder rückgängig machen:

```
root@arch-deb:/etc/rc2.d# update-rc.d cron enable
update-rc.d: using dependency based boot sequencing
```

Mit der Option `remove` werden die Start- und Stopplinks stattdessen entfernt und nicht nur umbenannt. Je nach verwendeter Programmversion kann hier die Option `-f` (force) notwendig werden. Das liegt daran, dass `remove` eigentlich nur dann verwendet werden sollte, wenn ein Startskript bereits gelöscht worden ist und man nun lediglich die verwaisten Links nachträglich entfernen will.

Runlevel-Wechsel mit init oder telinit

Zum Wechsel eines Runlevels verwenden Sie das Kommando `init` oder `telinit`. In den meisten Fällen ist `telinit` kein eigenständiges Programm mehr, sondern lediglich ein Softlink auf `init`. Es ist also egal, welchen der beiden Befehle Sie verwenden. Wenn Sie in ein anderes Runlevel wechseln wollen, geben Sie einfach `init` ein, gefolgt von der Nummer des gewünschten Runlevels. So wechselt `init 1` in den Single User Mode. `init s` und `init S` liefern dasselbe Ergebnis. Mit `init 0` kann man das System herunterfahren oder es mit `init 6` neu starten. In beiden Fällen werden eventuell verbundene Benutzer allerdings nicht benachrichtigt und das System wird ohne weitere Verzögerungen sofort heruntergefahren. Ein Datenverlust wäre nicht auszuschließen.

systemd

Vergleiche zu SysVinit

systemd ist eine moderne Alternative zu *SysVinit*. Inzwischen wird er von fast allen Linux-Distributionen standardmäßig verwendet. *systemd* startet Dienste parallel und beschleunigt dadurch den Systemstart gegenüber *SysVinit* erheblich. *systemd* parallelisiert auch Dienste, die voneinander abhängig sind. Damit das funktioniert, stellt *systemd* Sockets zur Verfügung und speichert Anfragen an noch nicht laufende Dienste zwischen, die an deren Sockets gerichtet wurden.

systemd ist grundsätzlich kompatibel mit *System-V*-basierten Skripten. Sie können auch hier in der Übergangszeit weiterhin mit den altbekannten Kommandos von *System-V* arbeiten (z. B. `runlevel`, `init` und `sysctl`). Sie sollten sich allerdings daran gewöhnen, vorzugsweise mit den zu *systemd* gehörenden Administrationstools zu arbeiten.

Wenn Sie z. B. zum Runlevel 3 wechseln wollen, dann können Sie dazu (anstatt `init 3`) folgendes Kommando verwenden:

```
[root@fedora24 /]# systemctl isolate runlevel3.target
```

Das Target `runlevel3.target` ist in Wirklichkeit ein Link auf das tatsächlich existierende Target `multi-user-target`.

Für den Single User Mode wird das Target `rescue.target` angesteuert. Verwenden Sie hierfür also dieses Kommando:

```
[root@fedora24 /]# systemctl isolate rescue.target
```

Wenn Sie ein System herunterfahren müssen, können Sie den folgenden Aufruf ausführen:

```
[root@fedora24 /]# systemctl isolate poweroff.target
```

Und für einen Neustart verwenden Sie:

```
[root@fedora24 /]# systemctl isolate reboot.target
```

Sie können übrigens die Erweiterungen (wie in diesem Fall `.target`) bei der Eingabe von Kommandos weglassen.

Prüfungstipp

Das gerade beschriebene Ansteuern der Targets ist ein für die Prüfung wichtiges Thema. Auch das Einstellen des standardmäßigen Runlevels (bzw. jetzt Targets) sollten Sie beherrschen.

/usr/lib/systemd/, /etc/systemd/ und /run/systemd

Die sogenannten Units entsprechen bei *systemd* in etwa den Startskripten von *SysVinit*. *Units* sind allerdings vom Aufbau her wesentlich einfacher konstruiert. Der genaue Inhalt ist nicht prüfungsrelevant, aber Sie sollten wissen, dass die Dateien der *Units* (also auch die der Targets) im Verzeichnis */usr/lib/systemd/system* (auch */lib/systemd/system*) zu finden sind.

Konfigurierte Targets werden in das Verzeichnis */etc/systemd/system* verknüpft. Um etwa das standardmäßige Target für den Systemstart festzulegen, können Sie hier eine entsprechende Verknüpfung anlegen bzw. ändern. Damit der Computer nach einem Neustart lediglich die Multiuser-Umgebung, nicht aber die grafische Umgebung lädt, können Sie eine solche Verknüpfung erstellen:

```
[root@fedora24 /]# ln -s /usr/lib/systemd/system/multi-user.target /etc/
systemd/system/default.target
```

Ein weiteres prüfungsrelevantes Verzeichnis ist */run/systemd*. Sie können hier in der Regel keine administrativen Eingriffe vornehmen, sollten aber wissen, dass *systemd* an dieser Stelle einige Verzeichnisse und Sockets verwendet. Diese werden unter anderem zur Kommunikation zwischen den zu *systemd* gehörenden Prozessen verwendet.

systemctl

Abgesehen vom Wechsel der Runlevel können Sie das Programm `systemctl` auch zur Steuerung von Diensten verwenden. Um etwa den *sshd*-Daemon zu starten, können Sie dieses Kommando verwenden:

```
[root@fedora24 /]# systemctl start sshd.service
```

Beendet wird der Daemon entsprechend mit diesem Befehl:

```
[root@fedora24 /]# systemctl stop sshd.service
```

Mit der Option `status` können Sie unter anderem überprüfen, ob ein Dienst läuft. Es werden hierbei in Abhängigkeit vom abgefragten Dienst zusätzliche Informationen ausgegeben:

```
[root@fedora24 systemd]# systemctl status sshd
• sshd.service - OpenSSH Daemon
   Loaded: loaded (/usr/lib/systemd/system/
sshd.service; disabled; vendor preset
   Active: active (running) since Di 2016-07-12 20:28:39 CEST; 9s ago
 Main PID: 12776 (sshd)
    Tasks: 1 (limit: 512)
   CGroup: /system.slice/sshd.service
           └─12776 /usr/bin/sshd -D

Jul 12 20:28:39 fedora24 systemd[1]: Started OpenSSH Daemon.
Jul 12 20:28:39 fedora24 sshd[12776]: Server listening on 0.0.0.0 port 22.
Jul 12 20:28:39 fedora24 sshd[12776]: Server listening on :: port 22.
```

Damit ein Daemon beim Systemstart automatisch geladen wird, ist im Verzeichnis */etc/systemd/system* ein entsprechender Link erforderlich, der auf die jeweils dem Service entsprechende Datei in */usr/lib/systemd/system* zeigt. Ein solcher Link kann sehr bequem mittels `systemctl` erstellt bzw. bei Bedarf wieder gelöscht werden. Als Beispiel wieder auf den sshd-Daemon bezogen, würde man den Link zum automatischen Starten des Daemons so erstellen:

```
[root@fedora24 /]# systemctl enable sshd.service
```

Entsprechend würde folgendes Kommando den Link wieder entfernen:

```
[root@fedora24 /]# systemctl disable sshd.service
```

Im Prinzip ist das die Entsprechung zur Verwendung von `chkconfig` bzw. `update-rc.d` bei *SysVinit* basierten Systemen.

systemd-delta

Die Konfigurationsdateien von *systemd* sind jeweils in Unterverzeichnissen von */etc/*, */usr/lib/*, */lib/* und */run/* zu finden. Zu einigen Konfigurationsdateien gibt es zusätzlich **.d*-Verzeichnisse, die drop-in-Files enthalten, was die ganze Angelegenheit noch unübersichtlicher macht. Das Programm `systemd-delta` unterstützt Sie, wenn Konfigurationsdateien sich in Ihren Einstellungen überschneiden. Wenn die Konfiguration sauber ist, sollte das Programm eine solche Antwort liefern:

```
[root@fedora24 /]# systemd-delta
0 overridden configuration files found.
```

Bei abweichenden Konfigurationen kann das Ergebnis so aussehen:

```
root@archangel:~# systemd-delta
[EXTENDED]   /run/systemd/system/session-51.scope → /run/systemd/system/
session-51.scope.d/50-SendSIGHUP.conf
[EXTENDED]   /run/systemd/system/session-51.scope → /run/systemd/system/
session-51.scope.d/50-After-systemd-user-sessions\x2eser
[EXTENDED]   /run/systemd/system/session-51.scope → /run/systemd/system/
session-51.scope.d/50-After-systemd-logind\x2eservice.co
3 overridden configuration files found.
```

Linux Standard Base (LSB)

Die Linux Standard Base gehört zur Linux Foundation. Diese Arbeitsgruppe hat es sich zum Ziel gemacht, Standards zu setzen, die für alle Linux-Distributionen gültig sind. Schon im Laufe der 90er Jahre des letzten Jahrhunderts sind viele neue Linux-Distributionen und Derivate entstanden. Diese versuchten sich natürlich immer von den jeweils anderen Distributionen deutlich zu unterscheiden und abzugrenzen. Hierdurch ergaben sich zum Teil so starke Abweichungen, insbesondere bezüglich der Verzeichnisstruktur, dass es immer schwieriger wurde, Anwendungsprogramme zu schreiben, die dann auch auf allen Linux-Plattformen liefen. So konnte es vorkommen, dass eine Anwendung benötigte Tools oder Bibliotheken auf einem System nicht finden konnte. Diesem Problem wirkt die LSB entgegen. Ich würde sogar behaupten, dass sich in den letzten Jahren eine spürbare Verbesserung in dieser Problematik ergeben hat.

Wenn Sie sich genauer über die LSB informieren möchten, werden Sie unter den folgenden beiden URLs fündig:

http://www.linuxbase.org
http://www.linuxfoundation.org

202.2 Systemwiederherstellung

Wichtung: 4

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, ein Linux-System während des Bootprozesses und im Recovery-Modus in richtiger Weise zu bedienen. Dieses Lernziel umfasst sowohl die Benutzung des `init`-Befehls als auch die von `init`-bezogenen Kernel-Optionen. Die Kandidaten sollen dazu in der Lage sein, Ursachen für Fehlfunktionen in Bootloadern zu finden. Hierbei sind sowohl GRUB 2, als auch GRUB Legacy von Interesse. Sowohl BIOS, als auch UEFI-Systeme werden abgedeckt.

Wichtigste Wissensgebiete:

- BIOS und UEFI
- NVMe Bootprozess
- GRUB Version 2 und Legacy
- GRUB-Shell
- Start des Bootloaders und Übergabe der Kontrolle an den Kernel
- Laden des Kernels
- Hardwareinitialisierung und Einrichtung
- Initialisierung und Setup der Hintergrundprozesse (Daemons)
- Kenntnisse über die Installationspfade der verschiedenen Bootloader auf der Festplatte oder auf anderen Speichermedien
- Überschreiben von Standardoptionen des Bootloaders und Verwendung seiner Shell
- Verwenden von *systemd* in Notfall- und Wiederherstellungsmodi

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `mount`
- `fsck`
- *inittab*, `init` und `telinit` mit SysVinit
- Inhalt von */boot/*, */boot/grub/* und */boot/efi/*
- EFI System Partition (ESP)
- *GRUB*
- `grub-install`
- `efibootmgr`
- UEFI-Shell
- `initrd`, `initramfs`
- *Master Boot Record*
- `systemctl`

Allgemeines

Wenn ein System nicht mehr startet, kann das natürlich verschiedene Ursachen haben. Wenn der Systemstart schon in einer sehr frühen Phase fehlschlägt, stehen aber Probleme mit dem verwendeten Bootloader oder dem Dateisystem des Startmediums schnell im Verdacht. In solchen Situationen ist es immer gut, eine Live-CD griffbereit zu haben, auf der sich eine beliebige Linux-Distribution befindet. Einige Administratoren haben zwar ihre Präferenzen, was so ein Live-System anbelangt, aber letztendlich benötigt ein Profi ohnehin nur Standardwerkzeuge wie `fsck`, `fdisk`, `dd` und `vi`, um nur einige zu nennen. Einige der in diesem Topic genannten Komponenten kennen Sie bereits aus den vorangehenden Kapiteln. Das sind insbesondere die Kommandos `init`, `telinit` und `systemctl` sowie die Konfigurationsdatei */etc/inittab*. Diese Themen sollen hier nicht noch einmal wiederholt werden, sondern der Schwerpunkt soll auf die Überprüfung von Bootloadern und Dateisystemen gelegt werden.

Die frühe Phase des Systemstarts

BIOS

Das BIOS (Basic Input and Output System) ist ein Programm, das sich in einem ROM-Baustein auf der Hauptplatine jedes Computers befindet. Dieses Programm hat u. a. die Aufgaben, die Hardware eines Computers beim Systemstart zu initialisieren und ein paar grundlegende Tests durchzuführen. Wenn diese Tests, die als POST (Power On Self Test) bezeichnet werden, abgeschlossen sind, sucht das BIOS nach einem Startprogramm auf einem bootfähigen Datenträger. Im Fall, dass Linux das zu startende Betriebssystem ist, handelt es sich bei diesem Startprogramm wahrscheinlich um den Bootloader GRUB. Den Bootloader sucht das BIOS im MBR (Master Boot Record) von Datenträgern, die im BIOS als Startdatenträger angegeben sind.

UEFI

Bei vielen modernen Hauptplatinen kommt als Firmware nicht mehr das altbekannte BIOS zum Einsatz, sondern UEFI (Unified Extensible Firmware Interface). Der grundsätzliche Verwendungszweck von UEFI kommt aber dem eines BIOS gleich. Gegenüber dem BIOS weist UEFI neue Funktionen auf. Hierzu zählen:

- hohe Grafikauflösungen im Setup-Programm
- Unterstützung von GUID-Partitionstabellen für Festplatten mit einer Größe von bis zu 8 Zebibytes
- Netzwerkanbindung zur Fernwartung
- Integrationsmöglichkeit von Treibern auf Firmware-Ebene
- integrierter Bootloader
- Digital Right Management

Die Integration des Digital Right Managements bringt UEFI in die Kritik, weil es damit möglich wird, die Ausführung von unerwünschter Software zu verhindern. Hierbei stellt sich natürlich die Frage, ob die Software vom Hersteller oder vom User nicht erwünscht ist.

In Bezug auf Linux (oder überhaupt auf Betriebssysteme) sollten Sie einige Dinge wissen. UEFI benötigt eine eigene Partition, die als *EFI System Partition* (*ESP*) bezeichnet wird. Diese Partition wird von Betriebssystemen, die UEFI unterstützen, normalerweise während der Installation automatisch angelegt. Sie benötigt ein FAT-Dateisystem, wobei die Spezifikationen dieses Dateisystems unabhängig von den ursprünglichen FAT-Konventionen sind. Die EFI System Partition wird unter Linux im Verzeichnis */boot/efi* bereitgestellt. Sie enthält zumindest den Bootloader und eventuell andere zum Start benötigte Dateien. *GRUB* etwa kann den Kernel auch dann laden, wenn er sich nicht innerhalb der ESP befindet. *systemd-boot* hingegen kann das nicht. In diesem Fall muss also der Kernel auch innerhalb der ESP liegen.

efibootmgr

Mit dem Programm `efibootmgr` können Sie den *EFI Boot Manager* einsehen und bearbeiten. Der Aufruf des Programms ohne Optionen und Parameter gibt einen Einblick in die aktuelle Konfiguration:

```
[root@arch-book /]# efibootmgr
BootCurrent: 0003
Timeout: 0 seconds
BootOrder: 0005,0006,0004,0003,0000,2001,2002,2003
Boot0000* Windows Boot Manager
Boot0003* manjaro_grub
Boot0004* Network Boot: Realtek PXE B01 D00
Boot0005* SATA HDD    : ST500LM000-SSHD-8GB
Boot0006* SATA ODD    : PLDS    DVD-RW DA8A5SH
Boot2001* EFI USB Device
Boot2002* EFI DVD/CDROM
Boot2003* EFI Network
```

Die Ausgabe des Befehls ist fast selbsterklärend. Zuletzt wurde das System mit der Ordnungsnummer `0003` gestartet, also `manjaro_grub`. Der Boot Manager wird beim Systemstart nicht angezeigt, weil der `Timeout` auf `0 seconds` eingestellt ist. Ansonsten sehen Sie, in welcher Reihenfolge EFI beim Start nach Betriebssystemen sucht. Wenn Sie möchten, dass beim nächsten Start des Computers automatisch der Windows Boot Manager ausgeführt wird, können Sie, in Bezug auf diese Konfiguration, folgenden Befehl verwenden:

```
[root@arch-book /]# efibootmgr -n 0000
```

Wenn Sie allerdings das beim Kauf des Rechners vorinstallierte Windows bereits gelöscht haben, können Sie den verwaisten Eintrag mit diesem Kommando aus der Konfiguration entfernen:

```
[root@arch-book /]# efibootmgr -B 0000
```

UEFI-Shell

Die *UEFI-Shell* ist kein Bestandteil von Linux, sondern von *UEFI* selbst. Sie können diese Konsole während des Systemstarts aus dem EFI Boot Menü auswählen. Welche Kommandos in der Shell verfügbar sind, unterscheidet sich von Hersteller zu Hersteller. Von dieser Shell aus können auch dann noch Konfigurationen bzw. Reparaturen durchgeführt werden, wenn keines der installierten Betriebssysteme mehr lauffähig ist.

NVMe-Bootprozess

Was die Treiberunterstützung für *NVMe* anbelangt, brauchen Sie sich bei Linux keine Sorgen zu machen. Der Kernel unterstützt *NVMe* seit Version 3.1. Wenn Sie allerding vorhaben, von *NVMe* zu booten, sollten Sie darauf achten, dass im *BIOS* bzw. *UEFI* hierfür eine Unterstützung vorhanden und aktiviert ist. Sollte das nicht der Fall sein, muss der initiale Systemstart von einem anderen Datenträger aus erfolgen. Sobald der Kernel läuft, kann auch ohne Unterstützung durch *BIOS* oder *UEFI* auf *NVMe* zugegriffen werden.

Master Boot Record (MBR)

Davon ausgehend, dass der Startvorgang des Systems von einer Festplatte fortgesetzt wird (und nicht z. B. von einer CD, DVD oder einer PXE-fähigen Netzwerkkarte), sollten Sie verstehen, wie eine Festplatte grundsätzlich strukturiert ist, damit Sie den weiteren Verlauf des Startvorgangs nachvollziehen können.

Eine Festplatte kann in mehrere Partitionen unterteilt sein. Es ist möglich, bis zu vier »echte« Partitionen auf einer Festplatte zu erstellen. Sollten mehr als vier Partitionen, logische Laufwerke o. Ä. (je nach Betriebssystem) existieren, handelt es sich wahrscheinlich um logische Partitionen, um Laufwerke innerhalb einer erweiterten Partition oder um Volumen, die vom Betriebssystem außerhalb der Partitionstabelle verwaltet werden. Auch Bit-Slices können vortäuschen, man hätte mehr als vier Partitionen.

Man kann bei vielen Systemen, so auch bei Linux, drei primäre und eine erweiterte Partition erstellen. Diese erweiterte Partition kann dann ihrerseits logische Partitionen enthalten, die ebenfalls nicht von der Partitionstabelle verwaltet werden. Die Partitionstabelle befindet sich im *Master Boot Record* (*MBR*). Dieser MBR gehört selbst keiner Partition an, sondern er ist eigenständig. Er ist immer genau 512 Bytes

groß und befindet sich ganz am Anfang der Festplatte in Sektor 0, Spur 0. Nachdem der Aufbau der Festplatte nun zumindest ansatzweise geklärt ist (weitere Informationen folgen noch in diesem Kapitel), kann die Beschreibung des Computerstartvorgangs fortgesetzt werden. Das BIOS liest nun einfach in Spur 0 des Sektors 0 den Master Boot Record der Festplatte und findet dort beispielsweise Folgendes vor:

```
63eb 1090 d08e 00bc b8b0 0000 d88e c08e
befb 7c00 00bf b906 0200 a4f3 21ea 0006
be00 07be 0438 0b75 c683 8110 fefe 7507
ebf3 b416 b002 bb01 7c00 80b2 748a 8b01
024c 13cd 00ea 007c eb00 00fe 0000 0000
0000 0000 0000 0000 0000 8000 0001 0000
0000 0000 faff 07eb c2f6 7580 b202 ea80
7c74 0000 c031 d88e d08e 00bc fb20 64a0
3c7c 74ff 8802 52c2 80be e87d 011c 05be
f67c 80c2 4874 41b4 aabb cd55 5a13 7252
813d 55fb 75aa 8337 01e1 3274 c031 4489
4004 4488 89ff 0244 04c7 0010 8b66 5c1e
667c 5c89 6608 1e8b 7c60 8966 0c5c 44c7
0006 b470 cd42 7213 bb05 7000 76eb 08b4
13cd 0d73 c2f6 0f80 d884 be00 7d8b 82e9
6600 b60f 88c6 ff64 6640 4489 0f04 d1b6
e2c1 8802 88e8 40f4 4489 0f08 c2b6 e8c0
6602 0489 a166 7c60 0966 75c0 664e 5ca1
667c d231 f766 8834 31d1 66d2 74f7 3b04
0844 377d c1fe c588 c030 e8c1 0802 88c1
5ad0 c688 00bb 8e70 31c3 b8db 0201 13cd
1e72 c38c 1e60 00b9 8e01 31db bff6 8000
c68e f3fc 1fa5 ff61 5a26 be7c 7d86 03eb
95be e87d 0034 9abe e87d 002e 18cd feeb
5247 4255 0020 6547 6d6f 4800 7261 2064
6944 6b73 5200 6165 0064 4520 7272 726f
0a0d bb00 0001 0eb4 10cd 3cac 7500 c3f4
0000 0000 0000 0000 2f05 000b 0000 2080
0021 fe83 ffff 0800 0000 6800 027b fe00
ffff fe05 ffff 7000 027b 7ab0 22c7 0000
0000 0000 0000 0000 0000 0000 0000 0000
0000 0000 0000 0000 0000 0000 0000 aa55
```

In diesem Fall ist das ein Ladeprogramm in hexadezimaler Schreibweise, nämlich die erste Stufe von GRUB 2 und im unteren Bereich die Partitionstabelle. Es gibt in Abhängigkeit vom verwendeten Betriebssystem zwei absolut unterschiedliche Möglichkeiten dazu, wie der Startvorgang fortgesetzt werden kann. Wenn ein Bootloader im Master Boot Record vorgefunden wird, wie in diesem Fall, gibt das BIOS die Kon-

trolle an diesen ab und beendet sich. Falls im MBR kein Bootloader vorhanden ist (dann wären im oberen Bereich des MBRs übrigens nur Nullen zu sehen), konsultiert das BIOS die Partitionstabelle. Hier sucht das BIOS nach einem Verweis auf eine startfähige Partition.

Die Partitionstabelle beginnt in der fünften Zeile, von unten an gezählt, mit `80`. Der Eintrag für jede Partition entspricht genau einer Zeile. Das Ende der Partitionstabelle wird mit `55aa` markiert. Hierbei handelt es sich um eine Abschlussmarkierung, die nicht Bestandteil eines Partitionseintrags ist. Die offensichtliche »Vertauschung« der Bytes (`55aa` wird als `aa55` dargestellt) wird durch das Programm verursacht, das diese Partitionstabelle darstellt. Programme, die eine Big-Endian-First-Byte-Abfolge verwenden, zeigen das Byte mit den höchstwertigen Bits zuerst an. Andere Programme nutzen die Little-Endian-First-Byte-Abfolge. Sie zeigen das Byte mit den niedrigstwertigen Bits zuerst an. Dieser Unterschied ist historisch durch die Verwendung unterschiedlicher Prozessorarchitekturen begründet. Während Intel-basierte Systeme Little Endian verwenden, kommt beispielsweise bei Motorola 6800 Big-Endian zum Einsatz.

Die Partition, die mit `80` beginnt, ist die Startpartition. Sollte also kein Ladeprogramm im MBR vorliegen, würde das BIOS nun diesem Verweis folgen und den Bootsektor der Partition mit der Markierung `80` einlesen. (Machen Sie sich bitte den Unterschied zwischen Startsektor einer Partition und MBR klar!) Spätestens hier sollte sich dann entweder ein Bootloader oder das Betriebssystem selbst befinden. Nun kann der tatsächliche Vorgang, den man als Booten bezeichnet, beginnen.

Ein Backup des MBR können Sie übrigens einfach mithilfe des Programms `dd` durchführen:

```
archangel:/ # dd if=/dev/hda of=mbr.backup ibs=512 count=1
```

Sie können die Sicherungsdatei dann an einem sicheren Ort aufbewahren und im Notfall mit dem entsprechend umgekehrten Kommando wieder in das System kopieren.

/boot/, /boot/grub/ und /boot/efi/

Die Verzeichnisse */boot/*, */boot/grub/* und */boot/efi/* und deren Inhalte sind ausdrücklich als Prüfungsthema benannt, tauchen aber in diesem Buch ohnehin immer bei den jeweils zugehörigen Themen auf. Deshalb sollen an dieser Stelle ein paar Verweise reichen:

- */boot* gehört zum Thema Kernel (Topic 201).
- */boot/grub* finden Sie beim Thema GRUB (Legacy und 2) auf den nächsten Seiten.
- */boot/efi* ist Bestandteil des Themas UEFI in diesem Kapitel.

GRUB (Legacy)

GRUB ist ein zweistufiger Bootloader. Genau genommen sind es inzwischen drei Stufen, weil aus Gründen der Kompatibilität zu verschiedenen Dateisystemen irgendwann ein »Stage 1,5« hinzugefügt wurde.

Das Programm für den ersten Stage befindet sich im MBR der Festplatte, von der aus das System starten soll. Es gibt aber normalerweise auch noch eine Kopie in */boot/grub/stage1*. Es handelt sich hierbei um eine Binärdatei, weshalb ein normaler Pager eine Fehlermeldung ausgibt, wenn Sie versuchen, den Inhalt der Datei zu betrachten.

Für den Stage 1,5 wird dann ein Programm ausgeführt, das mit dem verwendeten Dateisystem übereinstimmt, z. B. *reiserfs_stage1_5*. Diese Datei befindet sich ebenfalls im Pfad */boot/grub*.

Die letzte Stufe des Bootloaders (Stage 2) befindet sich wiederum in der Datei */boot/grub/stage2*. Dieses Programm stellt das Bootmenü für den Benutzer bereit und ist auch für das Starten des Kernels zuständig.

GRUB (Legacy)-Prompt

GRUB verfügt über einen Prompt, den Sie verwenden können, um mit dem Bootloader zu interagieren. Das ist etwa dann nötig, wenn das System aufgrund einer Fehlkonfiguration nicht startet. Sie können dann mit GRUB das System manuell booten. Dazu benötigt GRUB Informationen über die Position des Hauptverzeichnisses, den Kernel und, falls vorhanden, die initiale RAM-Disk. Die GRUB-Eingabeaufforderung sieht so aus:

```
grub>
```

Wenn Sie ein System manuell starten müssen, können Sie eine Kommandofolge wie diese verwenden:

```
grub> root (hd0,0)
grub> kernel /boot/vmlinuz-2.6.23.1-10.fc7 root=/dev/sda2
grub> initrd /boot/initrd-2.6.23.1-10.fc7.img
grub> boot
```

Es wurde hier davon ausgegangen, dass sich das Verzeichnis */boot* auf einer SCSI-Festplatte, nämlich */dev/sda1*, und das Hauptverzeichnis »/« auf */dev/sda2* befindet.

GRUB-Legacy-Konfigurationsdateien

GRUB unterscheidet nicht zwischen SCSI-Festplatten und IDE-Geräten. Die erste Partition auf der ersten Festplatte, die ihm durch ein BIOS übermittelt wird, bezeichnet GRUB als `(hd0,0)`. Dabei spielt die Art des verwendeten Festplattensubsystems keine Rolle. Damit für den weiteren Startverlauf eine Zuordnung möglich ist, benötigt er

die Konfigurationsdatei */boot/grub/device.map*. In dieser Datei gibt es lediglich eine Zuordnung von der GRUB-Notation zu den »normalen« Linux-Geräten:

```
[root@fedora10 grub]# cat device.map
 (hd0)     /dev/sda
```

In der Datei *menu.lst* (normalerweise ein Softlink auf *grub.conf*) finden Sie im Normalfall Einträge wie den folgenden für jedes zu startende Betriebssystem:

```
#boot=/dev/sda
default=0
timeout=5
splashimage=(hd0,0)/grub/splash.xpm.gz
hiddenmenu
title Fedora (2.6.23.1-10.fc7)
        root (hd0,0)
        kernel /vmlinuz-2.6.23.1-10.fc7 ro root=/dev/VolGroup00/
        LogVol00 rhgb quiet
        initrd /initrd-2.6.23.1-10.fc7.img
```

GRUB 2

GRUB-2-Konfigurationsdateien

GRUB 2 ist eigentlich ein völlig neuer Bootloader und hat mit seinem Vorläufer nicht mehr besonders viel gemeinsam. Im Gegensatz zu GRUB Legacy positioniert GRUB 2 einige Konfigurationsdateien unterhalb von */etc*. Die Hauptkonfigurationsdatei befindet sich immer noch im Verzeichnis */boot/grub* (auch */boot/grub2*). Es ist die Datei *grub.cfg*, die durch das Programm `grub-mkconfig` (auch `grub2-mkconfig`) generiert wird. Als Vorlage kommen hier die Datei */etc/default/grub* und weitere Dateien unterhalb von */etc/grub.d* zum Einsatz. Sie sollten zur Konfiguration von GRUB 2 lediglich die genannten Dateien im */etc*-Verzeichnis editieren.

Wenn Sie einem System manuell einen neuen Kernel hinzugefügt haben, müssen Sie das Skript `update-grub2` ausführen. Diese Vorgehensweise entspricht also der Vorgehensweise von GRUB Legacy.

Die GRUB-2-Shell

Die GRUB-Shell benötigen Sie lediglich in Notfällen. Sie können mit dieser Shell notdürftig kleinere Reparaturen an der Startumgebung vornehmen. Das könnte zum Beispiel notwendig werden, wenn sich die UUID eines Datenträgers nach der Wiederherstellung eines Backups auf einem anderen System geändert hat oder wenn sich die ID einer Partition oder eines Datenträgers nach einem Systemumbau geändert haben sollte. Wenn so ein Notfall eintritt, brauchen Sie die GRUB-Shell nicht manuell

zu starten, weil sie dann automatisch geladen wird. Wenn Sie einen Eintrag im Startmenü editieren wollen, können Sie diesen aber auch aus dem Startbildschirm von GRUB heraus mit den Cursortasten auswählen und anschließend die Taste E betätigen. Es wird dann ein Editor geladen, mit dem Sie genau diesen Eintrag modifizieren können. Anschließend können Sie mit dem modifizierten Eintrag das System wie gewohnt starten. Die Änderungen, die Sie hier vornehmen, sind allerdings flüchtig. Wenn der Eintrag dauerhaft geändert werden soll, müssen Sie, wie gehabt, nach erfolgtem Systemstart die Datei *grub.cfg* (bzw. *grub.conf*) bearbeiten.

Wenn Sie im Startbildschirm von GRUB die Taste C betätigen, öffnet sich das GRUB Command Line Interface. Mit dem Kommando `help` bekommen Sie einen ersten Überblick über die verwendbaren Befehle. Mit `help <Kommando>` bekommen Sie entsprechend Hilfe zu jedem einzelnen Kommando der Shell. Erfreulicherweise vervollständigt die GRUB-Shell Kommandos, wenn Sie die Tabulatortaste verwenden. Das funktioniert genauso, wie Sie es von der Shell `bash` her gewohnt sind.

Die einzelnen Kommandos der GRUB-Shell müssen Sie für die Prüfung nicht auswendig können. Sie sollten sich dieses CLI aber zumindest einmal ansehen.

Dateisysteme prüfen und reparieren

Wenn Sie ein Dateisystem überprüfen oder gar reparieren wollen, sollten Sie zuerst überprüfen, welche Gerätedatei dem Verzeichnis zugeordnet ist. Das können Sie mit `mount` erledigen.

```
root@archangel:/# mount |grep usb-disk1
/dev/sdb1 on /media/usb-disk1 type ext4 (rw,noexec,nosuid,nodev)
```

Es muss also in diesem Fall nach dem Aushängen des Dateisystems die Gerätedatei */dev/sdb1* behandelt werden.

Als Nächstes müssen Sie dafür sorgen, dass kein Prozess mehr auf das Dateisystem zugreift. Das ist insofern einfach, als dass Sie ein Dateisystem sowieso aushängen müssen, um eine Überprüfung oder Reparatur durchzuführen. Sollten noch Dateien geöffnet sein, erhalten Sie beim Aushängen eine Fehlermeldung:

```
root@archangel:/# umount /media/usb-disk1/
umount: /media/usb-disk1: device is busy.
        (In some cases useful info about processes that use
         the device is found by lsof(8) or fuser(1))
```

Die Fehlermeldung beinhaltet sogar einen Hinweis darauf, wie Sie herausbekommen, welche Benutzer oder Prozesse Dateien geöffnet halten. Das Programm `lsof` soll hier zunächst Klarheit verschaffen:

```
root@archangel:/# lsof /media/usb-disk1/
COMMAND   PID  USER   FD   TYPE DEVICE SIZE/OFF NODE NAME
bash    23936 franz  cwd    DIR   8,17     4096    2 /media/usb-disk1
less    23967 franz  cwd    DIR   8,17     4096    2 /media/usb-disk1
less    23967 franz    4r   REG   8,17        0   12 /media/usb-disk1/
   MichisTagebuch.txt
```

Diese Informationen sind völlig ausreichend. Der User Franz liest in einer Datei, die *MichisTagebuch.txt* heißt. Er verwendet den Pager `less`. Sie können den User jetzt auffordern, die Datei zu schließen (die ist dem Namen nach zu urteilen ohnehin nicht für ihn bestimmt) und den Datenträger dann aushängen. Das Programm `lsof` sollte bei nochmaliger Eingabe keine Ergebnisse mehr liefern. Hängen Sie die Festplatte jetzt aus:

```
root@archangel:/# umount /media/usb-disk1/
```

Sollte es sich um eine USB-Festplatte handeln, wie in diesem Fall, können Sie diese jetzt gefahrlos vom System entfernen. Das ist insofern sinnvoll, als dass Sie die Ausgabe des Kernels hierbei beobachten können, was erste Rückschlüsse auf Hardwarefehler liefern kann. Insbesondere wenn Sie das Gerät wieder anstecken, können Sie mit `dmesg|tail` eine erste Diagnose stellen. Nach dem Aushängen ergibt sich:

```
root@archangel:/# dmesg | tail -n 1
[834238.130766] usb 1-3: USB disconnect, address 3
```

Interessanter ist aber, was der Kernel beim Wiedereinhängen der Festplatte feststellt:

```
root@archangel:/# dmesg | tail -n 20
[834252.561017] usb 1-3: new high speed USB device using ehci_
hcd and address 5
[834252.694851] usb 1-3: configuration #1 chosen from 1 choice
[834252.695146] scsi7 : SCSI emulation for USB Mass Storage devices
[834252.695256] usb-storage: device found at 5
[834252.695262] usb-storage: waiting for device to settle before scanning
[834257.692254] usb-storage: device scan complete
[834257.693967] scsi 7:0:0:0: Direct-
Access     WD       5000AAC External 1.06 PQ: 0 ANSI: 0
[834257.694558] sd 7:0:0:0: Attached scsi generic sg2 type 0
[834257.695351] sd 7:0:0:0: [sdb] 976773168 512-byte logical blocks: (500 GB/
465 GiB)
[834257.699478] sd 7:0:0:0: [sdb] Write Protect is off
[834257.699483] sd 7:0:0:0: [sdb] Mode Sense: 00 00 00 00
[834257.699486] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[834257.701833] sd 7:0:0:0: [sdb] Assuming drive cache: write through
```

```
[834257.701838]  sdb: sdb1
[834257.719828] sd 7:0:0:0: [sdb] Assuming drive cache: write through
[834257.719836] sd 7:0:0:0: [sdb] Attached SCSI disk
[834258.192779] EXT4-fs (sdb1): mounted filesystem with ordered data mode
```

Die Ausgabe des Kommandos ist so gekürzt, dass nur das Ab- und Anschließen der Festplatte dargestellt wird. In diesem Fall sind keinerlei Fehlermeldungen des Kernels zu sehen, was auf eine fehlerfreie Elektronik schließen lässt. Beachten Sie auch die letzte Zeile! Der Daemon `mounted` hat die Festplatte wieder eingehängt, was natürlich für die Diagnose und ggf. die Reparatur wieder rückgängig gemacht werden muss:

```
root@archangel:/# umount /dev/sdb1
```

Jetzt kann das Dateisystem gefahrlos überprüft und ggf. repariert werden:

```
root@archangel:/# fsck -t ext4 -V /dev/sdb1
fsck from util-linux-ng 2.17.2
[/sbin/fsck.ext4 (1) -- /media/usb-disk1] fsck.ext4 /dev/sdb1
e2fsck 1.41.11 (14-Mar-2010)
/dev/sdb1: sauber, 548212/30531584 Dateien, 29760882/122096000 Blöcke
```

Es wurden keine Fehler gefunden. In diesem Fall war die Angabe des Dateisystems (`-t ext4`) übrigens optional. Das Programm `fsck` hätte über einen entsprechenden Eintrag in der Datei */etc/fstab* selbst den Dateisystemtyp ermitteln können. Wenn ein Fehler gefunden worden wäre, hätte `fsck` bei jedem einzelnen Fehler nachgefragt, ob dieser behoben werden soll. Das kann bei einer großen Anzahl von Fehlern sehr lästig werden. Sie sollten deshalb unbedingt die folgenden Optionen kennen:

- `-a` repariert automatisch alle gefundenen Dateisystemfehler ohne Nachfrage.
- `-y` beantwortet automatisch alle Fragen, die `fsck` stellt, mit »yes«.
- `-A` überprüft automatisch alle Dateisysteme, die in der Datei */etc/fstab* gelistet sind.

Wenn alle eventuellen Schäden beseitigt sind, hängen Sie das Dateisystem wieder ein und prüfen anschließend mit `dmesg`, ob alles in Ordnung ist:

```
root@archangel:/# mount /dev/sdb1 && dmesg|tail
```

Sie können in der Konfigurationsdatei */etc/fstab* eine Option verwenden, die bei Zugriffsstörung auf ein Dateisystem dieses automatisch read-only mountet. Die Option heißt `errors=remount-ro` und soll verhindern, dass fehlerhafte Daten geschrieben werden. Sie können dann nur noch lesend auf das Dateisystem zugreifen. Wenn Sie vermuten, dass es sich lediglich um eine temporäre Störung handelt, können Sie das Dateisystem beschreibbar remounten. Verwenden Sie dazu folgendes Kommando:

```
root@archangel:/# mount /dev/sdb1 -o remount,rw
```

Umgekehrt können Sie auch selbst (z. B. zu Diagnosezwecken) ein Dateisystem read-only remounten, indem Sie dieses Kommando verwenden:

```
root@archangel:/# mount /dev/sdb1 -o remount,ro
```

Probleme beim Laden des Kernels

Wenn Sie einen angepassten Kernel für ein System erstellt haben und das System anschließend nicht startet, kann das daran liegen, dass Sie Teile des Kernels, die während des Systemstarts erforderlich sind, modular einkompiliert haben. Das gilt insbesondere für Dateisystemtreiber, weil der Systemstart von einer Festplatte ohne passende Treiber für das Dateisystem, auf dem sich das Betriebssystem befindet, natürlich nicht fortgesetzt werden kann. Zur Behebung dieses Dilemmas gibt es die initiale RAM-Disk, die als Image in der Datei */boot/initrd* abgelegt ist. Zur Erzeugung der initialen RAM-Disk kommen, in Abhängigkeit von der verwendeten Linux-Distribution, unterschiedliche Programme zum Einsatz. Bei Red Hat-basierten Systemen verwenden Sie `mkinitrd`. Wenn Sie Debian oder ein Derivat von Debian konfigurieren, benötigen Sie `mkinitramfs`. Nähere Informationen zu diesem Thema finden Sie in Abschnitt 201.2, »Einen Linux-Kernel kompilieren«.

202.3 Alternative Bootloader

Wichtung: 2

Beschreibung: Die Kandidaten sollten auch andere Bootloader und deren wichtigste Eigenschaften kennen.

Wichtigste Wissensgebiete:

- SYSLINUX, ISOLINUX, PXELINUX
- Verstehen von PXE sowohl für BIOS als auch UEFI
- Kenntnis von systemd-Boot und U-Boot

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- *syslinux*
- *extlinux*
- *isolinux.bin*
- *isolinux.cgf*
- *isohdpfx.bin*

- *efiboot.img*
- *pxelinux.0*
- *prelinux.cfg/*
- *uefi/shim.efi*
- *uefi/grubx64.efi*

Allgemeines

Abgesehen von den gängigen Bootloadern GRUB 2 und GRUB Legacy ist insbesondere auf älteren Systemen immer noch LILO im Einsatz. Außerdem verwendet Slackware bis heute den Linux Loader. Ansonsten gibt es noch weitere Mechanismen, um Linux zu laden. Einige werden z. B. für den Start über das Netzwerk oder von optischen Medien benötigt. Da Live-Systeme, die von CDs, DVDs oder USB-Sticks aus gestartet werden können, immer gängiger werden, sollten Sie natürlich auch deren Bootloader kennen.

Das SYSLINUX-Projekt

Der schwedische Programmierer Hans Peter Anvin hat u. a. eine ganze Reihe von schlanken, spezialisierten Bootloadern zur FOSS (Free and Open Source Software) beigesteuert. Mit Sicherheit haben Sie schon den einen oder anderen seiner Bootloader verwendet, wahrscheinlich ohne dass es Ihnen überhaupt bewusst war.

SYSLINUX

Der Namensgeber des Projekts SYSLINUX wurde schon sehr früh für den Systemstart von Disketten verwendet. In den neunziger Jahren des letzten Jahrhunderts waren einige CD-ROM-Laufwerke nicht darauf ausgerichtet, als Bootlaufwerk zu fungieren, weshalb die Installation von Betriebssystemen meist von Startdisketten mit FAT-Dateisystem ausgeführt wurde oder werden musste. Später wurden Disketten zum Teil nur noch als reine Startmedien verwendet und das eigentliche Setup dann von CDs aus fortgesetzt.

Mit der Spezifikation *Eltorito* wurden CD-ROM-Laufwerke startfähig gemacht, was einen neuen Bootloader (ISOLINUX) erforderlich werden ließ. Schließlich ist SYSLINUX ausschließlich für FAT-Dateisysteme konzipiert worden! Dennoch hat SYSLINUX nicht an Bedeutung verloren, weil z. B. Rettungsdisketten u. Ä. weiterhin mit diesem Bootloader starteten. Heute wird SYSLINUX verwendet, um von USB-Medien zu starten, die mit dem FAT-Dateisystem formatiert sind.

Die Erstellung der Startmedien ist verhältnismäßig einfach. Sie benötigen lediglich das Programm `syslinux` selbst bzw. `syslinux.exe` unter Windows. Beide Varianten stehen auf der Website des SYSLINUX-Projekts zur Verfügung:

http://www.syslinux.org/

Da die genaue Vorgehensweise zur Erstellung dieser Medien nicht prüfungsrelevant ist, soll hier auf detaillierte Ausführungen verzichtet werden.

ISOLINUX

ISOLINUX wird zum Starten von ISO-9660-Dateisystemen verwendet. Es wird hierfür ein CD-ROM-Laufwerk benötigt, das der *Eltorito*-Spezifikation genügt. Da dieser Standard bereits 1995 herausgegeben worden ist, brauchen Sie sich hierüber nicht mehr den Kopf zu zerbrechen.

Wenn Sie einmal selbst eine startfähige CD mit ISOLINUX erstellen wollen, müssen Sie folgende Vorbereitungen treffen:

- Laden Sie *syslinux* von *http://www.syslinux.org/* herunter. Alternativ können Sie *https://www.kernel.org/pub/linux/utils/boot/syslinux/* durchsuchen.
- Erstellen Sie ein Verzeichnis mit dem Namen *cd-inhalt*.
- Kopieren Sie alle Daten, die Sie später auf der CD benötigen, in das Verzeichnis *cd-inhalt*.
- Erstellen Sie unterhalb von *cd-inhalt* die Verzeichnisse *isolinux*, *kernel* und *images*.
- Kopieren Sie aus dem heruntergeladenen Paket die Datei *core/isolinux.bin* in Ihr Verzeichnis *cd-inhalt/isolinux*.
- Erstellen Sie im Verzeichnis *cd-inhalt/isolinux* die Konfigurationsdatei *isolinux.cfg*. Ein Beispiel für eine solche Datei finden Sie weiter unten.
- An dieser Stelle benötigen Sie ein fertiges Disketten-Image, das Sie in das Verzeichnis *cd-inhalt/images* kopieren.
- Kopieren Sie jetzt *memdisk/memdisk* aus dem SYSLINUX-Paket in Ihr Verzeichnis *cd-inhalt/kernel*.
- Installieren Sie `mkisofs`, falls dieses Programm noch nicht auf Ihrem Computer vorhanden ist.

Nachdem alle Vorbereitungen getroffen sind, können Sie das ISO für Ihre Boot-CD erstellen. Verwenden Sie dafür dieses Kommando:

```
root@archangel:~# mkisofs -o output.iso -b \
isolinux/isolinux.bin -c isolinux/boot.cat \
-no-emul-boot -boot-load-size 4 -boot-info-table cd-inhalt
```

Ein Beispiel für die Konfigurationsdatei *isolinux.cfg*:

```
display boot.txt
prompt 1
default 1
```

```
# Boot other devices
label a
    localboot 0x00
label b
    localboot 0x80
label c
    localboot -1
# PC-DOS
label 1
    kernel /kernel/memdisk
    append initrd=/images/tools.imz
# Darik´s Boot and Nuke
label 2
    kernel /kernel/memdisk
    append initrd=/images/bootnuke.imz
# MemTest
label 3
    kernel /kernel/memtp170
```

Diese Konfiguration stammt von der Website *http://www.syslinux.org*. Die entsprechenden Datei *boot.txt* ist so angegeben:

```
☼09a☼07 - Boot A:
☼09b☼07 - Boot first HDD
☼09c☼07 - Boot next device

☼091☼07 - ☼0fPC-DOS☼07
☼092☼07 - Darik's Boot and Nuke
☼093☼07 - memtest86
```

Isohybrid

ISO-9660-Dateisysteme sind mittels ISOLINUX startfähig, wie Sie gerade gelesen haben. Voraussetzung dafür ist aber, dass diese auf einem optischen Medium, wie einer CD oder einer DVD, vorliegen. Soll der Start stattdessen von einem USB-Stick aus erfolgen, ist zusätzlich ein MBR notwendig. Einen solchen MBR können Sie mithilfe von Isohybrid einem fertigen ISO-9660-Dateisystem hinzufügen. Dazu können Sie alternativ auch das Programm *xorriso* verwenden.

Die genaue Vorgehensweise brauchen Sie für die Prüfung nicht zu kennen, aber Sie sollten wissen, welche beiden Dateien des SYSLINUX-Projekts Sie benötigen, die den MBR enthalten, nämlich:

- *isohdpfx.bin*, wenn von einem konventionellen BIOS gestartet wird.
- *efiboot.img*, wenn der Start von UEFI durchgeführt wird.

EXTLINUX

Ein verhältnismäßig neuer Bootloader des SYSLINUX-Projekts ist EXTLINUX. Es handelt sich hierbei um ein SYSLINUX-Derivat und dient dem Starten von Betriebssystemen, die sich auf Linux-Dateisystemen befinden. Bisher werden *ext2*, *ext3*, *ext4* und *btrfs* unterstützt.

Die Installation geschieht im Telegrammstil so:

- Laden Sie von *https://www.kernel.org/pub/linux/utils/boot/syslinux/* die benötigten Pakete herunter, falls noch nicht geschehen.
- Erstellen Sie das Verzeichnis */boot/extlinux*.
- Führen Sie `extlinux --install /boot/extlinux` aus. Das Programm finden Sie in dem heruntergeladenen SYSLINUX-Paket.
- Erstellen Sie in dem eben erstellten Unterverzeichnis */boot/extlinux* die Konfigurationsdatei *extlinux.conf*. Die Syntax dieser Datei ist identisch mit der von *syslinux.conf*.
- Jetzt wird es sehr roh: Der Inhalt der Datei *mbr.bin* muss in den Master Boot Record implementiert werden. Mit `cat mbr.bin > /dev/sda` ist das schnell erledigt. Sie können natürlich auch `dd` verwenden, wenn es Ihnen angenehmer ist. Die genaue Syntax für diesen Befehl kennen Sie ja bereits.

Die Datei *mbr.bin* enthält übrigens lediglich 440 Bytes Daten (je nach Version leicht abweichend). Das ist auch der Grund, warum die Partitionstabelle nicht durch das oben angegebene Kommando überschrieben und somit unbrauchbar gemacht wird.

PXELINUX

Das *PXE*-Protokoll ist eine Kombination aus den Protokollen DHCP und TFTP mit ein paar kleinen Anpassungen. *PXE* steht für Preboot Execution Environment und wird für Computer verwendet, auf denen lokal kein Betriebssystem installiert ist und möglicherweise auch nicht installiert werden soll. Voraussetzung ist eine PXE-fähige Netzwerkkarte und eine entsprechende Unterstützung im *BIOS*. Computer, die bereits über *UEFI* verfügen, sind grundsätzlich *PXE*-fähig.

Beim Systemstart führt die *PXE*-fähige Netzwerkkarte zunächst einen DHCP-Client aus. Es werden dann zunächst die normalen vier Phasen eines DHCP-Lease-Vorgangs ausgeführt (DHCPDISCOVER, DHCPOFFER, DHCPREQUEST und DHCPACK). Zusätzlich zum normalen DHCP-Vorgang wird aber anschließend ein *Boot Service Discover* gesendet, in dem der Client nach einem TFTP-Server fragt, der ein für den Systemstart geeignetes Betriebssystem-Image bereitstellt. Der DHCP-Server sollte dann mit einem *Boot Service Ack* antworten, das die Information zu dem benötigten TFTP-Server enthält.

Der Client verfügt nun über eine gültige IP-Adresse und die Information dazu, von welcher Quelle er ein Systemabbild erhält. Er kontaktiert den TFTP-Server mit einem *Bootstrap Program Download Request*. Wenn alles wie geplant funktioniert, antwortet der TFTP-Server mit einem *Bootstrap Program Download*, und der Client erhält sein Betriebssystem. Dieser Mechanismus wird heutzutage auch eingesetzt, um netzwerkgestützte Betriebssysteme auf viele Computer gleichzeitig zu verteilen. Diese Computer starten dann später ganz normal von den Festplatten aus.

Wenn Sie eine PXE-Umgebung mit PXELINUX nachbauen wollen, benötigen Sie zunächst einen DHCP-Server und einen TFTP-Server. Die Konfiguration von DHCP-Servern ist im zweiten Teil dieses Buchs detailliert beschrieben. Es sind allerdings einige spezielle Anpassungen erforderlich, wenn ein Bootservice unterstützt werden soll. Als TFTP-Server kommt z. B. *atftpd* infrage. Das ist ein ausgezeichneter TFTP-Server, der verhältnismäßig einfach zu konfigurieren ist. Man legt für diesen Server ein Hauptverzeichnis (üblicherweise */tftpboot*) an. In dieses Verzeichnis kopieren Sie die Datei *pxelinux.0* aus dem Paket SYSLINUX. Zusätzlich legen Sie hier ein Unterverzeichnis *pxelinux.cfg* an. Dieses Unterverzeichnis dient später zur Ablage diverser Konfigurationsdateien, die Äquivalente zur Datei *syslinux.cfg* darstellen. Die Client-Systeme suchen jeweils ihre eigene Konfigurationsdatei, und zwar zunächst gemäß ihrer MAC-Adressen. Die entsprechende Konfigurationsdatei muss dann einen Namen in Kleinschreibweise (!), wie in folgendem Beispiel haben:

/tftpboot/pxelinux.cfg/e0-b9-a5-7f-0e-7b

Wenn eine solche Datei nicht vorhanden ist, sucht der Client nach einer Datei, die seiner IP-Adresse, jedoch in hexadezimaler Schreibweise entspricht. Für die IP-Adresse 192.0.2.91 wäre das z. B. *C000025B*. Die entsprechende Konfigurationsdatei (diesmal übrigens in Großschreibweise) wäre dann:

/tftpboot/pxelinux.cfg/C000025B

Wird der Client wieder nicht fündig, entfernt er bei weiteren Anfragen jeweils ein Zeichen der HEX Adresse, also *C000025*, *C00002*, *C0000* usw. Als letzten Ausweg fragt der Client nach der Konfigurationsdatei *default*.

Die genaue Konfiguration einer PXE-Umgebung ist für die Prüfung nicht relevant. Wenn Sie jetzt neugierig geworden sind, finden Sie genauere Informationen unter:

http://www.syslinux.org/wiki/index.php/PXELINUX

Shim-Bootloader

Damit *EFI Secure-Boot* verwendet werden konnte, war es ursprünglich notwendig, dass ein Bootloader von Microsoft signiert wurde. Wenn man auf Secure-Boot verzichten will, kann man es einfach deaktivieren und zum Systemstart die nicht signierte Binärdatei *grubx64.efi* verwenden.

Wenn allerdings Secure-Boot verwendet werden soll, muss die Datei *grubx64.efi* mit einem Schlüssel versehen und anschließend in *shim.efi* umbenannt werden. Danach ist sie durch den Bootloader *Shim* verwendbar.

systemd-Boot und U-Boot

Über die Bootloader *systemd-Boot* und *U-Boot* brauchen Sie sich für die Prüfung kein Detailwissen anzueignen.

systemd-Boot (ursprünglich *Gummiboot* genannt) ist ausschließlich dazu geeignet, EFI-Images auszuführen. Da *systemd-Boot* keine Kernel aus anderen Partitionen starten kann, ist ein ausführbarer Kernel innerhalb der EFI System Partition von Nöten.

Bei *U-Boot* (eigentlich »Das U-Boot«) handelt es sich um einen Bootloader, der unter anderem auf Microcontrollern lauffähig ist. Deshalb wird *U-Boot* häufig auf Embedded Systemen eingesetzt.

203 Dateisystem und Devices

Das Langzeitgedächtnis eines Computers befindet sich auf unterschiedlichsten Datenträgern. Damit der Computer diese Datenträger benutzen kann, müssen sie mit einem Dateisystem versehen und entsprechend eingerichtet worden sein.

203.1 Arbeiten mit dem Linux-Dateisystem

Wichtung: 4

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, ein Standard-Linux-Dateisystem zu konfigurieren und darin zu arbeiten. Dieses Lernziel enthält die Konfiguration und das Einbinden verschiedener Dateisysteme.

Wichtigste Wissensgebiete:

- Konzept der *fstab*-Konfiguration
- Werkzeuge und Dienstprogramme für das Arbeiten mit Swap-Partitionen und Swap-Dateien
- Benutzung von UUIDs zur Identifikation und zum Einhängen von Dateisystemen
- Verständnis der *systemd-Mountunits*

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/fstab*
- */etc/mtab*
- */proc/mounts*
- `mount` und `umount`
- `blkid`
- `sync`
- `swapon`
- `swapoff`

Allgemeines

Wie Sie wissen, werden Laufwerke unter Linux bei Bedarf ein- oder ausgehängt. Dieser Vorgang kann manuell ausgeführt oder durch Einträge in der Datei */etc/fstab*

automatisiert werden. Im Normalfall darf das Ein- und Aushängen eines Dateisystems nur vom root-Benutzer durchgeführt werden. Ausnahmen von dieser Regel können in der Datei */etc/fstab* festgelegt werden.

Manuelles Mounten und Unmounten

Eigentlich ist zu vermuten, dass Sie dieses Thema längst aus dem Effeff beherrschen, aber ich fange hier trotzdem noch einmal von Anfang an, damit das Thema bei Ihnen prüfungssicher sitzt.

In der deutschen Umgangssprache wird der Vorgang des Einhängens eines Dateisystems als »mounten« bezeichnet. Hierbei handelt es sich um einen »verbogenen« englischen Ausdruck. Der Begriff »mount« bedeutet in diesem Kontext so viel wie »montieren« oder »anbringen«. Das Kommando, das zum Einhängen eines Dateisystems verwendet wird, heißt aus verständlichen Gründen ebenfalls `mount`. Soll das Dateisystem wieder ausgehängt werden, verwendet man `umount`. Achtung: Es heißt nicht `unmount`.

mount

Wenn Sie das Kommando `mount` ohne Optionen und Argumente verwenden, erhalten Sie eine Auflistung der momentan in den Verzeichnisbaum eingehängten Dateisysteme. Beispiel:

```
archangel:~ # mount
/dev/hda2 on / type ext3 (rw,acl,user_xattr)
/dev/hda4 on /storage type reiserfs (rw,usrquota,grpquota)
/dev/hda1 on /boot type ext2 (rw,acl, user_xattr)
proc on /proc type proc (rw)
/dev/fd0 on /media/floppy type subfs (rw, sync,fs=floppyfss)
//connor/daten on /mnt/connor type smbfs (0)
arch-srv:/storage on /storage type nfs (rw,sync,addr=192.168.50.47)
```

Wenn Sie mit dem Kommando `mount` Dateisysteme in den Verzeichnisbaum einhängen wollen, erwartet `mount` die Angabe von Optionen, den Pfad zum Gerät, das eingehängt werden soll, und den Pfad zum Einhängepunkt (Mountpoint). Beispiel:

```
[root@fedora /]# mount -t ext3 /dev/sdb1 /dateien
```

Die Option `-t ext3` gibt den Dateisystemtyp an, */dev/sdb1* ist die Gerätedatei der Partition, die eingehängt werden soll, und das Verzeichnis */dateien* ist der Mountpoint. Wenn die Zuordnung von Dateisystem zu Mountpoint und Dateisystemtyp bereits vorab in der Datei */etc/fstab* (die Funktionsweise der Datei */etc/fstab* wird auf den nächsten Seiten ausführlich beschrieben) vorgenommen wurde, reicht es aus, nur

den Mountpoint oder nur das einzuhängende Dateisystem anzugeben. Folgende Kommandos führen zu demselben Ergebnis, wenn das Dateisystem aus dem vorangehenden Beispiel in die Datei *fstab* eingetragen worden ist:

```
[root@fedora /]# mount /dateien
[root@fedora /]# mount /dev/sdb1
```

Auf dieselbe Art werden auch Diskettenlaufwerke, CD-ROMs und DVD-Laufwerke eingehängt. Beim Einhängen von USB-Memory-Sticks ist zu beachten, dass diese dieselben Gerätedateien verwenden wie SCSI-Laufwerke. Eine CD können Sie z. B. so einhängen:

```
root@ubuntu-server:~# mount -t iso9660 /dev/hdc /media/cdrom0
```

Sie können u. a. die folgenden Kommandozeilenoptionen mit dem `mount`-Befehl verwenden:

- `-a` (all) hängt alle Dateisysteme ein, die in der Datei */etc/fstab* gelistet sind.
- `-r` (read-only) hängt ein Dateisystem schreibgeschützt ein.
- `-w` (writeable) hängt ein Dateisystem im Read/Write Mode ein.
- `-t` (type) gibt den Dateisystemtyp an, falls erforderlich. Folgende Arten von Dateisystemen werden unterstützt:

  `adfs`, `affs`, `autofs`, `coda`, `coherent`, `cramfs`, `devpts`, `efs`, `ext`, `ext2`, `ext3`, `hfs`, `hpfs`, `iso9660`, `jfs`, `minix`, `msdos`, `ncpfs`, `nfs`, `ntfs`, `proc`, `qnx4`, `ramfs`, `reiserfs`, `romfs`, `smbfs`, `sysv`, `tmpfs`, `udf`, `ufs`, `umsdos`, `usbfs`, `vfat`, `xenix`, `xfs`, `xiafs`
- `-v` (verbose) sorgt für eine Bestätigung des Vorgangs. Der `mount`-Befehl zeigt normalerweise keine Erfolgsmeldungen an.
- `-o` (options) übergibt Mount-Optionen. Die wichtigsten dieser Optionen sollten Sie kennen.

Mount-Optionen (mount -o)

Manchmal ist es notwendig, während des Einhängens eines Dateisystems Optionen zu übergeben. Diese Optionen werden kommagetrennt mit der Kommandozeilenoption `-o` übermittelt. Das kann z. B. nötig sein, wenn Sie ein Laufwerk, das mit der Option `-r` (read-only) gemountet worden ist, beschreiben müssen. Wenn Sie dieses Laufwerk einfach aushängen und dann schreibbar wieder einhängen, gehen nicht gespeicherte Daten von Benutzern, die eventuell über das Netzwerk auf das System zugreifen, verloren. Ein anderer Grund könnte sein, dass Sie das Dateisystem neu einhängen wollen, auf dem sich der `mount`-Befehl selbst befindet. Wenn Sie dieses Dateisystem zuerst aushängen, haben Sie keine Möglichkeit mehr, das Kommando `mount` erneut auszuführen. Das geschieht nicht, wenn Sie das Dateisystem stattdessen beschreibbar remounten:

```
[root@fedora /]# mount -v -o remount,rw /dateien
```

Beachten Sie die kommagetrennte Übergabe beider Optionen. Wenn eine Authentifizierung beim Zugriff auf entfernte Ressourcen notwendig ist, können Benutzername und Passwort ebenfalls als kommaseparierte Optionen übergeben werden:

```
archangel:~ # mount -t smbfs -o username=martha,password=geheim //connor/
daten /mnt/connor/
```

Weitere Optionen finden Sie auf den nächsten Seiten im Zusammenhang mit der Datei */etc/fstab*, denn in dieser Konfigurationsdatei finden dieselben Optionen Verwendung wie auf der Kommandozeile. In der Prüfung werden Mount-Optionen oft mit der Datei */etc/fstab* in Zusammenhang gebracht.

Netzwerkressourcen einhängen

Mit dem Kommando `mount` können auch entfernte Dateisysteme eingehängt werden. Wenn Sie auf einen NFS-Server zugreifen wollen, verwenden Sie den `mount`-Befehl wie folgt:

```
root@ubuntu-server:~# mount -t nfs archangel:/storage /daten
```

Als Dateisystemtyp wird also einfach `nfs` angegeben. Der Mountpoint ist in diesem Fall das lokale Verzeichnis */daten*. Merken Sie sich unbedingt, wie die Syntax für das einzuhängende Dateisystem bei NFS aussieht:

```
hostname:/exportiertes_Verzeichnis
```

Im vorangehenden Beispiel ist demnach auf dem Host *archangel* das Verzeichnis */storage* exportiert worden. Es gibt einen erheblichen Unterschied zum Einhängen von SMB-Dateisystemen: Mit dem Dateisystemtyp `smbfs` können sowohl Windows-Freigaben als auch Shares auf einem Samba-Server angesprochen werden. Die Syntax für den `mount`-Befehl sieht dann so aus:

```
archangel:~ # mount -t smbfs //connor/daten /mnt/connor
```

In diesem Fall existiert auf dem Host *connor* eine Freigabe mit der Bezeichnung *daten*. Diese Freigabe wird in das lokale Verzeichnis */mnt/connor* auf dem Host *archangel* eingehängt. Im Normalfall muss beim Zugriff auf eine SMB-Share ein Benutzername und ein Passwort angegeben werden. Verwenden Sie dazu die Option `-o` wie folgt:

```
archangel:~ # mount -t smbfs -o username=martha,password=geheim //connor/
daten /mnt/connor/
```

umount

Wenn ein Dateisystem nicht mehr benötigt wird, können Sie es mit dem Kommando `umount` aushängen. Das kann auch dann erforderlich sein, wenn Sie ein Dateisystem mit `fsck` überprüfen wollen. Sie können auch Optionen mit `umount` verwenden, aber das ist in der Regel nicht nötig. Sie können beim Aushängen eines Dateisystems entweder die Gerätedatei oder den Mountpoint des zu entfernenden Dateisystems als Argument übergeben. Wenn z. B. das Dateisystem */dev/hda1* auf dem Mountpoint */boot* eingehängt worden ist, führen die folgenden beiden Kommandos zu demselben Ergebnis:

```
archangel:/ # umount /dev/hda1
archangel:/ # umount /boot
```

Sie können auch alle Dateisysteme, die in der Datei */etc/fstab* eingetragen sind, mit einem einzigen Kommando aushängen, nämlich:

```
archangel:/ # umount -a
```

Das ist aber normalerweise nicht sinnvoll.

Prüfungstipp

In der Prüfung geht es eigentlich eher selten um das Kommando `umount` als solches, sondern vor allem darum, wer es verwenden darf. Im Normalfall darf man nur root-Dateisysteme ein- bzw. aushängen. Ausnahmen von dieser Regel werden in der Datei */etc/fstab* über die entsprechenden Optionen festgelegt.

Automatisches Mounten über die Datei /etc/fstab

Wenn auf Dateisysteme regelmäßig zugegriffen wird, ist es natürlich sinnvoll, dass diese schon beim Systemstart in den Verzeichnisbaum eingehängt werden. Aber auch dann, wenn ein Dateisystem erst im laufenden Betrieb eingehängt werden soll, bringt die Verwendung der Datei */etc/fstab* einige Vorteile mit sich. Sie können Dateisysteme, die hier aufgeführt sind, mit dem verkürzten `mount`-Befehl wesentlich bequemer einhängen. So entfällt die Angabe des Dateisystemtyps und wahlweise des Mountpoints oder der einzuhängenden Partition. Darüber wurde aber bereits im Zusammenhang mit dem Kommando `mount` ausführlich berichtet. Hier sehen Sie ein Beispiel für eine *fstab*-Datei:

```
archangel:/ # cat /etc/fstab
/dev/hda2     /            ext3      acl,user_xattr     1 1
/dev/hda1     /boot        ext2      acl,user_xattr     1 2
/dev/hda4     /storage     reiserfs  defaults           1 2
```

```
/dev/hda3      swap          swap      pri=42          0 0
proc           /proc         proc      defaults        0 0
usbfs          /proc/bus/usb usbfs     noauto          0 0
host7:/share   /mnt/host7    nfs       defaults        0 0
```

Die sechs Spalten der Datei */etc/fstab* müssen Sie für die Prüfung genau kennen:

1. Die Spalte enthält das einzuhängende Gerät. Hierbei kann es sich auch um Ressourcen auf einem anderen System handeln, wie Sie in der letzten Zeile des Beispiels sehen.
2. Die Spalte enthält den Mountpoint.
3. Die Spalte bezeichnet den Dateisystemtyp.
4. Die Spalte enthält Mount-Optionen (eine Auflistung folgt).
5. Die Spalte ist eine Information für das Sicherungsprogramm `dump`. Wenn hier eine Null steht, wird das Dateisystem von `dump` nicht gesichert. Dateisysteme, die nicht ständig eingehängt sind, sollten nicht über die Datei *fstab* zur Sicherung mit `dump` vorgemerkt werden.
6. Die Spalte enthält Informationen für `fsck`. Hier wird festgelegt, ob und in welcher Reihenfolge das Programm `fsck` die Dateisysteme beim Systemstart prüfen soll:
   - `0` wird nicht geprüft.
   - `1` wird vorzugsweise geprüft.
   - `2` wird geprüft, nachdem Dateisysteme mit dem Wert `1` in diesem Feld bereits geprüft worden sind.

Mount-Optionen

Mount-Optionen können im vierten Feld der Datei */etc/fstab* verwendet oder beim manuellen Einhängen von Dateisystemen mit `mount -o` übergeben werden. Einige dieser Optionen sind aber erst bei der Verwendung mit der Datei *fstab* sinnvoll. Wichtige Optionen sind:

- `auto` ermöglicht ein automatisches Einhängen mittels `mount -a`.
- `noauto` verhindert ein automatisches Einhängen mittels `mount -a`.
- `usrquota` aktiviert die Möglichkeit der Verwendung von Quota auf der Benutzerebene.
- `grpquota` aktiviert die Möglichkeit der Verwendung von Quota auf der Gruppenebene.
- `suid` ermöglicht die Funktion der SUID-Bits.
- `nosuid` verhindert aus Sicherheitsgründen die Funktion der SUID-Bits.
- `exec` erlaubt das Ausführen von Dateien auf diesem Dateisystem.

- `noexec` verhindert das Ausführen von Dateien auf diesem Dateisystem.
- Mit `ro` wird das Dateisystem im Read-only Mode eingehängt.
- Mit `rw` wird das Dateisystem im Read/Write Mode eingehängt.
- `user` erlaubt es einem normalen Benutzer, dieses Dateisystem einzuhängen. Das Dateisystem kann dann nur von demselben Benutzer oder vom root ausgehängt werden.
- `nouser` verhindert, dass ein normaler Benutzer ein Dateisystem einhängen kann.
- `users` erlaubt es einem normalen Benutzer, dieses Dateisystem einzuhängen. Das Dateisystem kann von einem beliebigen Benutzer wieder ausgehängt werden.
- `defaults` setzt die Standardeinstellungen. Ausnahmen werden kommasepariert übergeben. Defaults sind: `rw`, `suid`, `dev`, `exec`, `auto`, `nouser`, `async`.

Prüfungstipp

Achten Sie bitte besonders darauf, die Optionen `user` und `users` nicht miteinander zu verwechseln.

Die Datei /etc/mtab

Wenn ein Dateisystem eingehängt wird, trägt das Programm `mount` das Dateisystem, den Mountpoint und eventuell übergebene Optionen in die Datei */etc/mtab* ein. Eine Ausnahme stellen Dateisysteme dar, die mit der Option `-n` gemountet worden sind. Beispiel:

```
archangel:~ # cat /etc/mtab
/dev/hda2 / ext3 rw,acl,user_xattr 0 0
proc /proc proc rw 0 0
sysfs /sys sysfs rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
devpts /dev/pts devpts rw,mode=0620,gid=5 0 0
/dev/hda4 /storage reiserfs rw,usrquota,grpquota 0 0
usbfs /proc/bus/usb usbfs rw 0 0
nfsd /proc/fs/nfsd nfsd rw 0 0
/dev/hda1 /boot ext2 rw,acl,usrquota,grpquota,user_xattr 0 0
```

Im */proc*-Dateisystem werden eingehängte Dateisysteme in jedem Fall aufgeführt. Auf diese Art können Sie sich auch die Dateisysteme anzeigen lassen, denen beim Einhängen mit `mount` die Option `-n` übergeben worden ist.

```
archangel:~ # cat /proc/mounts
rootfs / rootfs rw 0 0
/dev/root / ext3 rw 0 0
```

```
proc /proc proc rw,nodiratime 0 0
sysfs /sys sysfs rw 0 0
devpts /dev/pts devpts rw 0 0
tmpfs /dev/shm tmpfs rw 0 0
/dev/hda4 /storage reiserfs rw 0 0
usbfs /proc/bus/usb usbfs rw 0 0
nfsd /proc/fs/nfsd nfsd rw 0 0
/dev/hda1 /boot ext2 rw 0 0
```

Prüfungstipp

Eignen Sie sich für die Prüfung auch besonders die Fähigkeit an, die Datei */etc/fstab* genau auswerten zu können. Sie müssen z. B. antworten können, wenn Sie gefragt werden, was das fünfte Feld in dieser Datei bewirkt.

Dateisystempuffer entleeren

Es kann passieren, dass Sie das System veranlassen müssen, den Schreibcache zu entleeren. Wenn Sie etwa eine USB-Festplatte vom System entfernen wollen, obwohl der `umount`-Befehl aus irgendeinem Grund fehlgeschlagen ist, können Sie durch die Verwendung des Kommandos `sync` erzwingen, dass geänderte Blöcke auf die Festplatte geschrieben werden und der Superblock aktualisiert wird. Auf diese Weise können Sie halbwegs sicher sein, dass Sie keine Daten verlieren, weil sich diese noch im Dateisystempuffer befanden. Ein anderer Grund könnte sein, dass ein Computer beim Herunterfahren hängt. Auch in dieser Situation sollten Sie die Dateisysteme syncen, bevor Sie das System hart ausschalten. Es ist natürlich extrem unwahrscheinlich, dass Sie das jemals erleben werden, wenn Sie Linux einsetzen. Das Kommando `sync` versteht übrigens keine Optionen, die Sie für die Prüfung kennen müssen.

Das Paging ein- bzw. ausschalten

Als Paging bezeichnet man den Vorgang, bei dem der Kernel Seiten aus dem Arbeitsspeicher auf ein Swap-Dateisystem oder in eine Swap-Datei aus- oder einlagern muss. Diesen Vorgang führt der Kernel durch, wenn nicht genügend physikalischer Arbeitsspeicher zur Verfügung steht. Unter Linux wird man im Normalfall eine Partition für die Auslagerung nicht benötigter Speicherseiten erstellen. Es besteht aber auch die Möglichkeit, lediglich eine Auslagerungsdatei zu verwenden, wie es auch Windows macht. Da eine solche Auslagerungsdatei fragmentieren kann und das Paging hierdurch langsam wird, ist das aber nicht die empfohlene Vorgehensweise. Die Erstellung der Dateisysteme für das Paging wird auf den nächsten Seiten genauer beschrieben. An dieser Stelle soll es reichen, das Paging zu aktivieren bzw. zu deaktivieren. Sie können, in dem Fall, dass Sie mehrere Dateisysteme oder Dateien für das

Paging konfiguriert haben, diese einzeln ein- bzw. ausschalten, oder Sie können alle auf einmal ansprechen. Da die hierfür benötigten Programme von Haus aus recht schweigsam sind, sehen Sie hier je ein Beispiel mit der Option `-v`:

```
root@arch-deb-book:/# swapoff -v /dev/sda7
swapoff on /dev/sda7
```

Jetzt ist das Paging auf */dev/sda7* deaktiviert. Mit `swapon` wird es wieder eingeschaltet:

```
root@arch-deb-book:/# swapon -v /dev/sda7
swapon on /dev/sda7
swapon: /dev/sda7: found swap signature: version 1, page-
size 4, same byte order
swapon: /dev/sda7: pagesize=4096, swapsize=6226444288, devsize=6226444288
```

Weitere wichtige Optionen für `swapon` sind:

- `-a` aktiviert alle Swap-Dateisysteme, die in der Datei */etc/fstab* definiert sind.
- `-L` wird verwendet, wenn Sie ein Swap-Dateisystem über ein Label ansprechen wollen.
- `-U` spricht ein Swap-Dateisystem über den UUID an.
- `-s` zeigt die Verwendung der Swap-Dateisysteme an.

Mit dem Kommando `swapon -s` kann man überprüfen, welche Dateien bzw. Dateisysteme ein Computer gerade für das Paging verwendet.

```
root@arch-deb-book:/home/harald# swapon -s
Filename        Type         Size      Used      Priority
/dev/sda7       partition    6080504    0         -1
```

UUIDs verwenden

Bevor es UUIDs gab, konnte es passieren, dass ein System Partitionen an den falschen Mountpoints eingehängt hat, weil z. B. eine Festplatte an einen anderen Anschluss innerhalb des Systems angeschlossen wurde. Das konnte ein anderer IDE-, SCSI- oder auch einfach ein USB-Anschluss sein. Heutzutage wird jeder Partition oder jedem Volumen ein eindeutiger Bezeichner gegeben. Man nennt diese Bezeichner Universally Unique Identifier. Aufgrund ihrer Komplexität und Länge ist es sehr unwahrscheinlich, dass zwei Geräte zufällig dieselbe ID erhalten. Sie können die UUIDs Ihrer Geräte ermitteln, indem Sie das Kommando `blkid` verwenden:

```
root@arch-deb-book:/home/harald# blkid
/dev/sda1: UUID="c8e624bb-5ec6-4e69-afab-8b0e7df33067" TYPE="ext3"
/dev/sda5: UUID="90fd4b0c-49d8-4da3-8649-6f5dfe0d83f7" TYPE="ext3"
/dev/sda6: UUID="9ff36df5-6bff-4d4c-9243-c163e3e8d24b" TYPE="ext3"
```

```
/dev/sda7: UUID="40ea2bd8-fc6e-4c25-88e6-5855f69668f8" TYPE="swap"
/dev/sda8: UUID="37609c74-f447-4399-bf8c-f80b3a93f0d0" TYPE="ext3"
/dev/sda9: UUID="6958a017-26e6-4b00-80df-e60298e4d7c8" TYPE="ext3"
```

Bei vielen heutigen Systemen werden die Mountpoints innerhalb der Datei */etc/fstab* nicht mehr an den Gerätedateien festgemacht, sondern eben an den UUIDs. Da diese IDs auch auf den Geräten selbst abgespeichert werden, können die Dateisysteme völlig unabhängig von ihrem Anschluss identifiziert und immer an derselben Stelle des Verzeichnisbaums eingehängt werden. Hierdurch werden auch Systemstartprobleme vermieden, die sonst durch Umkonfigurieren der Hardware eines Computers entstehen könnten.

Die Zuordnung der UUIDs zu den klassischen Gerätedateien geschieht durch Softlinks. Im Verzeichnis */dev/disk/by-uuid* existiert für jede Partition bzw. jedes Volumen ein Link mit dem Namen des UUID, der auf die richtige Gerätedatei verweist:

```
root@arch-deb-book:/dev/disk/by-uuid# ls -l
insgesamt 0
lrwxrwxrwx 1 root root 10 27. Apr 17:43 37609c74-f447-4399-bf8c-f80b3a93f0d0
-> ../../sda8
lrwxrwxrwx 1 root root 10 27. Apr 19:55 40ea2bd8-fc6e-4c25-88e6-5855f69668f8
-> ../../sda7
lrwxrwxrwx 1 root root 10 27. Apr 17:43 6958a017-26e6-4b00-80df-e60298e4d7c8
-> ../../sda9
lrwxrwxrwx 1 root root 10 27. Apr 17:43 90fd4b0c-49d8-4da3-8649-6f5dfe0d83f7
-> ../../sda5
lrwxrwxrwx 1 root root 10 27. Apr 17:43 9ff36df5-6bff-4d4c-9243-c163e3e8d24b
-> ../../sda6
lrwxrwxrwx 1 root root 10 27. Apr 17:43 c8e624bb-5ec6-4e69-afab-8b0e7df33067
-> ../../sda1
```

Auf den nächsten Seiten werden Sie sehen, wie man bei Bedarf den UUID eines Dateisystems ändern kann.

systemd-Mountunits

Mit *systemd* kommen auch einige *Mountunits*. Diese können entweder über Unit-Dateien oder über Einträge in der Datei */etc/fstab* angesteuert werden. Wenn Sie solche Einträge in der Datei */etc/fstab* vornehmen, werden diese während des Systemstarts dynamisch in Units umgewandelt. Der Vorteil gegenüber normalen Einträgen in der Datei *fstab* ist der, dass *systemd-Mountunits* mit Abhängigkeiten konfiguriert werden können. So kann man sicherstellen, dass beim Einbinden von Netzwerkshares das Netzwerk bereits gestartet wurde. Das ist besonders bei mobilen

Computern wichtig, weil WLAN in der Regel überhaupt erst nach der Benutzeranmeldung zur Verfügung steht – also deutlich nach dem Abarbeiten der Datei *fstab*. Die *systemd-Mountunits* sorgen dann dafür, dass Netzwerkressourcen nachträglich eingebunden werden.

Wie das nachträgliche Einbinden ohne *systemd* mittels *Automounter* funktioniert, können Sie später in diesem Kapitel lesen. Der folgende Eintrag einer *fstab*-Datei wurde auf zwei Zeilen aufgeteilt und kommentiert:

```
archangel:/home/harald          /home/harald/Schreibtisch/archangel          nfs
```

Bei `archangel:/home/harald` handelt es sich um eine Netzwerkressource. `/home/harald/Schreibtisch/archangel` ist der zugehörige Mountpoint. Das Netzwerkdateisystem ist `nfs`. Bis hierher gibt es also nichts, was Sie nicht schon kennen. Weiter geht es mit:

```
noauto,x-systemd.automount,x-systemd.device-timeout=10,x-systemd.idle-timeout=
1min 0 0
```

Zunächst wird mit `noauto` festgelegt, dass das Dateisystem nicht automatisch eingehängt werden soll. Diese Aufgabe wird im Anschluss mit `x-systemd.automount` sofort an *systemd* übergeben. `x-systemd.device-timeout=10` legt fest, dass *systemd* nach 10 Sekunden aufgibt, wenn sich das Gerät nicht einhängen lässt. `x-systemd.idle-timeout=1min` legt fest, dass *systemd* diese Ressource aushängen soll, wenn diese seit einer Minute nicht mehr verwendet wurde. Die abschließenden Nullen kennen Sie bereits.

203.2 Pflege des Linux-Dateisystems

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, ein Standard-Linux-Dateisystem mit den entsprechenden Systemwerkzeugen zu pflegen. Dieses Lernziel beinhaltet das Verändern der Standarddateisysteme.

Wichtigste Wissensgebiete:

- Werkzeuge und Dienstprogramme für *ext2-*, *ext3-* und *ext4*-Dateisysteme
- Werkzeuge und Dienstprogramme für grundlegende *Btrfs*-Operationen, inklusive der Erstellung von *Subvolumes* und *Snapshots*
- Werkzeuge und Dienstprogramme für das *XFS*-Dateisystem
- Kenntnis von ZFS

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `mkfs` (`mkfs.*`)
- `mkswap`
- `fsck` (`fsck.*`)
- `tune2fs`, `dumpe2fs`, `debugfs`
- `btrfs`, `btrfs-convert`
- `xfs_info`, `xfs_check`, `xfs_repair`, `xfsdump` und `xfsrestore`
- `smartd`, `smartctl`

Allgemeines

Auf den letzten Seiten haben Sie einiges über das Einhängen von Dateisystemen erfahren. Sie sollten aber auch dazu in der Lage sein, die Integrität dieser Dateisysteme sicherzustellen und kleinere Probleme aufzufinden bzw. zu beseitigen. Hierzu zählt auch die Fähigkeit, Engpässe bei der Speicherkapazität schnell zu ermitteln und bei einem überfüllten Dateisystem festzustellen, in welchen Verzeichnissen der Speicherplatz verbraucht wird.

Sicherstellen der Integrität des Dateisystems und Problembehebung

fsck und e2fsck

Das Programm `fsck` ist ein Frontend zur Überprüfung von Dateisystemen. Es ruft zur Laufzeit andere Programme auf, die auf die Überprüfung der jeweiligen Dateisysteme spezialisiert sind. Zur Überprüfung von *ext2*-, *ext3*- oder *ext4*-Dateisystemen gibt es zusätzlich das Programm `e2fsck`. Hierbei handelt es sich lediglich um einen Hardlink mit den Programmen `fsck.ext2`, `fsck.ext3` und `fsck.ext4`. Die Beziehungen zwischen den Frontend- und den Backendprogrammen zur Dateisystemprüfung ähneln stark denen zwischen Formatierungsfrontends und deren Backends. Die Backends zur Dateisystemprüfung können natürlich auch direkt angesprochen werden. Da wären:

- `fsck.ext2` prüft *ext2*- und *ext3*-Dateisysteme. Dieses Backend ist hart verlinkt mit `fsck.ext3` und `e2fsck`.
- `fsck.ext3` ist ein Hardlink zu `fsck.ext2` und `e2fsck`.
- `fsck.ext4` ist ebenfalls lediglich ein Link zu den oben genannten Artgenossen.
- `fsck.minix` prüft MINIX-Dateisysteme.
- `fsck.cramfs` prüft CramFS-Dateisysteme.
- `fsck.xfs` prüft XFS-Dateisysteme.
- `fsck.jfs` prüft JFS-Dateisysteme.

- `fsck.msdos` und `fsck.vfat` sind Links zu `dosfsck`.

Sie sollten die folgenden Optionen von `fsck` kennen:

- `-f` (force) erzwingt die Prüfung, auch wenn das Dateisystem sauber erscheint.
- `-A` (all) testet alle Dateisysteme, die in */etc/fstab* aufgeführt sind.
- `-t Dateisystemtyp` (type) sorgt dafür, dass das zum Dateisystem passende Backend gestartet wird.
- `-c` (check) sucht nach defekten Blöcken.
- `-b Blocknummer` (block) gibt einen alternativen Superblock an.
- `-y` (yes) beantwortet alle Fragen des Programms mit »yes«, damit eine Reparatur unbeaufsichtigt durchgeführt werden kann.

Wenn Sie `fsck` ausführen, sollte das zu prüfende Dateisystem nicht eingehängt sein. Das gilt insbesondere, wenn zu Reparaturzwecken Schreibzugriffe von `fsck` durchgeführt werden müssen. Sie erhalten vor der Prüfung eines eingehängten Dateisystems eine Warnmeldung, wie das Beispiel zeigt:

```
[root@fedora ~]# fsck -f /dev/sdb1
fsck 1.40.2 (12-Jul-2007)
e2fsck 1.40.2 (12-Jul-2007)
/dev/sdb1 is mounted.
WARNING!!!  Running e2fsck on a mounted filesystem may cause
SEVERE filesystem damage.
Do you really want to continue (y/n)? n
```

Sicherheitshalber wird das Dateisystem also vorher mit `umount` ausgehängt:

```
[root@fedora ~]# umount /dev/sdb1
```

Anschließend kann die Überprüfung gefahrlos durchgeführt werden:

```
[root@fedora ~]# fsck -f /dev/sdb1
fsck 1.40.2 (12-Jul-2007)
e2fsck 1.40.2 (12-Jul-2007)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/sdb1: 11/124928 files (9.1% non-contiguous), 18084/497980 blocks
```

Mit dem Dateisystem war in diesem Fall alles in Ordnung. Ein Dateisystem wird standardmäßig alle 36 Mounts oder alle 180 Tage beim Systemstart geprüft (diese Werte

können in Abhängigkeit von der verwendeten Distribution abweichen). Um diese Prüfung für den nächsten Systemstart zu unterbinden, verwenden Sie `shutdown` mit der Option `-f`:

```
archangel:~ # shutdown -hf now
```

Das ist besonders wichtig, wenn Sie einen Server z. B. für Hardwarearbeiten herunterfahren müssen, dieser aber sehr schnell wieder verfügbar sein muss. Umgekehrt können Sie die Überprüfung für den nächsten Start erzwingen, wenn Sie die Option `-F` verwenden:

```
archangel:~ # shutdown -hF now
```

tune2fs

Das Programm `tune2fs` ermöglicht das Bearbeiten der Dateisystemparameter bei *ext2*-, *ext3*- oder *ext4*-Dateisystemen. So ist es z. B. möglich, die Intervalle für die automatische Überprüfung der Dateisysteme mittels `fsck` zu beeinflussen. Sie können die Intervalle entweder ändern oder die aktuellen Werte in den entsprechenden Zählern verändern. Die hierfür benötigten Optionen sind:

- `-c` legt die maximale Anzahl der Mount-Vorgänge zwischen den Dateisystemüberprüfungen fest.
- `-C` legt fest, wie oft das Dateisystem nach der letzten Überprüfung tatsächlich gemountet worden ist.
- `-i` legt das Intervall zwischen den Dateisystemprüfungen in Tagen, Wochen oder Monaten fest. Ohne Zeitangabe schaltet diese Option die regelmäßige Überprüfung aus.
- `-T YYYYMMDD[[HHMM]SS]` legt den Zeitpunkt fest, wann die letzte Dateisystemprüfung tatsächlich stattgefunden hat.
- `-U` wird verwendet, um den UUID (Universally Unique Identifier) eines Dateisystems zu ändern.

Wenn Sie ein *ext2*-Dateisystem in ein *ext3*-Dateisystem konvertieren wollen, verwenden Sie die Option `-j` (journal). Bei einem *ext3*-Dateisystem handelt es sich, wie bereits erläutert, um ein *ext2*-Dateisystem mit Journal. Deshalb ist die Konvertierung recht einfach:

```
[root@fedora ~]# tune2fs -j /dev/hdb3
tune2fs 1.40.2 (12-Jul-2007)
Erstelle Journal-Inode: erledigt
```

Das Dateisystem wird automatisch alle 36 Mounts bzw. alle 180 Tage überprüft, je nachdem, was zuerst eintritt. Veränderbar ist dieses Verhalten mit `tune2fs -c` oder `-t`.

Um die Informationen des Superblocks anzuzeigen, verwenden Sie einfach die Option `-l`. Da die vollständige Ausgabe dieses Kommandos mindestens zwei Buchseiten füllen würde, soll hier auf ein Beispiel verzichtet werden.

debugfs

Das Programm `debugfs` dient dazu, ein *ext2*-, *ext3*- oder *ext4*-Dateisystem interaktiv zu untersuchen oder zu modifizieren. Es gibt auch die Möglichkeit, mit `debugfs` gelöschte Dateien wiederherzustellen. Dazu muss zunächst das betroffene Dateisystem ausgehängt werden:

```
[root@fedora /]# umount /dev/sdb1
```

Anschließend wird eben dieses Dateisystem mit `debugfs` geöffnet:

```
[root@fedora /]# debugfs /dev/sdb1
```

Am interaktiven Prompt von `debugfs` geben Sie das Kommando `lsdel` ein, um gelöschte Inodes anzuzeigen:

```
debugfs:  lsdel
 Inode  Owner  Mode    Size    Blocks                Time deleted
    12      0 100600  12288       12/  12 Sun Dec 23 00:02:48 2007
    13      0 100644   7416        8/   8 Sun Dec 23 00:02:57 2007
2 deleted inodes found.
```

Davon ausgehend, dass die wiederherzustellende Datei auf Inode 13 befindlich war, muss jetzt noch folgendes Kommando verwendet werden:

```
debugfs:  dump <13> /tmp/restored
```

Anschließend kann das Dateisystem wieder gemountet und die Datei an ihren ursprünglichen Platz kopiert werden.

dumpe2fs

Mit `dumpe2fs` können detaillierte Informationen über ein *ext2*-, *ext3*- oder *ext4*-Dateisystem eingeholt werden. Um einen Überblick zu erhalten, starten Sie das Programm am besten ohne Optionen, aber mit einer Dateisystemangabe als Argument. Da die Ausgabe des Programms recht umfangreich ist, sollten Sie die Ausgabe an `less` weitergeben. Beispiel:

```
archangel:~ # dumpe2fs /dev/hda1 | less
```

Da auch `dumpe2fs` in der Prüfung eher peripher abgefragt wird, soll es hier nicht weiter thematisiert werden.

mke2fs

Sie können auch mit dem Formatierungsprogramm `mke2fs` die Integrität eines Dateisystems überprüfen. Verwenden Sie dazu die Option `-c`:

```
[root@fedora ~]# mke2fs -c /dev/sdb1
mke2fs 1.40.2 (12-Jul-2007)
Dateisystem-Label=
OS-Typ: Linux
Blockgröße=1024 (log=0)
Fragmentgröße=1024 (log=0)
124928 Inodes, 497980 Blöcke
24899 Blöcke (5.00%) reserviert für den Superuser
... der restliche Teil der Ausgabe wurde abgeschnitten ...
```

btrfs-tools

Ein relativ neues Dateisystem ist *btrfs*. Es wird spekuliert, ob dieses Dateisystem in Zukunft *ext2*, *ext3* und *ext4* vollständig ersetzen wird, weil es gegenüber den älteren Dateisystemen einige Vorteile aufweist. Dazu zählen:

- Dateisystemgrößen bis zu 16 EiB
- Schnappschüsse
- Dateisystemprüfung im laufenden Betrieb
- Datenkompression
- integriertes RAID
- mehrere Wurzelverzeichnisse (Subvolumen)

Bestehende *ext3*- und *ext4*-Dateisysteme können konvertiert werden. Zur Verwaltung der Dateisysteme benötigen Sie das Paket *btrfs-tools*. Anschließend steht Ihnen das Programm `fsck.btrfs` zur Dateisystemprüfung zur Verfügung. Hierbei handelt es sich nicht um einen Link auf `fsck`, sondern um ein eigenständiges Programm mit dem Namen `btrfs`.

Erzeugen der Dateisysteme

Wenn Sie Partitionen auf einer Festplatte erstellen wollen, stehen Ihnen normalerweise unabhängig von der verwendeten Distribution die Werkzeuge `fdisk` und `cfdisk` zur Verfügung. Das Programm `cfdisk` ist menügeführt und intuitiv bedienbar. Da in den LPI-Prüfungen ausschließlich der traditionelle `fdisk` thematisiert wird, soll auf die Beschreibung von `cfdisk` in diesem Buch verzichtet werden. Wenn `fdisk` gestartet wird, müssen Sie zumindest als Argument die zu bearbeitende Festplatte

angeben. Wenn Sie lediglich die aktuelle Partitionierung der Festplatte einsehen wollen, können Sie die Option `-l` (list) verwenden:

```
archangel:~ # fdisk /dev/hda -l
Disk /dev/hda: 80.0 GB, 80026361856 bytes
255 heads, 63 sectors/track, 9729 cylinders
Units = cylinders of 16065 * 512 = 8225280 bytes
 Device Boot      Start         End      Blocks   Id  System
/dev/hda1   *           1           7       56196   83  Linux
/dev/hda2               8        2401    19229805   83  Linux
/dev/hda3            2402        2495      755055   82  Linux swap
/dev/hda4            2496        9729    58107105   83  Linux
```

Wenn Sie beabsichtigen, die Partitionierung eines Datenträgers zu ändern, sollten Sie die aktuelle Belegung genau kennen. Am besten drucken Sie die Liste aus, oder Sie schreiben sie ab. Eine Übersicht über alle Partitionen sämtlicher Festplatten eines Computers können Sie im */proc*-System abfragen:

```
archangel:~ # cat /proc/partitions
major minor     #blocks   name
   3     0     78150744   hda
   3     1        56196   hda1
   3     2     19229805   hda2
   3     3       755055   hda3
   3     4     58107105   hda4
   8    16    156290904   sdb
   8    17    156288321   sdb1
```

Wenn Sie mit `fdisk` eine Festplatte partitionieren wollen, starten Sie `fdisk` und geben die Festplatte mit ihrer Gerätedatei an:

```
archangel:~ # fdisk /dev/hda
```

Durch einfaches Betätigen der Taste [n] (new) erstellen Sie, interaktiv geführt, eine neue Partition. Das folgende Beispiel demonstriert die Erstellung einer 500 MB großen primären Partition auf einem SCSI-Laufwerk. Die vom root getätigten Eingaben sind durch Fettdruck hervorgehoben.

```
[root@fedora ~]# fdisk /dev/sdb
```

Das Programm `fdisk` ist gestartet und wartet auf weitere Befehle:

```
Befehl (m für Hilfe): n
Befehl  Aktion
   e      Erweiterte
   p      Primäre Partition (1-4)
```

n leitet das Erstellen einer neuen Partition ein. Als Nächstes muss festgelegt werden, ob die Partition eine primäre oder eine erweiterte Partition werden soll. Betätigen Sie p für primär:

```
p
Partitionsnummer (1-4):  1
```

Die Partitionsnummer wurde also auf 1 festgelegt. Hätte es sich um eine Festplatte mit bestehenden Dateisystemen gehandelt, wäre spätestens jetzt eine genaue Kenntnis der bestehenden Partitionen erforderlich. Als Nächstes wird der Startzylinder angegeben. Bei einer fabrikneuen Festplatte bietet sich Zylinder 1 förmlich an. Also wird der Vorgabewert einfach durch Betätigen der Eingabetaste übernommen:

```
Erster Zylinder (1-1044, Vorgabe: 1):
```

Das Programm bestätigt diese Angabe und fragt anschließend nach dem Ende der zu erstellenden Partition. Mit dem Pluszeichen wird die Einheit für die Größe der Partition in Bytes, Kilobytes oder Megabytes festgelegt. Eine numerische Eingabe ohne Pluszeichen steht für die Angabe des Endzylinders.

Die Angabe des Endzylinders ist während der Installation besonders interessant, damit man das Dateisystem für */boot* bequem innerhalb der ersten 1.024 Zylinder positionieren kann. Man braucht sich dann nicht mehr zu überlegen, wie groß die Partition wohl höchstens (in Megabytes gerechnet) sein darf. In diesem Beispiel soll aber eine 500-MB-Partition für die Daten erstellt werden:

```
Benutze den Standardwert 1
Letzter Zylinder oder +Größe, +GrößeK oder +GrößeM
(1-1044, Vorgabe: 1044): +500MB
```

Das Ergebnis sollte anschließend unbedingt geprüft werden:

```
Befehl (m für Hilfe): p

Platte /dev/sdb: 8589 MByte, 8589934592 Byte
255 heads, 63 sectors/track, 1044 cylinders
Einheiten = Zylinder von 16065 × 512 = 8225280 Bytes

   Gerät  boot.     Anfang      Ende      Blöcke   Id  System
/dev/sdb1               1        62      497983+  83  Linux
```

Nachdem die Konfiguration geprüft und für gut befunden wurde, kann die Partitionstabelle auf die Festplatte geschrieben werden. Bis jetzt hat noch kein Schreibzugriff stattgefunden!

```
Befehl (m für Hilfe): w
Die Partitionstabelle wurde verändert!

Rufe ioctl() um Partitionstabelle neu einzulesen.
Synchronisiere Platten.
```

Der Vorgang ist nun abgeschlossen und die Partition kann formatiert werden. Vorher sollen aber noch ein paar Kommandos für `fdisk` erläutert werden. Um alle verfügbaren Optionen von `fdisk` anzuzeigen, betätigen Sie die Taste [m] (menu).

- `a` (activate) legt die aktive Partition fest. Dieses Flag wird vom BIOS ausgewertet und für Betriebssysteme benötigt, die über kein Startprogramm im MBR verfügen.
- `d` (delete) löscht eine Partition.
- `l` (list) zeigt alle von `fdisk` unterstützten Dateisysteme an.
- `m` (menu) zeigt das Menü an.
- `n` (new) erstellt eine neue Partition.
- `p` (print) zeigt die aktuelle Partitionstabelle an.
- `q` (quit) verwirft alle Änderungen und beendet `fdisk`.
- `t` (type) ändert den Dateisystemtyp einer Partition.
- `u` (units) ändert die Einstellung für die Einheit der Anzeige (z. B. für `p`).
- `v` (verify) überprüft die Partitionstabelle.
- `w` (write) schreibt alle Änderungen in die Partitionstabelle.
- `x` (extra) ergibt ein zusätzliches Menü mit hardwarenahen Optionen.

Wenn Sie [l] betätigen, werden Sie feststellen, dass Linux sehr viele Dateisystemarten unterstützt. Sie sollten allerdings dem Typ 83 (Linux) den Vorzug geben, damit Sie bei der Formatierung ein für Linux optimales Dateisystem (*ext2*, *ext3*, *ext4* oder *btrfs*) erstellen können. Für ein Swap-Dateisystem ist der Typ 82 erforderlich.

Sie sollten die anderen verfügbaren Dateisystemtypen nur dann verwenden, wenn Sie mit anderen Betriebssystemen über die zu erstellende Partition Dateien austauschen müssen. Die gängigsten Kennungen sind:

- `7` HPFS/Windows NTFS
- `c` Windows 95 FAT 32 (LBA)
- `f` Windows 95, erweiterte Partition (LBA)
- `82` Linux Swap
- `83` Linux (für alle Linux-Dateisysteme, z. B. *ext2*, *ext3*, *ext4*, *btrfs* usw.)
- `85` Linux Extended
- `8e` Linux LVM

Formatieren der Dateisysteme

Nachdem Sie eine Partition mit `fdisk` erstellt haben, muss diese noch formatiert werden. Hierbei vollzieht sich die eigentliche Erstellung des Dateisystems. Das Programm `fdisk` modifiziert lediglich die Partitionstabelle. Das wichtigste Formatierungswerkzeug unter Linux ist `mkfs`. Eigentlich handelt es sich hierbei um ein Frontend, das, je nach zu formatierendem Dateisystem, ein passendes Backendprogramm ausführt. Sie finden `mkfs` mit den zugehörigen Backends im Verzeichnis */sbin*. Einige dieser Backends sind lediglich Hardlinks oder Softlinks, welche die Bedienung, unabhängig vom verwendeten Dateisystem, vereinheitlichen sollen. Diese sind:

- `mkfs.ext2` erstellt sowohl *ext2-*, *ext3-* als auch *ext4*-Dateisysteme. Zur Formatierung mit *ext3* wird die Option `-j` (Journal) verwendet.
- `mkfs.ext3` ist normalerweise ein Hardlink auf `mkfs.ext2`. Durch den Aufruf erkennt das Programm, welches Dateisystem formatiert werden soll.
- `mkfs.ext4` ist ebenfalls im Normalfall lediglich ein Hardlink auf das Werkzeug `mkfs.ext2`.
- `mkfs.msdos` erstellt MS-DOS-Dateisysteme unter Linux. Sie können von Partitionen, die mit `mkfs.msdos` erstellt worden sind, kein Betriebssystem starten. Normalerweise handelt es sich bei `mkfs.msdos` um einen Softlink zu `mkdosfs`.
- `mkfs.vfat` ist ein weiterer Softlink auf `mkdosfs`.
- `mkfs.ntfs` erstellt NTFS-Dateisysteme unterschiedlicher Versionen. Momentan werden alle Windows-Systeme unterstützt.
- `mkfs.xfs` erstellt das ursprünglich für IRIX entwickelte 64-Bit-Dateisystem XFS. XFS pflegt ein Journal ähnlich wie ReiserFS und unterstützt die Verwendung von Zugriffssteuerungslisten. Um Fragmentierungen zu minimieren, hält XFS relativ große Datenmengen im Schreibcache. Dadurch wird die Wahrscheinlichkeit erhöht, für Datenblöcke relativ genau passende Bereiche zu finden. Leider werden hierdurch größere Datenverluste bei Stromausfall wahrscheinlicher.
- `mkfs.cramfs` erstellt das Dateisystem CramFS. Hierbei handelt es sich um ein komprimiertes Read-only-Dateisystem, das eigentlich ausschließlich für Embedded Systems eingesetzt wird, auf die nicht schreibend zugegriffen werden muss.
- `mkfs.jfs` erstellt das Dateisystem JFS, das ursprünglich für das Betriebssystem AIX entwickelt worden ist.

Prüfungstipp

Für die Prüfung müssen Sie sich vor allem mit der Bedienung der Frontendprogramme auseinandersetzen.

mkfs und mke2fs

Die beiden wichtigsten Frontends zur Formatierung sind `mkfs` und `mke2fs`. Bei `mke2fs` handelt es sich allerdings erneut lediglich um einen Hardlink, nämlich mit `mkfs.ext2`, `mkfs.ext3` bzw. `mkfs.ext4`.

Wenn Sie `mkfs` verwenden, benötigen Sie vor allem die folgenden Optionen:

- `-t` (type) legt den zu formatierenden Dateisystemtyp fest.
- `-c` (check) prüft das Gerät auf fehlerhafte Sektoren.
- `-v` (verbose) erzeugt eine informativere Ausgabe.
- `-j` (journal) erstellt ein Journal für *ext3*.
- `-L` Bezeichnung (Label) erstellt ein Volume-Label.

ext2/ext3

Beachten Sie die Besonderheit bei der Erstellung von *ext3*-Dateisystemen. Wie bereits erwähnt, stellen die Formatierungsbackends für *ext2* und *ext3* ein und dasselbe Programm dar. Das ist auch sinnvoll, weil es sich nämlich bei *ext3* um ein *ext2*-Dateisystem handelt, das lediglich um ein Journal erweitert worden ist. Sie können sogar ein bestehendes *ext2*-Dateisystem um ein Journal erweitern. Sie erhalten dann ein vollwertiges *ext3*-Dateisystem. Wenn Sie z. B. die *ext2*-Partition */dev/hdb3* nach *ext3* konvertieren wollen, geben Sie folgendes Kommando ein:

```
[root@fedora ~]# tune2fs -j /dev/hdb3
tune2fs 1.40.2 (12-Jul-2007)
Erstelle Journal-Inode: erledigt
Das Dateisystem wird automatisch alle 26 Mounts bzw. alle 180 Tage überprüft,
je nachdem, was zuerst eintritt. Veränderbar mit tune2fs -c oder -t.
```

Bei der Formatierung können Sie deshalb entweder *ext3* als Dateisystem angeben, oder Sie wählen *ext2* mit der Option `-j`. Sie können also wahlweise eines der folgenden beiden Kommandos verwenden:

```
[root@fedora ~]# mkfs -t ext3 /dev/sdb1
```

oder:

```
[root@fedora ~]# mkfs -t ext2 -j /dev/sdb1
```

Beide Kommandos erzeugen zudem die gleiche Ausgabe. Beachten Sie bitte, dass in Wirklichkeit `mke2fs` aufgerufen wird:

```
mke2fs 1.40.2 (12-Jul-2007)
Dateisystem-Label=
OS-Typ: Linux
Blockgröße=1024 (log=0)
```

```
Fragmentgröße=1024 (log=0)
124928 Inodes, 497980 Blöcke
24899 Blöcke (5.00%) reserviert für den Superuser
erster Datenblock=1
Maximum filesystem blocks=67633152
61 Blockgruppen
8192 Blöcke pro Gruppe, 8192 Fragmente pro Gruppe
2048 Inodes pro Gruppe
Superblock-Sicherungskopien gespeichert in den Blöcken:
        8193, 24577, 40961, 57345, 73729, 204801, 221185, 401409
Schreibe Inode-Tabellen: erledigt
Erstelle Journal (8192 Blöcke): erledigt
Schreibe Superblöcke und Dateisystem-Accountinginformationen: erledigt
Das Dateisystem wird automatisch alle 36 Mounts bzw. alle 180 Tage überprüft,
je nachdem, was zuerst eintritt. Veränderbar mit tune2fs -c oder -t.
```

XFS

Wenn Sie einen Datenträger mit dem XFS-Dateisystem formatieren wollen, müssen Sie diesen ebenfalls zuvor mit dem Partitionstyp Linux (83) partitionieren. Sie können das Journal, das XFS verwendet, bei Bedarf auf einem andern Datenträger erstellen und auch die Größe festlegen. Die eigentliche Formatierung ähnelt ansonsten der Formatierung von anderen Dateisystemen. Sie sollten (auch zu Wartungszwecken) die passenden Werkzeuge an Bord haben. Diese werden im weiteren Verlauf des Kapitels ohnehin noch benötigt:

```
root@arch-deb-book:/# apt-get install xfsprogs
root@arch-deb-book:/# apt-get install xfsdump
```

Nun aber zur eigentlichen Formatierung:

```
root@arch-deb-book:/# mkfs.xfs /dev/sdb1 -l logdev=/dev/sda1
meta-data=/dev/sdb1              isize=256    agcount=4, agsize=245534 blks
         =                       sectsz=512   attr=2, projid32bit=0
data     =                       bsize=4096   blocks=982134, imaxpct=25
         =                       sunit=0      swidth=0 blks
naming   =version 2              bsize=4096   ascii-ci=0
log      =Internes Protokoll     bsize=4096   blocks=2560, version=2
         =                       sectsz=512   sunit=0 blks, lazy-count=1
realtime =keine                  extsz=4096   blocks=0, rtextents=0
```

Hinweis

Wenn Sie zu Testzwecken einen USB-Stick verwenden wollen, bedenken Sie, dass der dauerhafte Einsatz eines Dateisystems mit Journal den Verschleiß des Sticks erhöht.

ZFS

ZFS impliziert zwar aufgrund seines Namens (*Zettabyte File System*), dass es sich um ein reines Dateisystem handeln würde, es beinhaltet aber gleichzeitig die Funktionalität eines logischen *Volumemanagers*, eines Software-RAID und ein *Copy-on-Write-Dateisystem*. ZFS arbeitet schneller als ein vergleichbares Hardware-RAID. Da es sich um ein 128-Bit Dateisystem handelt, sind die Speicherkapazitäten geradezu unvorstellbar groß. Chefentwickler *Jeff Bonwick* hat es einmal so ausgedrückt:

> *»Ein 128-Bit-Dateisystem zu füllen würde die quantenmechanische Grenze irdischer Datenspeicherung übersteigen. Man könnte einen 128-Bit-Speicher-Pool nicht füllen, ohne die Ozeane zu verdampfen.«*

Genaue Kenntnisse zu ZFS sind in der Prüfung nicht erforderlich.

Btrfs

Das Dateisystem *Btrfs* (sprich: B-tree FS) ist gegenüber ext4 mit weniger Einschränkungen behaftet, wenn es um die Verwaltung großer Festplatten bzw. Partitionen geht. Wenn Sie die *btrfs-tools* (bei Red Hat btrfs-progs) installiert haben, steht Ihnen zur Formatierung von *Btrfs*-Dateisystemen das Programm `mkfs.btrfs` zur Verfügung. Hierbei handelt es sich nicht um einen Link auf `mkfs`, sondern um ein eigenständiges Programm. Da das Handling keine Besonderheiten beinhaltet, soll hier auf eine Vorführung verzichtet werden.

Interessanter ist die Konvertierung eines bestehenden *ext2*, *ext3* oder *ext4*-Dateisystems in *Btrfs*. Das geht mit folgendem Befehl:

```
[root@scientific7 ~]# btrfs-convert /dev/sdb1
creating btrfs metadata.
copy inodes [o] [         0/        11]
creating ext2fs image file.
cleaning up system chunk.
conversion complete.
```

Auf der ext-Partition vorhandene Daten bleiben bei der Konvertierung erhalten. Die Konvertierung ist durch den Schalter `-r` einfach reversibel:

```
[root@scientific7 ~]# btrfs-convert /dev/sdb1 -r
rollback complete.
```

Machen Sie das Rollback bitte wieder rückgängig, um weitere Kommandos nachvollziehen zu können. Zum Lieferumfang der *btrfs*-tools gehört, unter etlichen anderen, das Programm `btrfs` selbst. Sie können damit Subvolumen, aber auch Snapshots eines bestehenden *Btrfs*-Volumes erzeugen. Hängen Sie das eben erzeugte Volumen ein:

```
[root@scientific7 ~]# mount /dev/sdb1 /mnt
```

Sie können nun feststellen, dass während der Konvertierung eines bestehenden Dateisystems gleichzeitig ein *Subvolumen* mit einem *Snapshot* erstellt worden ist. Verwenden Sie dazu folgenden Befehl:

```
[root@scientific7 ~]# btrfs subvolume list /mnt
ID 256 gen 6 top level 5 path ext2_saved
```

Sie können diesen Snapshot einfach in ein Verzeichnis einhängen. Es handelt sich im Prinzip um eine Image-Datei, die das Dateisystem aufweist, die vor der Konvertierung vorhanden war. Beispiel:

```
[root@scientific7 /]# mkdir /snapshot
[root@scientific7 /]# mount -t ext4 -o loop /mnt/ext2_saved/image /snapshot/
```

Sie können überprüfen, dass der Snapshot den Inhalt der ursprünglichen ext4-Partition aufweist.

mkswap

Wenn Sie eine Swap-Partition formatieren wollen, benötigen Sie das Programm `mkswap`. Die Partition (Typ 82) muss zuvor mit `fdisk` erzeugt werden.

```
[root@fedora ~]# mkswap /dev/sdb4
Swapbereich Version 1 wird angelegt, Größe 1011703 KBytes
```

Damit Linux die neu erstellte Swap-Partition verwendet, muss sie mit dem Kommando `swapon` aktiviert werden.

```
[root@fedora ~]# swapon -v /dev/sdb4
swapon für /dev/sdb4
```

Ohne die Option `-v` erhalten Sie keine Bestätigung darüber, dass die neue Swap-Partition verwendet wird. Umgekehrt können Sie die Verwendung einer Swap-Partition abschalten, indem Sie das Kommando `swapoff` verwenden. Was die Redseligkeit von `swapoff` betrifft, gilt das Gleiche wie für `swapon`. Die Option `-v` kann also auch hier nicht schaden:

```
[root@fedora ~]# swapoff -v /dev/sdb4
swapoff für /dev/sdb4
```

Damit die neue Swap-Partition gleich nach dem Systemstart verfügbar wird, muss die Datei */etc/fstab* bearbeitet werden. Tragen Sie die Swap-Partition hier folgendermaßen ein (der genaue Aufbau der Datei */etc/fstab* wurde bereits an anderer Stelle erläutert):

```
/dev/sdb4       swap        swap       pri=42         0 0
```

Erstellen einer Swap-Datei

Im Gegensatz zu Windows verwendet Linux im Normalfall keine Swap-Datei, sondern ganze Swap-Partitionen, wie bereits beschrieben. Das hat vor allem den Vorteil, dass es nicht zu einer Fragmentierung des Auslagerungsbereichs kommen kann. Sollte es aber kurzfristig im Speicher einmal eng werden, kann man ohne großen Aufwand schnell eine Swap-Datei (mit all ihren Nachteilen) erzeugen. Verwenden Sie dazu folgendes Kommando:

```
archangel:~ # dd if=/dev/zero of=/swapfile bs=1024 count=524288
524288+0 Datensätze ein
524288+0 Datensätze aus
```

Der `dd`-Befehl bewirkt hier Folgendes: Aus dem Zero Device (*/dev/zero*) werden Nullen in die noch nicht existierende Datei *swapfile* kopiert. Die Blockgröße ist auf `1024` Bytes festgelegt. Das Ganze geschieht `524288`-mal. Durch diese Prozedur entsteht eine Datei, die aus 512 MB Nullen besteht. Diese Datei kann anschließend als Swap-Datei formatiert werden. Das geht deshalb so einfach, weil Linux nicht zwischen einer normalen Datei (hier: */swapfile*) und einer Gerätedatei (z. B. */dev/sdb4* für eine Swap-Partition) unterscheidet.

```
archangel:~ # mkswap /swapfile
Swapbereich Version 1 wird angelegt, Größe 524288 KBytes
```

Zur Aktivierung der Swap-Datei geben Sie, wie gehabt, dieses Kommando ein:

```
archangel:~ # swapon /swapfile
```

Damit eine Swap-Datei gleich nach dem Systemstart verfügbar ist, muss die Datei */etc/fstab* bearbeitet werden. Tragen Sie die Swap-Datei hier ein, wie jedes andere Dateisystem:

```
/swapfile       swap          swap    defaults       0 0
```

Nach dem Hinzufügen der neuen Swap-Datei und ihrer Aktivierung vergewissern Sie sich, dass sie wirklich aktiv ist, indem Sie die Ausgabe der Befehle `cat /proc/swaps` oder `free` prüfen:

```
archangel:~ # cat /proc/swaps
Filename          Type          Size    Used    Priority
/dev/hda3         partition     755044  0       42
/dev/hdb2         partition     755044  0       41
/swapfile         file          524288  0       40
```

Prüfungstipp

Für die Prüfung sollten Sie sich besonders mit der Verwendung der Frontends und den Besonderheiten bei der Erstellung von *ext3* im Verhältnis zu *ext2* vertraut machen.

xfs_info, xfs_check und xfs_repair

Eigentlich ist XFS ein sehr robustes Dateisystem, das normalerweise nicht repariert werden muss. In der Praxis ist es wahrscheinlicher, dass Sie regelmäßig das Programm `xfs_fsr` ausführen, um das Dateisystem zu reorganisieren. Um die Geometrie des Dateisystems zu überprüfen, können Sie `xfs_info` verwenden. Sie sollten das Dateisystem während der Abfrage einhängen und dem Programm `xfs_info` den Mountpoint des Dateisystems übergeben:

```
root@arch-deb-book:/# xfs_info /xfsdisk
Metadaten =/dev/sdb1              isize=256    agcount=4, agsize=245534 blks
          =                       sectsz=512   attr=2
Daten     =                       bsize=4096   Blöcke=982134, imaxpct=25
          =                       sunit=0      swidth=0 blks
Benennung =Version 2              bsize=4096   ascii-ci=0
Protokoll =Intern                 bsize=4096   Blöcke=2560, Version=2
          =                       sectsz=512   sunit=0 blks, lazy-count=1
Echtzeit  =keine                  extsz=4096   Blöcke=0, rtextents=0
```

Wenn Sie einen Fehler in einem XFS-Dateisystem vermuten, können Sie `xfs_check` heranziehen. In diesem Fall muss das Dateisystem allerdings zunächst ausgehängt werden. Sollte sich das Journal auf einer anderen Partition befinden, müssen Sie diese über die Option `-l` angeben. Beispiel:

```
root@arch-deb-book:/# xfs_check /dev/sdb1 -l /dev/sda1
```

Sollte ein XFS-Dateisystem beschädigt sein (was eher unwahrscheinlich ist), können Sie es mit `xfs_repair` reparieren. Dieser Vorgang kommt der Verwendung von `fsck` bei anderen Dateisystemen gleich. Die Reparatur wird in sieben Phasen durchgeführt. Hier ein Beispiel mit gekürzter Ausgabe:

```
root@arch-deb-book:/# xfs_repair /dev/sdb1
Phase 1 - Superblock finden und überprüfen...
Phase 2 - ein internes Protokoll benutzen
        - Null-Protokoll...
        - freier Speicher und Inode-Karten des Dateisystems werden gescannt...
```

Hier wurden die weniger interessanten Zeilen entfernt.

```
Inodes werden zurückgesetzt
        - Dateisystem wird durchquert ...
        - durchqueren beendet ...
        - nicht verbundene Inodes werden nach lost+found verschoben ...
Phase 7 - Verweisanzahl wird geprüft und berichtigt...
erledigt
```

xfsdump/xfsrestore

Zur Datensicherung eines XFS-Dateisystems sollten Sie `xfsdump` verwenden. Der Vorteil gegenüber anderen Sicherungsprogrammen ist, dass `xfsdump` die Dateiattribute ebenfalls sichert, sodass diese mithilfe des Programms `xfsrestore` wiederhergestellt werden können. Im Übrigen unterstützt `xfsdump` inkrementelle Sicherungen, was beim täglichen Backup Zeit einspart. Als Ziel für eine Sicherung kommen Dateien, Speichermedien oder die Standardausgabe infrage.

```
root@arch-deb-book:/# xfsdump -v trace -f /media/backup/sicherung1 /xfsdisk
xfsdump: using file dump (drive_simple) strategy
xfsdump: version 3.0.4 (dump format 3.0) - Running single-threaded
===============dump label dialog ========================
please enter label for this dump session (timeout in 300 s.)
 -> Sicherung A
============== media label dialog ======================
please enter label for media in drive 0 (timeout in 300 sec)
 -> Medium 1
media label entered: "Medium 1"
-------------- end dialog --------------------------------
xfsdump: creating dump session media file 0 (media 0, file 0)
xfsdump: dumping ino map
xfsdump: dumping directories
xfsdump: dumping directory ino 128
xfsdump: dumping non-directory files
xfsdump: ending media file
xfsdump: media file size 21016 bytes
xfsdump: ending stream: 99 seconds elapsed
xfsdump: dump size (non-dir files) : 0 bytes
xfsdump: dump complete: 99 seconds elapsed
xfsdump: Dump Status: SUCCESS
```

Eine Wiederherstellung ist mit dem Tool `xfsretore` durchführbar. Wenn Sie dem Programm die Quelle und das Ziel für die Wiederherstellung angeben, beginnt es ohne weitere Rückfragen mit der Arbeit:

```
root@arch-deb-book:/# xfsrestore -f /media/backup/sicherung1 /xfsdisk/
xfsrestore: using file dump (drive_simple) strategy
```

```
xfsrestore: version 3.0.4 (dump format 3.0) - Running single-threaded
xfsrestore: searching media for dump
xfsrestore: examining media file 0
xfsrestore: dump description:
xfsrestore: hostname: arch-deb-book
xfsrestore: mount point: /xfsdisk
xfsrestore: volume: /dev/sdb1
xfsrestore: session time: Sat Feb 25 18:25:01 2012
xfsrestore: level: 0
xfsrestore: session label: "Sicherung A"
xfsrestore: media label: "Medium1"
xfsrestore: Restore Status: SUCCESS
```

smartd und smartctl

S.M.A.R.T. (*Self-Monitoring, Analysis and Reporting Technology*) ist eine Technologie, mit der moderne Festplatten wichtige Parameter von sich selbst überwachen und aufzeichnen. Es sollen anhand dieser gesammelten Daten Ausfälle prognostiziert werden. Bisher können etwa 64 % aller Festplattenausfälle mit S.M.A.R.T. vorhergesagt werden. Damit das funktioniert, benötigen Sie zunächst das entsprechende Werkzeug, mit dem Sie die Festplattenparameter einlesen und dann auswerten können:

```
root@archangel:~# apt-get install smartmontools
```

In diesem Paket sind zwei Komponenten enthalten. Die eine Komponente ist *smartctl* und dient hauptsächlich der manuellen Prüfung von Festplattenparametern. Die andere Komponente ist der Daemon *smartd*. Er läuft, wenn Sie ihn aktivieren, permanent im Hintergrund und informiert Sie (z. B. per E-Mail), wenn ein Problem mit einer Festplatte besteht oder zu erwarten ist.

smartctl

Da das Kommando `smartctl` recht umfangreiche Ausgaben liefert, soll sich hier wieder auf das Nötigste beschränkt werden. Um grundlegende Informationen, wie z. B. Hersteller, Modell, Seriennummer und die Firmware-Version zu erhalten, verwenden Sie `smartctl` wie folgt:

```
root@archangel:~# smartctl /dev/sda -i
```

Weitere wichtige Optionen für `smartctl` sind:

- `-a` gibt alle S.M.A.R.T.-Parameter eines angegebenen Laufwerks aus.
- `-x` gibt alle S.M.A.R.T.- und Nicht-S.M.A.R.T.-Parameter eines Laufwerks aus.
- `--scan` sucht nach Laufwerken und gibt für jedes jeweils das Gerät, den Gerätetyp und das verwendete Protokoll aus.

Es gibt noch einige weitere Optionen, die Sie vielleicht auch noch ausprobieren möchten, aber dazu muss ich Sie wieder auf die entsprechende Manpage verweisen. Einen schnellen Check für den Gesundheitszustand (Health) will ich Ihnen aber nicht vorenthalten:

```
root@archangel:~# smartctl /dev/sda -H
smartctl 5.43 2012-06-30 r3573 [i686-linux-3.9.4] (local build)
Copyright (C) 2002-12 by Bruce Allen, http://smartmontools.sourceforge.net
=== START OF READ SMART DATA SECTION ===
SMART overall-health self-assessment test result: PASSED
```

Wenn der Status eines Laufwerks, für das Sie verantwortlich sind, hier nicht den Status `PASSED` aufweist, haben Sie höchstwahrscheinlich jetzt oder innerhalb der nächsten Stunden ein schwerwiegendes Problem.

smartd

Damit Sie *smartd* einsetzen können, ist zunächst ein wenig Konfigurationsarbeit zu leisten. Der Daemon wird sonst nicht starten. Bearbeiten Sie zunächst die Datei */etc/default/smartmontools*. Hier müssen Sie zumindest die folgenden Optionen festlegen:

```
enable_smart="/dev/hda"
```

Diese Option sagt *smartd*, welche Laufwerke er überwachen soll.

```
start_smartd=yes
```

Sie müssen diese Option auf `yes` setzen und das Kommentarzeichen am Anfang der Zeile entfernen, damit *smartd* beim Systemstart geladen wird. Optional können noch einige Optionen gesetzt werden, die in der Manpage genauer beschrieben sind. Die folgende Option sorgt für eine Überprüfung durch *smartd* alle 3.600 Sekunden:

```
smartd_opts="--interval=3600"
```

Im Prinzip reicht diese Konfiguration schon aus, um *smartd* zu starten. Sie sollten aber für den Praxiseinsatz noch einige Einstellungen in der eigentlichen Konfigurationsdatei */etc/smartd.conf* prüfen und ggf. konfigurieren. Es gibt in dieser Datei viele auskommentierte Beispielzeilen, die Sie vielleicht nur noch an Ihre Bedürfnisse anpassen müssen.

Starten Sie *smartd* anschließend mit dem Kommando:

```
root@archangel:~ # /etc/init.d/smartmontools start
```

Sie können die Aktivitäten des Daemons am besten auf dem Syslog beobachten, indem Sie eine entsprechende Filterung durch `grep` vornehmen:

```
root@archangel:~ # grep smartd /var/log/syslog
```

203.3 Anlegen und Konfigurieren von Dateisystemen

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, *automount*-Dateisysteme mittels *autofs* zu konfigurieren. Dieses Lernziel beinhaltet die Konfiguration von *automount* für Netzwerk- und Gerätedateisysteme. Des Weiteren ist das Anlegen von Dateisystemen für Geräte wie z. B. CD-ROMs und Grundwissen über die Eigenschaften von verschlüsselten Dateisystemen in diesem Lernziel enthalten.

Wichtigste Wissensgebiete:

- `autofs`-Konfigurationsdateien
- Verständnis der *automount units*
- UDF- und ISO9660-Werkzeuge und -Dienstprogramme
- Kenntnis der CD-ROM-Dateisysteme (UDF, ISO9660, HFS)
- Kenntnis der CD-ROM Dateisystemerweiterungen (Joliet, Rock Ridge, El Torito)
- Grundwissen über verschlüsselte Dateisysteme (*dm-crypt* / *LUKS*)

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/auto.master*
- */etc/auto.[dir]*
- `mkisofs`
- `cryptsetup`

Allgemeines

Ein Teil der Einrichtungen, die Sie in diesem Kapitel kennenlernen werden, sind in den Standardinstallationen der meisten Linux-Distributionen gar nicht mehr zu finden. Das liegt daran, dass diese Tools durch andere Programme ersetzt worden sind, die von den meisten Benutzern bevorzugt werden. Es gibt aber manchmal Aufgaben, die nur von diesen älteren Programmen automatisch erledigt werden können. Gerade solche Tools sind oft Gegenstand von Prüfungsfragen, und Sie sollten sich mit diesen Themen beschäftigen.

Automatisches Mounten

Die Dateisysteme, die während des Startvorgangs bereits präsent sind, können Sie über die Konfigurationsdatei */etc/fstab* so einbinden, dass sie automatisch beim Systemstart gemountet werden. Anders ist das bei Ressourcen, auf die Sie über ein Netzwerk zugreifen. Am deutlichsten wird das, wenn Sie ein Notebook mit WLAN nutzen.

Hier wird die Netzwerkverbindung normalerweise erst hergestellt, nachdem Sie sich am System angemeldet haben. Wenn eventuelle Netzwerkverbindungen, z. B. über NIS oder SMB in der Datei */etc/fstab* eingetragen sind, können Sie diese zwar dann recht einfach mit dem Kommando `mount -a` einbinden, aber wirklich komfortabel ist das nicht. Ein *Automounter* hängt ein Dateisystem immer erst dann ein, wenn Sie darauf zugreifen. Das ist in einem solchen Falle natürlich die wesentlich elegantere Lösung.

Sie sollten das Paket *autofs* installieren, um die nachfolgenden Schritte nachvollziehen zu können:

```
root@arch-deb:~# apt-get install autofs
```

Oder wenn Sie ein Red Hat-Derivat verwenden:

```
[root@arch-fc ~]# yum install autofs
```

Die Hauptkonfigurationsdatei von `autofs` ist */etc/auto.master*. Hier werden die Haupteinhängepunkte definiert und gleichzeitig festgelegt, welche Konfigurationsdateien die Dateisysteme für diese Haupteinhängepunkte beschreiben. Am besten versteht man das, wenn man die Funktionsweise der Beispielkonfiguration von `automount` analysiert, auch wenn diese heutzutage nicht mehr benötigt wird. In der Datei *auto.master* gibt es den folgenden Eintrag:

```
/misc        /etc/auto.misc
```

Damit dieser funktioniert, muss das Verzeichnis */misc* existieren. Der Eintrag verweist auf die Datei *auto.misc*, die die folgenden Einträge enthalten könnte:

```
cd          -fstype=iso9660,ro,nosuid,nodev    :/dev/cdrom
floppy      -fstype=auto                       :/dev/fd0
```

Wenn eine CD eingelegt wird, hängt `automount` diese automatisch beim Zugriff unter dem Verzeichnis */misc/cd* ein. Entsprechend würde eine Diskette nach */misc/floppy* gemountet, sobald jemand auf dieses Verzeichnis zugreift. Im folgenden Beispiel wird demonstriert, wie eine NFS-Freigabe automatisch eingehängt wird, wenn der Benutzer einen Doppelklick auf eine entsprechende Verknüpfung ausführt, die sich auf seinem Desktop befindet. Wenn der Benutzer keine Dateien mehr geöffnet hält, wird die Ressource auch automatisch wieder ausgehängt.

Zuerst sollte der Mountpoint erstellt werden. Auf höchster Ebene wird ein Verzeichnis */netz* für Netzwerkressourcen angelegt. (Das muss übrigens nicht sein; Sie können das Verzeichnis anlegen, wo Sie wollen.) Die Unterverzeichnisse für die konkreten Ressourcen werden in diesem Verzeichnis von *automount* selbst erstellt und sollten deshalb nicht von Hand angelegt werden.

```
root@arch-deb-book:/# mkdir /netz
```

In der Datei */etc/auto.master* wird ein Eintrag erstellt, der auf eine Map-Datei zeigt. Diese wird im nächsten Arbeitsschritt hier angelegt.

Der Eintrag in der *auto.master*-Datei sieht folgendermaßen aus:

```
/netz           /etc/auto.netz    --timeout=60
```

Dieser Eintrag bedeutet, dass der Einhängepunkt für `autofs` das Verzeichnis */netz* sein soll. Die Dateisysteme, die an diesem Ort eingehängt werden sollen, sind in der Datei */etc/auto.netz* definiert. Wenn ein Dateisystem dieser Map-Datei für 60 Sekunden nicht verwendet wurde, wird es automatisch ausgehängt. Die Datei */etc/auto.netz* soll für das Beispiel lediglich den folgenden Inhalt haben:

```
storage        -fs=nfs,defaults        archangel:/storage
```

Diese Map-Datei veranlasst den Automounter, automatisch den Zugriffspunkt *storage* unterhalb von */netz* anzulegen. An dieser Stelle wird die NFS-Freigabe *storage* eingehängt, die der Computer *archangel* zur Verfügung stellt (exportiert).

Damit das Ganze für den Benutzer noch komfortabler wird, soll auf dem Desktop ein Link erstellt werden, sodass der Mount-Vorgang einfach mit der Maus initialisiert werden kann:

```
root@arch-deb-book:/# ln -s /netz/storage /home/dominik/Desktop/storage
```

Änderungen in den Map-Dateien werden übrigens sofort dynamisch übernommen. Sie müssen nichts manuell neu starten oder einlesen lassen. Nur dann, wenn Sie neue Mountpoints in der Datei *auto.master* erstellen, müssen Sie `autofs` neu starten.

ISO-Dateien und CDs erstellen

Eigentlich werden CDs und DVDs auch unter Linux heutzutage mit Programmen erstellt, die über eine grafische Benutzeroberfläche verfügen. Es kann aber durchaus vorkommen, dass man (z. B. für eine kleine Backuplösung) eine CD skriptgesteuert beschreiben muss. Dann benötigen Sie gute Kommandozeilenwerkzeuge.

mkisofs

Da eine CD nicht mit denselben Dateisystemen verwendet werden kann wie eine Festplatte, muss zunächst ein ISO9660-Dateisystem erzeugt werden, damit die Daten auf eine CD geschrieben werden können. Das gilt für eine DVD natürlich sinngemäß genauso. Bei DVDs kommt allerdings schon aufgrund der Größe das Dateisystem UDF (Universal Disk Format) zum Einsatz. Sie können recht einfach mithilfe des Programms `mkisofs` ein ISO9660- (oder UDF-)Dateisystem erstellen. Dieses Dateisys-

tem liegt anschließend in Form einer ISO-Datei vor und kann dann auf eine CD oder DVD gebrannt werden. Ein Beispiel:

```
root@archangel:~# mkisofs -r -J -l -ldots -o etc.iso /etc
Warning: creating filesystem that does not conform to ISO-9660.
 37.22% done, estimate finish Sat Apr 30 15:05:53 2011
 74.54% done, estimate finish Sat Apr 30 15:05:53 2011
Total translation table size: 0
Total rockridge attributes bytes: 458964
Total directory bytes: 1306624
Path table size(bytes): 7560
Max brk space used 309000
13435 extents written (26 MB)
```

Dieses Kommando erstellt im aktuellen Verzeichnis (hier das Heimatverzeichnis des root) die Datei *etc.iso*. Sie enthält eine Sicherung des Verzeichnisses */etc*. Es sind hier wesentliche Optionen übergeben worden, die für Unix-Derivate wichtig sind. Diese und weitere Optionen haben folgende Funktionen:

- `-r` ist ein Kürzel für Rock Ridge. Diese Erweiterungen sorgen dafür, dass die Dateisystemberechtigungen (Unix-Attribute) für die in der ISO-Datei gespeicherten Dateien und Verzeichnisse erhalten bleiben.
- `-l` erlaubt die Verwendung von langen Dateinamen mit bis zu 31 Zeichen. ISO9660 unterstützt ohne diese Option lediglich das 8.3-Format von MS-DOS.
- `-ldots` erlaubt die Verwendung von Dateinamen, die mit einem Punkt beginnen. Wenn Sie diese Option nicht verwenden, werden Punkte zum Beginn eines Dateinamens durch einen Unterstrich ersetzt.
- `-o` ist die Option für die Ausgabedatei.
- `-J` (Joliet) wird benötigt, wenn Sie die CD später auf einem Windows-Computer verwenden wollen. Es werden auch Dateinamen mit bis zu 64 Zeichen im UTF8-Format unterstützt.
- `-udf` erstellt ein UDF-Dateisystem. Das ist z. B. zur Unterstützung von DVDs erforderlich.

Da es sich sowohl bei Joliet, als auch bei Rock Ridge lediglich um Erweiterungen für ISO9660 handelt, können Sie die Optionen `-r` (bzw. `-R`) und `-J` ohne Probleme gemeinsam verwenden, wenn Sie die CD auf beiden Systemen lesen können wollen. Joliet kann allerdings nur von Windows und Linux gelesen werden. Andere Betriebssysteme können mit Joliet-Erweiterungen nichts anfangen.

Sie können die ISO-Datei jetzt einlagern oder auf eine CD brennen. Wenn Sie den Inhalt der ISO-Datei einsehen möchten, um die Sicherung zu überprüfen, können Sie die Datei einfach mounten. Verwenden Sie zu diesem Zweck beim Einhängen die Option `loop`:

```
root@archangel:~# mkdir /test
root@archangel:~# mount -t iso9660 -o loop etc.iso /test
root@archangel:~# cd /test
root@archangel:~# ls -l /test
insgesamt 1316
-r--r--r--  1 root root 15070 2008-04-08 18:24 a2ps.cfg
-r--r--r--  1 root root  2563 2008-04-08 18:24 a2ps-site.cfg
dr-xr-xr-x  3 root root  6144 2010-10-31 14:35 acpi
-r--r--r--  1 root root  2975 2008-04-22 19:49 adduser.conf
-r--r--r--  1 root root  2563 2010-04-20 15:41 aiccu.conf
```

Sieht aus, als wäre die Sicherung vollständig.

```
root@archangel:~# rm -Rf /test
```

cdrecord

Ein bewährtes Programm, mit dem Sie ISO-Dateien auf CDs schreiben können, ist `cdrecord`. Da der Vorgang des Schreibens mit einem einzigen Kommando initiiert werden kann, ist es natürlich auch sehr gut zur Verwendung in Skripten geeignet. Bei einigen Linux-Distributionen (z. B Debian und Ubuntu) können Sie über das Paketmanagement lediglich `wodim` installieren. Hierbei handelt es sich um eine Alternative zu `cdrecord`, die sich aufrufkompatibel verhält. Ein Link mit dem Namen `cdrecord` zeigt einfach auf `wodim`. Wegen lizenzrechtlicher Streitigkeiten ist zwar das ursprüngliche `cdrecord` von Jörg Schilling in den Repositories von Debian und Ubuntu nicht mehr enthalten, aber Sie können bei Bedarf das Paket `wodim` installieren. Spätestens wenn Sie Blu-ray- oder Double-Layer-DVDs erstellen wollen, stoßen Sie mit `wodim` an Grenzen.

Verschlüsselte Dateisysteme

Wenn Sie vertrauliche Daten auf Computern speichern, die vor einem Diebstahl nicht sicher geschützt werden können, sollten Sie darüber nachdenken, diese Daten zu verschlüsseln. Besonders Notebooks und Netbooks sind diebstahlgefährdet, wenn sie z. B. auf dem Rücksitz eines Autos liegen gelassen werden. Auch wenn das Gerät selbst durch ein Passwort geschützt ist und Dateien und Verzeichnisse durch entsprechende Berechtigungen abgesichert worden sind, können Daten ausgelesen werden. Die Festplatte muss dazu lediglich in einen anderen Computer als zusätzliches Laufwerk eingebaut werden.

Anstatt einzelne Dateien, z. B. mittels GnuPG, zu verschlüsseln, können Sie auch komplette Dateisysteme verschlüsseln. Das erleichtert die Handhabung vertraulicher Daten natürlich enorm. Sie sollten allerdings auch bedenken, dass die Ver- und Entschlüsselung von Daten Rechenzeit benötigt und der Computer in Abhängigkeit von der verwendeten Schlüsselgröße langsamer werden kann.

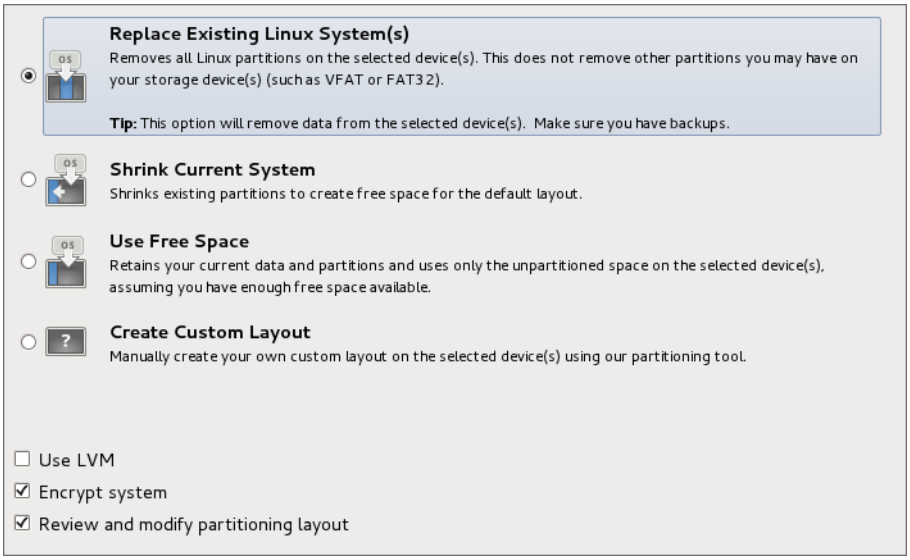

Abbildung 203.1 Der Installationsassistent von Fedora (Anaconda) bietet Verschlüsselung (Encrypt System) schon bei der Partitionierung an.

Wenn Sie beabsichtigen, Dateisysteme zu verschlüsseln, empfiehlt es sich, die entsprechende Konfiguration schon während der Installation des Betriebssystems durchzuführen, weil eine nachträgliche Implementierung einen erheblichen Aufwand mit sich bringt. Die meisten heutigen Distributionen bieten diese Option.

Wenn auf einem Computer besonders vertrauliche Dateien gespeichert und bearbeitet werden, sollten Sie unbedingt auch die Swap-Partition verschlüsseln. Das können Sie normalerweise ebenfalls mit dem Assistenten der Erstinstallation durchführen. Anaconda (das Setup-Programm von Fedora) schlägt die Verschlüsselung der Swap-Partition übrigens von sich aus vor. Die gleichzeitige Verwendung von LVM ist zumindest bei Fedora nicht erforderlich.

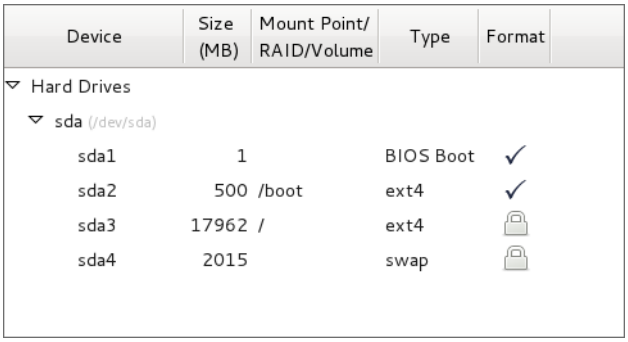

Abbildung 203.2 Verschlüsseln Sie auch die Swap-Partition, wenn Daten als besonders vertraulich zu betrachten sind.

Als Nächstes müssen Sie eine Passphrase für die Verschlüsselung angeben. Es sind hierbei mindestens acht Zeichen erforderlich. Es ist zwar sicherer, eine deutlich längere Passphrase zu verwenden (zwanzig Zeichen sollten normalerweise ausreichen), bedenken Sie aber auch, dass Sie diese Passphrase bei jedem Systemstart eingeben müssen.

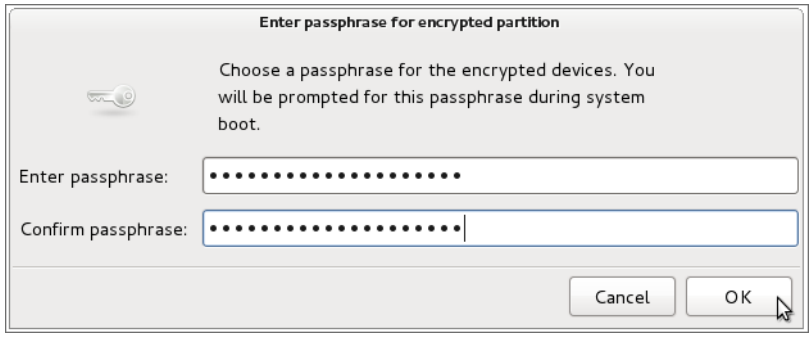

Abbildung 203.3 Das ist der Dialog zur Angabe der Passphrase für die Verschlüsselung. Mindestens acht Zeichen sind erforderlich.

Weitere Konfigurationsschritte sind nicht erforderlich. Wenn Sie die Eingabe der Passphrase mit einem Klick auf OK bestätigen, beginnt die Formatierung und anschließend die Verschlüsselung der ausgewählten Datenträger. Sie sollten darauf vorbereitet sein, dass die Verschlüsselung, insbesondere bei großen Dateisystemen, erhebliche Zeit in Anspruch nehmen kann.

Abbildung 203.4 Die Verschlüsselung kann bei großen Datenträgern sehr lange dauern. Hier ist Geduld gefragt.

Bei der Verschlüsselung von Dateisystemen kommt heutzutage LUKS (Linux Unified Key Setup) zum Einsatz. Um einen Datenträger mit LUKS zu verschlüsseln, können Sie das Hilfsprogramm `cryptsetup` verwenden. Bei der Einrichtung gehen alle Daten auf dem Datenträger verloren, was die Schwierigkeiten bei der nachträglichen Implementierung erklärt. Wenn Sie ein externes USB-Gerät verschlüsseln wollen, können Sie das aber ohne Probleme machen. Da LUKS selbst ein fester Bestandteil des Kernels ist, müssen Sie lediglich das Programm `cryptsetup` nachinstallieren. Unter der Annahme, dass ein USB-Gerät mit der Gerätedatei */dev/sdb1* verschlüsselt werden soll, führen Sie die folgenden Schritte durch:

```
root@arch-deb-book:/# apt-get install cryptsetup
```

Anschließend muss das Kernel-Modul für den Device Mapper geladen werden:

```
root@arch-deb-book:/# modprobe dm_mod
```

Jetzt können Sie die Verschlüsselung der Partition vornehmen. Achten Sie darauf, dass das Gerät nicht eingehängt ist, weil die Verschlüsselung sonst kommentarlos (also ohne Fehlermeldung) fehlschlägt.

```
root@arch-deb-book:/# cryptsetup luksFormat /dev/sdb1
```

Nachdem Sie das Kommando ausgeführt haben, werden Sie nach einer Passphrase gefragt. Diese müssen Sie wieder eingeben, wenn Sie das Gerät später verwenden wollen.

Als Nächstes muss eine Gerätedatei für den Device Mapper erzeugt werden. Er abstrahiert das verschlüsselte Gerät für Anwendungen, sodass diese von der Verschlüsselung praktisch nichts bemerken. Die Gerätedatei heißt hier */dev/mapper/safe*:

```
root@arch-deb-book:/# cryptsetup luksOpen /dev/sdb1 safe
```

Nachdem die Gerätedatei für den Device Mapper angelegt ist, können Sie diese nun formatieren. Weil für das Beispiel ein USB-Speicherstick verwendet worden war, war ein Dateisystem ohne Journal sinnvoll:

```
root@arch-deb-book:/# mkfs.ext2 /dev/mapper/safe
```

Jetzt benötigen Sie nur noch einen Mountpoint, an dem Sie das Gerät einhängen können:

```
root@arch-deb-book:/# mkdir /safe
root@arch-deb-book:/# mount /dev/mapper/safe /safe
```

Sie können das Gerät nun verwenden. Wenn Sie den Speicher wieder aushängen, sind die Daten für Dritte nicht mehr zugänglich.

Es gibt übrigens Alternativen zu dem oben beschriebenen Verfahren, wie z. B. `dm-crypt`, `cryptoloop`, `loop-aes` und `truecrypt`. LUKS in Verbindung mit `cryptsetup` ist jedoch heutzutage am gebräuchlichsten.

204 Erweiterte Administration von Storage Devices

Performance und Datensicherheit sind Faktoren, die beim Speichersubsystem kritisch zu betrachten sind. Hier kommen RAID-Systeme in den Blickwinkel – Redundant Arrays of Independent Disks.

204.1 RAID-Konfiguration

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Software-RAID zu konfigurieren und zu implementieren. Dieses Lernziel beinhaltet die Konfiguration von RAID 0, 1 und 5.

Wichtigste Wissensgebiete:

- Software-RAID-Konfigurationsdateien und -Dienstprogramme

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- *mdadm.conf*
- `mdadm`
- */proc/mdstat*
- Partitionstyp 0xFD

Allgemeines

Wenn man mehrere physikalisch vorhandene Festplatten zu einer logischen Festplatte zusammenschließt, spricht man von einem RAID-Verbund. Die ursprüngliche Bedeutung dieser Abkürzung war *Redundant Array of Inexpensive Disks*. Man konnte durch den Zusammenschluss kostengünstiger Festplatten große logische Datenträger erzeugen. Später wurde die Aussprache des Akronyms in *Redundant Array of Independent Disks* geändert, weil man nicht wollte, dass durch die Verwendung der Bezeichnung »Inexpensive« eine Billiglösung suggeriert wird.

Die in diesem Kapitel beschriebenen Lösungen sind Software-RAIDs. Sie unterscheiden sich von Hardware-RAIDs vor allem in den folgenden Punkten:

- Software-RAIDs benötigen keine speziellen RAID-Controller und sind deshalb kostengünstig.
- Da sich bei Software-RAIDs die CPU um die Verwaltung der Datenträger kümmern muss, ist ein Software-RAID vergleichsweise langsam.
- Im Gegensatz zum Hardware-RAID können beim Software-RAID die Festplatten unterschiedlich groß sein.
- Ein Hardware-RAID verwendet ganze Festplatten, während ein Software-RAID lediglich Partitionen unterschiedlicher Festplatten zu einem RAID-System zusammenfügt.

RAID-Level

Es gibt unterschiedliche RAID-Level, die numerisch bezeichnet werden. Für die Prüfung sind allerdings nur die gängigen RAID-Level 0, 1 und 5 von Belang. Diese drei RAID-Level unterscheiden sich wie folgt:

- Bei *RAID-0-Systemen* werden mehrere Festplatten so zusammengeschaltet, dass diese gleichzeitig gelesen und beschrieben werden können. Die Speicherkapazität von RAID 0 entspricht der Summe der Kapazitäten der ursprünglichen Festplatten. Das bedeutet eine Zusammenlegung der Kapazitäten und eine Steigerung der Geschwindigkeit des Systems, bietet aber keine Ausfallsicherheit.
- Bei *RAID-1-Systemen* werden zwei Festplatten gespiegelt (Mirroring). Das heißt, dass ein RAID-1-Verbund auch nur genau zwei Festplatten enthalten kann und diese dieselben Daten beinhalten. Dieser RAID-Verbund bietet ausschließlich Ausfallsicherheit durch redundante Daten.
- Ein *RAID-5-System* bietet Ausfallsicherheit und Performancezuwachs zugleich. Ähnlich wie bei RAID 0 werden die Daten hier auf mehrere Festplatten verteilt. Es werden aber Paritätsdaten durch logische XOR-Verknüpfungen erzeugt und auf die Datenträger verteilt geschrieben. Hierdurch ergibt sich, dass der Ausfall eines Datenträgers ohne Datenverlust toleriert werden kann. Da bei Schreibzugriffen zunächst die redundanten Daten berechnet werden müssen, ist der Geschwindigkeitsvorteil hauptsächlich bei Lesezugriffen bemerkbar. Wenn Sie z. B. vier Festplatten mit einer Kapazität von je 2 TB zu einem RAID-5-Verbund zusammenschalten, erhalten Sie nur eine Gesamtkapazität von 6 TB. Die fehlenden 2 TB werden für Paritätsinformationen verwendet.

Andere RAID-Level sind für die Prüfung nicht relevant, sollen aber dennoch der Vollständigkeit halber zumindest kurz erwähnt und mit den drei oben stehenden Systemen verglichen werden:

- *RAID 4* arbeitet ähnlich wie RAID 5. Die Paritätsinformationen werden hier aber nicht über alle Datenträger verteilt, sondern auf einen einzigen Datenträger geschrieben.

- *RAID 6* ähnelt ebenfalls RAID 5, kann aber auch den gleichzeitigen Ausfall von zwei Datenträgern tolerieren.
- *RAID 10* ist ein Zusammenschluss von mehreren RAID-1-Systemen zu einem RAID-0-Verbund. So erhält man die Vorteile von RAID 1 und RAID 0 in einem einzigen System.

Ein Software-RAID erstellen

Wenn Sie die Erstellung eines Software-RAID unter Linux nachvollziehen wollen, benötigen Sie mehrere Festplatten oder eine virtuelle Maschine, der Sie noch Festplatten hinzufügen können. Wenn Sie mit virtuellen Maschinen arbeiten, ist es sinnvoll, einen virtuellen SCSI-Controller einzusetzen. Sie sind dann bei der Bestückung mit Festplatten weniger eingeschränkt. Für die Erstellung von RAID 5 benötigen Sie mindestens drei Festplatten. Zur Erstellung der Beispielkonfiguration habe ich vier Platten genommen und diesmal Debian verwendet, aber es dürfte Ihnen vermutlich inzwischen keine Schwierigkeiten mehr bereiten, `yum` anstelle von `apt-get` zu verwenden, falls Sie ein Red Hat-basiertes System einsetzen.

Installieren Sie zunächst das Paket *mdadm*:

```
root@arch-deb:/# apt-get install mdadm
```

Prüfen Sie nach der Installation, ob das für Multiple Disks benötigte Kernel-Modul geladen worden ist:

```
root@arch-deb:/# lsmod|grep md_mod
md_mod                 67377  0
```

Sollte das Modul bei Ihnen (wider Erwarten) nicht geladen worden sein, laden Sie es bitte manuell mit `insmod` oder `modprobe`.

Im nächsten Arbeitsschritt müssen alle zu verwendenden Festplatten mit `fdisk` vorbereitet werden. Es müssen Partitionen von Typ FD (Linux-RAID) angelegt werden. Unter der Annahme, dass die erste Ihrer für RAID vorgesehenen Festplatten */dev/sdb* ist, gehen Sie folgendermaßen vor:

```
root@arch-deb-book:/# fdisk /dev/sdb
```

Sie werden beim Programmstart nach einem Kommando gefragt. Geben Sie ein `n` für »new partition« ein:

```
Command (m for help): n
```

Sie werden gefragt, ob Sie eine erweiterte oder eine primäre Partition erstellen wollen. Geben Sie hier `p` ein:

```
Command action
   e   extended
   p   primary partition (1-4)
p
```

Da auf der Festplatte noch keine Partitionen existieren, beantworten Sie die Frage nach der gewünschten Partitionsnummer mit `1`:

```
Partition number (1-4): 1
```

Wenn Sie bei den Abfragen nach dem ersten und letzten zu verwendenden Zylinder jeweils die Eingabetaste betätigen, werden die Standardwerte übernommen, und die neue Partition erstreckt sich über die komplette Festplatte. Geben Sie anschließend `t` ein, um den Partitionstyp festzulegen:

```
Command (m for help): t
Selected partition 1
Hex code (type L to list codes):
```

Der HEX-Code für Linux-RAID ist `fd`:

```
Hex code (type L to list codes): fd
Changed system type of partition 1 to fd (Linux raid autodetect)
```

Geben Sie abschließend ein `w` ein, damit die neue Partitionstabelle geschrieben wird:

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Wenden Sie dasselbe Verfahren für alle Festplatten an, die Sie für RAID verwenden wollen. Sollten Sie eine veraltete Version von `fdisk` verwenden, kann es sein, dass diese die Partitionierung von SCSI-Geräten nicht unterstützt. Sie müssten dann alternativ auf das Programm `sfdisk` ausweichen.

Nachdem die Vorbereitungen abgeschlossen sind, kann das RAID nun endlich erstellt werden. Die Vorgehensweise ist bei unterschiedlichen RAID-Leveln fast identisch. Im folgenden Beispiel habe ich mich für RAID 5 mit vier Festplatten entschieden. Die vier Festplatten sind dynamisch unter VirtualBox virtualisiert und weisen eine Kapazität von jeweils 8 GB auf. Es sollte also im RAID-5-Verbund eine Gesamtkapazität von lediglich 24 GB feststellbar sein. Die Erstellung erfolgt so:

```
root@arch-deb:/# mdadm --create --verbose /dev/md0 --auto=yes --level=5
--raid-devices=4 /dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1
```

```
mdadm: layout defaults to left-symmetric
mdadm: chunk size defaults to 512K
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: layout defaults to left-symmetric
mdadm: size set to 8384512K
mdadm: Defaulting to version 1.2 metadata
mdadm: array /dev/md0 started.
```

Das RAID ist jetzt schon unter der Gerätedatei */dev/md0* erreichbar und kann formatiert werden. Die einzelnen Elemente des obigen Kommandos haben folgende Aufgaben:

- `mdadm --create` sorgt für die Erstellung eines neuen RAID-Arrays.
- `--verbose` zeigt umfangreiche Informationen während der Erstellung.
- `/dev/md0` ist die zu verwendende Gerätedatei für das neue RAID-Array.
- `--auto=yes` sorgt für die automatische Erstellung der Gerätedatei, die übrigens nicht partitionierbar sein wird.
- `--level=5` besagt, dass ein RAID-5-Array erstellt werden soll. Mögliche Level sind übrigens 0, 1, 4, 5, 6 und 10.
- `--raid-devices=4`: Das Array wird vier Geräte enthalten.
- `/dev/sdb1 /dev/sdc1 /dev/sdd1 /dev/sde1` ist eine durch Leerzeichen getrennte Aufzählung der Geräte, die zusammengeschaltet werden sollen.

RAID-Arrays verwenden, konfigurieren und überprüfen

Damit das RAID-Array verwendbar wird, muss es jetzt formatiert und in den Verzeichnisbaum integriert werden:

```
root@arch-deb:/# mkfs.ext4 /dev/md0
```

Die Formatierung sieht genauso aus wie bei einem konventionellen Datenträger und soll deshalb hier nicht abgedruckt werden.

Erstellen Sie ein Verzeichnis, das Sie als Einhängepunkt verwenden möchten, und mounten Sie das RAID:

```
root@arch-deb:/# mkdir /raid5
root@arch-deb:/# mount -t ext4 /dev/md0 /raid5/
```

Damit das RAID nach einem Systemstart automatisch zur Verfügung steht, können Sie in der Datei */etc/fstab* einen entsprechenden Eintrag vornehmen. Der Eintrag sieht genauso aus wie bei normalen Festplatten:

```
/dev/md0     /raid5      ext4      defaults   0  0
```

Jetzt kann überprüft werden, ob der Datenträger tatsächlich die erwartete Kapazität aufweist. Zur Erinnerung: Es handelt sich hier um ein RAID-5-Array mit vier Festplatten zu je 8 GB. Es ist also nach Abzug der Speichermenge, die für Paritätsinformationen benötigt wird, von einer Gesamtkapazität von 24 GB auszugehen. Das Kommando `df` wird Klarheit bringen:

```
root@arch-deb:/# df -h /dev/md0
Dateisystem           Size  Used Avail Use% Eingehängt auf
/dev/md0               24G  172M   23G   1% /raid5
```

Die Kapazität stimmt also. Der bereits belegte Speicherbereich wird durch Verwaltungsinformationen von *ext4* benötigt.

Sie können Informationen über RAID-Arrays beim */proc*-Device abfragen. Hier sehen Sie auch, welche Partitionen der physikalischen Festplatten in welchen RAID-Systemen verwendet werden:

```
root@arch-deb:/# cat /proc/mdstat
Personalities : [raid6] [raid5] [raid4]
md0 : active raid5 sde1[4] sdd1[2] sdc1[1] sdb1[0]
      25153536 blocks super 1.2 level 5, 512k chunk, algorithm 2 [4/4] [UUUU]

unused devices: <none>
```

Wesentlich genauere Informationen erhalten Sie mithilfe des Programms `mdadm`. Sie können mit diesem Programm jede erdenkliche Detailinformation über RAID-Arrays abfragen. Außerdem wird `mdadm` verwendet, um RAID-Arrays zu erweitern oder auch zu reparieren. Hier sehen Sie das Ergebnis einer detaillierten Abfrage:

```
root@arch-deb:/# mdadm --detail /dev/md0
  Creation Time : Sat Apr 30 22:10:35 2011
     Raid Level : raid5
     Array Size : 25153536 (23.99 GiB 25.76 GB)
  Used Dev Size : 8384512 (8.00 GiB 8.59 GB)
   Raid Devices : 4
  Total Devices : 4
    Persistence : Superblock is persistent
    Update Time : Sat Apr 30 22:52:05 2011
          State : clean
 Active Devices : 4
Working Devices : 4
 Failed Devices : 0
  Spare Devices : 0
```

```
        Layout : left-symmetric
    Chunk Size : 512K
          Name : arch-deb:0  (local to host arch-deb)
          UUID : 5871b763:1cd3bf9e:ef0bd4c5:44b0546f
        Events : 34
 Number  Major  Minor  RaidDevice State
    0      8      17       0      active sync   /dev/sdb1
    1      8      33       1      active sync   /dev/sdc1
    2      8      49       2      active sync   /dev/sdd1
    4      8      65       3      active sync   /dev/sde1
```

Die Konfigurationsdatei mdadm.conf

In der Konfigurationsdatei */etc/mdadm/mdadm.conf* können Sie für das RAID einige Einstellungen vornehmen. So können Sie z. B. eine E-Mail-Benachrichtigung konfigurieren, für den Fall, dass ein Fehler auftritt:

```
MAILADDR raid-admin@meinedomaene.tld
```

Mit der folgenden Zeile konfigurieren Sie die Absenderadresse des Servers, damit Sie im Störfall wissen, welcher Computer überhaupt das Problem hat:

```
MAILFROM fileserver-07@meinedomaene.tld
```

Bei den meisten aktuellen Linux-Distributionen sind keine weiteren Konfigurationsschritte nötig. Das RAID-Array muss auch nicht mehr (so wie früher) in die Konfigurationsdatei eingetragen werden, damit es automatisch startet. Sollten Sie nach einem Systemstart keinen Zugriff auf Ihr RAID haben und Sie können einen Fehler in der Datei */etc/fstab* ausschließen, kann noch Folgendes versucht werden:

```
root@arch-deb:/# mdadm --detail --scan >> /etc/mdadm/mdadm.conf
```

Es werden dann die RAID-Array-Informationen in die Datei *mdadm.conf* eingetragen, sodass sich das *mdadm*-System selbst um das Aktivieren des RAID beim Systemstart kümmert. Beachten Sie unbedingt den doppelten Redirektor (>>), damit Sie nicht versehentlich die *mdadm.conf*-Datei in einen unbrauchbaren Zustand versetzen.

Ein RAID-Array erweitern

Wenn die Benutzer es geschafft haben, Ihr RAID-System bis an die Kapazitätsgrenze zu füllen, können Sie das Array vergrößern, indem Sie weitere Laufwerke hinzufügen. Eine neu hinzugefügte Festplatte muss natürlich auch zunächst mit einer Partition versehen werden. Hier können Sie einen Kunstgriff anwenden und die Partitionstabelle einer der bestehenden Festplatten auf die neue Festplatte kopieren. Im folgen-

den Beispiel heißt die Gerätedatei der neuen Festplatte */dev/sdf*. Mit `sfdisk -d` wird die Partitionstabelle der Festplatte */dev/sde* ausgelesen und die Ausgabe des Kommandos anschließend mithilfe einer Pipe an eine zweite Instanz von `sfdisk` weitergereicht, um damit die neue Festplatte zu partitionieren:

```
root@arch-deb:/# sfdisk /dev/sde -d |sfdisk /dev/sdf
```

Verwenden Sie anschließend das Programm `mdadm`, um die neue Festplatte (eigentlich die einzige Partition dieser Festplatte) in das bestehende RAID-Array aufzunehmen:

```
root@arch-deb:/# mdadm --add /dev/md0 /dev/sdf1
mdadm: added /dev/sdf1
```

Anschließend sollten Sie überprüfen, ob die neue Partition im RAID-Array erscheint:

```
root@arch-deb:/# mdadm --detail /dev/md0
 ...
 Number  Major  Minor  RaidDevice State
    0      8      17       0      active sync   /dev/sdb1
    1      8      33       1      active sync   /dev/sdc1
    2      8      49       2      active sync   /dev/sdd1
    4      8      65       3      active sync   /dev/sde1

    5      8      81       -      spare   /dev/sdf1
```

Die Ausgabe des Programms ist diesmal stark gekürzt, sodass nur die relevanten Informationen abgebildet werden. Man kann sehen, dass die neue Partition */dev/sdf1* in das Array integriert wurde. Sie wird aber zunächst lediglich als *Spare Disk* (Reserveplatte) betrachtet. Um die Festplatte aktiv im RAID verwenden zu können, muss das Array bezüglich der Anzahl der Festplatten vergrößert werden. Das erreichen Sie mit folgendem Kommando:

```
root@arch-deb:/# mdadm --grow --raid-devices=5 /dev/md0
mdadm: Need to backup 6144K of critical section..
```

Die Option `--grow` bewirkt das Wachstum von vier auf fünf Geräte innerhalb des RAID-Arrays (`--raid-devices=5`). Beachten Sie bitte, dass dieser Vorgang (das sogenannte *Reshape*) sehr viel Zeit in Anspruch nehmen kann. Der Computer darf während des Reshape-Vorgangs natürlich nicht heruntergefahren oder ausgeschaltet werden. Wenn Sie sehen wollen, wie weit der Vorgang fortgeschritten ist, können Sie im */proc*-Device nachsehen:

```
root@arch-deb:/# cat /proc/mdstat
Personalities : [linear] [multipath] [raid0] [raid1] [raid6] [raid5] [raid4]
[raid10]
```

```
md0 : active raid5 sdf1[5] sdb1[0] sde1[4] sdd1[2] sdc1[1]
      25153536 blocks super 1.2 level 5, 512k chunk, algorithm 2 [5/5] [UUUUU]
      [==================>..] reshape = 90.1% (7562860/8384512) finish=
0.6min       speed=20393K/sec
unused devices: <none>
```

In diesem Fall ist der Vorgang so gut wie abgeschlossen (nur noch 0,6 Minuten). Als zusätzlichen Indikator können Sie übrigens das Tool `top` verwenden. Da der Vorgang des Reshape die CPU stark belastet, können Sie auch hier Beobachtungen vornehmen.

Nach Abschluss des Reshape-Vorgangs ist es noch erforderlich, das Dateisystem (also die Partition des RAID-Systems) zu vergrößern. Für diese Aufgabe ist ein passendes Bordmittel verfügbar. Auch wenn der Name es nicht vermuten lässt: Sie können mithilfe von `resize2fs` auch ein *ext3*- oder, wie in diesem Fall, ein *ext4*-Dateisystem vergrößern. Verwenden Sie einfach den folgenden Befehl:

```
root@arch-deb:/# resize2fs /dev/md0
resize2fs 1.41.12 (17-May-2010)
Das Dateisystem auf /dev/md0 ist auf /raid5 eingehängt; Online-
Grössenveränderung nötig
old desc_blocks = 2, new_desc_blocks = 2
Führe eine Online-Grössenänderung von /dev/md0 auf 8384512 (4k) Blöcke durch.
Das Dateisystem auf /dev/md0 ist nun 8384512 Blöcke groß.
```

Wie Sie sehen, können Sie die Größenänderung sogar ausführen, ohne das RAID zu unmounten. Die Änderung wird online durchgeführt. Das wird allerdings nur von relativ neuen 2.6er-Kernel-Versionen (und neuer) unterstützt; in diesem Fall Version 2.6.32. Eine Prüfung mit `df` soll zum Schluss noch zeigen, ob alles geklappt hat:

```
root@arch-deb:/# df -h /dev/md0
Dateisystem           Size  Used Avail Use% Eingehängt auf
/dev/md0               32G  176M   30G   1%         /raid5
```

32 GB entsprechen der erwarteten Größe eines RAID-5-Arrays mit fünf Festplatten.

204.2 Konfiguration von Storage Devices

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Kernel-Optionen für verschiedene Speichermedien zu konfigurieren. Dieses Lernziel beinhaltet die Verwendung von Werkzeugen zur Darstellung und Verwaltung von Festplatteneinstellungen, inklusive iSCSI-Geräte.

Wichtigste Wissensgebiete:

- Werkzeuge und Dienstprogramme zur DMA-Konfiguration von IDE-Geräten, einschließlich ATAPI und SATA
- Werkzeuge und Dienstprogramme zur Konfiguration von Solid-State-Drives einschließlich AHCI und NVMe
- Werkzeuge und Dienstprogramme zur Manipulation oder Analyse von Systemressourcen (z. B. Interrupts)
- Kenntnis von `sdparm` und seiner Verwendung
- Werkzeuge und Hilfsmittel für iSCSI
- Kenntnis von SAN einschließlich der hierfür relevanten Protokolle (AoE, FCoE)

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `hdparm`, `sdparm`
- `nvme`
- `tune2fs`
- `fstrim`
- `sysctl`
- */dev/hd*, /dev/sd*, /dev/nvme**
- `iscsiadm`, *scsi_id*, *iscsid* und *iscsi.conf*
- *WWID*, *WWN*, *LUN* numbers

Allgemeines

Was die Konfiguration von Speichergeräten anbelangt, finden Sie in der anstehenden Prüfung einige Themen aus dem LPIC-1-Bereich wieder. Das gilt insbesondere bei der Zuordnung von Gerätedateien für Festplatten und der darin enthaltenen Partitionen. Man kann sagen, dass dieses Thema allerdings auch für die Praxis gut verinnerlicht sein muss. Bestimmt haben Sie auch schon mal von jemandem gehört, der in einer Stresssituation die falsche Partition formatiert hat.

hdparm

Mit dem Werkzeug `hdparm` kann man sowohl Festplattenparameter auslesen als auch einstellen. Diese Parameter beziehen sich nicht, wie auf den vorangehenden Seiten, auf Partitionen. Es geht hier eher um physikalische Merkmale einer Festplatte. Wenn Sie `hdparm` aufrufen und als einzigen Parameter die Gerätedatei einer Festplatte angeben, erhalten Sie die folgende Ausgabe:

```
root@archangel:/# hdparm -v /dev/sda
/dev/sda:
 multcount    = 16 (on)
 IO_support   =  1 (32-bit)
 readonly     =  0 (off)
 readahead    = 256 (on)
 geometry     = 60801/255/63, sectors = 976773168, start = 0
```

Sie sehen also, dass hier sehr hardwarenahe Informationen ausgegeben werden, wie z. B. die Laufwerksgeometrie und die Einstellungen für den Read-ahead-Cache. Im Gegensatz zu vielen anderen Programmen bewirkt die Option `-v` hier keine Verbose-Ausgabe. In vielen Situationen werden Sie nicht einmal einen Unterschied feststellen, wenn Sie die Option verwenden oder weglassen.

Falls Sie Informationen benötigen, wie z. B. Festplattenmodell, Seriennummer, Puffergröße, unterstützte DMA-, UDMA- und PIO-Modi, Schreibcache und Advanced Power Management, können Sie die Option `-i` verwenden. Die gewünschten Informationen werden beim Kernel abgefragt.

```
root@archangel:/# hdparm -i /dev/sda
/dev/sda:
Model=STM3500418AS, FwRev=CC37, SerialNo=9VM3Z1GZ
Config={ HardSect NotMFM HdSw>15uSec Fixed DTR>10Mbs RotSpdTol>.5% }
RawCHS=16383/16/63, TrkSize=0, SectSize=0, ECCbytes=4
BuffType=unknown, BuffSize=16384kB, MaxMultSect=16, MultSect=16
CurCHS=16383/16/63, CurSects=16514064, LBA=yes, LBAsects=976773168
IORDY=on/off, tPIO={min:120,w/IORDY:120}, tDMA={min:120,rec:120}
PIO modes:  pio0 pio1 pio2 pio3 pio4
DMA modes:  mdma0 mdma1 mdma2
UDMA modes: udma0 udma1 udma2 udma3 udma4 udma5 *udma6
AdvancedPM=no WriteCache=enabled
Drive conforms to: unknown:  ATA/ATAPI-4,5,6,7
signifies the current active mode
```

Diese Angaben sind aber lediglich Informationen, die dem Kernel mitgeteilt worden sind. Sie können durch die Verwendung eines großen `-I` direkt die Festplatte abfragen. Die Ausgabe des Befehls wird erheblich umfangreicher, weshalb das folgende Beispiel auch wieder gekürzt worden ist:

```
root@archangel:/# hdparm -I /dev/sda
/dev/sda:
ATA device, with non-removable media
        Model Number:       STM3500418AS
       Serial Number:      9VM3Z1GZ
        Firmware Revision:  CC37
```

```
        Transport:          Serial
Standards:
        Used: unknown (minor revision code 0x0029)
        Supported: 8 7 6 5
        Likely used: 8
Configuration:
        Logical          max        current
        cylinders        16383      16383
        heads            16         16
        sectors/track    63         63
        --
        CHS current addressable sectors:   16514064
        LBA    user addressable sectors:  268435455
        LBA48  user addressable sectors:  976773168
        Logical/Physical Sector size:           512 bytes
        device size with M = 1024*1024:      476940 MBytes
        device size with M = 1000*1000:      500107 MBytes
(500 GB)
```

Auf diese Art können Sie sich also auch bei einem Server, der sich an einem entfernten Standort befindet, schnell Informationen über die verwendeten Speichergeräte beschaffen. Man wird schließlich oft genug an schlecht oder gar nicht dokumentierten Maschinen arbeiten müssen.

Einige wichtige Optionen für `hdparm` sind:

- `-a` fragt die Read-ahead-Konfiguration ab oder konfiguriert diese.
- `-Bnum` ändert die Einstellungen für APM. Hierbei kann `num` Werte zwischen 1 und 255 annehmen. Die größte Energieersparnis wird mit dem Wert 1 erreicht; der Wert 255 schaltet das Powermanagement ab.
- `-g` zeigt die Festplattengeometrie an (wie vom Kernel gesehen).
- `-i` zeigt die Identifikationsinformationen an.
- `-r [0|1]` fragt das Read-only-Flag ab oder setzt dieses. Im Gegensatz zur Option `ro` des `mount`-Befehls wird hier eine ganze Festplatte schreibgeschützt.
- `-z` zwingt den Kernel, die Partitionstabelle der angegebenen Festplatte neu einzulesen.
- `-d` aktiviert oder deaktiviert den DMA-Modus einer Festplatte. Bei Verwendung von DMA wird der Prozessor entlastet, was eine erhebliche Performancesteigerung des Systems bewirken kann. Beispiel:

```
root@archangel:~# hdparm -d1 /dev/sda
/dev/sda:
setting using_dma to 1 (on)
```

- `-X` ändert die Einstellungen einer Festplatte bezüglich MDMA und SDMA. Diese Option wird, wenn überhaupt, am ehesten mit dem Schalter `-d1` zusammen verwendet. Beachten Sie bitte, dass es sich um ein großes `X` handelt.

Normalerweise setzt das System die optimalen Parameter für die Festplatten automatisch. Sie sollten beim Testen des Programms vorsichtig sein. Es ist sonst möglich, dass Sie Ihre Festplatte in einen unbrauchbaren Zustand versetzen.

sdparm

Das Programm `sdparm` kann verwendet werden, um die SCSI-Mode-Pages von Geräten auszulesen, die solche Abfragen unterstützen. Hierbei muss es sich nicht zwingend um SCSI-Geräte handeln. Auch einige CD-/DVD-Laufwerke und USB-Geräte unterstützen diese Kommandos. Um einen Überblick zu erhalten, welche Seiten und Felder dieses Tool auswerten kann, können Sie mit folgendem Befehl eine Aufstellung erzeugen:

```
root@archangel:/# sdparm --enumerate
Mode pages:
  addp 0x0e,0x02  DT device primary port (ADC)
  adlu 0x0e,0x03  logical unit (ADC)
  adtd 0x0e,0x01  Targer device (ADC)
  adts 0x0e,0x04  Targer device serial number (ADC)
  apo  0x1a,0xf1  SAT ATA Power condition
  atag 0x0a,0xf0  Application tag (SBC)
  bc   0x1c,0x01  Background control (SBC)
  ca   0x08       Caching (SBC)
```

Die ausgegebene Liste ist eigentlich erheblich länger und wurde hier nur aus Platzgründen abgeschnitten. Um Informationen zu einem bestimmten Gerät zu erhalten, geben Sie dieses einfach in der Kommandozeile an:

```
root@archangel:/# sdparm /dev/sda
    /dev/sda: ATA       STM3500418AS      CC37
Read write error recovery mode page:
  AWRE        1
  ARRE        0
  PER         0
Caching (SBC) mode page:
  WCE         1
  RCD         0
Control mode page:
  SWP         0
```

Sie können mittels `sdparm` auch einfache SCSI-Kommandos senden und Konfigurationsänderungen vornehmen. Solche Eingriffe sind aber nur selten erforderlich.

tune2fs

Das Tool `tune2fs` haben Sie schon früher kennengelernt. Sie sollen hier nur noch ein paar weitere Möglichkeiten erfahren, die Ihnen dieses Programm bietet. Es gibt im aktuellen Sachzusammenhang die folgenden Optionen:

- `-e error-behaviour` legt fest, wie der Kernel reagieren soll, wenn ein Fehler in dem angegebenen Dateisystem entdeckt wird. Es gibt drei Möglichkeiten für `error-behavior`:
  - `continue` bewirkt, dass der Fehler ignoriert wird. Das System versucht einfach fortzufahren.
  - `remount-ro` remountet die Festplatte read-only. Das ist normalerweise eine sehr gute Variante.
  - `panic` löst eine Kernelpanic aus. Das System wir dann sofort angehaltet.
- `-m reserved-blocks-percentage` legt den Speicherplatz (in Prozent) fest, den das System als Schutz vor Überflutung des Dateisystems durch Benutzer reserviert. Ein komplett gefülltes Dateisystem kann im schlimmsten Fall den Systemstart verhindern. Der reservierte Speicherplatz kann nur durch den User `root` verwendet werden.
- `-r reserved-block-count` legt die Anzahl der zu reservierenden Blöcke für ein Dateisystem fest. Der Verwendungszweck ist mit `-m` identisch.
- `-u user` legt Benutzer fest, die reservierte Blöcke verwenden dürfen. Es kann wahlweise der Benutzername oder die UID angegeben werden.
- `-g group` legt Gruppen fest, deren Mitglieder reservierte Blöcke beschreiben können. Auch hier kann der Name der Gruppe oder der GID angegeben werden.
- `-s [0|1]` aktiviert (`1`) bzw. deaktiviert (`0`) die Verwendung von Reserve- (Spare-) Superblocks. Man kann zwar durch die Deaktivierung der Spare-Superblocks Speicherkapazitäten einsparen, aber der Sicherheitsverlust, der hierdurch in Kauf genommen wird, ist absolut unverhältnismäßig.

Wenn Sie überprüfen wollen, wie ein Datenträger in Bezug auf diese Parameter konfiguriert ist, verwenden Sie die Option `-l`. Da die Ausgabe dieses Kommandos sehr umfangreich ist, wurde das folgende Beispiel so geschnitten, dass hauptsächlich die für das aktuelle Thema relevanten Einträge angezeigt werden:

```
root@archangel:/# tune2fs -l /dev/sda1
Filesystem state:         clean
Errors behavior:          Continue
Filesystem OS type:       Linux
Inode count:              4685824
Block count:              18729774
Reserved block count:     936488
Reserved GDT blocks:      1019
```

```
Reserved blocks uid:      0 (user root)
Reserved blocks gid:      0 (group root)
Journal inode:            8
Journal backup:           inode blocks
```

SSD und NVMe

Für neuere Speichertechnologien, wie *SSD (Solid State Disc)* und *NVMe (Non Volatile Memory over PCI-Express)* sind zusätzliche Werkzeuge erforderlich. Handelsübliche SSDs, die auch bereits Einzug in PCs und Notebooks der Privatanwender gefunden haben, werden bisher an gewöhnlichen *SATA*-Anschlüssen betrieben und im günstigsten Fall von einem *AHCI*-Controller verwaltet. Beim Betrieb von SSDs sollte entsprechend im BIOS bzw. UEFI der Betrieb des SATA-Anschlusses auf AHCI konfiguriert werden. Diese Controller sind jedoch ursprünglich für magnetische Speichermedien konzipiert und für die Performance von SSD-Speicher wenig förderlich.

Abhilfe bietet hier *NVMe*. Bei dieser Technologie wird im Prinzip eine *Solid State Disc* direkt an den *PCI-Express*-Bus angeschlossen, was natürlich erhebliche Performancevorteile mit sich bringt.

fstrim

Das Kommando `fstrim` wird verwendet um nicht mehr genutzte Speicherbereiche auf einem SSD-Speicher wieder freizugeben. Es ist hierfür erforderlich, das zu trimmende Dateisystem einzuhängen.

Bei älteren SSDs sollte man sich zunächst erkundigen, ob der Einsatz von `fstrim` ratsam ist, weil dieses Tool die Lebensdauer der Geräte herabsetzen kann.

nvme

Das Kommandozeilentool `nvme` bietet etliche Funktionen, die zur Verwaltung von *NVMe*-Controllern und Speichergeräten verwendet werden können. Es können gezielt Register gelesen oder auch beschrieben werden. Der Funktionsumfang der enthaltenen Unterkommandos würde den Rahmen dieses Kapitels sprengen.

Die Gerätedateien für Festplatten und CD-ROMs

Sowohl physikalische als auch logische Laufwerke werden unter Linux als Gerätedateien unterhalb von */dev* dargestellt. Hierbei handelt es sich nicht um ein gewöhnliches Verzeichnis auf der Festplatte, sondern um eine Präsentation der Geräte durch den Kernel. Man sagt, die Geräte werden ins User-Land exportiert. Die Präsentation war bei älteren Kernel-Versionen statisch (bis *devfs* durch `udev` abgelöst wurde), weshalb auch Geräte dargestellt wurden, die der Computer gar nicht besaß. Für die Prüfung müssen Sie diese Gerätedateien genau kennen.

/dev/hd*

Dateien für Geräte, die am IDE-Bus angeschlossen sind, beginnen normalerweise mit */dev/hd*. Der nächste Buchstabe sagt etwas über die Position des Gerätes am Controller aus, aber nichts über das tatsächlich verwendete Gerät. Sie können Festplatten, CD-ROM- oder DVD-Laufwerke nicht anhand der Gerätedateien unterscheiden. Die meisten (E)IDE-Controller verfügen über zwei Kanäle: Primary IDE und Secondary IDE. An jeden der beiden Kanäle können jeweils zwei Geräte (ein Master und ein Slave) angeschlossen werden. Moderne IDE-RAID-Controller verwalten mehr Geräte. Diese sollen hier aber aufgrund der Irrelevanz für die Prüfung nicht weiter thematisiert werden. Die IDE-Gerätedateien sind:

- */dev/hda* – Primary Master
- */dev/hdb* – Primary Slave
- */dev/hdc* – Secondary Master
- */dev/hdd* – Secondary Slave

Wenn ein CD-ROM-Laufwerk z. B. als Secondary Master angeschlossen wird, wird oft ein Softlink von */dev/cdrom* nach */dev/hdc* erstellt. Das Gerät bleibt aber unter */dev/hdc* weiterhin ansprechbar.

/dev/sd*

Die Gerätedateien für SCSI-Laufwerke beginnen mit */dev/sd*. Ansonsten ist die Bezeichnung ähnlich:

- */dev/sda* – erstes SCSI-Gerät
- */dev/sdb* – zweites SCSI-Gerät
- ...und so weiter

SATA-Festplatten werden gehandhabt wie SCSI-Laufwerke und bekommen die entsprechend gleichen Gerätedateien zugewiesen. Beachten Sie bitte auch, dass manche Debian-Ableger (z. B. Ubuntu, Joli OS, Mint) auch IDE-Festplatten mit den Gerätedateien */dev/sda* usw. ansprechen.

/dev/nvme*

Bei NVMe-Geräten sieht es mit der Aufzählung der Gerätedateien etwas anders aus. Die Aufzählung der physikalischen Geräte läuft nach diesem Muster ab:

- */dev/nvme0n1*
- */dev/nvme0n2*
- ...und so weiter

Die Gerätedateien für Partitionen

Aufgrund des Aufbaus eines Master Boot Records und der darin enthaltenen Partitionstabelle können auf einer Festplatte nur vier Partitionen erstellt werden. Es wird unterschieden zwischen primären Partitionen und erweiterten Partitionen. Eine primäre Partition ist direkt ansprechbar. Sie können eine solche Partition formatieren und benutzen. Es können bis zu vier primäre Partitionen auf einer Festplatte koexistieren.

Eine erweiterte Partition kann nicht direkt verwendet werden. Sie dient lediglich als eine Art »Behälter« für logische Partitionen. Sie können nur eine einzige erweiterte Partition auf einer Festplatte anlegen. In der erweiterten Partition können dann bis zu 60 logische Partitionen angelegt werden, wenn es sich um ein IDE-Gerät handelt. Auf SCSI- und SATA-Geräten können in einer erweiterten Partition lediglich 12 logische Partitionen erstellt werden.

Partitionen auf IDE-Festplatten

Primäre und erweiterte Partitionen werden von 1 bis 4 durchnummeriert. Der ersten logischen Partition wird die Ordnungszahl 5 zugewiesen, auch wenn es nur eine primäre und eine erweiterte Partition geben sollte. Eine Partitionierung kann also z. B. so aussehen:

- */dev/hda1* – erste primäre Partition auf dem Primary Master
- */dev/hda2* – zweite primäre Partition
- */dev/hda3* – einzige erweiterte Partition
- */dev/hda5* – erste logische Partition
- */dev/hda6* – zweite logische Partition
- */dev/hda7* – dritte logische Partition

Es sind nach der Partitionierung fünf Laufwerke tatsächlich nutzbar. Die erweiterte Partition */dev/hda3* ist nicht verwendbar.

Viele Administratoren lassen sich davon verwirren, dass es üblicherweise maximal vier IDE-Geräte gibt (*/dev/hda* bis */dev/hdd*) und auch maximal vier »echte« Partitionen. Es soll deshalb erneut darauf hingewiesen werden, dass diese beiden Fakten nichts miteinander zu tun haben. Aus diesem Grund erfolgt die Nummerierung der Partitionen (auch und insbesondere der logischen Partitionen) bei SCSI-Festplatten genauso wie bei IDE-Festplatten.

Partitionen auf SCSI- und SATA-Festplatten

Das Beispiel zeigt die Partitionierung der fünften SCSI-Festplatte, durch die fünf Laufwerke zur Verfügung stehen:

- */dev/sde1* – erste primäre Partition
- */dev/sde2* – zweite primäre Partition
- */dev/sde3* – dritte primäre Partition
- */dev/sde4* – einzige erweiterte Partition
- */dev/sde5* – erste logische Partition
- */dev/sde6* – zweite logische Partition

Wie Sie sehen, erfolgt die Partitionierung von SCSI-Laufwerken nach demselben Schema wie bei IDE-Geräten.

Partitionen auf NVMe-Festplatten

Die Aufzählung der Partitionen von NVMe-Geräten unterscheidet sich von denen der klassischen Geräte. So würden drei Partitionen auf dem ersten physikalischen NVMe-Gerät diese Gerätedateien erhalten:

- */dev/nvme0n1p1*
- */dev/nvme0n1p2*
- */dev/nvme0n1p3*

iSCSI

iSCSI ist die Übertragung des *SCSI-Protokolls* über TCP-Verbindungen. Hierbei gibt es eine serverseitige Komponente, nämlich das *iSCSI-Target*, auf dem Speicherplatz bereitgestellt wird, und eine Client-Komponente, den *iSCSI-Initiator*, der über das Netzwerk auf den bereitgestellten Speicher zugreift. Dem Client-Betriebssystem wird dieser Speicher blockorientiert zur Verfügung gestellt. Der große Unterschied zum Zugriff auf eine »normale« Netzwerkfreigabe über SMB oder NFS ist der, dass der Client die Speicherressource z. B. über das Netzwerk partitionieren und formatieren kann. Er kann beim Zugriff auf mehrere sogenannte *iSCSI LUN*s (Logical Unit Number) sogar RAID-Systeme zusammenstellen. Hier eröffnen sich also gleich eine ganze Reihe von Möglichkeiten, um besondere Lösungen zu entwickeln.

iSCSI-Target konfigurieren

Es gibt verschieden Möglichkeiten, um Speicher auf einem iSCSI-Target bereitzustellen. Sie können z. B. ganze Festplatten, Partitionen, optische Laufwerke oder mit LVM erstellte Volumen bereitstellen. Für eine kleine Testumgebung zeige ich Ihnen (schon aus ökonomischen Gründen) eine Variante, in der lediglich eine Datei als Target dient. Aus Platzgründen werde ich diese Konfiguration nur unter Debian beschreiben, aber das wird ausreichen, um die prüfungsrelevanten Themen nachvollziehen zu können. Beginnen Sie mit der Installation der Target-Software:

```
root@archangel:~# apt-get install iscsitarget
```

Bei den meisten Distributionen müssen Sie zusätzlich das passende DKMS-Paket installieren, weil sonst die Unterstützung durch den Kernel fehlt:

```
apt-get install iscsitarget-dkms
```

Die Software ist per Voreinstellung deaktiviert. Sie müssen deshalb die Datei */etc/default/iscsitarget* bearbeiten und `false` in `true` abändern:

```
ISCSITARGET_ENABLE=true
```

Als Nächstes benötigen Sie einen Speicherplatz, in dem Sie die Targets bzw. LUNs erstellen können:

```
root@archangel:~# mkdir /targets && cd /targets
```

Zwei LUNs sollen als Testobjekte ausreichen. Die folgenden beiden Kommandos erzeugen jeweils 10 GB große Image-Dateien, die mit Nullen gefüllt sind. Bedenken Sie beim Erstellen der Dateien, dass die angegebenen Speicherkapazitäten auf Ihrem Datenträger tatsächlich sofort belegt werden, auch wenn die Images jetzt noch keine Daten enthalten:

```
root@archangel:/targets# dd if=/dev/zero of=speicher-lun0 count=0 obs=1 seek=10G
root@archangel:/targets# dd if=/dev/zero of=speicher-lun1 count=0 obs=1 seek=10G
```

Erstellen Sie nun in der Konfigurationsdatei */etc/iet/ietd.conf* die folgenden Einträge:

```
Target iqn.2013-11.tld.domain:imagespeicher
Lun 0 Path=/targets/speicher-lun0,Type=fileio,ScsiId=lun0,ScsiSN=lun0
Lun 1 Path=/targets/speicher-lun1,Type=fileio,ScsiID=lun1,ScsiSN=lun1
```

Bei der Bezeichnung für das Target handelt es sich um den *iSCSI Qualified Name (IQN)*. Der Aufbau dieses Namens ist wie folgt strukturiert:

- `iqn` als selbstdeutendes Symbol
- das Datum im Format YYYY-MM
- der Name der Domäne in umgekehrter hierarchischer Abfolge
- ein optionaler Doppelpunkt, gefolgt von einem optional frei wählbaren Namen

Starten Sie jetzt das Target:

```
root@archangel:/targets# /etc/init.d/iscsitarget start
```

iSCSI-Initiator konfigurieren

Die Client-Komponente ist der iSCSI-Initiator. Sie können ihn wie gehabt über das Paketmanagement installieren:

```
arch-book harald # apt-get install open-iscsi
```

Damit der Initiator (repräsentiert durch den Daemon *iscsid*) automatisch starten kann, müssen Sie in der Konfigurationsdatei */etc/iscsi/iscsid.conf* den folgenden Eintrag vornehmen (bzw. anpassen):

```
node.startup = automatic
```

Sie können das Kommando `iscsiadm` verwenden, um zu erfragen, welche iSCSI-Targets ein Remote-System bereitstellt:

```
arch-book harald # iscsiadm -m discovery -t st -p 192.168.50.1
192.168.50.1:3260,1 iqn.2013-11.tld.domain:imagespeicher
```

Verwenden Sie dasselbe Werkezeug, um den iSCSI-Initiator mit dem iSCSI-Target zu verbinden:

```
arch-book harald # iscsiadm -m node --login
Logging in to [iface: default, target: iqn.2013-
11.tld.domain:imagespeicher, portal: 192.168.50.1,3260] (multiple)
Login to [iface: default, target: iqn.2013-
11.tld.domain:imagespeicher, portal: 192.168.50.1,3260] successful.
```

Hinweis

Sie sollten in der Praxis authentifizierte und verschlüsselte Verbindungen einrichten. Die hier gezeigte Beispielkonfiguration ist nur eine Vereinfachung und als nicht sicher einzustufen!

Mit dem Kommando `dmesg` können Sie nachprüfen, ob der Kernel die beiden auf dem Target bereitgestellten LUNs registriert hat:

```
arch-book harald # dmesg|tail -n 4
[  663.925325]  sdb: unknown partition table
[  663.930017]  sdc: unknown partition table
[  663.936187] sd 6:0:0:0: [sdb] Attached SCSI disk
[  663.939605] sd 6:0:0:1: [sdc] Attached SCSI disk
```

Offensichtlich sind die Geräte *sdb* und *sdc* erkannt worden. Aus der Sicht des Kernels handelt es sich hierbei um herkömmliche SCSI-Geräte. Diese sind nun wie fabrikneue Festplatten zu behandeln. Es geht also mit der Partitionierung weiter:

```
arch-book harald # fdisk /dev/sdb
arch-book harald # fdisk /dev/sdc
```

Da Sie mit `fdisk` bereits bestens vertraut sind, will ich hier auf die Details der Partitionierung verzichten. Fahren Sie also gleich mit der Formatierung fort:

```
arch-book harald # mkfs.ext4 /dev/sdb1
arch-book harald # mkfs.ext4 /dev/sdc1
```

Jetzt müssen Sie nur noch zwei Bereitstellungspunkte erstellen und die neuen Partitionen dort einhängen:

```
arch-book harald # mkdir /iscsi-LUN1 && mount /dev/sdb1 /iscsi-LUN1
arch-book harald # mkdir /iscsi-LUN2 && mount /dev/sdc1 /iscsi-LUN2
```

Eine Überprüfung mit `df` zeigt die leeren eingehängten Partitionen an:

```
arch-book harald # df -h |grep /dev/sd
/dev/sda5       146G   42G   97G  31% /
/dev/sdb1       9,8G   23M  9,2G   1% /iscsi-LUN1
/dev/sdc1       9,8G   23M  9,2G   1% /iscsi-LUN2
```

Alternativ können Sie auf die unter Linux bereitgestellten LUNs auch von einem Windows-System aus zugreifen. Die Konfiguration des iSCSI-Initiators unter Windows ist verhältnismäßig einfach.

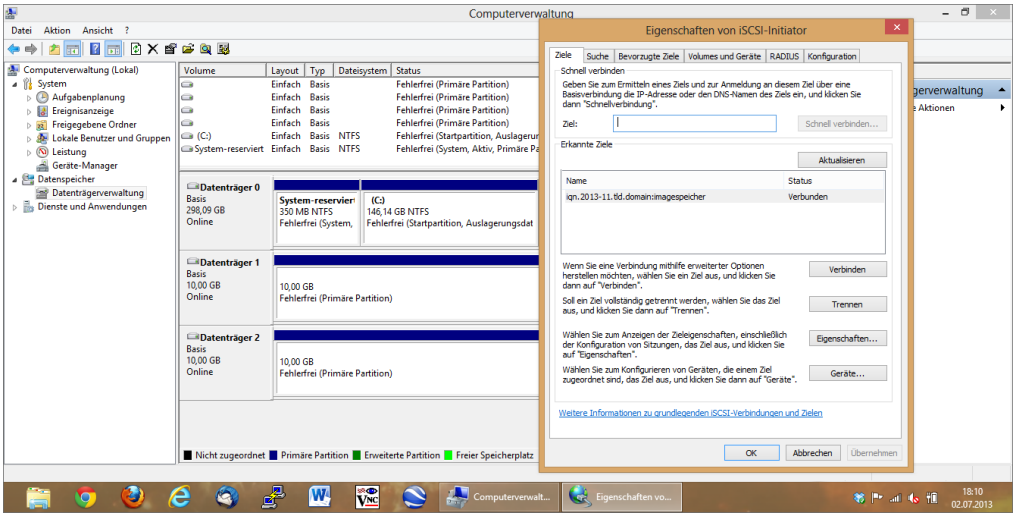

Abbildung 204.1 Der in Windows 8 integrierte iSCSI-Initiator hat beim Zugriff auf die unter Linux bereitgestellten LUNs kein Problem.

Wenn Sie die hier vorgestellte Konfiguration nachgestellt haben, können Sie die Datenträger allerdings so nicht verwenden, weil Windows für *ext4*-Dateisysteme keine Treiberunterstützung bietet.

WWID, WWN, LUN und das Werkzeug scsi_id

Allen SCSI-, ATA-, SAS- und Fibre-Channel-Geräten ist jeweils eine WWID bzw. ein WWN und eine LUN zugeordnet. Das gilt natürlich auch für iSCSI-Geräte. Begriffserklärungen:

- *WWID* (World Wide ID) ist eine eindeutige Bezeichnung. Hierbei handelt es sich um eine 8- oder 16-Bytes-Zahl.
- *WWN* (World Wide Name) entspricht WWID.
- *LUN* (Logical Unit Number) bezeichnet eine logische Verwaltungseinheit innerhalb eines Speichergerätes, zumeist beginnend mit LUN 0. Die meisten SCSI-Geräte weisen überhaupt nur die LUN 0 auf. Ein Beispiel für mehrere LUNs innerhalb eines physischen Gerätes wäre ein DVD-Wechsler, bei dem jedem DVD-Schacht eine eigene LUN zugeordnet ist. Bei der Verwendung mehrerer virtueller Geräte innerhalb eines iSCSI-Targets werden zur Unterscheidung ebenfalls LUNs verwendet.

Mit dem Werkzeug `scsi_id` können Sie die oben beschriebenen Parameter einsehen. Zu diesem Zweck fragt `scsi_id` die Geräte mittels *SCSI Inquiry* ab. Es können zwei Seiten, nämlich `0x80` und `0x83`, von *VPD* (Vital Product Data) abgefragt werden. Die Seite `0x80` enthält lediglich Grundinformationen über den Gerätetyp und die LUN. Die folgenden Abfragen stammen von den im vorangehenden Beispiel erstellten iSCSI-Geräten:

```
arch-book # /lib/udev/scsi_id --page=0x80 --whitelisted /dev/sdb
SIET     VIRTUAL-DISK              lun0
arch-book # /lib/udev/scsi_id --page=0x80 --whitelisted /dev/sdc
SIET     VIRTUAL-DISK              lun1
```

Die Option `--whitelisted` ist hier erforderlich, weil in diesem Szenario keine Konfigurationsdatei den Standardwert (blacklisted) aufhebt. Dementsprechend würde das Kommando ohne diese Option keine Ausgaben liefern. Die folgenden beiden Befehle zeigen die WWIDs der Geräte, indem die Seite `0x83` abgerufen wird:

```
arch-book # /lib/udev/scsi_id --page=0x83 --whitelisted /dev/sdb
14945540000000006c756e3000000000000000000000000000
arch-book # /lib/udev/scsi_id --page=0x83 --whitelisted /dev/sdc
14945540000000006c756e3100000000000000000000000000
```

SAN (AoE, FCoE)

Der Verwendungszweck von *SAN*s (*Storage Area Networks*) ähnelt dem von *iSCSI*. Auch hier will man die physikalische Infrastruktur von Ethernet verwenden, um auf

Speichermedien zuzugreifen. Der Hauptunterschied ist der, dass zum Transport nicht IP zum Einsatz kommt, sondern eigene Protokolle verwendet werden, die auf den Layer 2 (Osi-Modell) aufgesetzt werden. Sie sollten zwei dieser Protokolle kennen, nämlich:

- *AoE* steht für ATA over Ethernet. Dieses Protokoll sendet gewöhnliche ATA-Kommandos, wie sie bei SATA und PATA vorkommen, eingekapselt in Ethernet-Frames.
- *FCoE* kapselt entsprechend Fibre-Channel-Frames in Ethernet-Frames.

Genauere Kenntnisse über diese Protokolle sind für die Prüfung nicht erforderlich.

204.3 Logical Volume Manager

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Logical Volumes (logische Volumen), Volume Groups (Volumengruppen) und Physical Volumes (physikalische Volumen) anzulegen und zu löschen. Dieses Lernziel beinhaltet auch die Erstellung von Snapshots und die Größenveränderung von Logical Volumes.

Wichtigste Wissensgebiete:

- Werkzeuge der LVM-Suite
- Anpassen der Größe, Umbenennen, Anlegen und Löschen von logischen Volumen, Volumengruppen und physikalischen Volumen
- Anlegen und Pflegen von Snapshots
- Aktivieren von Volumengruppen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `/sbin/pv*`
- `/sbin/lv*`
- `/sbin/vg*`
- `mount`
- */dev/mapper/*
- *lvm.conf*

Allgemeines

LVM ist ein Akronym für Logical Volume Manager. Der Sinn von LVM ist die Abstraktion physikalischer Datenträger gegenüber dem Dateisystem. Im Gegensatz zu RAID-

Systemen bieten logische Volumen keine Redundanz. Trotzdem gibt es eine Parallele, nämlich die Abstraktion. Bei der Konfiguration von RAID haben Sie mehrere Datenträger zusammengefasst. Anschließend haben Sie das RAID-Array aber »am Stück« formatiert. Dem Dateisystem wurde also verschwiegen, dass es sich um mehrere Datenträger handelte. Bei LVM ist das ähnlich, wenn auch der Verwendungszweck ein anderer ist. Aus der Sicht des Dateisystems wird beim Einsatz von LVM auf logische Volumen zugegriffen. Der Vorteil dieser Technologie liegt in der flexiblen Verwendung von Speicherressourcen. Ein logisches Volumen kann nachträglich vergrößert oder auch verkleinert werden. Sollte einem Server der Speicherplatz in einem logischen Volumen ausgehen, können Sie einfach weitere Festplatten hinzufügen und für das bestehende Volumen verfügbar machen. Umgekehrt können Sie einem überdimensionierten Server wieder Festplatten entnehmen (was man in der Praxis allerdings kaum machen wird).

Wenn Sie die Vorteile von RAID und LVM miteinander kombinieren möchten, können Sie LVM auf einem bestehenden RAID-Array einrichten.

LVM-Komponenten und Zusammenhänge

LVM besteht aus drei Komponenten. Wenn Sie den Zusammenhang dieser drei Komponenten verstanden haben, fällt es Ihnen auch nicht mehr schwer, die relativ zahlreichen Kommandos, die in diesem Thema verwendet werden, voneinander zu unterscheiden. *Physikalische Volumen* (`pv`) sind vergleichbar mit echten Partitionen auf einer Festplatte. Tatsächlich werden diese Volumen auch mittels `fdisk` vorbereitet. Der Dateisystemtyp ist `8E`. *Volumengruppen* (`vg`) sind ein Zusammenschluss aus mehreren physikalischen Volumen. Sie können diese Gruppen nachträglich mit weiteren physikalischen Volumen erweitern. *Logische Volumen* (`lv`) werden innerhalb der Volumengruppen erstellt. Aus der Sicht des Dateisystems handelt es sich hierbei um Partitionen. Sie werden letztendlich in den Dateisystembaum gemountet und verwendet.

Wenn Sie den Zusammenhang dieser drei Bauelemente von LVM verstanden haben, wird es Ihnen nicht weiter schwer fallen, die zugehörigen Kommandos zu verwenden. Sie beginnen jeweils mit den beiden in Klammern stehenden Zeichen der Begriffserklärung. Sie können sich über die Befehle einen schnellen Überblick verschaffen:

```
root@arch-deb-book:/# ls -l /sbin/pv* /sbin/lv* /sbin/vg*
```

Sie werden feststellen, dass es sich bei fast allen Kommandos um Softlinks handelt, die auf */sbin/lvm* zeigen und dann in Abhängigkeit vom verwendeten Befehl (die Variable `$0` wird ausgewertet) ihre Arbeit verrichten. Sollte der `ls`-Befehl bei Ihnen kein brauchbares Ergebnis liefern (ca. 40 Zeilen), dann müssen Sie das Paket `lvm2` nachinstallieren:

```
[root@arch-fc /] yum install lvm2
```

oder bei Debian und dessen Ablegern:

```
root@arch-deb:/# apt-get install lvm2
```

LVM-Konfiguration

Auf dieser und den folgenden Seiten finden Sie alle Schritte, die notwendig sind, um LVM einzurichten. Die Vorgehensweise ist hierbei darauf ausgerichtet, das System zu verstehen und die dazu passenden Prüfungsfragen beantworten zu können. In diesem Beispiel kommen vier Festplatten zum Einsatz. Sie können das Szenario aber auch mit zwei Festplatten komplett nachvollziehen. An dieser Stelle möchte ich Sie aber noch einmal darauf hinweisen, dass LVM standardmäßig keine Ausfallsicherheit bietet. Wenn Sie vier Festplatten zusammenschalten und zum Schluss in einem einzigen logischen Volumen bereitstellen, wird die Wahrscheinlichkeit von Datenverlust bei Ausfall einer Festplatte um den Faktor vier erhöht.

Wenn Sie nicht gerade eine extrem alte Linux-Distribution einsetzen, ist die Unterstützung von LVM in Ihrem Kernel bereits integriert. Sie benötigen, wie im vorangehenden Abschnitt bereits erwähnt, das Paket `lvm2`, um die Konfiguration nachvollziehen zu können.

Vorbereitung

Für einen ersten Funktionstest können Sie das Kommando `vgscan -v` verwenden:

```
[root@arch-fc /]# vgscan -v
    Wiping cache of LVM-capable devices
    Wiping internal VG cache
  Reading all physical volumes.  This may take a while...
    Finding all volume groups
    Finding volume group "vg_archfc"
  Found volume group "vg_archfc" using metadata type lvm2
```

Das Programm `vgscan` durchsucht alle angeschlossenen blockorientierten Geräte nach physikalischen LVM-Volumen und nach Volumengruppen. Das Ergebnis dieser Abfrage wird in der Datei */etc/lvm/cache/.cache* gespeichert. Sie sollten diese Datei nicht von Hand bearbeiten. Die einzige Konfigurationsdatei, die zur manuellen Bearbeitung infrage kommt, ist die Datei *lvm.conf*. Sie finden diese Datei im Verzeichnis */etc/lvm*. Sie können hier z. B. festlegen, welche Geräte `vgscan` nicht durchsuchen soll. Da es keine Probleme verursacht, wenn `vgscan` z. B. ein CD-ROM-Laufwerk anspricht, können Sie diese Filterung aber auch einfach weglassen.

Im vorliegenden Fall hat `vgscan` bereits existierende Volumengruppen gefunden, weil das Betriebssystem darauf installiert worden ist. Das kann bei Ihnen natürlich ganz anders aussehen.

Damit Sie neue Festplatten für LVM verfügbar machen können, müssen Sie zunächst mittels `fdisk` (oder auch `cfdisk`) geeignete Partitionen vom Typ Linux LVM erstellen. Der Code für diesen Partitionstyp ist `8E`. In diesem Beispiel kommen Festplatten mit den Gerätenamen */dev/sdb*, */dev/sdc*, */dev/sdd* und */dev/sde* zum Einsatz. Alle erhalten eine primäre Partition mit der Kennung 1. Sie können das aber natürlich Ihren eigenen Wünschen entsprechend anpassen. Die Partitionierung kennen Sie schon; deshalb gibt es hier nur noch Service im Telegrammstil:

```
[root@arch-fc /]# fdisk /dev/sdb
Befehl (m für Hilfe): n
```

Betätigen Sie [n] für »neue Partition«.

```
Befehl  Aktion
   e      Erweiterte
   p      Primäre Partition (1-4)
p
```

Betätigen Sie [p] für »primäre Partition«.

```
Partitionsnummer (1-4, Vorgabe: 1): 1
```

Es soll die Partitionskennung `1` verwendet werden.

```
Erster Sektor (2048-16777215, Vorgabe: 2048):
Benutze den Standardwert 2048
Last Sektor, +Sektoren or +size{K,M,G} (2048-16777215, Vorgabe: 16777215):
Benutze den Standardwert 16777215
```

Den ersten und den letzten Sektor können Sie jeweils mit [↵] quittieren, wenn Sie die ganze Festplatte verwenden möchten.

```
Befehl (m für Hilfe): t
```

Mit [t] wird der Partitionstyp festgelegt.

```
Partition 1 ausgewählt
Hex code (L um eine Liste anzuzeigen): 8e
Der Dateisystemtyp der Partition 1 ist nun 8e (Linux LVM)
```

Wählen Sie [8e], damit eine Partition vom Typ Linux LVM erstellt wird.

```
Befehl (m für Hilfe): w
```

Zum Abschluss schreiben Sie mit der Taste [w] die Änderungen in die Partitionstabelle.

```
Die Partitionstabelle wurde verändert!
Rufe ioctl() um Partitionstabelle neu einzulesen.
Synchronisiere Platten.
```

Um unnötige Arbeit zu sparen, kann man die Partitionstabelle der gerade bearbeiteten Festplatte mittels `sfdisk` auf die anderen Festplatten kopieren:

```
[root@arch-fc /]# sfdisk -d /dev/sdb | sfdisk /dev/sdc
[root@arch-fc /]# sfdisk -d /dev/sdb | sfdisk /dev/sdd
[root@arch-fc /]# sfdisk -d /dev/sdb | sfdisk /dev/sde
```

Sollten Sie beim Kopieren der Partitionstabelle Probleme haben, versuchen Sie dem zweiten `sfdisk`-Kommando die Option `--force` zu übergeben.

Als Nächstes teilen Sie dem System mit, dass es neue physikalische Volumen gibt. Das Kommando kennen Sie ja bereits:

```
[root@arch-fc /]# vgscan -v
    Wiping cache of LVM-capable devices
    Wiping internal VG cache
  Reading all physical volumes.  This may take a while...
    Finding all volume groups
    Finding volume group "vg_archfc"
  Found volume group "vg_archfc" using metadata type lvm2
```

Erstellung der physikalischen Volumen

Als Nächstes wird das Dateisystem auf dem physikalischen Volumen erzeugt. Das kommt einer Formatierung des Datenträgers gleich:

```
[root@arch-fc ~]# pvcreate -v /dev/sdb1
Set up physical volume for "/dev/sdb1" with 16775168 available sectors
  Zeroing start of device /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
```

Sie haben jetzt ein physikalisches Volumen erstellt. Spätestens an dieser Stelle ist klar, was dieser etwas abstrakte Begriff bedeutet. Mit dem Kommando `pvdisplay` können Sie das Ergebnis überprüfen:

```
[root@arch-fc ~]# pvdisplay
  "/dev/sdb1" is a new physical volume of "8,00 GiB"
  --- NEW Physical volume ---
```

```
  PV Name               /dev/sdb1
  VG Name
  PV Size               8,00 GiB
  Allocatable           NO
  PE Size               0
  Total PE              0
  Free PE               0
  Allocated PE          0
  PV UUID               l3AxSw-p6WC-eTOO-7DFv-WO2b-JdGq-hgvJ3O
```

Nachdem die Erstellung des ersten physikalischen Volumens gut funktioniert hat, können die verbleibenden drei Festplatten genauso behandelt werden. Erstellen Sie die verbleibenden physikalischen Volumen:

```
[root@arch-fc ~]# pvcreate -v /dev/sdc1
[root@arch-fc ~]# pvcreate -v /dev/sdd1
[root@arch-fc ~]# pvcreate -v /dev/sde1
```

Erstellung der Volumengruppe

Zu diesem Zeitpunkt verfügen Sie über vier physikalische Volumen (pv). Diese werden im nächsten Arbeitsgang zu einer Volumengruppe (vg) mit der Bezeichnung *vg_big* zusammengeführt. Hierbei wird die Gerätedatei */dev/vg_big* erzeugt. Sie können eine Volumengruppe nicht benutzen. Es handelt sich lediglich um eine Verwaltungseinheit des LVM-Systems. Sie können die Gruppe übrigens benennen wie Sie wollen, aber es empfiehlt sich das Präfix vg_ zu verwenden, damit man der Gerätedatei gleich ansieht, wozu sie verwendet wird:

```
[root@arch-fc ~]# vgcreate vg_big /dev/sdb1 /dev/sdc1 \
/dev/sdd1 /dev/sde1
  Volume group "vg_big" successfully created
```

Sie können das Ergebnis der Erstellung wieder überprüfen. Diesmal verwenden Sie den Befehl vgdisplay:

```
[root@arch-fc /]# vgdisplay
  --- Volume group ---
  VG Name               vg_big
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  28
  VG Access             read/write
  VG Status             resizable
```

Die Ausgabe des Kommandos wurde aus Platzgründen deutlich gekürzt, aber Sie können es ja selbst ausprobieren.

Erstellung logischer Volumen

Das finale Produkt, auf dem später auch die Daten abgelegt werden, ist das logische Volumen. Bei der Benennung des logischen Volumens sollten Sie `lv_` als Präfix verwenden. Das hilft Ihnen später, den Sinn der Gerätedatei sofort zu erkennen, die Sie gleich erzeugen werden. Im folgenden Beispiel ist also `lv_big_1` das logische Volumen und `vg_big` die Volumengruppe, in der das logische Volumen erstellt wird. Die Option `-L` bestimmt die Größe des zu erzeugenden Volumens und legt diese auf 20 GB fest:

```
[root@arch-fc ~]# lvcreate -n lv_big_1 -L 20G vg_big
  Logical volume "lv_big_1" created
```

Das logische Volumen wurde erstellt und die Gerätedatei entsprechend dem Kommando als */dev/vg_big/lv_big_1* angelegt. Das Ergebnis können Sie mit dem Befehl `lvdisplay` überprüfen:

```
[root@arch-fc /]# lvdisplay /dev/vg_big/lv_big_1
  --- Logical volume ---
  LV Name                /dev/vg_big/lv_big_1
  VG Name                vg_big
  LV UUID                lg2yS4-YveV-REmP-PCMo-5SO1-aDix-3D7Ufg
  LV Write Access        read/write
  LV Status              available
  # open                 1
  LV Size                20,00 GiB
```

Logische Volumen verwenden

Wie jeder andere Datenträger muss das Volumen formatiert werden, bevor Sie es verwenden können. Da es sich aus der Sicht des Dateisystems um eine gewöhnliche Partition handelt, ist bei der Formatierung auch nichts Ungewöhnliches zu beachten:

```
[root@arch-fc ~]# mkfs.ext4 /dev/vg_big/lv_big_1
```

Um den Datenträger zu testen, legen Sie einfach ein Testverzeichnis an, mounten den Datenträger und kopieren ein paar Daten. Es sollte alles so funktionieren wie bei einem konventionellen Datenträger:

```
[root@arch-fc /]# mkdir /lvtest
[root@arch-fc /]# mount /dev/vg_big/lv_big_1 /lvtest/
[root@arch-fc /]# cp /boot/* /lvtest/
```

Logische Volumen erweitern

Achtung

Sie sollten vor der Erweiterung von logischen Volumen auf jeden Fall eine Datensicherung durchführen. Auch wenn die hier gezeigten Beispiele leicht von der Hand zu gehen scheinen, können in der Hektik immer mal Fehler gemacht werden, die dann zu vollständigem Datenverlust führen.

Der Hauptvorteil von LVM ist die Erweiterbarkeit von Volumen. Die Volumengruppe besteht in unserem Beispiel aus vier physikalischen Volumen zu je 8 GB. Es ergibt sich also eine maximale Kapazität von 32 GB. Das einzige bisher existierende logische Volumen hat eine Größe von lediglich 20 GB. Es gibt also noch etwas Spielraum. Mit `lvextend` können Sie das bestehende logische Volumen erweitern. Die Option `-L` gibt hierbei die neue Größe des Volumens (`25G`) nach der Erweiterung an:

```
[root@arch-fc ~]# lvextend -L 25G /dev/vg_big/lv_big_1
  Extending logical volume lv_big_1 to 25,00 GiB
  Logical volume lv_big_1 successfully resized
```

Wenn Sie */dev/vg_big/lv_big_1* mittels `df` überprüfen würden, wäre die neue Größe noch nicht feststellbar und die Kapazität auch nicht nutzbar. Das Dateisystem muss nämlich ebenfalls in seiner Größe angepasst werden. Hier kommt das Werkzeug `resize2fs` zum Einsatz, das Sie ja bereits kennen. Wenn Sie diesem Programm keine Zielgröße als Parameter angeben, wird es das Dateisystem auf die Gesamtgröße der angegebenen Partition erweitern. Neue Versionen von `resize2fs` unterstützen auch die *ext3*- und *ext4*-Dateisysteme. Sie können die Größe des Dateisystems sogar ändern, ohne es vorher auszuhängen, wie Sie gleich sehen werden:

```
[root@arch-fc ~]# resize2fs /dev/vg_big/lv_big_1
resize2fs 1.41.12 (17-May-2010)
Das Dateisystem auf /dev/vg_big/lv_big_1 ist auf /lvtest eingehängt;
Online-Grössenveränderung nötig
old desc_blocks = 2, new_desc_blocks = 2
Führe eine Online-Grössenänderung von /dev/vg_big/lv_big_
1 auf 6553600 (4k) Blöcke durch.
Das Dateisystem auf /dev/vg_big/lv_big_1 ist nun 6553600 Blöcke groß.
```

Sie können jetzt die Kapazität von 25 GB nutzen. Eine Abfrage mit dem Programm `df` wird das nachweisen:

```
[root@arch-fc ~]# df -h /lvtest
Dateisystem           Size  Used Avail Use% Eingehängt auf
/dev/mapper/vg_big-lv_big_1
                       25G  191M   24G   1% /lvtest
```

Logische Volumen verkleinern

Achtung

Auch vor der Verkleinerung von logischen Volumen sollten Sie immer eine Datensicherung durchführen. Vollständiger Datenverlust ist nämlich auch hier nicht unwahrscheinlich.

Die Verkleinerung eines logischen Volumens funktioniert genau andersherum als die Vergrößerung. Sie müssen nämlich zuerst das (oder die, wenn es mehrere sind) Dateisystem(e) verkleinern. Das muss deshalb sein, weil die Dateisysteme sonst nach der Verkleinerung des logischen Volumens nicht mehr auf dieses passen würden. Die Werkzeuge, die hier zum Einsatz kommen, würden es allerdings nicht zulassen, dass Sie einen solchen Fehler begehen. Sie würden durch eine entsprechende Fehlermeldung rechtzeitig gestoppt. Eine Verkleinerung kann im Gegensatz zu einer Vergrößerung auch nicht online durchgeführt werden. Deshalb beginnt die Arbeit mit dem Aushängen des Dateisystems:

```
[root@arch-fc ~]# umount /lvtest/
```

Anschließend muss das Dateisystem mit `e2fsck` überprüft werden. Sie können sonst nicht fortfahren, weil `resize2fs` diesen Arbeitsschritt voraussetzt. Die Dateisysteme *ext3* und *ext4* werden von `e2fsck` inzwischen problemlos unterstützt, auch wenn der Name des Programms das nicht erwarten lässt.

```
[root@arch-fc ~]# e2fsck -f /dev/vg_big/lv_big_1
e2fsck 1.41.12 (17-May-2010)
Durchgang 1: Prüfe Inodes, Blocks, und Größen
Durchgang 2: Prüfe Verzeichnis Struktur
Durchgang 3: Prüfe Verzeichnis Verknüpfungen
Durchgang 4: Überprüfe die Referenzzähler
Durchgang 5: Überprüfe Gruppe Zusammenfassung
/dev/vg_big/lv_big_1: 18/1638400 Dateien (0.0% nicht zusammenhängend),
151612/6553600 Blöcke
```

Nach der gründlichen Prüfung des Dateisystems kann dieses nun verkleinert werden. Diesmal muss natürlich auch eine Zielgröße angegeben werden (1.024.000 Blöcke zu je 4k ergeben 4 GB). Zur Erinnerung: Die aktuelle Größe des Dateisystems beträgt 25 GB:

```
[root@arch-fc ~]# resize2fs /dev/vg_big/lv_big_1 1024000
resize2fs 1.41.12 (17-May-2010)
Die Grösse des Dateisystems auf /dev/vg_big/lv_big_
1 wird auf 1024000 (4k) Blöcke geändert.
Das Dateisystem auf /dev/vg_big/lv_big_1 ist nun 1024000 Blöcke groß.
```

Nach der Verkleinerung des Dateisystems kann auch das logische Volumen verkleinert werden. Die neue Größe soll, passend zum Dateisystem, nach der Änderung 4 GB betragen:

```
[root@arch-fc ~]# lvreduce -L 4G /dev/vg_big/lv_big_1
  WARNING: Reducing active logical volume to 4,00 GiB
  THIS MAY DESTROY YOUR DATA (filesystem etc.)
Do you really want to reduce lv_big_1? [y/n]: y
  Reducing logical volume lv_big_1 to 4,00 GiB
  Logical volume lv_big_1 successfully resized
```

Der Ausgabe des Programms nach zu urteilen, hat alles geklappt. Beachten Sie auch nochmal die Warnung: `THIS MAY DESTROY YOUR DATA`. Sie sollten also mit Produktionsdaten wirklich vorsichtig sein.

Physikalische Volumen aus einer Volumengruppe entfernen und physikalische Volumen zu einer Gruppe hinzufügen

Im Augenblick befindet sich in der Speichergruppe *vg_big* nur noch ein logisches Volumen mit einer Größe von 4 GB. Da die physikalischen Volumen alle jeweils eine Kapazität von 8 GB aufweisen, müssten theoretisch drei der Datenträger entnehmbar sein. Bei der Entfernung kommt `vgreduce` zum Einsatz. Wenn die Option `-a` verwendet wird, werden automatisch alle physikalischen Volumen aus der Volumengruppe entfernt, die leer sind. Es müssen dann keine physikalischen Volumen angegeben werden:

```
[root@arch-fc ~]# vgreduce -a vg_big
  Physical volume "/dev/sdb1" still in use
  Removed "/dev/sdc1" from volume group "vg_big"
  Removed "/dev/sdd1" from volume group "vg_big"
  Removed "/dev/sde1" from volume group "vg_big"
```

Im Prinzip können die drei Festplatten, auf denen sich die entfernten physikalischen Volumen befinden, jetzt ausgebaut werden. Da es hier aber nur um die Vorführung der LVM-Kommandos geht, soll stattdessen eines der entfernten physikalischen Volumen der Volumengruppe wieder hinzugefügt werden. Hierbei wird der Befehl `vgextend` verwendet. Als Parameter benötigt `vgextend` den Namen der Volumengruppe (*vg_big*) und die Gerätedatei des physikalischen Volumens (*/dev/sdc1*), das hinzugefügt werden soll:

```
[root@arch-fc ~]# vgextend vg_big /dev/sdc1
  Volume group "vg_big" successfully extended
```

Durch diese Erweiterung umfasst die Volumengruppe jetzt eine Kapazität von 16 GB. Von diesem Speicherpool sind allerdings nur 4 GB für das logische Volumen *lv_big_1* in Gebrauch.

LVM-Snapshots

Wenn Sie eine große Datenmenge sichern müssen, ergibt sich das Problem, dass sich die zu sichernden Daten während des Sicherungsprozesses fortlaufend ändern könnten. Daraus können sich bei einer Wiederherstellung der Daten Inkonsistenzen ergeben. Diese Problematik wird durch LVM-Snapshots elegant gelöst. Es wird hierbei zunächst eine Momentaufnahme erstellt (jedenfalls sieht es nach außen hin so aus) und diese dann gesichert. Nach der Sicherung kann (bzw. sollte) der Snapshot wieder entfernt werden.

Da ein Snapshot in Wirklichkeit nur die Änderungen der Daten während des Backups »auffängt«, kann das logische Volumen, das für den Snapshot verwendet wird, relativ klein gehalten werden. Das in diesem Beispiel noch existierende logische Volumen *lv_big_1* ist 4 GB groß. Für den Snapshot sollte, wenn man lediglich geringe Änderungen an den Daten erwartet, eine Größe von 1 GB absolut ausreichen. Das folgende Kommando erstellt ein Snapshot-Volumen, das unter der Gerätedatei */dev/vg_big/big1backup* erreichbar sein wird. Die Quelle für den Snapshot ist das logische Volumen */dev/vg_big/lv_big_1*. Die Größe des Snapshots wird mit 1 GB (`-L 1G`) festgelegt.

```
[root@arch-fc /]# lvcreate-L 1G -s -n big1backup /dev/vg_big/lv_big_1
  Logical volume "big1backup" created
```

Das Volumen wurde erfolgreich erstellt. Bei einigen Kernel-Versionen kommt es bei der Erstellung des Snapshots zu Fehlermeldungen, die in etwa so aussehen:

```
Internal error: Maps lock 35007744 < unlock 35269888
```

Sie können aber dennoch fortfahren, denn die Fehlermeldung hat keine negativen Auswirkungen auf den Snapshot und kann ignoriert werden.

Sie können den Snapshot jetzt in den Verzeichnisbaum einhängen:

```
[root@arch-fc /]# mount /dev/vg_big/big1backup /backup/
```

Mit welchem Programm Sie die Datensicherung letztendlich durchführen, ist unerheblich. Eine einfache Möglichkeit ist die Verwendung von `tar`:

```
[root@arch-fc /]# tar -pzcf backup.tar.gz /backup/
tar: Entferne führende "/" von Elementnamen
```

Wenn `tar` seine Arbeit getan hat, können Sie den Snapshot wieder aushängen und anschließend mithilfe des Programms `lvremove` aus der Volumengruppe entfernen:

```
[root@arch-fc /]# umount /backup
[root@arch-fc /]# lvremove /dev/vg_big/big1backup
Do you really want to remove active logical volume big1backup? [y/n]:
y
Logical volume "big1backup" successfully removed
```

Auch hier können Sie in Abhängigkeit von der verwendeten Kernel-Version eine solche Fehlermeldung erhalten:

```
Internal error: Maps lock 35007744 < unlock 35269888
```

Aber auch diesmal können Sie die Fehlermeldung getrost ignorieren. Es ist nicht mit Problemen zu rechnen, die mit dieser Fehlermeldung in Zusammenhang stehen.

Es ist im Übrigen absolut sinnvoll, einen Snapshot sofort nach dem Backup wieder zu entfernen, weil sich die Existenz des Snapshots negativ auf die Performance des logischen Volumens auswirkt. Davon abgesehen müssen Sie beim nächsten Backup ohnehin einen neuen Snapshot erstellen, weil ein solcher Snapshot lediglich eine Momentaufnahme darstellt – und die haben Sie ja bereits gesichert. Für eine regelmäßige Datensicherung sollten Sie ein Skript schreiben, das die obigen Kommandos (sinngemäß) enthält. Dieses Skript können Sie dann regelmäßig von `cron` ausführen lassen.

Device Mapper

Während der Konfiguration auf den letzten Seiten sind Sie mit dem Device Mapper gar nicht in Berührung gekommen. Bei der Auflistung der Prüfungsthemen wird */dev/mapper* aber ausdrücklich genannt, weshalb hier wenigstens kurz ein paar Worte darüber verloren werden sollen.

Der Device Mapper ist eine Komponente des Kernels, welche die Gerätedateien aufeinander abbildet. Er ist eine Grundvoraussetzung für das logische Volumenmanagement. Sie können die Wirkungsweise des Device Mappers sehen. Das logische Volumen, das auf den vorangehenden Seiten erstellt worden ist, haben Sie wie folgt eingehängt:

```
[root@arch-fc /]# mount /dev/vg_big/lv_big_1 /lvtest/
```

Wenn Sie mit dem Kommando `mount` prüfen, wo das Volumen montiert ist, wäre demzufolge ein solches Ergebnis zu erwarten:

```
/dev/vg_big/lv_big_1 on /lvtest type ext4 (rw)
```

In Wirklichkeit wird es aber zu einer solchen Ausgabe kommen:

```
/dev/mapper/vg_big-lv_big_1 on /lvtest type ext4 (rw)
```

Das logische Gerät mit der Bezeichnung */dev/vg_big/lv_big_1* wurde ganz offensichtlich auf */dev/mapper/vg_big-lv_big_1* abgebildet.

Der Sinn dieser Abbildungen ist letztendlich die Abstraktion der physikalischen Volumen gegenüber Programmen. Programme »sehen« also nur noch die logischen Volumen, während ihnen die physikalischen Volumen verborgen bleiben.

205 Netzwerkkonfiguration

Nachdem das X Window System nach Linux portiert worden war, bot es sich an, auch das Netzwerk zu implementieren. Schließlich basiert die Kommunikation mit einem X-Server ohnehin auf Unix-Sockets, die auch für das Netzwerk nötig sind.

205.1 Grundlagen der Netzwerkkonfiguration

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, ein Netzwerkgerät zu konfigurieren und dieses in ein lokales kabelgebundenes oder Funknetzwerk einzubinden. Dieses Lernziel beinhaltet die Fähigkeit, eine Verbindung zwischen verschiedenen Subnetzen eines Netzwerks unter IPv4 und IPv6 herzustellen.

Wichtigste Wissensgebiete:

- Dienstprogramme zur Konfiguration und Manipulation von Ethernet-Schnittstellen
- Konfiguration von Funknetzwerken

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `route`
- `ifconfig`
- `ip`
- `arp`
- `iw`
- `iwconfig`
- `iwlist`

Allgemeines

Eine grundlegende Fertigkeit ist die Konfiguration und Überprüfung von Netzwerkeinstellungen im laufenden Betrieb. Die dazu erforderlichen Werkzeuge sind in allen Linux-Distributionen und auch standardmäßig in der Grundinstallation enthalten. Einige Themen, denen sich dieses und die nachfolgenden Kapitel widmen, waren

schon Bestandteil des LPIC-Levels 1. Deshalb soll an dieser Stelle auch nicht noch einmal auf die Grundlagen von TCP/IP eingegangen werden, sondern prüfungsnah die Verwendung der relevanten Werkzeuge besprochen werden. In diesem ersten Teil des Kapitels nehmen Sie nur ein paar einfache Konfigurationen vor, die übrigens alle nicht dauerhaft sind, sondern nach einem Systemneustart verloren gehen.

Werkzeuge zur Netzwerkkonfiguration

ifconfig

Das Programm `ifconfig` wird verwendet, um Netzwerkkarten zu konfigurieren oder deren Konfiguration zu prüfen. Sie können mit `ifconfig` auch Netzwerkschnittstellen aktivieren bzw. deaktivieren. Aber Vorsicht: Es gehen dann alle Einträge in den Routing-Tabellen, die mit dieser Netzwerkschnittstelle assoziiert sind, verloren. Diese Einträge werden beim erneuten Start der Netzwerkschnittstelle nicht automatisch neu erstellt. Das Programm `ifconfig` wird keine Erfolgsmeldungen von sich geben, wenn Sie es verwenden. Deshalb müssen Sie die vorgenommenen Einstellungen selbst prüfen. Hier zur Veranschaulichung ein paar Anwendungsbeispiele:

```
root@ipcop:~ # ifconfig eth1 down
```

Die Netzwerkschnittstelle `eth1` ist nun deaktiviert. Es folgt eine Neukonfiguration der Karte mit IP-Adresse und Subnetzmaske:

```
root@ipcop:~ # ifconfig eth1 20.14.5.88 netmask 255.255.255.192
```

Nach Rekonfiguration einer Schnittstelle ist keine Aktivierung erforderlich. Das geschieht implizit von selbst. Wäre die IP-Konfiguration nicht verändert worden, hätten Sie `eth1` durch folgendes Kommando wieder gestartet:

```
root@ipcop:~ # ifconfig eth1 up
```

Eine IPv6-Adresse können Sie ebenfalls mit `ifconfig` setzen. Allerdings müssen Sie dann zusätzlich die Adressfamilie angeben:

```
root@ipcop:~ # ifconfig eth1 inet6 add 2001:6f8:1d2d::4
```

Die aktuelle Konfiguration aller Netzwerkschnittstellen können Sie durch die Eingabe von `ifconfig` ohne Parameter ansehen. Wenn Sie der Übersichtlichkeit halber nur eine einzige Schnittstelle betrachten wollen, geben Sie deren Namen mit auf der Befehlszeile an. Das Ergebnis der vorangehenden Beispiele sollte schließlich noch überprüft werden:

```
root@ipcop:~ # ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:0C:29:9E:F5:EF
inet addr:20.14.5.88 Bcast:20.14.5.255 Mask:255.255.255.192
```

```
inet6-Adresse: 2001:6f8:1cfe::4/64 Gültigkeitsbereich:Global
inet6-Adresse: fe80::20d:60ff:feae:de4/64 Gültigkeitsbereich:Verbindung
UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
RX packets:0 errors:0 dropped:0 overruns:0 frame:0
TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1000
RX bytes:0 (0.0 b)  TX bytes:0 (0.0 b)
Interrupt:10 Base address:0x1400
```

Alle Parameter sehen aus wie erwartet. Bleibt nur noch die Frage nach den verloren gegangenen Routing-Tabelleneinträgen zu klären. Doch dazu gleich mehr.

route

Mit dem Kommando `route` können Sie Routing-Tabellen überprüfen und anpassen. Aus einem vorangehenden Beispiel zu `ifconfig` steht immer noch die Prüfung und Rekonfiguration des Routings aus. Durch die Deaktivierung der Schnittstelle `eth1` waren nämlich alle mit dieser Netzwerkkarte assoziierten Routing-Einträge entfernt worden. Ein Blick in die Routing-Tabelle des Rechners zeigt das Dilemma:

```
root@ipcop:~ # route
Kernel IP routing table
Destination   Gateway      Genmask         F. M. Ref. Use Iface
20.14.5.64    *            255.255.255.192 U  0  0    0 eth1
192.168.0.0   *            255.255.255.0   U  0  0    0 eth0
```

Es sind nur noch die Standardeinträge vorhanden, die mit jenen Netzwerken in Verbindung stehen, in denen der Computer selbst vertreten ist. Es fehlt das Standard-Gateway, das vor allem für die Kommunikation mit dem Internet erforderlich ist. Außerdem gibt es noch zwei interne Netze, die dieser Host über einen anderen Router erreichen können muss.

Achtung

Sie müssen die folgenden Kommandos Ihrer Umgebung anpassen, damit Sie die Aufgaben nachvollziehen können. Sie können sich zwar Netzwerke als Routing-Ziele frei ausdenken, aber die IP-Adresse eines angegebenen Gateways muss für den Computer zumindest theoretisch erreichbar sein, damit es nicht zu einer Fehlermeldung kommt.

[!]

Die fehlenden Einträge werden mit den folgenden Befehlen ergänzt:

```
root@ipcop:~ # route add default gw 20.14.5.65
```

Das Standard-Gateway zum Internet steht somit fest. Fehlen nur noch die Einträge, die den Weg in die beiden privaten Netze aufzeigen:

```
# route add -net 172.16.0.0 netmask 255.255.0.0 gw 192.168.0.10
# route add -net 172.20.0.0 netmask 255.255.0.0 gw 192.168.0.10
```

Sie können bei aktuellen Betriebssystemen auch die *CIDR-Schreibweise* (Classless Inter-Domain Routing) der Netzwerkmaske verwenden, wenn Sie Routing-Einträge hinzufügen. Die Kommandos würden dann stattdessen so aussehen:

```
# route add -net 172.16.0.0/16 gw 192.168.0.10
# route add -net 172.20.0.0/16 gw 192.168.0.10
```

Das ist eine erhebliche Arbeitserleichterung, wenn man die Erstkonfiguration eines Routers vornimmt, der von vielen Netzwerken aus erreichbar sein muss.

Die daraus resultierende Routing-Tabelle sieht dann so aus:

```
root@ipcop:~ # route
Kernel IP routing table
Destination   Gateway      Genmask         F. M. Ref. Use If
20.14.5.64    *            255.255.255.192 U  0  0    0 eth1
192.168.0.0   *            255.255.255.0   U  0  0    0 eth0
172.16.0.0    192.168.0.10 255.255.0.0     U  0  0    0 eth0
172.20.0.0    192.168.0.10 255.255.0.0     U  0  0    0 eth0
default       20.14.5.65   0.0.0.0         UG 0  0    0 eth1
```

An dieser Tabelle können Sie leicht erkennen, wie das Routing abläuft. In den ersten beiden Netzen 20.14.5.64 und 192.168.0.0 ist der Computer selbst vertreten. Der Stern (*) als Gateway-Eintrag weist darauf hin, dass der Rechner für die Auslieferung der Pakete in diese Netze selbst zuständig ist. Die beiden privaten Netze 172.16.0.0/16 und 172.20.0.0/16 erreicht er über das Gateway mit der IP-Adresse 192.168.0.10. Alle Pakete, die nicht aufgrund anderer Routing-Einträge zuzuordnen sind, schickt der Computer an das Standard-Gateway. Dieses ist mit `default` bezeichnet und hat die IP-Adresse 20.14.5.65.

Wenn Sie Routing-Einträge für IPv6 festlegen wollen, müssen Sie die Option `-A inet6` verwenden. Das entsprechende Kommando könnte dann etwa so aussehen:

```
# route -A inet6 add 2001:6f8:1ce:1::/64 gw 2001:6f8:1ce:0::5
```

Bei der Überprüfung der IPv6-Routing-Einträge benötigen Sie ebenfalls die Option `-A inet6`:

```
root@chrouter1:~# route -A inet6 | head -4
Kernel-IPv6-Routentabelle
```

```
Destination           Next Hop         Flag Met Ref Use  If
2001:6f8:1ce::/64     ::               U    256 0   1    eth1
2001:6f8:1ce:1::/64   2001:6f8:1ce::5  UG   1   0   1369 eth1
```

Sollten Sie einen Computer konfigurieren, der direkt über ein DSL-Modem an das Internet angebunden ist, werden zusätzlich Routen dynamisch generiert. Sie können diese sich im Cache des Kernels befindenden Routing-Einträge einsehen:

```
root@chrouter1:~# route -C
Kernel-IP-Routencache
Ziel            Genmask         Flags Metric Ref    Ben Iface
212.224.0.189   212.224.0.189         0      0       12 ppp0
time.nist.gov   time.nist.gov   i     0      0        3 ppp0
ptbtime1.ptb.de ptbtime1.ptb.de i     0      0        0 ppp0
europium.canoni europium.canoni       0      0        1 ppp0
europium.canoni europium.canoni       0      0        1 ppp0
```

arp

Das Kommando `arp` kann dazu verwendet werden, den ARP-Cache eines Computers einzusehen oder einzelne bzw. alle Einträge aus dem ARP-Cache zu löschen. Wenn Sie lediglich den Cache einsehen wollen, verwenden Sie das Kommando einfach ohne Optionen oder Parameter:

```
root@chrouter1:~# arp
Adresse         Hardware-Typ Hardware-Adresse Optionen  Maske  Schnittstelle
192.168.0.244   ether        00:15:5d:00:fe:49          C      eth1
192.168.0.98    ether        c4:7d:4f:a9:7b:10          C      eth1
192.168.0.90    ether        00:13:10:d7:f3:84          C      eth1
192.168.0.99    ether        00:22:6b:19:cb:2a          C      eth1
192.168.0.100   ether        00:13:46:fc:9e:00          C      eth1
192.168.0.31    ether        00:15:5d:00:fe:3a          C      eth1
```

Würde jetzt bei einem der aufgelisteten Systeme die IP-Adresse geändert, dann wäre die Zuordnung von IP-Adresse zu MAC-Adresse im ARP-Cache dieses Computers vorübergehend falsch. Sie können dann den Cacheeintrag für das betroffene System löschen, indem Sie den Schalter `-d` verwenden:

```
root@chrouter1:~# arp -d 192.168.0.104
```

Sie müssen dies für die Prüfung wissen, auch wenn man in der Praxis davon ausgehen kann, dass sich der ARP-Cache schon von selbst aktualisiert hat, bevor Sie überhaupt einen Fehler bemerken.

ip

Das Kommando `ip` beinhaltet verschiedene Optionen zur Überprüfung, aber auch zur Konfiguration diverser Netzwerkeinstellungen. Es deckt die Möglichkeiten aller vorangehenden Tools ab und weist auch noch weitere Funktionen auf. Es handelt sich praktisch um das »Schweizer Messer« unter den Netzwerkprogrammen. Einige Funktionalitäten sind gegenüber den spezialisierten Tools modernisiert. So zeigt das Kommando `ip` schon per Voreinstellung Routing-Tabellen in CIDR-Notation an:

```
root@chrouter1:~# ip route show
192.168.7.0/26 via 192.168.0.98 dev eth1
192.168.3.0/26 via 192.168.0.10 dev eth1
192.168.8.0/26 via 192.168.0.98 dev eth1
192.168.4.0/26 via 192.168.0.11 dev eth1
```

Die Ausgabe ist hierbei auf das Wesentliche beschränkt und deshalb leichter ablesbar. So sagt zum Beispiel der erste Eintrag dieser Routing-Tabelle aus, dass Pakete, die für das Netzwerk 192.168.7.0/26 bestimmt sind, über das Gateway mit der Adresse 192.168.0.98 zugestellt werden müssen. Im direkten Vergleich sieht dieselbe Routing-Tabelle, wenn man sie mit dem `route`-Kommando ausgibt, so aus:

```
root@chrouter1:~# route -n
Kernel-IP-Routentabelle
Ziel         Router       Genmask         Flags Metric Ref  Use  Iface
192.168.7.0  192.168.0.98 255.255.255.192 UG    0      0    0    eth1
192.168.3.0  192.168.0.10 255.255.255.192 UG    0      0    0    eth1
192.168.8.0  192.168.0.98 255.255.255.192 UG    0      0    0    eth1
192.168.4.0  192.168.0.11 255.255.255.192 UG    0      0    0    eth1
```

Auch für das Kommando `arp` bietet `ip` eine Entsprechung. Da die Kommunikation mittels MAC-Adressen nur innerhalb eines Netzwerksegments stattfindet, kann man sagen, dass es sich bei Computern, die sich auf dieser Ebene unterhalten, um Nachbarn handelt. Dementsprechend fällt auch die Syntax für das Kommando aus, das die Zuordnung von IP-Adressen zu MAC-Adressen anzeigt:

```
root@chrouter1:~# ip neighbour show
fe80::20d:60ff:feae:de4 dev eth1 lladdr 00:0d:60:ae:0d:e4
fe80::a58c:f3f7:59ad:bd27 dev eth1 lladdr 00:30:48:f9:8e:7c
2001:6f8:1cfe::4 dev eth1 lladdr 00:04:75:82:69:28
192.168.0.101 dev eth1 lladdr 00:25:9c:75:ec:bd
192.168.0.244 dev eth1 lladdr 00:15:5d:00:fe:49
```

Beachten Sie bitte, dass `ip` im Gegensatz zu `arp` auch die Zuordnung zu den IPv6-Adressen anzeigt.

Selbstverständlich gibt es auch eine Entsprechung für das Werkzeug `ifconfig`. Sie können sich die Parameter einer Schnittstelle ansehen, indem Sie folgendes Kommando verwenden:

```
root@archangel:~# ip addr show eth1
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1280 qdisc pfifo_
fast state UNKNOWN qlen 1000
    link/ether 00:e0:4c:39:03:91 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.1/24 brd 192.168.50.255 scope global eth1
    inet6 2a01:198:5dd::1/64 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::2e0:4cff:fe39:391/64 scope link
       valid_lft forever preferred_lft forever
```

Auch hier sieht man, dass die Netzwerkmaske ausschließlich in CIDR-Notation und nicht mehr in Dotted-Quad-Schreibweise dargestellt wird.

Eine weitere Komponente, die mit dem Kommando `ip` überprüft bzw. auch konfiguriert werden kann, sind Tunnel, bei denen IPv6-Pakete in IPv4-Paketen gekapselt werden:

```
root@archangel:~# ip tunnel show
sit0: ipv6/ip remote any local any ttl 64 nopmtudisc
sixxs: ipv6/ip remote 91.184.37.98 local 88.73.9.183 ttl 64
```

Wie Sie sehen, ist der Aufbau der Kommandos zur Überprüfung von Netzwerkkomponenten immer gleich, unabhängig davon, ob Sie IP-Adressen, Routing-Einträge oder den ARP-Cache eines Computers diagnostizieren müssen. Sie können übrigens mit dem Befehl `ip monitor` auch live überwachen, was sich bezüglich der Zuordnung von MAC-Adressen zu IP-Adressen in einem Netzwerksegment gerade tut:

```
root@archangel:~# ip monitor
192.168.50.100 dev eth1 lladdr 00:1d:7e:a9:c6:17 STALE
fe80::2e0:4cff:fe39:391 dev eth1 lladdr 00:e0:4c:39:03:91 router STALE
fe80::2e0:4cff:fe39:391 dev eth1 lladdr 00:e0:4c:39:03:91 router STALE
192.168.50.100 dev eth1 lladdr 00:1d:7e:a9:c6:17 REACHABLE
```

Wie mit diesem Kommando Konfigurationsänderungen vorgenommen werden, können Sie auf den nächsten Seiten nachlesen.

iwconfig

Das Werkzeug `iwconfig` wird verwendet, um Drahtlosnetzwerk-Einstellungen zu konfigurieren. Der Unterschied zu `ifconfig` ist der, dass Drahtlosnetzwerke Parameter benötigen, die es in drahtgebundenen Netzwerken nicht gibt. Das wären vor allem

SSID, Sendeleistung, Frequenz, Übertragungskanal, Schlüssel und Empfangsempfindlichkeit. Diese Optionen können Sie zusätzlich zur IP-Konfiguration mit `iwconfig` festlegen. Die Abfrage der aktuellen Einstellungen eines WLAN-Adapters können Sie mit folgendem Kommando durchführen:

```
root@arch-deb-book:/# iwconfig wlan0
wlan0     IEEE 802.11abg  ESSID:"arch-net"
  Mode:Managed  Frequency:2.427 GHz  Access Point: 00:21:29:EB:66:35
  Bit Rate=1 Mb/s   Tx-Power=15 dBm
  Retry  long limit:7   RTS thr:off   Fragment thr:off
  Encryption key:off
  Power Management:off
  Link Quality=40/70  Signal level=-70 dBm
  Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
  Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Die Sendeleistung der Karte können Sie manuell festlegen, wenn es nötig sein sollte. Geben Sie dazu z. B. folgendes Kommando ein:

```
root@arch-deb-book:/home/harald# iwconfig wlan0 txpower 30mW
```

Die meisten Notebooks verfügen über einen Schalter, mit dem Sie die Drahtlosnetzwerkverbindung deaktivieren können. Ein Abschalten kann erforderlich sein, wenn man Angriffe befürchtet oder einfach nur, um Akkukapazitäten einzusparen. Ein WLAN-Adapter verbraucht nämlich verhältnismäßig viel Strom. Wenn Sie den Sender der Netzwerkkarte ausschalten müssen, aber keinen entsprechenden Schalter am Notebook finden, können Sie dieses Kommando verwenden:

```
root@arch-deb-book:/home/harald# iwconfig wlan0 txpower off
```

Um den Sender wieder einzuschalten, verwenden Sie entsprechend:

```
root@arch-deb-book:/home/harald# iwconfig wlan0 txpower on
```

iwlist

Mit `iwlist` können Sie detaillierte Informationen über Ihre Drahtlosnetzwerkkarte, Access Points in deren Umgebung oder auch benachbarte Ad-hoc-Netzwerke erhalten. Geben Sie dazu das Kommando `iwlist`, gefolgt vom Namen der Drahtlosnetzwerkkarte und zum Schluss den abzufragenden Parameter an. Im folgenden Beispiel werden z. B. die verwendbaren Frequenzen abgefragt:

```
root@arch-deb-book:/# iwlist wlan0 frequency
wlan0     14 channels in total; available frequencies :
          Channel 01 : 2.412 GHz
```

```
Channel 02 : 2.417 GHz
Channel 03 : 2.422 GHz
Channel 04 : 2.427 GHz
Channel 05 : 2.432 GHz
Channel 06 : 2.437 GHz
Channel 07 : 2.442 GHz
Channel 08 : 2.447 GHz
Channel 09 : 2.452 GHz
Channel 10 : 2.457 GHz
Channel 11 : 2.462 GHz
Channel 12 : 2.467 GHz
Channel 13 : 2.472 GHz
Channel 14 : 2.484 GHz
Current Frequency=2.427 GHz (Channel 4)
```

Das Programm versteht viele Optionen. Wenn Ihnen WLAN zur Verfügung steht, sollten Sie ein wenig damit experimentieren. Der Zweck der meisten im Folgenden genannten Parameter ist selbsterklärend:

- `scanning`
- `frequency`
- `rate`
- `keys`
- `power`
- `txpower`
- `retry`
- `event`
- `auth`
- `wpakeys`
- `genie`
- `modulation`

iw

Ähnlich wie das Programm `ip` die Werkzeuge `ifconfig` und `route` ablösen soll, ist `iw` als Neuerung zu `iwconfig` und `iwlist` gedacht. Es gibt zu den alten Kommandos jeweils Entsprechungen, z. B.:

- `iw dev wlan0 link` entspricht `iwconfig wlan0`
- `iw wlan0 connect meine-ssid` entspricht `iwconfig wlan0 essid meine-ssid`

Es ist hilfreich, die Manpages der Programme einander gegenüberzustellen.

205.2 Fortgeschrittene Netzwerkkonfiguration

Wichtung: 4

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, ein Netzwerkgerät so zu konfigurieren, dass dieses mit verschiedenen Netzwerk-Authentifizierungsschemata betrieben werden kann. Dieses Lernziel umfasst auch die Einrichtung eines Netzwerkgerätes mit mehreren IP-Adressen, die Konfiguration eines VPN-Clients und die Behebung von Netzwerkproblemen.

Wichtigste Wissensgebiete:

- Dienstprogramme zur Manipulation von Routing-Tabellen
- Dienstprogramme zur Konfiguration und Manipulation von Ethernet-Schnittstellen
- Dienstprogramme zur Analyse des Netzwerkstatus eines Netzwerkgerätes
- Dienstprogramme zur Überwachung und Analyse des TCP-/IP-Verkehrs

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `route`
- `ifconfig`
- `netstat`
- `ss`
- `ping`, `ping6`
- `arp`
- `tcpdump`
- `lsof`
- `nc`
- `ip`
- `nmap`

Allgemeines

Auf den folgenden Seiten finden Sie ein paar alte Bekannte wieder, was die Konfigurationstools betrifft. Sie finden hier allerdings auch ein paar Optionen und Möglichkeiten für diese Programme, die einem Administrator bei seinen täglichen Aufgaben die Arbeit erleichtern können. Außerdem sollen Sie an die Implementierung des IPv6-Protokolls herangeführt werden.

Werkzeuge und Konfigurationsdateien

ifconfig

Die Grundkonfiguration von Netzwerkadaptern mithilfe des Programms `ifconfig` kennen Sie bereits aus dem vorangehenden Kapitel. Es kann während Ihrer täglichen Arbeit aber auch Situationen geben, in denen es sinnvoll ist, eine weitere IP-Adresse temporär an einen Netzwerkadapter zu binden. Das gilt für IPv4 genauso wie für IPv6. Was die IPv6-Adressen anbelangt, werden diese ohnehin schon in der Grundkonfiguration (zumindest was die Syntax angeht) ähnlich vergeben wie zusätzliche IPv4-Adressen.

IPv4-Adressen hinzufügen

Stellen Sie sich vor, Sie müssen einen neuen Access Point konfigurieren. Im Auslieferungszustand ist die IP-Adresse des Gerätes auf 192.168.0.50 eingestellt. Der Computer, an dem Sie arbeiten, ist mit der IP-Adresse 192.168.50.1 konfiguriert. Da sich beide Geräte in unterschiedlichen Netzwerksegmenten befinden, können Sie den Access Point nicht erreichen, wenn Sie ihn mit dem Netzwerk verbinden. Wenn Sie nun einfach die IP-Adresse Ihres Computers entsprechend ändern, können Sie zwar den Access Point erreichen, verlieren aber die Anbindung zum eigentlichen Netzwerk. Das kann sehr lästig sein, wenn Sie während der Gerätekonfiguration parallel etwas auf der Website des Herstellers nachlesen müssen oder ein Firmware-Update aus dem Internet herunterladen wollen. Die Konfiguration der zusätzlichen IP-Adresse ist ziemlich einfach:

```
root@archangel:/# ifconfig eth1:1 192.168.0.60
```

Es wurde mit diesem Kommando einfach ein weiterer Alias zu der Schnittstelle `eth1` hinzugefügt. Der Alias `eth1:1` wurde gleichzeitig mit der IP-Adresse 192.168.0.60 konfiguriert. Diese IP-Adresse ist so ziemlich willkürlich gewählt. Eine beliebige andere Adresse aus dem Netzwerk 192.168.0.0/24 hätte natürlich ebenfalls zu einem brauchbaren Ergebnis geführt. Es sollte jetzt möglich sein, den Access Point zu konfigurieren. Sie können leicht nachprüfen, ob der neue Schnittstellenalias an die richtige Netzwerkkarte gebunden ist:

```
root@archangel:/# ifconfig|grep ^eth1
eth1      Link encap:Ethernet  Hardware Adresse 00:e0:4c:39:03:91
eth1:1    Link encap:Ethernet Hardware Adresse 00:e0:4c:39:03:91
```

Wie man sieht, wird für beide Aliasse dieselbe MAC-Adresse ausgegeben. Es ist also alles in Ordnung. Wenn die zusätzliche IP-Adresse nicht mehr benötigt wird, können Sie Ihr System schnell in den ursprünglichen Zustand zurückversetzen:

```
root@archangel:/# ifconfig eth1:1 down
root@archangel:/# ifconfig|grep ^eth1
eth1      Link encap:Ethernet  Hardware Adresse 00:e0:4c:39:03:91
```

Wie man sieht, ist der zusätzliche Alias und damit auch die zugehörige IP-Adresse wieder entfernt worden. Sie hätten den Alias auch behalten und nur die IP-Adresse entfernen können, aber das ist in den wenigsten Fällen sinnvoll. Falls Sie das tatsächlich mal verwenden müssen, machen Sie das mit diesem Befehl:

```
root@archangel:/# ifconfig eth1:1 del 192.168.0.60
```

Der Alias bleibt dann erhalten:

```
root@archangel:/# ifconfig eth1:1
eth1:1    Link encap:Ethernet  Hardware Adresse 00:e0:4c:39:03:91
          UP BROADCAST RUNNING MULTICAST  MTU:1280  Metrik:1
          Interrupt:19 Basisadresse:0xdf00
```

Vergessen Sie bitte nicht, dass eine Konfiguration mittels `ifconfig` nur temporär ist. Nach dem Neustart sind diese Einstellungen nicht mehr verfügbar.

IPv6-Adressen hinzufügen bzw. konfigurieren

Wenn Sie mithilfe des Werkzeugs `ifconfig` eine IPv6-Adresse konfigurieren, wird diese dem Adapter hinzugefügt. Bereits vorhandene Adressen werden nicht ersetzt. Das liegt u. a. daran, dass eine Netzwerkschnittstelle in Bezug auf IPv6 sowohl dynamische als auch statische Adressen gleichzeitig handhaben kann. Außerdem muss auch die linklokale IPv6-Adresse, die zur Kommunikation innerhalb desselben Netzwerksegments verwendet werden kann, beibehalten werden. Sie können eine IPv6-Adresse mit folgendem Kommando hinzufügen:

```
root@archangel:/# ifconfig eth1 inet6 add 2a01:198:5dd:50::200/64
```

Die Adresse kann mit folgendem Kommando wieder entfernt werden:

```
root@archangel:/# ifconfig eth1 inet6 del 2a01:198:5dd:50::200/64
```

Da Sie für eine Netzwerkschnittstelle mit Ihrem »normalen« Alias mehrere IPv6-Adressen konfigurieren können, ist das Anlegen eines zusätzlichen Alias, so wie bei IPv4, in der Regel nicht notwendig.

route

Genauso wie für das `ifconfig`-Kommando sollten Sie auch noch ein paar weitere Fähigkeiten des `route`-Programms kennenlernen.

Achtung

Sie müssen die folgenden Kommandos Ihrer Umgebung anpassen, damit Sie die Aufgaben nachvollziehen können. Sie können sich zwar Netzwerke als Routing-Ziele frei ausdenken, aber die IP-Adresse eines angegebenen Gateways muss für den Computer zumindest theoretisch erreichbar sein, damit es nicht zu einer Fehlermeldung kommt.

Die neueren Versionen von `route` unterstützen z. B. die Verwendung der CIDR-Notation (Classless Inter-Domain Routing). Sie können also z. B. beim Hinzufügen einer neuen Route zu einer Routing-Tabelle anstatt `netmask 255.255.0.0` die Notation `/16` verwenden:

```
[root@r1 /]# route add -net 172.16.0.0/16 gw 192.168.0.98
```

Wenn Sie einen neuen Router konfigurieren, der für viele Netzwerke zuständig ist, erspart das eine Menge Tipparbeit.

Manchmal ist es notwendig, eine Route zu einem einzigen Host anstatt zu einem Netzwerk anzulegen (Hostroute). Das könnte z. B. dann der Fall sein, wenn in einem abgesicherten Netzwerksegment nur ein einziger Computer von anderen Netzwerken aus erreichbar sein muss. Verwenden Sie hierzu folgendes Kommando:

```
[root@r1 /]# route add -host 192.168.1.7 gw 192.168.0.4
```

Beachten Sie, dass die Angabe einer Netzwerkmaske in diesem Falle nicht nötig ist, weil es sich um einen einzelnen Host handelt. Sie können, wenn Sie unbedingt wollen, eine Netzwerkmaske mit dem Wert `/32` angeben. Eine andere Netzwerkmaske hätte eine Fehlermeldung zur Folge.

Wenn Sie einen Routing-Eintrag aus einer Routing-Tabelle löschen müssen, verwenden Sie die Option `del`, unabhängig davon, ob es sich bei dem Routing-Ziel um eine Hostroute oder eine Route zu einem Netzwerk handelt. Die Syntax ist ansonsten weitestgehend identisch mit der Syntax zum Hinzufügen einer Route. Um die Routing-Tabelleneinträge der beiden vorangehenden Beispiele wieder zu löschen, verwenden Sie folgende Kommandos:

```
[root@r1 /]# route del -net 172.16.0.0/16 gw 192.168.0.98
[root@r1 /]# route del -host 192.168.1.7 gw 192.168.0.4
```

Wenn Sie Routing-Tabelleneinträge für IPv6-Netzwerke konfigurieren, ist die Syntax fast genauso wie bei IPv4-Routen zu verwenden. Sie müssen lediglich die Option `-A` mit dem Parameter `inet6` angeben. Es entfällt die Option `-net` gegenüber IPv4, weil die Kombination aus IP-Adresse und Netzwerkmaske automatisch impliziert, ob es sich bei dem Ziel um einen Host oder um ein Netzwerk handelt. Das Kommando zum Hinzufügen einer neuen Route zu einer Routing-Tabelle könnte also z. B. so aussehen:

```
[root@r1 /]# route -A inet6 add 2001:6f8:1cfe:1::/64 gw 2001:6f8:1cfe:0::4
```

Eine Hostroute würde, wie schon gesagt, durch die Netzwerkmaske (`/128`) automatisch erkannt:

```
root@r1:~# route -A inet6 add 2001:6f8:1cfe:7::47/128 gw 2001:6f8:1cfe:0::47
```

Das Löschen der Route entspricht wiederum eins zu eins dem Hinzufügen einer Route, nur dass Sie diesmal die Option `del` verwenden müssen. Um die Routing-Einträge aus den Übungsbeispielen zu löschen, verwenden Sie also:

```
[root@r1 /]# route -A inet6 del 2001:6f8:1cfe:1::/64 gw 2001:6f8:1cfe:0::4
root@r1:~# route -A inet6 del 2001:6f8:1cfe:7::47/128 gw 2001:6f8:1cfe:0::47
```

netstat

Eine der wichtigsten Aufgaben von `netstat` ist wohl die Anzeige von bestehenden Netzwerkverbindungen. So können Sie z. B. bei Kommunikationsproblemen, bei denen ein Client-Programm keine brauchbare Fehlermeldung ausgibt, auf dem entsprechenden Server sehen, ob überhaupt eine Netzwerkverbindung zwischen den Computern besteht. Weitere Anzeigemöglichkeiten von `netstat` sind z. B. Routing-Tabellen oder Statistiken der Netzwerkschnittstellen. Einige wichtige Optionen von `netstat` sind:

- `-n` sorgt für eine rein numerische Anzeige, beschleunigt also die Ausgabe, weil keine Namensauflösung stattfinden muss.
- `-l` bewirkt, dass lediglich Sockets, die sich im Status `LISTEN` befinden, angezeigt werden.
- `-p` sorgt zusätzlich für die Anzeige der Prozesse, die einen Socket geöffnet halten, und die Anzeige der zugehörigen PIDs.
- `-r` zeigt die Routing-Tabelle an.
- `-i` zeigt eine Liste der Netzwerkschnittstellen (Interfaces).
- `-a` zeigt alle Verbindungen an, und nicht nur die, an denen der Host selbst lauscht.
- `-c` sorgt für eine fortlaufende Aktualisierung der Anzeige. Diese Funktion kann mit Strg + C abgebrochen werden.

Da `netstat` nicht nur Netzwerkverbindungen im Sinne von TCP/IP anzeigt, sondern z. B. auch die systeminterne Kommunikation über Unix-Sockets, wird die Ausgabe des Programms sehr umfangreich und dadurch auch unübersichtlich. Sie sollten deshalb eine Filterung über das Programm `grep` durchführen, damit Sie nur die Informationen erhalten, die Sie wirklich benötigen. Im folgenden Beispiel wird überprüft, welche Dienste ein Server über das Protokoll IPv6 anbietet:

```
root@archangel:/# netstat -an | grep tcp6
tcp6   0  0 :::22                   :::*      LISTEN
tcp6   0  0 2a01:198:5dd::1:139     :::*      LISTEN
tcp6   0  0 fe80::2e0:4cff:fe39:139 :::*      LISTEN
tcp6   0  0 :::5900                 :::*      LISTEN
tcp6   0  0 :::80                   :::*      LISTEN
tcp6   0  0 :::53                   :::*      LISTEN
tcp6   0  0 :::25                   :::*      LISTEN
tcp6   0  0 :::631                  :::*      LISTEN
tcp6   0  0 ::1:953                 :::*      LISTEN
tcp6   0  0 2a01:198:5dd::1:445     :::*      LISTEN
tcp6   0  0 fe80::2e0:4cff:fe39:445 :::*      LISTEN
```

Man kann der Ausgabe des Kommandos entnehmen, dass der Server folgende Dienste über IPv6 anbietet: ssh-Zugriff (Port 22), Dateidienste für Windows-Clients über Samba (Ports 139 und 445), VNC (Port 5900), Webserver (Port 80), DNS-Server (Port 53), SMTP (Port 25), CUPS-Druckdienste (Port 631) und rndc (Port 953).

Prüfungstipp

Sie sollten u. a. die Ports, die dieser Server abhört, kennen. Das ist Grundwissen für einen Administrator und kann in der Prüfung natürlich jederzeit abgefragt werden.

ss

Der Verwendungszweck des Kommandos `ss` ähnelt dem von `netstat`. Bei diesem neuen Werkzeug zur Anzeige von Sockets wurden etliche Optionen von `netstat` übernommen, was den Umstieg erheblich erleichtert. Da Sie sich gerade in `netstat` eingearbeitet haben, empfehle ich Ihnen einfach ein wenig mit `ss` herumzuexperimentieren. Die Optionen `-i`, `-a`, `-n`, `-p` und `-l` sind hierfür eine gute Basis.

ping, ping6

Mit dem Kommando `ping` senden Sie ICMP-Anfragen (Echo-Requests) an einen Computer. In den meisten Fällen will man auf diese Art lediglich feststellen, ob der entsprechende Computer noch reagiert. Zusätzlich kann man aus der Antwort aber auch Rückschlüsse auf die Antwortzeit und damit die Anbindung eines Zielhosts ziehen.

```
root@arch-deb-book:/# ping 24.215.7.162
PING 24.215.7.162 (24.215.7.162) 56(84) bytes of data.
64 bytes from 24.215.7.162: icmp_req=1 ttl=56 time=137 ms
64 bytes from 24.215.7.162: icmp_req=2 ttl=56 time=138 ms
```

Der ICMP-Antwort kann man entnehmen, dass das anfordernde Paket 64 Bytes Daten enthielt. Dieser Wert ist übrigens nicht genormt. Windows-Rechner senden z. B. üblicherweise 32 Bytes. Die Antworten sind sequenziell durchnummeriert. So lässt sich leicht feststellen, ob vielleicht ein Paket verloren gegangen ist.

Aus der TTL (Time to Live) lässt sich ablesen, wie viele Router das Paket auf seinem Weg durchlaufen hat. Jeder Router, der durchlaufen wird, reduziert die TTL um den Wert 1. Auf diese Art soll ein Router Bounce verhindert werden. Wenn ein Router, der selbst als Standard-Gateway fungiert, falsch konfiguriert ist und deshalb ein Paket an einen sendenden Router irrtümlich zurückgibt, schickt der sendende Router das Paket noch einmal an das Standard-Gateway. Das Paket wird also zwischen den beiden Routern ständig hin und her gesendet. Gäbe es die TTL nicht, würde ein solches Paket für immer zwischen den beiden Routern hin- und herlaufen. Dieses Problem wird also durch die TTL gelöst. Wenn die TTL den Wert 0 erreicht hat, wird das Paket verworfen, und der Absender des Pakets erhält eine ICMP-Fehlermeldung.

Da ein ping-Paket von einem Linux-Computer mit einer TTL von 64 abgesendet wird, lagen in obigem Beispiel offensichtlich acht Router auf dem Weg zur Zieladresse.

Die Laufzeiten von 137 ms bzw. 138 ms sind nicht besonders aussagekräftig. Man müsste die Verbindung schon über einen längeren Zeitraum überwachen, um eine qualitative Bewertung vornehmen zu können.

Wenn Sie eine IPv6-Adresse anpingen wollen, müssen Sie das auf dieses Protokoll zugeschnittene `ping6` verwenden. Es gibt hierbei ansonsten keinen nennenswerten Unterschied zum IPv4-ping. Sehen Sie selbst:

```
root@arch-deb-book:/# ping6 2001:6f8:1d2d::10
PING 2001:6f8:1d2d::10(2001:6f8:1d2d::10) 56 data bytes
64 bytes from 2001:6f8:1d::10: icmp_seq=1 ttl=50 time=149 ms
64 bytes from 2001:6f8:1d::10: icmp_seq=2 ttl=50 time=245 ms
```

tcpdump

Das Programm `tcpdump` gibt den Verkehr von Netzwerkverbindungen auf dem Bildschirm aus. Mithilfe dieses Programms können Sie sehr genau diagnostizieren, wenn Sie bei der Kommunikation einer Netzwerk-Client-Anwendung mit der zugehörigen Serveranwendung ein Problem vermuten. Häufig verwendete Optionen von `tcpdump` sind:

- `-i` legt fest, welche Schnittstelle (Interface) überwacht werden soll.
- `-n` bewirkt eine numerische Anzeige der IP-Adressen und Ports. Dieser Modus sorgt für eine beschleunigte Anzeige, weil keine Reverse Lookups durchgeführt werden müssen.
- `-w` leitet die Ausgabe des Befehls in eine angegebene Datei um (write).
- `-F` liest nicht von einer Netzwerkschnittstelle, sondern aus einer Datei (File).

Da die Ausgabe von `tcpdump` ohnehin recht umfangreich ausfällt, sollten Sie bei der Konstruktion Ihres Kommandos von vornherein eine enge Filterung durch Optionen und Parameter vorsehen. Sie können auch die Aufzeichnung des Verkehrs auf ein bestimmtes Protokoll, einen Port, einen Zielhost oder sogar alles auf einmal beschränken. Im folgenden Beispiel soll nur der HTTP-bezogene Datenverkehr in der Schnittstelle `eth1` aufgezeichnet werden:

```
root@archangel:/# tcpdump tcp port 80 -i eth1
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth1, link-type EN10MB (Ethernet), capture size 96 bytes
19:42:07.258622 IP6 2a01:198:5dd:0:a00:27ff:fef7:f3cb.40364 > 2a01:198:5dd::1.
www: Flags [S], seq 744820740, win 5760, options
[mss 1440,sackOK,TS val 160826194 ecr 0,[|tcp]>
19:42:07.258653 IP6 2a01:198:5dd::1.www > 2a01:198:5dd:0:a00:27ff:fef7:f3cb.40
364: Flags [S.], seq 2068209594, ack 744820741, win 4832, options
[mss 1220,sackOK,TS val 149320145 ecr 160826194,[|tcp]>
19:42:07.261077 IP6 2a01:198:5dd:0:a00:27ff:fef7:f3cb.40364 > 2a01:198:5dd::1.
www: Flags [.], ack 337, win 214, options
[nop,nop,TS val 160826197 ecr 149320146], length 0
19:42:07.262212 IP6 2a01:198:5dd:0:a00:27ff:fef7:f3cb.40364 > 2a01:198:5dd::1.
www: Flags [F.], seq 228, ack 338, win 214, options
[nop,nop,TS val 160826198 ecr 149320146], length 0
19:42:07.262233 IP6 2a01:198:5dd::1.www > 2a01:198:5dd:0:a00:27ff:fef7:f3cb.40
364: Flags [.], ack 229, win 185, options
[nop,nop,TS val 149320146 ecr 160826198], length 0
^C
5 packets captured
5 packets received by filter
0 packets dropped by kernel
```

Die Aufzeichnung wurde hier sehr frühzeitig durch Betätigen von Strg + C unterbrochen. Die fünf aufgezeichneten Pakete wurden innerhalb von vier Millisekunden gesammelt. Sie können sich also vorstellen, was passiert, wenn Sie auf einem Produktionsserver ungefiltert über einen längeren Zeitraum aufzeichnen.

lsof

Das Programm `lsof` (List Open Files) ist Ihnen bereits ein Begriff. Im Zusammenhang mit Dateisystemen wird es verwendet, um festzustellen, ob noch Dateien durch Benutzer geöffnet sind, z. B. wenn sich ein Datenträger nicht aushängen lässt. Im Netzwerk ist der Verwendungszweck sehr ähnlich. Da aus der Sicht von Linux ohnehin alles eine Datei oder ein Prozess ist, kann man geöffnete Netzwerkverbindungen als geöffnete Dateien betrachten und mit `lsof` auch entsprechend auflisten. Für die-

sen Verwendungszweck gibt es in `lsof` eigene Optionen. Eine gute Kombination dieser Optionen zeigt das Beispiel:

```
root@arch-deb-book:/# lsof -i -n -P
COMMAND    PID    USER    FD  TYPE  DEVICE SIZE/OFF NODE NAME
portmap    1120   daemon  4u  IPv4  4702   0t0      UDP  *:111
portmap    1120   daemon  5u  IPv4  4714   0t0      TCP  *:111
rpc.statd  1132   statd   4w  IPv4  4739   0t0      UDP  *:884
rpc.statd  1132   statd   6u  IPv4  4748   0t0      UDP  *:48020
rpc.statd  1132   statd   7u  IPv4  4751   0t0      TCP  *:34387
sshd       1846   root    3u  IPv4  6078   0t0      TCP  *:22
sshd       1846   root    4u  IPv6  6080   0t0      TCP  *:22
cupsd      1874   root    8u  IPv4  6319   0t0      UDP  *:631
dhclient   2206   root    6u  IPv4  7830   0t0      UDP  *:68
```

Wie Sie sehen, kann man der Ausgabe des Programms entnehmen, welche Dienste unter der Verwendung welches Benutzernamens und welcher Prozess-IDs IP-Sockets geöffnet halten. Die in diesem Beispiel verwendeten Optionen haben die folgenden Auswirkungen:

- `-i` sorgt für eine Filterung, sodass nur IP-Sockets ausgegeben werden. Sie können dieser Option noch Parameter übergeben. Sie können auf diese Art die Filterung auf TCP, UDP oder Adressen beschränken.
- `-n` bewirkt wieder eine numerische Ausgabe von IP-Adressen, falls vorhanden, und verhindert damit die Reverse-Auflösung der Adressen im DNS.
- `-P` verhindert die Auflösung der Portnummern in Servicenamen.

Die Ausgabe des `lsof`-Kommandos im obigen Beispiel ist erheblich gekürzt worden. Weitere einzelne Zeilen demonstrieren die Aussagekraft dieses Programms:

```
icedove-b 11337      harald   40u  IPv4 101352
0t0  TCP 192.168.50.134:38591->80.67.31.47:143 (ESTABLISHED)
```

Man kann hier sehen, dass der User `harald` den E-Mail-Client `icedove` verwendet, um bei einem Server mit der IP-Adresse 80.67.31.47 seine E-Mail abzuholen. Der Zielport 143 verrät, dass es sich um einen IMAP-Server handeln muss. Die IP-Adresse des verwendeten Client-Computers lautet 192.168.50.134. Die nächste Zeile ist nicht weniger informativ:

```
chromium- 11349      harald   62u  IPv4 101971
0t0  TCP 192.168.50.134:57537->192.168.50.1:3128 (ESTABLISHED)
```

Derselbe User surft mit demselben Computer im Internet. Er verwendet den Browser Chromium. Der Zugriff erfolgt über einen Squid-Proxy (der Port 3128 verrät es) mit der IP-Adresse 192.168.50.1.

nc

Ein nettes Werkzeug für diverse Funktionstests ist `netcat` (`nc`). Da die Kommandos `nc` und `netcat` lediglich miteinander verlinkt sind, ist es im Übrigen egal, welches der beiden Kommandos Sie verwenden. Bei `netcat` handelt es sich eigentlich um ein Backend im Netzwerkbereich und man kann es, abgesehen von der Diagnose, für alles Erdenkliche verwenden. Sie können mit diesem Programm Listener (Abhörer) für beliebige Ports auf einem Computer erstellen. Mit einem anderen Computer, der ebenfalls `netcat` ausführt, kann man diesen Port dann ansprechen. Wenn die Verbindung zustande kommt, ist sichergestellt, dass auf der Strecke zwischen Quell- und Zielcomputer keine Firewalls positioniert sind, die den Zugriff auf den verwendeten Port blockieren. Das ist ein sehr einfach durchzuführender Funktionstest. Abgesehen davon können Sie durch die bestehende Verbindung aber auch alles Mögliche hindurchschicken.

Damit Sie diese Funktion ausprobieren können, benötigen Sie zwei Computer. Um eine Kommunikation über den Port 4444 (willkürlich gewählt) zu testen, führen Sie auf dem Computer, der als Server fungieren soll, dieses Kommando aus:

```
root@archangel:/# netcat -l -p 4444
```

Der Cursor springt dann in die nächste Zeile und der Computer wartet nun auf Daten an den TCP-Port 4444.

Führen Sie auf dem anderen Computer den folgenden Befehl aus:

```
root@arch-deb-book:/# netcat archangel 4444
```

Die Verbindung ist nun eingerichtet. Da `netcat` per Default von *stdin* liest und nach *stdout* schreibt, können Sie jetzt auf beiden Terminals Tastatureingaben machen, die dann auf dem jeweils anderen Terminal sichtbar werden.

Wie schon gesagt, können Sie `netcat` aber auch als Netzwerkbackend für Programme verwenden, die selbst nicht netzwerkfähig sind. Ein Anwendungsbeispiel wäre die Sicherung einer kompletten Festplattenpartition mittels `dd` auf einem anderen Computer. Wie Sie wissen, ist `dd` von Haus aus nicht netzwerkfähig. Sie können auf dem Zielcomputer z. B. Folgendes eingeben, damit `dd` das Netzwerk als Eingabedatei verwenden kann:

```
root@archangel:/# netcat -l -p 4444 | dd of=remotedisk.img
```

In den Computer, der die zu sichernde Festplattenpartition enthält, geben Sie ein solches Kommando ein:

```
root@arch-deb-book:/# dd if=/dev/sdb1 | netcat archangel 4444
```

Das Programm `dd` auf `arch-deb-book` liest also die Partition mit der Gerätedatei */dev/sdb1* ein und sendet die Daten via `nc` an den Port `4444` des Computers `archangel`. Der

Datenstrom, der auf `archangel` via `netcat` empfangen wird, wird anschließend mittels `dd` in die Ausgabedatei *remotedisk.img* geschrieben.

Sie können `netcat` hervorragend benutzen, um Vorgänge im Netzwerk besser analysieren zu können. Versuchen Sie doch einmal, mit einem POP3-Server zu kommunizieren, ohne ein E-Mail-Client-Programm zu verwenden. Das folgende Beispiel können Sie an Ihren eigenen E-Mail-Account übertragen. Die durch den Benutzer gemachten Eingaben sind jeweils fett gedruckt:

```
root@archangel:/# nc pop3.mail-provider.de 110
+OK Dovecot ready.
user harald@mail-provider.de
+OK
pass S3cret!
+OK Logged in.
list
+OK 2 messages:
1 18554
2 3744
quit
+OK Logging out.
```

Nach dem Aufruf der Liste hätte man übrigens z. B. mit `retr 1` die erste E-Mail lesen und diese anschließend mit `dele 1` löschen können.

Mit `netcat` können Sie praktisch fast jeden Servertyp, zumindest testweise, ansprechen und sich dann mit seiner Sprache vertraut machen.

ip

Bisher haben Sie das Kommando `ip` nur verwendet, um Informationen aus dem System auszulesen. Sie können `ip` aber auch zur Konfiguration der Netzwerkeinstellungen verwenden. Das betrifft z. B. IPv4- und IPv6-Adressen sowie Routing-Tabelleneinträge und IPv6 in IPv4 Tunnel. Mit diesem Programm können Sie außerdem Einstellungen für Multicast vornehmen und auf Ethernet-Ebene operieren.

Im Gegensatz zu `ifconfig` kann das Programm `ip` mehrere IP-Adressen an eine Schnittstelle binden, ohne einen zusätzlichen Alias zu verwenden. Die Syntax für diese Operation sieht wie folgt aus:

```
[root@arch-fc /]# ip address add 172.16.0.15/16 dev eth0
```

Beachten Sie bitte, dass Ihnen diese zusätzliche IP-Adresse auch nicht angezeigt wird, wenn Sie anschließend `ifconfig eth0` ausführen. Sie müssen zur Anzeige ebenfalls auf `ip` zurückgreifen:

```
[root@arch-fc /]# ip address show dev eth0
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_
fast state UP qlen 1000
 link/ether 08:00:27:f7:f3:cb brd ff:ff:ff:ff:ff:ff
 inet 192.168.50.12/24 brd 192.168.50.255 scope global eth0
 inet 172.16.0.15/16 scope global eth0
 inet6 2a01:198:5dd:50::12/64 scope global
 valid_lft forever preferred_lft forever
```

Wenn Sie die IP-Adresse nur temporär verwenden wollen, können Sie das Kommando `ip` natürlich auch verwenden, um die Adresse wieder zu löschen:

```
[root@arch-fc /]# ip address del 172.16.0.15/16 dev eth0
```

Abgesehen von der Schnittstellenkonfiguration können Sie mittels `ip`-Kommando auch die Routing-Tabelle bearbeiten. Die Syntax ähnelt der Syntax des `route`-Befehls, wie Sie im direkten Vergleich sehen können:

```
root@r1:~# ip route add 172.20.0.0/16 via 192.168.0.98
```

entspricht eins zu eins:

```
root@r1:~# route add -net 172.20.0.0/16 gw 192.168.0.98
```

Wenn Sie die Route wieder entfernen müssen, verwenden Sie entsprechend das Kommando:

```
root@r1:~# ip route del 172.20.0.0/16 via 192.168.0.98
```

Eine ausgezeichnete Eigenschaft des Programms `ip` ist, dass die Syntax bei der Konfiguration der Protokolle IPv4 und IPv6 nicht nur ähnlich, sondern sogar identisch ist. Also wird eine IPv6-Adresse mit diesem Kommando zu einer Schnittstelle hinzugefügt:

```
[root@r4 /]# ip address add 2a01:198:5dd::47/64 dev eth0
```

Wenn Sie eine Routing-Tabelle um einen IPv6-Eintrag erweitern müssen, sieht die Syntax so aus:

```
root@r1:~# ip route add 2001:6f8:1cfe:4f44::/64 via 2001:6f8:1cfe::4
```

Wahrscheinlich können Sie sich schon vorstellen, wie Sie die beiden letzten Beispiele wieder rückgängig machen, um Ihr System wieder in den ursprünglichen Zustand zu versetzen. Falls nicht, verwenden Sie abschließend einfach die beiden folgenden Kommandos:

```
[root@r4 /]# ip address del 2a01:198:5dd::47/64 dev eth0
root@r1:~# ip route del 2001:6f8:1cfe:4f44::/64 via 2001:6f8:1cfe::4
```

nmap

nmap ist ein Werkzeug, mit dem Sie ein Netzwerk untersuchen können. Primär handelt es sich hier um einen Portscanner. Die einfachste Verwendung sieht so aus, dass Sie nmap lediglich einen Zielhost angeben, den es zu untersuchen gilt. Hierbei können Sie entweder IP-Adressen oder auch Hostnamen verwenden. Am folgenden Beispiel wird schnell klar, wozu Sie nmap einsetzen können:

```
root@archangel:/# nmap 79.193.173.223
Starting Nmap 5.00 ( http://nmap.org ) at 2011-05-17 09:37 CEST
Interesting ports on p4FC1ADDF.dip.t-dialin.net (79.193.173.223):
Not shown: 995 closed ports
PORT    STATE    SERVICE
22/tcp  open     ssh
25/tcp  filtered smtp
80/tcp  open     http
110/tcp filtered pop3
443/tcp filtered https
Nmap done: 1 IP address (1 host up) scanned in 36.75 seconds
```

Die Ausgabe des Kommandos lässt einige Rückschlüsse zu:

- Auf dem Zielcomputer laufen die Dienste ssh, smtp, http, pop3 und https.
- Die Dienste ssh und http sind vom Internet aus erreichbar.
- Intern werden die Dienste smtp, pop3 und https angeboten.
- Der Computer reagiert auf ICMP-Echo-Anforderungen (ping).

Die letzte Schlussfolgerung basiert darauf, dass nmap per Default zunächst eine ICMP-Echo-Anforderung sendet. Wenn der Zielhost keine ICMP-Antwort sendet, wird davon ausgegangen, dass der Zielhost nicht läuft. Der Host muss also eine ICMP-Antwort gesendet haben.

Wenn ein Host nicht auf ICMP antwortet, geht nmap davon aus, dass dieser Computer ausgeschaltet ist:

```
root@archangel:/# nmap testrechner.testdomaene.net
Starting Nmap 5.00 ( http://nmap.org ) at 2011-05-17 14:14 CEST
Note: Host seems down. If it is really up, but blocking our ping probes,
try -PN
Nmap done: 1 IP address (0 hosts up) scanned in 3.51 seconds
```

Wenn Sie aber wissen, dass der zu überprüfende Host läuft und mit dem Netzwerk verbunden ist, können Sie mit der Option -P0 (großes P und eine Null) den Portscan erzwingen. Das Kommando sähe also so aus:

```
root@archangel:/# nmap -P0 testrechner.testdomaene.net
```

Sie können auch ein ganzes Netzwerksegment oder sogar noch mehr mit einem einzigen Kommando abscannen. Das ist nötig, wenn Sie in einem Netzwerk, das Sie verwalten, Sicherheitslücken aufspüren wollen. Einen Scan über ein komplettes Klasse-C-Netzwerk können Sie so ausführen:

```
root@archangel:/# nmap -PO 192.168.0.0/24
```

Da die Ausgabe des Kommandos recht umfangreich ausfallen kann, sollten Sie eine Umleitung in eine Textdatei in Erwägung ziehen. Wenn Sie nach bestimmten Diensten auf der Suche sind, können Sie `nmap` die zu den gesuchten Anwendungen passenden Ports mit der Option `-p` übergeben. Es wird dann keine ICMP-Anfrage vorab gesendet. Die Abfrage eines mutmaßlichen Webservers könnte so aussehen:

```
root@arch-deb:/# nmap -p 80,443 www.rheinwerkcomputing.de
Starting Nmap 5.00 ( http://nmap.org ) at 2011-05-17 20:37 CEST
Interesting ports on grobi.rheinwerk-verlag.de (85.88.3.146):
PORT    STATE SERVICE
80/tcp  open  http
443/tcp open  https
Nmap done: 1 IP address (1 host up) scanned in 0.51 seconds
```

Die Ports 80 und 443 sind auf diesem Host offen. Das ist auch nicht verwunderlich, wenn man den Namen (bzw. Alias) des Zielhosts bedenkt.

Wenn Sie die Option `-O` verwenden, bekommen Sie auch Informationen über das Betriebssystem des geprüften Computers:

```
root@arch-deb-book:/# nmap 192.168.50.10 -O | grep ^OS
OS details: Microsoft Windows XP SP2 or SP3, or Windows Server 2003
OS detection performed. Please report any incorrect results at
http://nmap.org/submit/ .
```

Die meisten neueren Linux-Systeme werden sich diese Informationen allerdings aus Sicherheitsgründen nicht mehr entlocken lassen. Lediglich sehr alte Installationen erweisen sich noch als geschwätzig:

```
root@arch-deb-book:/# nmap -PO -O alterSuSE.computer.net
... irrelevante Zeilen entfernt ...
Warning: OSScan results may be unreliable because we could not find at least 1
 open and 1 closed port
Running (JUST GUESSING) : Linux 2.2.X (86%)
Aggressive OS guesses: Linux 2.2.13 (SuSE 6.3) (86%)
No exact OS matches for host (test conditions non-ideal).
... irrelevante Zeilen entfernt ...
```

Obwohl die Bedingungen für den Test ungünstig waren (auf dem Zielsystem läuft eine `ipchains`-Firewall), ist das Betriebssystem exakt ermittelt worden. Wenn man allerdings bedenkt, dass dieser Scan in der Jahresmitte 2011 lief und das Zielsystem schon im Dezember 1999 aufgesetzt worden ist, erscheint die Situation in einem völlig anderen Licht.

205.3 Kernpunkte der Fehlerbehebung in Netzwerken

Wichtung: 4

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, häufig auftretende Probleme bei der Konfiguration von Netzwerken zu erkennen und zu beheben. Dazu gehört auch das Wissen, wo sich grundlegende Konfigurationsdateien und Befehle befinden.

Wichtigste Wissensgebiete:

- Speicherorte und Inhalte von Dateien zur Zugriffsbeschränkung
- Dienstprogramme zur Konfiguration und Manipulation von Ethernet-Schnittstellen
- Dienstprogramme zur Verwaltung von Routing-Tabellen
- Dienstprogramme zur Auflistung von Netzwerkzuständen
- Dienstprogramme zur Ermittlung der Netzwerkkonfiguration
- Methoden zur Informationsbeschaffung über erkannte und benutzte Hardware
- Systeminitialisierungsdateien und deren Inhalte (Systemd und SysVinit-Prozess)
- Wissen über den NetworkManager und seinen Einfluss auf die Netzwerkkonfiguration

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `ifconfig`, `ip`
- `route`
- `netstat`, `ss`
- */etc/network* bzw. */etc/sysconfig/network-scripts/*
- Systemprotokolldateien wie */var/log/syslog*, */var/log/messages* und das Systemd-Journal
- `ping`, `ping6`
- */etc/resolv.conf*
- */etc/hosts*

- */etc/hostname* oder */etc/HOSTNAME*
- `hostname`
- `traceroute`, `traceroute6`
- `mtr`
- `dmesg`
- */etc/hosts.allow* und */etc/hosts.deny*

Allgemeines

Die Fehlerbehebung in Netzwerken ist eine der Kernaufgaben eines Administrators überhaupt. Da Fehler und Störungen in Netzwerken Ausfallzeiten bedeuten und somit unter Umständen viel Geld kosten, muss ein guter Administrator in einer hektischen Situation logisch und strukturiert an ein Problem herangehen können. Es ist also absolut begründet, dass dieses Thema in der Prüfung hoch gehandelt wird. Viele alltägliche Störungen sind auf grundlegende TCP-/IP-Probleme oder DNS-Fehler zurückzuführen. Die Zusammenstellung der Tools in diesem Topic ist also durchaus als wirklichkeitsgetreu zu betrachten.

Werkzeuge und Konfigurationsdateien

/etc/network

Die Netzwerkkonfiguration eines Debian-Systems und dessen Derivaten finden Sie im Verzeichnis */etc/network*. Dieses Verzeichnis hat normalerweise diesen Inhalt:

```
root@archangel:/etc/network# ls -l
insgesamt 20
drwxr-xr-x 2 root root 4096 2011-05-19 11:05 if-down.d
drwxr-xr-x 2 root root 4096 2011-04-17 14:02 if-post-down.d
drwxr-xr-x 2 root root 4096 2011-04-17 14:02 if-pre-up.d
drwxr-xr-x 2 root root 4096 2011-05-19 11:05 if-up.d
-rw-r--r-- 1 root root  490 2011-01-20 09:43 interfaces
```

Die Unterverzeichnisse *if-down.d*, *if-post-down.d*, *if-pre-up.d* und *if-up.d* enthalten jeweils Skripte, die zu den Zeiten ausgeführt werden, die der jeweilige Verzeichnisname impliziert. Hierbei kommen hauptsächlich Skripte zum Einsatz, die ihrerseits wiederum `init`-Skripte aufrufen, die Netzwerkkomponenten starten. Eigene Skripte, die z. B. der Konfiguration der Firewall oder der Routing-Tabelle dienen, erstellen Sie am besten im Verzeichnis *if-up.d*. Zur Konfiguration von Netzwerkadaptern verwenden Sie die Datei *interfaces*. Die folgende Konfiguration gehört zu einem Router, der ein DSL-Modem verwendet. Zusätzlich ist er mit IPv6 konfiguriert. Es sind also viele wesentliche Einstellungen in dieser Konfiguration enthalten:

```
root@chrouter1:/etc/network# cat interfaces
# Loopbackdevice
auto lo
iface lo inet loopback
# Statische IPv6 Konfiguration der Schnittstelle eth1
# Die MTU wurde aus Performancegründen verringert.
iface eth1 inet6 static
address 2001:6f8:1cfe:0000::1
netmask 64
mtu 1280
# Die statische IPv4 Konfiguration von eth1
iface eth1 inet static
address 192.168.0.30
netmask 255.255.255.0
auto eth1
# eth0 wird für das DSL-Modem verwendet. Die statische
# Konfiguration soll lediglich den Systemstart
# beschleunigen und wird später nicht mehr verwendet.
auto eth0
iface eth0 inet static
address 192.168.77.77
netmask 255.255.255.0
# DSL-Modemeinstellungen
iface ppp0 inet ppp
provider ppp0
auto ppp0
```

In dieser Datei sind keine Schnittstellen enthalten, die ihre Konfiguration von einem DHCP-Server beziehen. Wenn Sie einen DHCP-Client konfigurieren wollen, können Sie der Datei *interfaces* z. B. den folgenden Eintrag hinzufügen:

```
iface eth2 inet dhcp
auto eth2
```

Wenn Sie der Datei *interfaces* eine weitere Schnittstelle hinzugefügt haben (sagen wir `eth2`), können Sie diese Schnittstelle anschließend mit folgendem Kommando aktivieren:

```
root@arch-deb-book:/# ifup eth2
```

Wenn die Schnittstelle vorübergehend deaktiviert werden soll, verwenden Sie diesen Befehl:

```
root@arch-deb-book:/# ifdown eth2
```

/etc/sysconfig/network-scripts/

Computer, auf denen Red Hat oder ein entsprechendes Derivat läuft, erhalten ihre Netzwerkkonfiguration hauptsächlich aus Dateien, die sich im Verzeichnis */etc/sysconfig/network-scripts/* befinden. Bei den meisten Dateien in diesem Verzeichnis handelt es sich um ausführbare Skripte, die Netzwerkschnittstellen starten oder beenden können. So lassen sich unter den wenigen nicht ausführbaren Dateien die Konfigurationsdateien leicht ausmachen. In der Regel haben diese Dateien Namen, wie z. B. *ifcfg-eth0* oder Ähnliches. Es gibt jedenfalls für jede Netzwerkschnittstelle eine eigene Konfigurationsdatei. Für die Schnittstelle `eth0` könnte eine solche Datei so aufgebaut sein:

```
TYPE=Ethernet
BOOTPROTO=none
IPADDR=192.168.50.12
PREFIX=24
GATEWAY=192.168.50.1
DNS1=192.168.50.1
DOMAIN=homelinux.net
DEFROUTE=yes
IPV4_FAILURE_FATAL=yes
IPV6INIT=yes
NAME="Auto eth0"
UUID=2e59b52b-f335-4525-ae08-a24578d56128
ONBOOT=yes
HWADDR=08:00:27:F7:F3:CB
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=yes
```

Die meisten Einstellungen in dieser Datei erklären sich von selbst. Beachten Sie aber die Besonderheit gegenüber der Einstellung bei Debian-Systemen, dass hier der Alias der Netzwerkschnittstelle (eth0) an einen UUID und an die MAC-Adresse des Adapters gebunden ist. Diese Zuordnung wird zusätzlich auch durch `udev` getroffen, sodass Sie beim Austausch einer defekten Netzwerkkarte einerseits die Datei *70-persistent-net.rules* und andererseits die oben abgedruckte Konfigurationsdatei bearbeiten müssen.

Nach den Konfigurationsänderungen können Sie die Netzwerkschnittstellen genauso wie bei Debian-Systemen mit dem Skript `ifup` aktivieren bzw. mit `ifdown` deaktivieren.

/etc/systemd/network/*.network

Bei Systemen, in welchen *Systemd* im Einsatz ist, finden Sie die Konfigurationsdateien für das Netzwerk unter */etc/systemd/network*. Netzwerkschnittstellen werden in Dateien mit der Erweiterung **.network* konfiguriert. Die Konfiguration mit einer statischen IP-Adresse könnte für einen kabelgebundenen Netzwerkadapter z. B. so aussehen:

```
[Match]
Name=enp1s0
[Network]
Address=192.168.50.70/24
Gateway=192.168.50.1
```

Für eine drahtlose mit DHCP konfigurierte Schnittstelle könnten Sie z. B. eine Datei mit der Bezeichnung *wlan.network* erstellen und einen solchen Eintrag vornehmen:

```
[Match]
Name=wlp2s0
[Network]
DHCP=ipv4
```

Nach einer Änderung an der Konfiguration muss `systemd-networkd.service` neu gestartet werden:

```
[root@arch-book /]# systemctl restart systemd-networkd.service
```

/etc/resolv.conf

Die Datei *resolv.conf* ist die Hauptkonfigurationsdatei für den DNS-Client. Sie enthält in erster Linie die IP-Adressen der zu verwendenden DNS-Server. Es gibt aber auch noch andere Eintragstypen, von denen zwei besonders wichtig und auch prüfungsrelevant sind.

Doch zur Erklärung zunächst ein Beispiel:

```
archangel:~ # cat /etc/resolv.conf
domain homelinux.net
search rheinwerkcomputing.de
search rheinwerk-verlag.de
nameserver 192.168.0.1
nameserver 192.168.0.2
```

- Die beiden Einträge vom Typ `nameserver` teilen dem DNS-Client mit, welche DNS-Server zu verwenden sind. Beachten Sie bitte unbedingt, dass hier keine Gleichheitszeichen oder Doppelpunkte zum Einsatz kommen.

- Der Eintrag vom Typ `domain` sagt dem Rechner, in welcher Domäne er selbst Mitglied ist. Das ist vor allem dazu nützlich, dass ein Computer, der sich in derselben Domäne befindet, einfach über seinen Hostnamen angesprochen werden kann. Der Suffix (hier `homelinux.net`) wird dann automatisch ergänzt.
- Etwas seltener wird der Eintrag `search` verwendet. Man braucht diesen Eintragstyp, wenn Computer unterschiedlicher Domänen untereinander über die kurzen Hostnamen erreichbar sein sollen. Die Verwendung von FQDNs ist dann nicht mehr notwendig.

Hinweis

Beachten Sie bitte, dass die Datei *resolv.conf* bei aktuellen Systemen nicht mehr von Hand bearbeitet werden sollte, weil sie ohnehin durch den Inhalt anderer Konfigurationsdateien regelmäßig überschrieben wird.

Eine weitere wichtige Konfigurationsdatei in diesem Zusammenhang ist die Datei */etc/nsswitch.conf*. Mit dieser Datei werden unterschiedliche Systeme in ihrem Verhalten beeinflusst. Das bezieht sich auf unterschiedliche Dienste, die unter anderem der Namensauflösung von Computern dienen (NIS, NIS+, DNS), aber auch auf Mechanismen, die zur Authentifizierung von Benutzern verwendet werden. Beispiel (gekürzt):

```
passwd: files
group:  files
hosts:          files dns
networks:       files dns
```

In diesem Fall verwendet das Authentifizierungsmodul (PAM) Dateien zur Authentifizierung für Benutzer und Gruppen. Das sind konkret die Dateien */etc/passwd* und */etc/group* sowie eventuelle Shadow-Dateien. Bei der Namensauflösung wird zunächst die */etc/hosts*-Datei und erst danach DNS kontaktiert.

/etc/hosts

Diese Datei ist historisch betrachtet ein Vorläufer von DNS. Bevor es überhaupt DNS gab, wurden in dieser Datei Zuordnungen zwischen voll qualifizierten Hostnamen (FQDN = Fully Qualified Domain Name) und IP-Adressen eingetragen und durch den Client verwendet. So war es schon sehr früh möglich, Computer über ihre Namen anstatt über die IP-Adressen anzusprechen. Heutzutage wird diese Datei hauptsächlich verwendet, um den Namen `localhost` mit der Loopback-Adresse zu verknüpfen oder um einen Ersatzeintrag zu schaffen, falls der Rechner die Datei */etc/hostname* nicht verwendet. Ein Beispiel:

```
root@ipcop:/etc # cat /etc/hosts
127.0.0.1       localhost
192.168.0.56    ipcop.homelinux.net     ipcop
```

Der Inhalt der Datei ist wohl absolut selbsterklärend.

/etc/hostname oder /etc/HOSTNAME

Diese beiden Konfigurationsdateien beinhalten den Namen des Computers. Welche der beiden Dateien (wenn überhaupt) verwendet wird sowie deren Inhalt, hängen von der verwendeten Distribution ab.

hostname

Bei den drei Kommandos `hostname`, `domainname` und `dnsdomainname` handelt es sich in Wirklichkeit um ein einziges Programm. Das tatsächliche Programm ist `hostname`. Bei den anderen beiden Dateien handelt es sich nur um Softlinks. Es wird beim Aufruf des Programms geprüft, ob dieses durch einen Link aufgerufen worden ist, und wenn ja, durch welchen. Die unterschiedlichen Aufrufe haben die folgende Wirkung:

- `hostname` zeigt den Namen eines Hosts oder ändert diesen
- `domainname` zeigt oder ändert den NIS-/YP-Domänennamen
- `dnsdomainname` zeigt oder ändert den DNS-Domänennamen eines Hosts

Die Ausgabe der Kommandos unterscheidet sich wieder einmal in Abhängigkeit von der verwendeten Distribution. Der Befehl `domainname` schneidet hierbei am schlechtesten ab. Er ist sowohl bei Debian als auch bei IPCop gar nicht erst vorhanden. Wenn Sie vorhaben, einen Computer umzubenennen, sollten Sie erst die Ausgabe von `hostname` auf Ihrer Distribution testen. Wenn hier ein FQDN zurückgeliefert wird, müssen Sie bei der Umbenennung ebenfalls einen FQDN angeben. Um den Zusammenhang zu zeigen, wird im folgenden Beispiel absichtlich eine Fehlkonfiguration durchgeführt:

```
root@ipcop:/ # hostname
ipcop.homelinux.net
root@ipcop:/ # dnsdomainname
homelinux.net
root@ipcop:/ #
```

`ipcop` gibt also den FQDN aus. Mal sehen, was geschieht, wenn man trotzdem nur den Hostnamen neu setzt:

```
root@ipcop:/ # hostname paketpolizist
root@ipcop:/ # hostname
paketpolizist
```

```
root@ipcop:/ # sh
root@paketpolizist:/ #
```

Nach der Umbenennung zeigt `hostname` also nur noch den reinen Hostnamen ohne Domänensuffix an. Damit der Prompt den Namen auch anzeigt, musste zunächst eine neue Shell geöffnet werden. Aber was ist jetzt aus dem Domänensuffix geworden? Das Ergebnis zeigt das Kommando `dnsdomainname`:

```
root@paketpolizist:/ # dnsdomainname
dnsdomainname: Host name lookup failure
```

Das Suffix ist offensichtlich nicht mehr vorhanden. Gehen Sie also vorsichtig mit diesen Programmen um!

traceroute, traceroute6

Um einen zu langsamen Router auf dem Weg zu einem Ziel zu identifizieren, benötigen Sie das Programm `traceroute`. Dieses Programm sendet zunächst ein UDP-Paket mit einer TTL von nur 1 an das endgültige Ziel. Da der erste Router das Paket verwirft und mit einem Fehler antwortet, ist die Paketlaufzeit zu diesem Router bekannt. Als Nächstes wird wieder ein Paket, diesmal mit einer TTL von 2, losgeschickt. Der zweite Router wird das Paket diesmal verwerfen und ebenfalls mit einer Fehlermeldung antworten. Nun ist auch die Paketlaufzeit zum zweiten Router bekannt. `traceroute` setzt diese Prozedur bis zum Zielhost fort. Der Router, bei dem die längste Antwortzeit ermittelt wird, ist dann als Engpass für das Netzwerk identifiziert:

```
[root@fedora15 ~]# traceroute -n 217.147.216.241
traceroute to 217.147.216.241, 30 hops max, 40 byte packets
 1  192.168.0.1       0.295 ms     0.294 ms     0.165 ms
 2  88.73.0.1         7.059 ms     8.746 ms     9.908 ms
 3  145.254.5.137    11.533 ms    12.850 ms    14.842 ms
 4  145.254.18.89  7234.327 ms  6541.484 ms  8046.301 ms
 5  145.254.16.182   29.524 ms    30.774 ms    32.764 ms
 6  80.81.192.141    58.472 ms    56.592 ms    55.878 ms
 7  195.141.190.118  43.765 ms    42.432 ms    43.378 ms
 8  217.147.223.35   44.350 ms    45.655 ms    45.141 ms
 9  217.147.216.241  45.298 ms    32.745 ms    33.116 ms
```

In diesem Beispiel ist also der vierte Router das Problem. Seine Antwortzeit liegt deutlich außerhalb des akzeptablen Bereichs. Die Option `-n`, die hier bei `traceroute` verwendet worden ist, steht übrigens für »numerical« und unterdrückt die Namensauflösung der einzelnen IP-Adressen. Dadurch wird die Ausgabe von `traceroute` erheblich beschleunigt.

Die Verwendung von `traceroute6` zur Diagnose von IPv6-Routen entspricht der von `traceroute`, wie das folgende Beispiel dokumentiert:

```
root@archangel:/home/harald# traceroute6 -n 2001:6f8:1d2d::10
traceroute to 2001:6f8:1d2d::10 (2001:6f8:1d2d::10), 30 hops max, 80 byte packets
 1  2a01:198:200:860::1  30.305 ms  31.047 ms  32.583 ms
 2  * * *
 3  * * *
 4  2001:2000:3080:6a::1  43.826 ms  45.074 ms  46.411 ms
 5  2001:2000:3018:90::1  51.876 ms  52.924 ms  55.013 ms
 6  2001:668:0:3::2000:ef1  55.565 ms  45.925 ms  35.694 ms
 7  2001:668:0:2::1:3642  47.571 ms  46.238 ms  46.263 ms
 8  2001:668:0:3::5000:152  47.522 ms  48.021 ms  48.571 ms
 9  2001:6f8:1:1:87:86:76:83  58.272 ms  59.994 ms  57.911 ms
10  2001:6f8:1:1:87:86:76:5  58.240 ms  58.392 ms  58.893 ms
11  2001:6f8:1:1:87:86:76:6  59.067 ms  58.200 ms  58.249 ms
12  2001:6f8:1:1:87:86:76:11  58.445 ms  58.440 ms  57.950 ms
13  2001:6f8:1:1:87:86:76:12  58.375 ms  57.936 ms  58.439 ms
14  2001:6f8:1:1:87:86:76:28  50.785 ms  50.791 ms  50.560 ms
15  2001:6f8:1:1:87:86:76:27  50.623 ms  51.519 ms  50.754 ms
16  2001:6f8:1d2d::10  83.668 ms  83.720 ms  83.577 ms
```

mtr

Der Verwendungszweck von `mtr` ist im Prinzip mit dem von `traceroute` und `traceroute6` identisch. Allerdings bietet das Programm einige entscheidende Vorzüge. Es gibt eine reine Konsolenversion, aber auch eine mit grafischer Oberfläche. Interessant ist aber vor allem, dass `mtr` regelmäßig Pakete an den adressierten Zielhost sendet und dabei ständig entsprechende Statistiken ausgibt.

```
                          My traceroute  [v0.85]
archangel (::)                                  Sat Aug 27 17:18:24 2016
Keys:  Help   Display mode   Restart statistics   Order of fields   quit
                                  Packets               Pings
 Host                           Loss%   Snt   Last   Avg  Best  Wrst StDev
 1. 2a01:198:200:860::1          0.0%   215   30.7  32.2  30.3  55.4   3.1
 2. ???
 3. ???
 4. 2001:2000:3080:6a::1         0.5%   215   31.1  31.8  30.4  60.6   3.0
 5. 2001:2000:3018:90::1         0.0%   215   35.3  36.6  34.5  60.9   3.5
 6. 2001:668:0:3::2000:ef1       0.5%   215   38.0  36.4  34.5  68.1   3.8
 7. 2001:668:0:2::1:3642         0.0%   215   48.8  50.2  46.5 110.2   9.0
 8. 2001:668:0:3::5000:152       0.5%   215   47.6  48.1  46.9  57.3   0.9
 9. 2001:6f8:1:1:87:86:76:83     0.0%   215   58.8  60.0  58.3 140.7   5.8
```

```
10. 2001:6f8:1:1:87:86:76:5          0.5%   215   58.6  60.1  58.3 103.9   3.8
11. 2001:6f8:1:1:87:86:76:6          0.0%   215   59.0  60.1  58.3  91.0   3.2
12. 2001:6f8:1:1:87:86:76:11         0.0%   215   58.8  60.7  58.3 174.9   8.3
13. 2001:6f8:1:1:87:86:76:12         0.5%   215   58.5  60.3  58.2 116.3   4.9
14. 2001:6f8:1:1:87:86:76:28         0.5%   214   51.3  53.4  50.4 165.5   8.4
15. 2001:6f8:1:1:87:86:76:27         0.5%   214   52.2  53.3  51.0 110.9   5.1
16. 2001:6f8:1d2d::10                0.0%   214   82.4  86.1  82.4 107.1   3.0
```

Da sich die Aktualisierung der Ausgabe des Kommandos in einem Buch nicht darstellen lässt, sollten Sie das Programm selbst ausprobieren, wenn Sie es noch nicht kennen.

/var/log/syslog, /var/log/messages, /bin/dmesg und journalctl

Zur Diagnose von Netzwerkproblemen können Sie natürlich auch die Systemprotokolle zurate ziehen oder den Kernel-Ringpuffer mittels `dmesg` auslesen. Wenn Sie eine Netzwerkschnittstelle deaktivieren und anschließend wieder aktivieren, wird `dmesg` z. B. so etwas ausgeben:

```
eth1: link down
eth1: link up, 100Mbps, full-duplex, lpa 0x45E1
```

Ob Sie bei der Diagnose eher die Datei */var/log/messages* oder stattdessen */var/log/syslog* auswerten, hängt natürlich von der verwendeten Distribution ab. Bei neueren Systemen können Sie entweder zusätzlich oder ausschließlich `journalctl` zu Rate ziehen.

Sie werden feststellen, dass selbst hardwarenahe Ereignisse, wie z. B. das Entfernen des Netzwerksteckers, aufgezeichnet werden.

ip, ifconfig, route, netstat, ss, ping und ping6

Diese Programme tauchen wiederholt als Prüfungsthemen auf. Da diese Kommandos aber bereits hinlänglich in den vorherigen Kapiteln beschrieben wurden, soll hier auf eine Wiederholung verzichtet werden.

/etc/hosts.allow und /etc/hosts.deny

Die Dateien */etc/hosts.allow* und */etc/hosts.deny* dienen der Konfiguration von *TCP-Wrappern*. Hierbei macht es keinen Unterschied, ob das System mit `inetd` arbeitet und der TCP-Wrapper somit `tcpd` ist oder ob `xinetd` zum Einsatz kommt, welches einen integrierten TCP-Wrapper besitzt. Die Verarbeitungsweise der beiden Konfigurationsdateien ist etwas außergewöhnlich.

- Wenn in der Datei */etc/hosts.allow* eine zutreffende Regel gefunden wird, wird der Zugriff basierend auf dieser Regel erlaubt. Die *hosts.deny*-Datei wird dann nicht mehr verarbeitet.
- Wenn in der Datei */etc/hosts.deny* eine zutreffende Regel gefunden wird, wird der Zugriff, basierend auf dieser Regel, verweigert.
- Wenn keine Einträge in den beiden Dateien zutreffen, wird der Zugriff erteilt.
- Ist eine oder sind beide Dateien nicht vorhanden, wird das gewertet, als wäre(n) die Datei(en) leer.

Es bringt also nichts, über die *hosts.allow*-Datei einem einzelnen Host explizit Zugriff auf einen Dienst zu gewähren, ohne es den anderen Hosts explizit zu verbieten. Eine wasserdichte Grundkonfiguration erreichen Sie, indem Sie in die Datei */etc/hosts.deny* die folgende Zeile schreiben:

```
ALL : ALL
```

Damit ist zunächst allen alles verboten. Sie müssen alle erlaubten Zugriffe explizit in die Datei */etc/hosts.allow* eintragen, z. B.:

```
ALL : LOCAL
```

Dieser Eintrag erlaubt allen lokalen Hosts den Zugriff auf alle Dienste. Als lokal werden alle Computer betrachtet, in deren Name kein Punkt vorkommt. Dieser muss dann folglich Mitglied derselben Domäne wie der Zielhost sein.

Prüfungstipp

Es ist wichtig zu wissen, dass ein Eintrag in der Datei *hosts.allow* Priorität gegenüber einem Eintrag in der Datei *hosts.deny* hat.

NetworkManager

Der NetworkManager ist ein Werkzeug, das insbesondere Anfänger bei der Netzwerkkonfiguration unterstützt. Er ist bei vielen Linux-Distributionen standardmäßig vorhanden, wenn eine grafische Oberfläche installiert worden ist. Insbesondere drahtlose Netzwerkverbindungen sind mithilfe des NetworkManagers leicht zu konfigurieren, obwohl er eine drahtgebundene Verbindung automatisch vorzieht, wenn Sie etwa ein Notebook an ein Netzwerk anschließen und der NetworkManager einen DHCP-Server vorfindet. Inzwischen liegt diese Software, die übrigens ursprünglich von Red Hat stammt, in der Version 0.9.8.2 vor (Juli 2013) und bietet Unterstützung für verschiedenste Netzwerkszenarien, inklusive mobile Breitbandanbindungen.

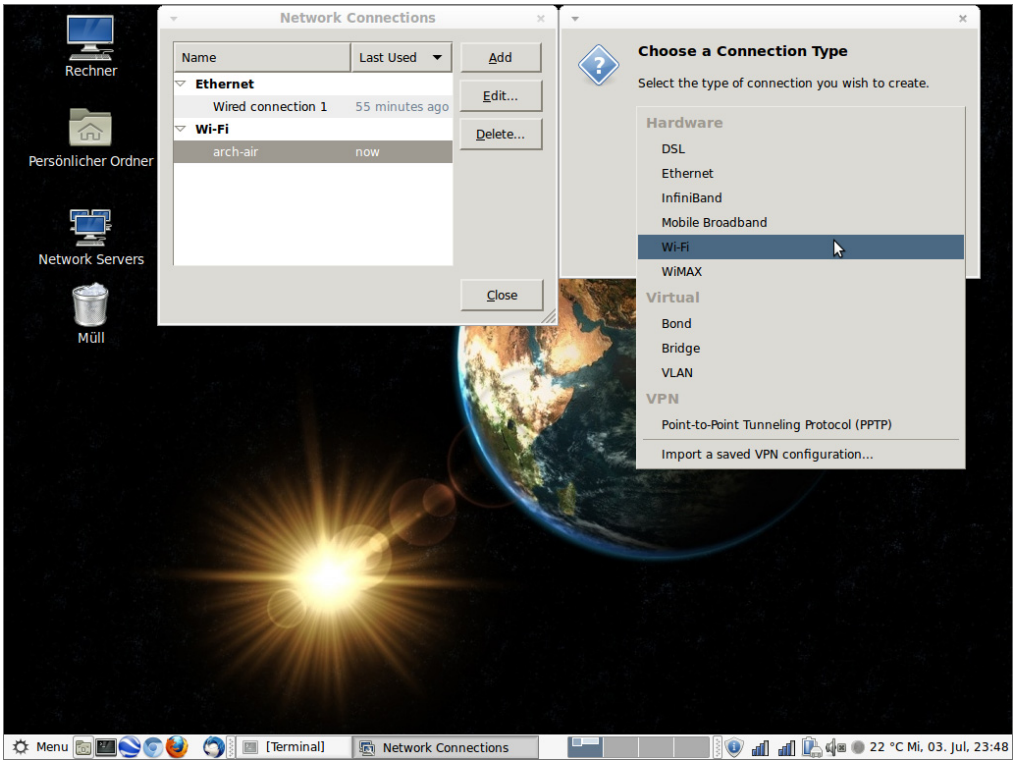

Abbildung 205.1 Mit dem NetworkManager ist es auch Anfängern leicht möglich, Netzwerkverbindungen zu verwalten.

Es gibt auch die Möglichkeit, den NetworkManager über die Kommandozeile zu bedienen. Das entsprechende Programm `nmcli` ist aber nicht als Ersatz für die grafische Konfiguration mit *nm-applet* gedacht. Sie können `nmcli` auf Computern einsetzen, die nicht über eine grafische Oberfläche verfügen. Die eigentliche Konfiguration der Netzwerkschnittstellen würde dann in den Konfigurationsdateien unterhalb von */etc/NetworkManager* erfolgen. Das eigentlich sinnvolle Einsatzgebiet des NetworkManagers ist aber die Konfiguration von Systemen, deren Netzwerkeinstellungen oft geändert werden müssen, also vor allem mobile Geräte.

206 Systemverwaltung und Wartung

Zur Sicherung, Aufbewahrung und auch für den Versand per E-Mail müssen Daten in Archive gepackt und ggf. komprimiert werden. Zum Glück bietet Linux hierfür eine ganze Menge Bordwerkzeuge an.

206.1 Programme aus dem Quellcode übersetzen und installieren

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, ein ausführbares Programm aus dem Quellcode zu übersetzen und zu installieren. Dieses Lernziel beinhaltet die Fähigkeit, eine Datei mit Quellcode zu entpacken.

Wichtigste Wissensgebiete:

- Entpacken von Quellcode mittels üblicher Komprimierungs- und Archivierungsbefehle
- grundlegendes Verständnis der Verwendung von `make`, um Programme zu kompilieren
- ein *configure*-Skript mit Parametern aufrufen
- wissen, wo Quellcode standardmäßig abgelegt wird

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */usr/src/*
- `gunzip`
- `gzip`
- `bzip2`
- `xz`
- `tar`
- `configure`
- `make`
- `uname`
- `install`
- `patch`

Allgemeines

Wenn Sie beabsichtigen, auf einem Computer ein neues Programm zu installieren, haben Sie verschiedene Möglichkeiten, um diese Aufgabe anzugehen. Sie können sich ein fertiges Paket des Programms beschaffen und es mittels `rpm`, `dpkg`, `pacman`, `zypper` oder was auch immer Ihre Distribution inklusive der dazu gehörenden Frontends anbietet installieren.

Die andere Variante ist es, sich einen entsprechenden tar-Ball zu besorgen, diesen auszupacken und (grob gesagt) dessen Inhalt zu kompilieren. Besorgen oder beschaffen bedeutet natürlich im Normalfall, dass Sie das jeweilige Produkt aus dem Internet herunterladen. Wenn Sie sich dafür entscheiden, das Programm selbst zu kompilieren, haben Sie den Vorteil, dass dieses Programm optimal auf Ihren PC abgestimmt wird. Im Übrigen sind auch nicht alle Programme für jede erdenkliche Distribution verfügbar, und Sie müssen unter Umständen zwangsläufig auf Programme zurückgreifen, die im Quelltext vorliegen.

Aufbau von tar-Balls

Ursprünglich wurde das Programm `tar` entwickelt, um Dateien zu archivieren und anschließend auf Magnetbändern zu sichern. Die Abkürzung »tar« steht für *Tape Archiver*. Heute verwendet man `tar` hauptsächlich, um Programme, die im Quellcode vorliegen, zu verpacken. In den meisten Fällen enthält der Dateiname die Bezeichnung des Programms, die Softwareversion und die Prozessorarchitektur, für die dieses Programm gedacht ist. Wenn Sie solche Pakete aus dem Internet herunterladen, sind diese normalerweise zusätzlich komprimiert. Ob und mit welchem Kompressionsverfahren ein tar-Ball komprimiert ist, erkennen Sie an der Dateierweiterung.

- Dateien, die nur die Erweiterung *.tar* aufweisen, sind nicht zusätzlich komprimiert und können einfach mit `tar -xvf dateiname.tar` entpackt werden.
- Dateien mit der Erweiterung *.tar.gz* sind mit `gzip` komprimiert. Sie können diese Dateien mit `tar -xvzf dateiname.tar.gz` dekomprimieren und entpacken.
- Dateien mit der Erweiterung *.tgz* sind ebenfalls mit `gzip` komprimiert. Sie können diese Dateien mit derselben Methode dekomprimieren und entpacken wie Dateien mit der Erweiterung *.tar.gz*. Es handelt sich hier lediglich um eine andere populäre Dateierweiterung für dieselbe Form von tar-Balls.
- tar-Balls mit der Erweiterung *.tar.bz2* sind mit `bzip2` komprimiert. Sie können solche Dateien mit `tar -xvjf dateiname.tar.bz2` dekomprimieren und entpacken.

Unabhängig vom Ausgangsprodukt erhalten Sie in jedem Falle eine Verzeichnishierarchie, die das Programm im Quellcode, eventuell einen initialen Makefile und eine Dokumentation zum Programm enthält.

Einen tar-Ball installieren

Wenn Sie einen tar-Ball zur Installation verwenden, muss er zunächst, so wie im vorangehenden Abschnitt beschrieben, entpackt werden. In welchem Verzeichnis Sie arbeiten möchten, ist natürlich Ihnen selbst überlassen, aber das offizielle (und damit prüfungskonforme) Verzeichnis ist */usr/src*. Im folgenden Beispiel wird ein Programm installiert, mit dem man die verschiedenen Temperaturen im System, Lüfterdrehzahlen und Mainboard-Spannungen überwachen kann. Zuerst erfolgt das Entpacken:

```
archangel:/usr/src # tar -xvzf xmbmon205.tar.gz
xmbmon205/
xmbmon205/AC-TOOLS/
xmbmon205/AC-TOOLS/config.guess
xmbmon205/AC-TOOLS/config.sub
xmbmon205/AC-TOOLS/install-sh
xmbmon205/smbuses.h
... weitere Zeilen wurden abgeschnitten ...
```

Die Anordnung der Optionen hätte auch anders aussehen können. Im weiteren Verlauf dieses Kapitels wird noch auf die genaue Syntax von `tar` eingegangen. Nach dem Auspacken des tar-Balls wechselt man in das entstandene Verzeichnis. Der Name dieses Verzeichnisses entspricht dem Namen des Pakets, allerdings ohne die Dateinamenerweiterung. Jetzt kann das Paket konfiguriert werden. In den meisten Fällen liegt dazu ein Skript mit dem Namen *configure* im Stammverzeichnis der Quellen. Sollte das einmal nicht der Fall sein, lesen Sie die README-Datei des Programms. Das Script *configure* prüft einige Gegebenheiten des Systems ab (z. B. welcher Compiler verfügbar ist) und erstellt das Makefile. Dieses enthält schließlich Informationen, die im nächsten Schritt bei der Kompilierung des Programms benötigt werden:

```
archangel:/usr/src/xmbmon205 # ./configure
checking for gcc... gcc
checking for C compiler default output... a.out
checking whether the C compiler works... yes
checking whether we are cross compiling... no
checking for suffix of executables...
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
... weitere Zeilen wurden abgeschnitten ...
```

Im nächsten Schritt erfolgt der eigentliche Vorgang des Kompilierens. Hierbei sind das Programm `make` und ein C-Compiler erforderlich. Im vorliegenden Fall kommt

der Compiler gcc zum Einsatz. Das Programm make wertet das im vorangehenden Schritt entstandene Makefile aus und steuert dementsprechend den Compiler:

```
archangel:/usr/src/xmbmon205 # make
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX getMBinfo.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX tyan_tiger.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX pci_pm.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX sensors.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX getMB-via.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX getMB-smb.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX getMB-isa.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX smbuses.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX smbus_piix4.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX smbus_amd.c
gcc -c -O3 -I. -DHAVE_CONFIG_H -Wall -DLINUX smbus_ali.c
... weitere Zeilen wurden abgeschnitten ...
```

Das Programm ist fertig und kann jetzt installiert werden. Die Installation wird normalerweise mit make install durchgeführt:

```
archangel:/usr/src/xmbmon205 # make install
install -o root -g wheel -m 4555 -c -p mbmon /usr/local/bin
install -o root -g wheel -m 4555 -c -p xmbmon /usr/X11R6/bin
```

Das Programm sollte jetzt betriebsbereit sein. In den meisten Fällen wird zum Abschluss der Installation angezeigt, in welches Verzeichnis das Programm installiert worden ist. Dadurch soll dem Benutzer das Auffinden erleichtert werden, wenn das Programm nicht über die PATH-Variable erreichbar ist.

uname zur Kernel-Quellen-Installation

In diesem Zusammenhang muss noch einmal das Kommando uname ins Spiel gebracht werden. Sie müssen nämlich die Version des laufenden Kernels ermitteln, wenn Sie passende Quellen in der Form eines tar-Balls besorgen wollen. Aber auch wenn Sie einen Paketmanager verwenden, kann uname gute Dienste leisten. Wenn Sie auf einem Debian-System die Linux-Header-Files benötigen, können Sie z. B. einen solchen Befehl verwenden:

```
root@debian:/#apt-get install linux-headers-$(uname -r)
```

Das kann etwa dann notwendig werden, wenn Sie Grafikkartentreiber oder Module für VirtualBox installieren wollen. Mit yum funktioniert das selbstverständlich ähnlich.

Archivierung im Allgemeinen

Nachdem Sie nun den praktischen Einsatz von tar-Balls kennengelernt haben, benötigen Sie noch einige theoretische Grundlagen zum Thema Archivierungsprogramme, um die Prüfungsfragen beantworten zu können. Der Übersichtlichkeit halber werden zunächst die Archivierungs- bzw. Komprimierungsprogramme mit ihren wichtigsten Optionen vorgestellt. Anschließend folgt nun eine Demonstration der praktischen Anwendung.

tar (Tape Archiver)

Das Archivierungsprogramm `tar` wurde, wie bereits erwähnt, eigentlich entwickelt, um Dateien zu einem Archiv zusammenzufassen und dieses Archiv dann auf einem Bandlaufwerk zu sichern. Sie sollten sich merken, dass man bei der Übergabe von Optionen an `tar` einen Strich verwenden kann, aber nicht muss. In Bezug auf das erste Beispiel bedeutet das, dass folgende Kommandos beide zulässig sind und dieselbe Aktion zur Folge haben:

```
archangel:/usr/src # tar -xvzf xmbmon205.tar.gz
archangel:/usr/src # tar xvzf xmbmon205.tar.gz
```

Sie können also bei sogenannten Fill-in-the-Blanks-Prüfungsaufgaben beide Schreibweisen verwenden. Wichtige Optionen sind:

- `-x` (extract) extrahiert ein Archiv.
- `-z` (gzip/gunzip) sorgt für eine Komprimierung mittels `gzip` bzw. eine Dekomprimierung mittels `gunzip`.
- `-j` (bzip2/bunzip2) sorgt für eine Komprimierung mittels `bzip2` bzw. für eine Dekomprimierung mittels `bunzip2`.
- `-J` (xz/unxz) sorgt für eine Komprimierung mittels `xz` bzw. für eine Dekomprimierung mittels `unxz`.
- `-v` (verbose) schaltet den Verbosemode ein.
- `-c` (create) erstellt ein neues Archiv.
- `-t` (table) listet den Inhalt eines Archivs auf.
- `-f` (file) zeigt an, dass zur Eingabe bzw. Ausgabe eine Datei verwendet wird. Standardmäßig arbeitet `tar` mit dem Bandlaufwerk */dev/rmt0*.

Sie sollten für die Prüfung mindestens die oben genannten Optionen kennen. Die Reihenfolge der Optionen spielt, wie bei den meisten Programmen, keine Rolle. Eine Ausnahme ist die Option `-f`. Diese muss immer als Letzte verwendet werden.

Das folgende Beispiel erzeugt ein mit `bzip2` komprimiertes Archiv der Konfigurationsdateien eines Linux-Systems:

```
archangel:/ # tar -cvjf backup.tar.bz2 /etc/*
```

Um den Inhalt eines solchen Archivs zu betrachten, benötigt man folgendes Kommando:

```
archangel:/ # tar -tvjf backup.tar.bz2 |less
```

In einem Notfall können die Dateien ohne Probleme wiederhergestellt werden:

```
archangel:/ # tar -xvjf backup.tar.bz2
```

gzip

Wenn Sie Dateien komprimieren wollen, können Sie das Programm `gzip` verwenden. Wie Sie im vorangehenden Abschnitt gelesen haben, kann `tar gzip` aufrufen. Wenn Sie mehrere Dateien zu einem Archiv zusammenfassen wollen, ist `tar` in Kombination mit den Optionen `z` oder `j` die bessere Lösung. Wenn `gzip` ohne Optionen verwendet wird, komprimiert es die angegebenen Dateien, hängt an die Zieldateien die Erweiterung `.gz` an und löscht anschließend die Originaldateien. Wenn die Originaldateien erhalten bleiben sollen, müssen Sie die Option `-c` verwenden. Damit man die Effizienz von `gzip` sehen kann, wird diese Option im folgenden Beispiel angewendet:

```
archangel:/demo # gzip lpi101.doc -c > lpi101.doc.gz
archangel:/demo # ls -lh
total 1.2M
drwxr-xr-x   2 root root 4.0K Nov 16 17:50 .
drwxr-xr-x  24 root root 4.0K Nov 16 17:45 ..
-rwxr--r--   1 root root 903K Nov 16 17:46 lpi101.doc
-rw-r--r--   1 root root 237K Nov 16 17:50 lpi101.doc.gz
```

Die Datei ist von 903 kB auf 237 kB komprimiert worden. Der Redirektor `>` war in diesem Fall notwendig, weil die Ausgabe von `gzip` auf dem Bildschirm erfolgt, wenn die Option `-c` verwendet wird.

Um eine Datei zu komprimieren, ohne das Original beizubehalten, geben Sie die Datei einfach ohne weitere Parameter an:

```
archangel:/demo # gzip lpi101.doc
```

Sie können auch mehrere Dateien durch Leerstellen voneinander getrennt angeben oder Platzhalter (Wildcards) verwenden.

Den Inhalt eines mit `gzip` komprimierten Archivs können Sie mit dem Schalter `-l` auflisten. Es wird Ihnen dann auch die komprimierte Größe der Einzeldateien angezeigt, falls das Archiv mehrere Dateien enthält:

```
archangel:/demo # gzip -l lpi101.doc.gz
    compressed        uncompressed  ratio uncompressed_name
        31                   0      0.0%         lpi101.doc
```

gunzip

Um eine oder mehrere Dateien, die mit `gzip` komprimiert wurde(n), wieder zu dekomprimieren, stehen Ihnen mehrere Möglichkeiten zur Verfügung:

- `gunzip` ist das offizielle Gegenstück zu `gzip`. Auf der Kommandozeile angegebene Dateien werden von `gunzip` dekomprimiert.
- `gzip -d` startet `gzip` im Modus »decompress«.
- `zcat` verhält sich ohne Optionen wie `gunzip -c`.

bzip2

Sie können zum Komprimieren von Dateien auch `bzip2` verwenden. In vielen Fällen erreicht man hierdurch eine höhere Kompression als mit `gzip`. Das bedeutet natürlich auch, dass mehr Prozessorressourcen verwendet werden, was aber nur bei großen Datenmengen von Bedeutung sein dürfte.

```
archangel:/demo # bzip2 lpi101.doc -c > lpi101.doc.bz2
archangel:/demo # ls -lh
total 1.2M
drwxr-xr-x   2 root root 4.0K Nov 16 18:00 .
drwxr-xr-x  24 root root 4.0K Nov 16 17:45 ..
-rwxr--r--   1 root root 903K Nov 16 17:46 lpi101.doc
-rw-r--r--   1 root root 213K Nov 16 18:00 lpi101.doc.bz2
```

Die Datei ist von 903 kB auf 213 kB komprimiert worden. Die Effizienz war also zumindest in diesem Fall etwas besser als bei der Verwendung von `gzip`.

bunzip2

Ähnlich wie bei `gzip` gibt es auch Kommandos zur Dekomprimierung von *bz2*-Dateien:

- `bunzip2`
- `bzip2 -d`
- `bzcat`

Es ist sogar so, dass es sich in allen drei Fällen um dasselbe Programm handelt, das jeweils über unterschiedliche Links aufgerufen wird.

xz und unxz

Das zurzeit aktuelle Datenkompressionsformat ist *xz*. Es eignet sich aufgrund des verwendeten Algorithmus besonders für Binärdateien. Die zu *xz* gehörenden Kommandos sind, wie Sie wahrscheinlich schon vermutet haben, `xz`, `unxz` und `xzcat`. Auf-

grund der identischen Verwendung mit den zuvor beschriebenen Tools soll hier auf eine Vorführung verzichtet werden.

patch

Wie ein Patch angewendet wird, kennen Sie bereits im Zusammenhang mit Kernel-Quellen. Das Thema soll deshalb hier nicht noch einmal wiederholt werden.

206.2 Datensicherung

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, ein Backup wichtiger Systemdaten mit Bordwerkzeugen zu erstellen.

Wichtigste Wissensgebiete:

- Kenntnis der Verzeichnisse, die in eine Datensicherung gehören
- Übersicht über Netzwerkbackuplösungen wie Amanda, Bacula, Bareos und BackupPC
- Vor- und Nachteile von Bändern, beschreibbaren CDs und anderen Backupmedien
- Anlegen von partiellen und manuellen Backups
- Prüfen der Integrität von Backups
- partielles oder vollständiges Wiederherstellen von Daten aus Backups

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */bin/sh*
- `dd`
- `tar`
- */dev/st** und */dev/nst**
- `mt`
- `rsync`

Allgemeines

In vielen Fachbüchern, in denen es ein Kapitel zum Thema Datensicherung gibt, wird eindringlich auf die Wichtigkeit dieses Themas hingewiesen. Davon ausgehend, dass Ihnen schon lange klar ist, dass Daten regelmäßig gesichert werden müssen, soll also diesmal auf eine solche Vorrede verzichtet werden.

Was muss gesichert werden?

Ein wichtiger Aspekt bei der Prüfung ist die Auswahl der zu sichernden Daten. In den meisten Fällen wird gefragt, welche Daten für ein Backup am unwichtigsten sind oder aus anderen Gründen nicht gesichert werden. Es sollen hier noch einmal die bekanntesten Verzeichnisse aufgelistet und in Bezug auf dieses Thema kommentiert werden.

- */bin* enthält Programme für Benutzer. Eine Sicherung wäre sinnvoll.
- */boot* enthält hauptsächlich den Kernel und sollte gesichert werden.
- */dev* enthält keine echten Daten, sondern nur Gerätedateien und ist deshalb eine Topantwort der nicht zu sichernden Verzeichnisse.
- */etc* enthält Konfigurationsinformationen. Die hier befindlichen Dateien sind normalerweise mühevoll konfiguriert worden. Eine Sicherung wäre dringend angebracht.
- */home* enthält die Dokumente der Benutzer, muss also unbedingt gesichert werden. Ein Verlust dieser Daten könnte unternehmenskritisch sein.
- */lib* enthält Programmbibliotheken und sollte gesichert werden.
- */mnt* enthält meist keine Daten, sondern fungiert als Mountpoint, muss also auch nicht gesichert werden.
- */opt* enthält optionale Programme, eine Datensicherung ist sinnvoll.
- */proc* ist ein Pseudoverzeichnis und ist deshalb eine Topantwort der nicht zu sichernden Verzeichnisse.
- */root* enthält Programme und Dokumente für den root. Eine Sicherung dieser Dateien ist sinnvoll.
- */sbin* enthält Programme für den root. Eine Sicherung ist auch hier sinnvoll.
- */tmp* enthält Temporärdateien. Vorsicht bei der Prüfung! Ob dieses Verzeichnis mitgesichert werden sollte oder nicht, hängt von den angebotenen Alternativen (zum Nicht-Sichern) ab. Einige Programme speichern hier ihre Statusinformationen. Diese sind zwar meist nicht überlebenswichtig, aber zumindest nützlich.
- */usr* enthält Programme für Benutzer. Eine Sicherung ist sinnvoll.
- */var* enthält u. a. eingehende E-Mails und Zonendaten für BIND; eine Sicherung ist absolut nötig.

Prüfungstipp

Seien Sie also in der Prüfung vorsichtig und machen Sie sich Gedanken zu den Prioritäten, die bei den zu sichernden Verzeichnissen zu setzen sind. Machen Sie sich immer klar, was ein Verzeichnis enthält und lassen Sie sich nicht von der Bezeichnung des Verzeichnisses irritieren.

Das Verzeichnis */usr* klingt für manche Prüflinge wichtiger als */var*. Man sollte aber berücksichtigen, dass */usr* im Gegensatz zu */var* keine selbst erstellten Daten enthält und somit bei einer Datensicherung als unwichtiger zu betrachten ist. Außerdem kann */usr* in vielen Fällen vom Installationsdatenträger wiederhergestellt werden.

Backupstrategien

Das Ausarbeiten einer Backupstrategie kann in Unternehmen, in denen ständig große Datenmengen erstellt oder modifiziert werden, eine anspruchsvolle Aufgabe darstellen. In den meisten Fällen müssen zumindest die folgenden Aspekte berücksichtigt werden:

- Da ein Backup nicht produktiv ist, sollte es möglichst wenig Kosten verursachen.
- Die Wiederherstellbarkeit muss sichergestellt sein und deshalb regelmäßig getestet werden.
- Das Backup soll das Tagesgeschäft nicht behindern und deshalb am besten außerhalb der Geschäftszeiten stattfinden.
- In einem Notfall muss das Backup so schnell wie möglich wiederherstellbar sein.
- Daten müssen auch in Fällen extremer Beschädigung (z. B. bei einem Brand, bei dem ein ganzes Gebäude zerstört wird) wiederherstellbar sein.

Es sind noch mehr Anforderungen an eine Datensicherungsstrategie denkbar, aber auch schon die fünf genannten schließen sich eigentlich zumindest teilweise gegenseitig aus. Wenn dann festgelegt worden ist, welche Kriterien in der zu sichernden Umgebung die höchste Priorität haben, kann man eine geeignete Kombination aus Sicherungsarten auswählen.

Sicherungsarten

Die Sicherungsarten, die unter Linux typischerweise verwendet werden, unterscheiden sich stark von denen, die auf einem Windows-basierten System zum Einsatz kommen. Wenn Sie die Datensicherungsverfahren von Windows kennen, wissen Sie auch, dass die Windows-basierten Dateisysteme (egal ob FAT, VFAT oder NTFS) mit einem Archiv-Flag arbeiten, wenn es darum geht, den Sicherungszustand einer Datei zu markieren. Ein solches Archiv-Flag gibt es unter Linux nicht, weshalb die Sicherungsmechanismen natürlich auch eine andere Strategie verfolgen müssen. Bei kommerziellen Backupsystemen spricht man von Sicherungsarten, die als Normalbackup (auch Vollbackup), differenzielles Backup, inkrementelles Backup oder Kopierbackup bezeichnet werden. Diese Sicherungsarten verhalten sich, davon ausgehend, dass eine Backupstrategie einem Wochenzyklus folgt, so:

- Das *Normalbackup* beinhaltet alle ausgewählten Daten. Es wird keine Rücksicht auf die Inhalte bereits vorhandener Sicherungen genommen.
- Das *differenzielle Backup* beinhaltet alle ausgewählten Daten, die sich nach der letzten Vollsicherung (Normalbackup) geändert haben.
- *Das inkrementelle Backup* beinhaltet alle ausgewählten Daten, die sich nach der letzten Vollsicherung (Normalbackup) oder dem letzten inkrementellen Backup geändert haben.
- Das *Kopierbackup* ist eine zusätzliche Sicherung, die den Verlauf der eigentlichen Backupstrategie nicht beeinflusst. Das Kopierbackup wird vor riskanten administrativen Eingriffen oder zur Aufbewahrung außer Haus durchgeführt.

In der Regel werden verschiedene Backupmethoden miteinander kombiniert. Bei der Kombination von Normalbackups mit differenziellen Backups ergibt sich in einem Wochenzyklus Folgendes:

- Das Normalbackup dauert relativ lange und sollte deshalb am Wochenende durchgeführt werden.
- Die Größe des differenziellen Backups wächst im Laufe der Woche und stört deshalb möglicherweise zunehmend den Produktionsbetrieb.
- Bei einer Wiederherstellung sind maximal zwei Datensätze nötig, weil das differenzielle Backup die volle Differenz zum Normalbackup enthält.

Eine andere verbreitete Kombination besteht aus Normalbackup und inkrementellen Backups:

- Das Normalbackup sollte natürlich auch hier am Wochenende durchgeführt werden.
- Da das inkrementelle Backup nur Änderungen zum vorangehenden Backup enthält, wächst die zu sichernde Datenmenge nicht täglich an. Die Störung des Produktionsbetriebs ist also eher gering.
- Eine Wiederherstellung benötigt alle Sicherungssätze ab der letzten Vollsicherung und kann somit viel Zeit in Anspruch nehmen.

Hardware und Verbrauchsmaterial

Für eine Datensicherung kommen eigentlich alle Medien infrage, die man dem Computer ohne großen Aufwand entnehmen und an einem dezentralen Ort aufbewahren kann. Wenn ein Computer durch einen Brand oder einen Wasserrohrbruch zerstört wird, hilft ein Backup, das sich auf einer zweiten Festplatte innerhalb desselben Gerätes befindet, nicht weiter. Welche Materialien für die Sicherung infrage kommen, ist letztendlich aber wahrscheinlich auch eine Frage der Kosten.

- Festplatten sind schnell und haben eine hohe Speicherkapazität. Leider ist eine Datensicherung auf Wechselfestplatten relativ teuer.
- Preisgünstig und schnell ist eine Sicherung auf CD-RW oder DVD-RW. Leider ist die verfügbare Kapazität auf einem einzelnen Medium sehr begrenzt.
- Für die Prüfung muss auch noch eine Datensicherung auf Disketten oder ZIP-Medien in Erwägung gezogen werden. Die Kapazität dieser Medien ist wohl in den meisten Fällen heutzutage nicht mehr ausreichend.
- Das wichtigste Medium für eine professionelle Backupstrategie ist wahrscheinlich das Bandlaufwerk. Dieses ist zwar in puncto Geschwindigkeit den Festplatten deutlich unterlegen, hat aber eine ausreichende Speicherkapazität und verursacht erheblich weniger Kosten.

Zur Sicherung benötigte Gerätedateien

Wie bereits beschrieben, sollten Sie in der Prüfung auch noch die Verwendung eines Diskettenlaufwerks in Erwägung ziehen. Die Gerätedatei für das erste (und vermutlich auch einzige) Diskettenlaufwerk ist */dev/fd0*. Das Kürzel »fd« steht für Floppy-Disk.

Bei einer Sicherung auf einer Festplatte oder in einem ZIP-Laufwerk werden die normalen Gerätedateien von IDE verwendet, die Ihnen sicherlich vertraut sind. Wenn ein ZIP-Laufwerk als Master am sekundären IDE-Kanal angeschlossen ist, steuert man dieses über die Gerätedatei */dev/hdc* an. Lesen Sie die Bezeichnung von IDE-Geräten im Zweifelsfall noch einmal im vorderen Teil des Buchs nach.

Interessanter ist die Verwendung eines Bandlaufwerks. Hier muss als Erstes zwischen einem *Floppy-Streamer* und einem *SCSI-Streamer* unterschieden werden. Ein Floppy-Streamer ist ein Bandlaufwerk, das an demselben Kabel angeschlossen wird, an dem sich auch das Diskettenlaufwerk befindet. Hieraus leitet sich auch der Name Floppy-Streamer ab. Ein SCSI-Streamer ist natürlich an einem SCSI-Controller angeschlossen, der aber leider nicht zur Grundausstattung eines handelsüblichen PCs gehört. Ein SCSI-Streamer ist die erheblich schnellere Variante. Auch gibt es für diese Geräte üblicherweise Medien mit wesentlich größeren Speicherkapazitäten als für Floppy-Streamer. Diese Vorteile schlagen sich bedauerlicherweise im Anschaffungspreis nieder. Für beide Varianten gibt es jeweils zwei Arten von Gerätedateien. Diese zu unterscheiden ist für die Prüfung unerlässlich. Die Gerätedateien unterscheiden sich im Rückspulverhalten des Bandlaufwerks. Es gibt jeweils eine rückspulende und eine nicht rückspulende Gerätedatei. Letztere wird benötigt, wenn man beabsichtigt Datensätze anzuhängen, anstatt bestehende Datensätze zu überschreiben. Wenn mehrere Geräte des gleichen Typs vorhanden sind, werden diese in der Reihenfolge durchnummeriert, in der sie am Bus gefunden werden. Beispiele:

- */dev/st0* – erstes SCSI-Tape, rückspulend
- */dev/nst0* – erstes SCSI-Tape, nicht rückspulend
- */dev/ft0* – erstes Floppy-Tape, rückspulend
- */dev/nft0* – erstes Floppy-Tape, nicht rückspulend

Die Erläuterungen sind, damit Sie sich diese besser merken können, direkt dem Kürzel der Gerätedatei entnommen. Also »ft« = Floppy-Tape usw.

Geeignete Programme zur Erstellung von Datensicherungen

In der Prüfung sollten Sie mit der Syntax der wichtigsten Programme, die zur Datensicherung verwendet werden können, vertraut sein. Die wichtigsten Programme sind in diesem Zusammenhang `dd`, `dump`, `restore`, `tar` und `mt`, wobei `mt` kein Programm zum Kopieren oder Sichern von Daten ist, sondern lediglich der Steuerung eines Bandlaufwerks dient.

Verwendung von dd

Mit `dd` kann man Daten kopieren oder konvertieren. Da die Konvertierung von Daten in diesem Zusammenhang keine Rolle spielt, sei nur erwähnt, dass es etwa möglich ist, Daten von einem Datenträger zu lesen und dann mit einer geänderten Blockgröße auf einen anderen Datenträger zu schreiben.

Wenn man `dd` (eigentlich sinnloserweise) ohne Parameter startet, liest dieser von der Standardeingabe und schreibt auf die Standardausgabe. Man kann dann erkennen, dass `dd` »stur« das schreibt, was eingegeben worden ist, ohne zu interpretieren oder Modifikationen vorzunehmen:

```
archangel:/ # dd
Mal was schreiben und mit Strg + D abschicken...
Mal was schreiben und mit Strg + D abschicken...
0+1 records in
0+1 records out
49 bytes (49 B) copied, 4.2909 seconds, 0.0 kB/s
```

Das ist auch schon einer der großen Vorzüge von `dd`, weil man auf diese Art problemlos empfindsame Betriebssysteme von einer Festplatte auf eine andere klonen kann. Es ist ratsam, dann gleich große Festplatten zu verwenden, weil selbst der MBR und die darin enthaltene Partitionstabelle mitkopiert werden. Dazu müssen natürlich Eingabekanal und Ausgabekanal geändert werden. Wenn die Originalfestplatte die Gerätedatei */dev/sda* verwendet und die erwünschte Kopie auf */dev/sdb* erfolgen soll, sieht die `dd`-Zeile so aus:

```
archangel:/ # dd if=/dev/sda of=/dev/sdb
```

Es muss allerdings darauf hingewiesen werden, dass die Duplizierung großer Datenträger mittels `dd` erhebliche Zeit in Anspruch nimmt.

Sehr beliebt ist auch das Sichern des Master Boot Records mittels `dd`. Das funktioniert dann folgendermaßen:

```
archangel:/ # dd if=/dev/hda of=mbr.backup ibs=512 count=1
1+0 records in
1+0 records out
512 bytes (512 B) copied, 0.037257 seconds, 13.7 kB/s
```

Hier wurde als Eingabe die Festplatte an */dev/hda* verwendet. Die Ausgabe erzeugt die Datei *mbr.backup* im aktuellen Verzeichnis, weil kein Pfad angegeben worden ist. Der Parameter `ibs` steht für »in block size« und sorgt dafür, dass ein gelesener Block immer 512 Bytes groß ist. Mit `count` wird die Anzahl der zu lesenden Blöcke auf 1 festgelegt, denn der MBR ist ja ausschließlich der erste Block auf einer Festplatte.

Soll `dd` zur Erstellung einer Datensicherung verwendet werden, muss als Ausgabekanal einfach der Streamer eingestellt werden. Denken Sie hierbei immer daran, dass es rückspulende und nicht rückspulende Geräte gibt. Beispiel:

```
archangel:/ # dd if=/home of=/dev/st0 cbs=16b
```

Dieses Kommando sichert alle Benutzerverzeichnisse auf dem ersten SCSI-Streamer und spult das Band anschließend zurück. Der `cbs`-Wert sorgt für eine dem Streamer angemessene Blockgröße.

Verwendung von tar

Natürlich kennen Sie `tar` schon, aber dieses Programm muss zur Wiederholung und Vertiefung (mit ein paar zusätzlichen Optionen) im Zusammenhang mit dem Thema Datensicherung noch einmal auf den Tisch gebracht werden. Die Abkürzung `tar` steht für »Tape Archiver« und es lässt sich leicht erkennen, welcher der ursprüngliche Verwendungszweck dieses Programms einmal war. Auch wenn `tar` heutzutage eher dafür bekannt ist, Programmpakete internetgerecht zu archivieren und, wenn nötig, auch zu komprimieren, handelt es sich dennoch um ein Backupprogramm. Sie sollten sowohl für die Prüfung als auch für Ihre täglichen Arbeiten mit den wichtigsten Optionen von `tar` vertraut sein.

- `-x` extrahiert bestehende Archive
- `-c` erstellt (create) ein neues Archiv
- `-t` zeigt den Inhalt (table) eines bestehenden Archivs
- `-z` führt eine Bearbeitung mittels `gzip`-Kompression durch
- `-v` steht für Verbose Mode

- `-f` ist immer der letzte Schalter vor der Datei (file), die mit `tar` bearbeitet werden soll. Die Option `-f` an der falschen Stelle hat immer eine Fehlermeldung zur Folge.
- `-w` führt den interaktiven Modus aus. Es wird vor der Archivierung bzw. Wiederherstellung jeder einzelnen Datei eine Bestätigung durch den Benutzer abgefragt. Die Bearbeitung großer Archive im interaktiven Modus kann viel Zeit in Anspruch nehmen.

Prüfungstipp

Der `tar`-Befehl versteht noch erheblich mehr Optionen. Sie sollten aber zumindest die hier genannten Möglichkeiten unbedingt kennen. Beachten Sie bitte, dass der Strich vor den Optionen bei `tar` auch weggelassen werden kann.

Datensicherung mit tar

Ein Backup aller Benutzerverzeichnisse kann mit dem `tar`-Befehl folgendermaßen ausgeführt werden:

```
archangel:/ # tar cvf /dev/st0 /home
```

Es wird eine Sicherung des Verzeichnisses */home*, inklusive aller Unterverzeichnisse und Dateien, auf dem SCSI-Streamer erstellt und das Band anschließend zurückgespult. Die Option `v` sorgt für umfangreiche Informationen über die zu sichernden Dateien auf dem Bildschirm. Das folgende Beispiel demonstriert lediglich die Verwendung des Striches vor den benötigten Optionen und lässt die Option `v` aus:

```
archangel:/ # tar -cf /dev/st0 /home
```

Will man nun den Inhalt des Archivs überprüfen, kann man das mit der Option `-t` durchführen:

```
archangel:/ # tar -tf /dev/st0 | less
```

Es folgt eine übersichtliche Ausgabe der gesicherten Dateien durch `less`.

Wiederherstellung von Daten mit tar

Um die Daten aus dem vorangehenden Beispiel wiederherzustellen, geht man folgendermaßen vor:

```
archangel:/ # tar -xvf /dev/st0
```

Die Angabe eines Zielverzeichnisses ist nicht erforderlich. Die Dateien werden unterhalb des aktuellen Verzeichnisses wiederhergestellt. Achten Sie also unbedingt darauf, in welchem Verzeichnis Sie sich gerade befinden. Es ist mit `tar` auch möglich,

eine einzelne Datei aus einem Archiv zu extrahieren. Dazu muss diese Datei lediglich auf der Kommandozeile mit angegeben werden:

```
archangel:/ # tar -xvf /dev/st0 /home/harald/steuererklärung.ods
```

Das Wiederherstellen einer einzelnen Datei von einem Bandlaufwerk kann erhebliche Zeit in Anspruch nehmen. Denn auch bei der Wiederherstellung einer einzelnen Datei muss zunächst die Bandkopfzeile gelesen und das Band anschließend an die Stelle gespult werden, an der sich die Datei befindet. Das kann schon Nerven kosten, wenn man eine wichtige Datei wiederherstellen muss.

Verwendung von mt

Damit die Positionierung eines Bandlaufwerks gesteuert werden kann, benötigt man das Programm `mt`. Die Abkürzung `mt` steht für »magnetic tape«. Typische Operationen sind:

- Mit diesem Kommando wird das Bandlaufwerk mit der Gerätedatei */dev/st0* zurückgespult:

  ```
  archangel:~ # mt -f /dev/st0 rewind
  ```

- Um weitere Daten an eine bestehende Sicherung anzuhängen, kann mit diesem Kommando an das Ende eines bestehenden Sicherungssatzes gespult werden:

  ```
  archangel:~ # mt -f /dev/st0 eom
  ```

Netzwerkbackuplösungen

Für den professionellen Einsatz in größeren Umgebungen gibt es mehrere fertige Programme, die Datensicherungen über das Netzwerk ausführen können. Dazu zählen *Amanda*, *Bacula*, *Bareos* (einem Fork von Bacula) und *BackupPC*. Es sollte für die Prüfung ausreichen, dass Sie wissen, worum es sich bei diesen Produkten handelt. Spezifische Fragen zu diesen Programmen sind in der Prüfung aufgrund deren Komplexität nicht zu erwarten.

rsync-Kommando

Wenn Sie Daten über das Netzwerk sichern wollen, bietet das Programm `rsync` einige hervorragende Eigenschaften für diese Aufgabe. So ist `rsync` z. B. bei wiederholten Datensicherungen gleicher Dateien dazu in der Lage, die Differenzen zwischen zwei Dateien zu bestimmen und lediglich die Änderungen zu übertragen. Hierbei kommt der Deltaalgorithmus zum Einsatz. Dieser Mechanismus hilft, Bandbreite zu sparen, wenn häufig an großen Dateien lediglich kleine Änderungen vorgenommen werden. Sie sehen das am besten an einem Beispiel:

```
root@archangel:/home/harald# rsync -avz /storage/buch/ \
root@192.168.50.12:/backup
root@192.168.50.12's password:
sending incremental file list
LPIC-2/
LPIC-2/206.odt
sent 49100 bytes  received 470 bytes  11015.56 bytes/sec
total size is 207050318  speedup is 4176.93
```

In diesem Fall hat `rsync` lediglich 49.100 Bytes Daten übertragen. Es ist also offensichtlich seit dem letzten Backup nur eine geringe Änderung an nur einem Dokument vorgenommen worden. Die Bandbreitenersparnis gegenüber anderen Sicherungsmethoden (z. B. `scp` oder `rcp`) liegt also auf der Hand.

Sie können mit `rsync` Daten auf einen entfernten Computer kopieren, so wie in dem Beispiel oben, oder umgekehrt, von einem entfernten Rechner auf das lokale System. Als dritte Variante ist auch ein Synchronisationsvorgang innerhalb eines Systems (z. B. auf einer lokal angeschlossenen Festplatte) möglich. Gegenüber einem gewöhnlichen Kopiervorgang bleiben bei der Verwendung von `rsync` Dateiberechtigungen, Timestamps, Eigentümer und Gruppe erhalten.

Achtung

Die Angabe der Quelle im obigem Beispiel (*/storage/buch/*) bewirkt, dass lediglich der Inhalt des Verzeichnisses *buch* transferiert wird. Lässt man den letzten Slash weg (*/storage/buch*), wird im Ziel ein Verzeichnis mit dem Namen *buch* angelegt und der Inhalt hineinkopiert.

Wenn die Daten aus dem vorangehenden Beispiel in einem lokalen Verzeichnis wiederhergestellt werden müssen, können Sie dieses Kommando verwenden:

```
root@archangel:/# rsync -azv 192.168.50.12:/backup /restore
```

Bei einer Wiederherstellung sollten Sie zunächst ein Verzeichnis (in diesem Fall */restore*) anlegen und nicht einfach in das Originalverzeichnis schreiben. Sollten nämlich aus irgendeinem Grund im Backup veraltete Daten liegen, haben Sie immer noch die Option, zumindest einen Teil der Originaldateien beizubehalten.

Eine Datensicherung auf einem lokal angeschlossenen Medium könnte z. B. mit einem solchen Kommando durchgeführt werden:

```
root@archangel:/# rsync -av /storage/buch/ /media/usb-hd
```

Wichtige Optionen für `rsync` sind:

- `-a` aktiviert den Archivmodus.
- `-z` komprimiert Daten während des Transfers. Am Zielort werden die Daten wieder dekomprimiert.
- `-r` – rekursive Befehlsausführung wird verwendet (ist in `-a` enthalten).
- `-v` sorgt für die verbose Ausgabe.
- `-u` (update) bewirkt, dass die Übermittlung von Dateien, die am Zielort neuer sind, ausgelassen wird.
- `-p` – Berechtigungen (Permissions) werden erhalten (ist in `-a` enthalten).
- `-4` verwendet IPv4 für den Transport, wenn möglich.
- `-6` verwendet IPv6 für den Transport, wenn möglich.
- `-e` wird verwendet, um eine Remote-Shell anzugeben. Sie sehen dazu gleich ein Beispiel.

Es gibt noch wesentlich mehr Optionen für `rsync`, aber diese Auflistung enthält die gebräuchlichsten davon. Sie können die Netzwerkübertragung des Backups absichern, indem Sie SSH verwenden. Die Option `-e` erlaubt es Ihnen, für `rsync` eine beliebige Remote-Shell zu verwenden:

```
root@archangel:/# rsync -azv -e ssh /storage/buch/ \
192.168.50.12:/backup
```

Wenn Sie ein solches Kommando eingegeben haben, werden Sie nach dem Passwort für den entsprechenden User auf dem Zielsystem gefragt. Um einen solchen Vorgang mit Skripten zu automatisieren, müssten Sie also mit Schlüsseln arbeiten, damit keine Passwortauthentifizierung mehr erforderlich ist. Dieses Verfahren ist Bestandteil des Kapitels »Systemsicherheit«.

rsync-Dämon

Wenn Sie aus irgendwelchen Gründen keine Remote-Shell einsetzen können oder wollen, müssen Sie mit einer Passwortauthentifizierung arbeiten. Das ist allerdings nur möglich, wenn Sie `rsync` auf dem Zielsystem als Daemon ausführen. Da diese Methode nicht prüfungsrelevant ist, soll dazu nur kurz Folgendes gesagt werden:

Sie können `rsync` als Daemon ausführen, indem Sie `inetd` oder `xinetd` einsetzen. Für einen schnellen Test können Sie aber auch einfach den Befehl `rsync --daemon` verwenden. Sie benötigen hierzu die Konfigurationsdatei */etc/rsyncd.conf*. Auf den meisten heutigen Systemen ist diese Datei standardmäßig nicht mehr vorhanden, weil die meisten Administratoren `rsync` aus Sicherheitsgründen eher über SSH verwenden.

Eine einfache *rsyncd.conf*-Datei zu Testzwecken könnte, passend zu dem Beispiel aus den vorangehenden Abschnitten, so aussehen:

```
[backup]
path = /backup
comment = backup
```

Wenn Sie die Datei angelegt haben, sollte sich `rsync --daemon` bereits ohne Fehlermeldung ausführen lassen.

Sie können jetzt das Passwort in einem Skript auf zwei verschiedene Arten verwenden: Entweder Sie legen eine Datei an, die das Passwort enthält und übergeben diese mithilfe des Schalters `--password-file` an das Kommando `rsync`, oder Sie hinterlegen das Passwort vor der Ausführung des `rsync`-Befehls in der Variablen *RSYNC_PASSWORD*. Beim Zugriff auf `rsync` als Daemon wird übrigens eine leicht geänderte Syntax verwendet, wie das Beispiel zeigt:

```
root@archangel:/home/harald# rsync -avz /storage/buch/ \
 root@192.168.50.12:backup --password-file=/etc/pw
```

Empfehlung

Sie sollten `rsync` vorzugsweise über SSH einsetzen. Der sichere Verschlüsselungsmechanismus von SSH sollte als Argument für dessen Verwendung selbst sprechen.

206.3 Benutzer über systembezogene Angelegenheiten benachrichtigen

Wichtung: 1

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Benutzer über aktuelle Informationen zum System in Kenntnis zu setzen.

Wichtigste Wissensgebiete:

- Automatisieren der Kommunikation mit Benutzern über Login-Meldungen
- Benachrichtigen der angemeldeten Benutzer über Systempflegemaßnahmen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/issue*
- */etc/issue.net*
- */etc/motd*

- `wall`
- `shutdown`
- `systemctl`

Allgemeines

In einem Netzwerk ist es häufig nötig, Benutzer über bestimmte Vorgänge zu benachrichtigen. Wenn in einem Unternehmen z. B. mehrere hundert Benutzer arbeiten, ist es einfach nicht mehr möglich, jeden einzelnen Benutzer anzurufen, weil zu Wartungszwecken ein Server heruntergefahren werden muss. Deshalb bietet Linux verschiedene Einrichtungen, mit denen der root seine Benutzer auf dem Laufenden halten kann.

Konfigurationsdateien und Werkzeuge

/etc/issue

Noch vor der Anmeldung wird der Inhalt der Datei */etc/issue* auf dem lokalen Bildschirm eines Computers ausgegeben. Der Inhalt dieser Datei sieht bei einem Ubuntu-System z. B. so aus:

```
Ubuntu 12.04.1 LTS \n \l
```

Der erste Teil des Textes wird dann so, wie er ist, von `getty` auf den Bildschirm geschrieben. `\n` wird durch den Hostnamen ersetzt. Anstatt (`\l`) wird die `tty`-Leitungsbezeichnung angezeigt, also z. B. `tty1`.

In dieser Datei kann nun beliebig Text platziert werden, wie etwa Hinweise auf Konsequenzen bei nicht erlaubter oder unsachgemäßer Benutzung des Computers. Außer den beiden Variablen, die oben schon beschrieben worden sind, können noch weitere Platzhalter eingesetzt werden:

- `\n` – Name bzw. Hostname
- `\o` – Domainname
- `\b` – Baudrate des Terminals
- `\l` – Line-Name (`tty`)
- `\s` – Systemname (Betriebssystem)
- `\m` – Maschinenarchitektur (z. B. i586)
- `\r` – Releasenummer des Kernels
- `\v` – Version, OS-Version
- `\d` – Datum

- `\t` – Time; aktuelle Uhrzeit
- `\u` – Users; Anzahl der angemeldeten Benutzer
- `\U` – Users; Anzahl der angemeldeten Benutzer

/etc/issue.net

Die Datei */etc/issue.net* ist mit der »normalen« */etc/issue*-Datei identisch. Der einzige Unterschied ist, dass der Inhalt der Datei *issue.net* nur bei Anmeldungen über das Netzwerk angezeigt wird. Einige Terminalemulatoren führen allerdings, sobald die Login-Aufforderung kommt, ein Clear Screen aus. Die Folge ist, dass die Meldung der */etc/issue.net*-Datei nur für Sekundenbruchteile angezeigt wird.

Der Grund für die Existenz der *issue.net*-Datei ist, dass Remote-User eventuell andere Informationen bekommen sollen als lokale User (z. B. abweichende Rechtsbelehrungen über Missbrauch).

/etc/motd

Eine weitere Methode der Information ist die */etc/motd*-Datei. Sie beinhaltet vom Begriff her das Motto des Tages (Motto of the Day). Der Inhalt dieser Datei wird bei jedem Benutzer (egal ob lokal oder remote angemeldet) nach erfolgreicher Anmeldung auf den Bildschirm geschrieben.

wall

Eine sehr direkte Art, seinen Benutzern sofort eine Nachricht zukommen zu lassen, ist der `wall`-Befehl. Man kann damit eine Nachricht aus einer Datei oder von der Standardeingabe sofort an alle angeschlossenen Terminals senden (einschließlich des eigenen). Eine an die Wand geworfene Nachricht liest sich dann folgendermaßen:

```
archangel:~ # echo "Heute 14.00 User-Versammlung\!" | wall
Broadcast Message from root@archangel
        (/dev/pts/0) at 11:02 ...
Heute 14.00 User-Versammlung!
```

Verwenden Sie übrigens lieber keine Umlaute. Denn diese können auf vielen Terminals nicht wiedergegeben werden und machen den Text daher unleserlich.

shutdown

Eine weitere Form der Benachrichtigung steckt im `shutdown`-Kommando, das ja eigentlich zum Herunterfahren des Systems dient. Im Gegensatz zum Beenden des Systems mittels `init 0` werden alle Benutzer, die über ein Terminal angeschlossen sind, darüber informiert, dass das System heruntergefahren wird. Auf diese Art haben Benutzer die Möglichkeit, geöffnete Dateien rechtzeitig zu schließen. In die-

sem Zusammenhang soll noch einmal auf die Option `-k` hingewiesen werden. Wenn Sie diese Option mit dem `shutdown`-Kommando verwenden, wird das System nicht wirklich heruntergefahren, sondern es werden lediglich eine Nachricht generiert und weitere Logins durch Benutzer verhindert:

```
root@archangel:/# shutdown -k 10
root@archangel:/#
An alle Benutzer verteilte Nachricht von harald@archangel
........(/dev/pts/0) um 18:35 ...
The system is going down for maintenance in 10 minutes!
```

systemctl

Bei modernen Systemen wird das Herunterfahren oder Neustarten des Computers von *Systemd* übernommen. Wie Sie wissen, wird das Kommando `systemctl` hierbei zur Steuerung verwendet. Auch hier werden angemeldete Benutzer auf allen Konsolen entsprechend benachrichtigt. Das ist immer dann der Fall, wenn mit `systemctl` eine der Optionen `halt`, `poweroff`, `reboot` oder `kexec` verwendet wird. Sie können die Benachrichtigung verhindern, indem Sie den Schalter `--no-wall` anhängen:

```
root@archangel:/# systemctl poweroff --no-wall
```

Übungsfragen zu LPI 117-201

Die folgenden Fragen sollen Ihnen helfen, sich an die Art der Fragestellung in der wirklichen Prüfung zu gewöhnen. Es hat keinen Zweck, die Fragen einfach auswendig zu lernen, denn es sind keine echten Prüfungsfragen. Sie sollten versuchen, die Antworten zu jeder einzelnen Frage zu verstehen. Deshalb werden sowohl die richtigen als auch die falschen Antworten im Lösungsteil des Buches detailliert besprochen. Das Üben mit diesen Fragen soll Ihnen auch die Herangehensweise bei eventuell Ihnen unbekannten Themen näher bringen. Ein unbekanntes Kommando in einer Frage ist nämlich noch längst kein Grund, eine Frage einfach nicht zu beantworten. Oft führt ein wenig Logik oder das Ausschlussverfahren dennoch zum Ziel.

Fragen

Frage 1:

Sie haben die Quellen für einen neuen Kernel von einem FTP-Server heruntergeladen. Unter welchem Verzeichnis werden Sie den neuen Kernel entpacken?

- ☐ A: */var/lib*
- ☐ B: */usr/src*
- ☐ C: */lib/modules*
- ☐ D: */boot*
- ☐ E: */etc*

Frage 2:

Sie haben auf einem Computer einen angepassten Kernel kompiliert. Viele Funktionen haben Sie bei der Konfiguration für den statischen Teil des Kernels angegeben. Der Kernel ist entsprechend groß. Welche Datei kopieren Sie in das Verzeichnis */boot*?

- ☐ A: *bzImage*
- ☐ B: *zImage*
- ☐ C: *image.bz*
- ☐ D: *image.gz*
- ☐ E: *kernel.st*

Frage 3:

Sie müssen die Version des laufenden Kernels auf einem System identifizieren. Welches Kommando zeigt ausschließlich die gewünschte Information?

- ☐ A: `uname -a`
- ☐ B: `ls /lib/modules`
- ☐ C: `uname -r`
- ☐ D: `dmesg`
- ☐ E: `cat /proc/cpuinfo`

Frage 4:

Sie beabsichtigen, einen angepassten Kernel für ein System zu konfigurieren. Die Konfiguration soll auf der Konsole stattfinden, aber Sie möchten dennoch eine Menüführung verwenden. Welches Kommando geben Sie ein?

- ☐ A: `make config`
- ☐ B: `make xconfig`
- ☐ C: `make gconfig`
- ☐ D: `make oldconfig`
- ☐ E: `make menuconfig`

Frage 5:

Sie haben die Quellen eines neuen Kernels aus dem Internet heruntergeladen und wollen zunächst die Konfiguration des bereits laufenden Kernels in den neuen Kernel übernehmen. Auf diese Art soll der Konfigurationsaufwand minimiert werden. Welches der folgenden Kommandos werden Sie verwenden?

- ☐ A: `make config`
- ☐ B: `make xconfig`
- ☐ C: `make cloneconfig`
- ☐ D: `make oldconfig`
- ☐ E: `make menuconfig`

Frage 6:

Eine neue Grafikkarte entfaltet nicht ihre volle Leistung. Sie haben von der Herstellerseite ein Softwarepaket heruntergeladen, das ein zusätzliches Kernelmodul zum laufenden Kernel hinzufügen soll. Welche Technologie muss der laufende Kernel unterstützen, damit das Modul installiert werden kann, ohne den gesamten Kernel zu kompilieren?

- ☐ A: *DKMS*
- ☐ B: *dracut*
- ☐ C: *systemd*
- ☐ D: */lib/modules*
- ☐ E: */var/lib/modules*

Frage 7:

Sie planen die Konfiguration eines angepassten Kernels. Sie haben die Quellpakete bereits beschafft und extrahiert. Welchen obligatorischen Softlink werden Sie erstellen, bevor Sie mit der Arbeit beginnen?

- ☐ A: */etc/src*
- ☐ B: *~/src/linux*
- ☐ C: */urs/src/linux*
- ☐ D: */usr/src/linux-headers-3.0.0-12*
- ☐ E: */lib/modules/3.0.0-12-generic*

Frage 8:

Welche beiden Programme werden wahlweise verwendet, wenn eine initiale RAM-Disk erstellt oder aktualisiert werden soll? (Wählen Sie drei Antworten.)

- ☐ A: `make`
- ☐ B: `mkinitrd`
- ☐ C: `dracut`
- ☐ D: `mkinitramfs`
- ☐ E: `init`

Frage 9:

Sie müssen die Geräteparameter des zweiten *NVMe*-Laufwerks eines Computers auslesen. Welches Kommando können Sie hierfür verwenden?

- ☐ A: `hdparm /dev/nvme2`
- ☐ B: `tune2fs /dev/nvme1`
- ☐ C: `hdparm /dev/nvme1`
- ☐ D: `tune2fs /dev/nvme2`
- ☐ E: `fsck /dev/nvme1`

Frage 10:

In welcher Konfigurationsdatei findet GRUB die Zuordnung seiner Bezeichnungen für Festplatten und Partitionen (z. B. `hd0,0`) zu den Gerätedateien von Linux?

- ☐ A: *mtab*
- ☐ B: *fstab*
- ☐ C: *device.map*
- ☐ D: *menu.lst*
- ☐ E: *grub.conf*

Frage 11:

Sie haben eine neue Kernel-Version in Form eines tar-Balls aus dem Internet heruntergeladen. In welchem Verzeichnis sollten Sie den tar-Ball auspacken, bevor Sie mit der Konfiguration beginnen?

Frage 12:

Sie haben eine neue Kernel-Version aus dem Internet heruntergeladen und in das Verzeichnis */usr/src/linux-3.0.0-15* entpackt. Sie wollen nun den obligatorischen Softlink */usr/src/linux* erstellen. Welches Kommando geben Sie ein?

Frage 13:

Sie planen die Konfiguration eines angepassten Kernels. Welches der folgenden Pakete müssen Sie installieren, um eine menügestützte Konfiguration auf der Konsole durchführen zu können?

- ☐ A: `make`
- ☐ B: `gcc`
- ☐ C: `mc`
- ☐ D: `dracut`
- ☐ E: `libncurses5-dev`

Frage 14:

Sie haben einen neu erstellten Kernel und die zugehörigen Module installiert. Der neue Kernel soll beim nächsten Systemstart verfügbar sein. Welches Kommando müssen Sie verwenden, wenn der Bootloader GRUB (Version 0.97) installiert ist?

Frage 15:

Mit welchen der folgenden Kommandos können Sie die geladenen Module eines laufenden Kernels anzeigen? (Wählen Sie zwei Antworten.)

- ☐ A: `lsmod`
- ☐ B: `modprobe -l`
- ☐ C: `insmod`
- ☐ D: `cat /proc/modules`
- ☐ E: `cat /etc/modules`

Frage 16:

Sie haben auf einem EFI-basierten Computer eine Neuinstallation von Linux durchgeführt. Leider sind im Bootmenü von UEFI weiterhin die zuvor installierten Betriebssysteme auswählbar. Welches Programm können Sie verwenden, um die verwaisten Einträge zu entfernen?

- ☐ A: `efivar`
- ☐ B: `efibootmgr`
- ☐ C: `fdisk`
- ☐ D: `vi`
- ☐ E: `update-grub`

Frage 17:

Sie müssen ein Modul in den laufenden Kernel laden. Module, von denen dieses Modul abhängt, sollen automatisch ebenfalls geladen werden. Welches Programm werden Sie verwenden?

- ☐ A: `modprobe`
- ☐ B: `insmod`
- ☐ C: `rmmod`
- ☐ D: `init`
- ☐ E: `start`

Frage 18:

Sie müssen die Datei *modules.dep* neu erstellen. Sicherheitshalber wollen Sie vorher einen Trockenlauf durchführen, um eventuelle Probleme im Vorfeld zu erkennen. Welches Kommando werden Sie ausführen?

- ☐ A: `modprobe -a`
- ☐ B: `depmod -A`
- ☐ C: `depmod -n`
- ☐ D: `insmod > modules.dep`
- ☐ E: `lsmod > modules.dep`

Frage 19:

Wie viele »echte« Partitionen können Sie aufgrund der Beschränkung einer Partitionstabelle auf einer Festplatte anlegen? (Geben Sie die Anzahl numerisch an.)

Frage 20:

Sie müssen die Datei *modules.dep* neu erstellen. Welches Programm werden Sie ausführen? (Geben Sie nur das Programm ohne Optionen an.)

Frage 21:

Welcher Daemon steht bei einem SysVinit-basierten System in der Hierarchie am höchsten und welche Prozess-ID verwendet dieser Daemon?

- ☐ A: `initd` mit PID 0
- ☐ B: `init` mit PID 1
- ☐ C: `init` mit PID 0
- ☐ D: `initd` mit PID 1
- ☐ E: `init` mit PID 1000

Frage 22:

Sie benötigen eine möglichst detaillierte Aufstellung über die in einem Computer verwendeten PCI-Geräte. Welches ist das beste Kommando, wenn Sie möglichst genaue Informationen über einzelne Geräte benötigen?

- ☐ A: `lspci`
- ☐ B: `pciinfo`
- ☐ C: `ps`
- ☐ D: `rpm`
- ☐ E: `lspci -vvv`

Frage 23:

Sie müssen herausfinden, von welchen Bibliotheken ein Programm abhängig ist, bevor Sie es zum ersten Mal starten. Welches Programm können Sie hierfür verwenden?

- ☐ A: `ldconfig`
- ☐ B: `ldd`
- ☐ C: `ltrace`
- ☐ D: `strace`
- ☐ E: `ld.so.conf`

Frage 24:

Sie haben einem System neue Bibliotheken hinzugefügt. Welches Programm werden Sie anschließend ausführen, damit die Bibliotheken verwendbar werden?

Frage 25:

Sie müssen auf einem SysVinit-basierten Linux-Computer die `init`-Skripte überprüfen. In welchen Verzeichnissen werden Sie nachsehen? (Wählen Sie zwei Antworten.)

- ☐ A: */etc/init*
- ☐ B: */etc/init.d*
- ☐ C: */etc/rc.d*
- ☐ D: */lib/modules*
- ☐ E: */boot*

Frage 26:

Sie verwenden das Kommando `runlevel`. Das Ergebnis des Kommandos sehen Sie hier:

```
archangel:~ # runlevel
1 3
```

Welche Rückschlüsse können Sie aus der Ausgabe des Kommandos ziehen? (Wählen Sie zwei Antworten.)

- ☐ A: Der Computer befindet sich in Runlevel 1.
- ☐ B: Der Computer befindet sich in Runlevel 3.
- ☐ C: Ein Neustart steht aus.
- ☐ D: Der Computer befand sich zuvor in Runlevel 3.
- ☐ E: Der Computer befand sich zuvor in Runlevel 1.

Frage 27:

Sie wollen einen SysVinit-basierten Computer neu starten. Während des Neustarts soll eine Überprüfung der Dateisysteme erzwungen werden. Welches Kommando können Sie hierfür verwenden?

- ☐ A: `shutdown -r now`
- ☐ B: `init 6`
- ☐ C: `init 0`
- ☐ D: `shutdown -rf now`
- ☐ E: `shutdown -rF now`

Frage 28:

Sie müssen sich über einige Standards belesen, die für alle Linux-Distributionen gelten sollten. Welche Webseiten sollten Sie hier besuchen? (Wählen Sie zwei Antworten.)

- ☐ A: *http://www.linuxbase.org*
- ☐ B: *http://www.linuxfoundation.org*
- ☐ C: *http://www.lpi.org*
- ☐ D: *http://www.kernel.org*
- ☐ E: *http://www.wikipedia.de*

Frage 29:

Sie vermuten ein Problem mit einer Festplatte. Bevor Sie die Dateisysteme einer gründlichen Überprüfung unterziehen, wollen Sie sehen, ob es in letzter Zeit Kernel-Meldungen zu dieser Festplatte gab. Welches Kommando werden Sie ausführen?

- ☐ A: `fsck -t ext4 /dev/sda1`
- ☐ B: `fdisk /dev/sda`
- ☐ C: `cfdisk /dev/sda`
- ☐ D: `dmesg|tail`
- ☐ E: `fsck -V /dev/sda1`

Frage 30:

Zu Überprüfungszwecken müssen Sie ein Dateisystem aushängen. Sie erhalten aber eine Fehlermeldung, die besagt, dass die Ressource beschäftigt sei. Welches Programm können Sie verwenden, um zu ermitteln, welche Programme bzw. welche Benutzer auf das Dateisystem zugreifen?

- ☐ A: `df`
- ☐ B: `free`
- ☐ C: `lsof`
- ☐ D: `dmesg`
- ☐ E: `du`

Frage 31:

Sie haben Änderungen an der Datei */etc/inittab* vorgenommen. Die Änderungen müssen sofort übernommen werden, ohne die Verfügbarkeit des Systems einzuschränken. Welches Kommando werden Sie verwenden?

- ☐ A: `init 6`
- ☐ B: `telinit-q`
- ☐ C: `reboot`
- ☐ D: `init -q`
- ☐ E: `init 0`

Frage 32:

Sie finden auf einem Computer ein beschädigtes Dateisystem vor. Welches Programm werden Sie für die Reparatur verwenden? (Geben Sie nur das Programm ohne Pfadangabe, Optionen oder Parameter, an.)

Frage 33:

Sie stellen fest, dass eine an ein System angeschlossene Festplatte plötzlich nicht mehr beschreibbar ist. Alle Partitionen sind nur noch lesbar. Sie müssen die zweite Partition der Festplatte sofort beschreibbar machen. Wie gehen Sie vor?

- ☐ A: `mount -o remount,rw /dev/sdb2`
- ☐ B: `mount -o remount,rw /dev/sdb1`
- ☐ C: `mount /dev/sdb1 /temp`
- ☐ D: `umask 022`
- ☐ E: `chmod 777 / -R`

Frage 34:

Sie starten ein System mit der GRUB-Shell. Sie nehmen Änderungen an den Kernel-Parametern eines Eintrags vor und starten anschließend das Betriebssystem. Nach einem weiteren Reboot stellen Sie fest, dass die Änderungen, die Sie in der GRUB-Shell vorgenommen haben, nicht mehr angewendet werden. Wie machen Sie die Änderungen dauerhaft?

- ☐ A: Schließen Sie Änderungen in der GRUB-Shell mit `Strg-X` ab.
- ☐ B: Schließen Sie Änderungen in der GRUB-Shell mit `:x` ab.
- ☐ C: Schließen Sie Änderungen in der GRUB-Shell mit `ZZ` ab.
- ☐ D: Editieren Sie die Datei */etc/grub.cfg*.
- ☐ E: Editieren Sie die Datei */boot/grub/grub.cfg*.

Frage 35:

Die dritte Partition einer Festplatte weist etliche Fehler auf. Sie wollen, dass die Fehler behoben werden, ohne bei jeder einzelnen Reparatur eine Bestätigung geben zu müssen. Welche Kommandos können Sie verwenden? (Wählen Sie zwei.)

- ☐ A: `fsck -a /dev/sdb2`
- ☐ B: `fsck -y /dev/sdb2`
- ☐ C: `fsck -a /dev/sdb3`
- ☐ D: `fsck -y /dev/sdb3`
- ☐ E: `cfdisk`

Frage 36:

Sie müssen ein ext4-Dateisystem überprüfen. Welche der folgenden Kommandos können Sie hierfür verwenden? (Wählen Sie zwei Antworten.)

- ☐ A: `ext4.mkfs /dev/sdb1`
- ☐ B: `mkfs -t ext4 /dev/sdb1`
- ☐ C: `fsck -t ext4 /dev/sdb1`
- ☐ D: `mkfs.ext4 /dev/sdb1`
- ☐ E: `fsck.ext4 /dev/sdb1`

Frage 37:

Sie wollen festlegen, dass die Partition */dev/sdb1* einer USB-Festplatte bei jedem Systemstart automatisch mit dem Verzeichnis */backup* verbunden wird. Welche Datei werden Sie bearbeiten?

- ☐ A: */etc/mtab*
- ☐ B: */proc/mounts*
- ☐ C: */dev/sdb1*
- ☐ D: */etc/fstab*
- ☐ E: */backup*

Frage 38:

Ein System lässt sich nicht herunterfahren. Bevor Sie das Gerät hart ausschalten, wollen Sie, dass der Inhalt des Festplattencaches auf die jeweiligen Datenträger geschrieben wird. Welches Kommando (ohne Optionen) werden Sie verwenden?

Frage 39:

Sie müssen überprüfen, welche Volumen und Partitionen eines Systems derzeit eingehängt sind. Mit welchen der folgenden Kommandos bekommen Sie die benötigten Informationen? (Wählen Sie drei Antworten.)

- ☐ A: `mount`
- ☐ B: `cat /proc/mounts`
- ☐ C: `sync`
- ☐ D: `cat /etc/fstab`
- ☐ E: `cat /etc/mtab`

Frage 40:

Welche der folgenden Einträge einer *fstab*-Datei sind nicht gültig? (Wählen Sie zwei Antworten.)

- ☐ A: `/dev/sda1 / ext3 defaults 0 0`
- ☐ B: `/dev/sdb1 /backup ext4 noauto 1 1`
- ☐ C: `proc /proc proc defaults 0 0`
- ☐ D: `sda1 / ext2 defaults 0 0`
- ☐ E: `/proc/sda1 / ext3 defaults 0 0`

Frage 41:

Sie arbeiten an einem System, dessen */etc/fstab*-Datei u. a. den folgenden Eintrag enthält:

```
/dev/sda3    /speicher    ext4    defaults,auto    0  0
```

Welche der folgenden Kommandos würden */dev/sda3* mit dem Verzeichnis */speicher* verbinden? (Wählen Sie drei Antworten.)

- ☐ A: `mount /dev/sda3`
- ☐ B: `mount /speicher /dev/sda3 -t ext4`
- ☐ C: `mount /speicher`
- ☐ D: `umount /dev/sda3 /speicher`
- ☐ E: `mount -a`

Frage 42:

Sie erstellen einen neuen Eintrag in der Datei */etc/fstab*. Sie müssen sicherstellen, dass das neu eingetragene Dateisystem von beliebigen Benutzern eingehängt werden kann. Gleichzeitig muss garantiert werden, dass dieses Dateisystem nur von demselben Benutzer oder durch root wieder ausgehängt werden kann. Welche Option werden Sie dem Eintrag hinzufügen?

- ☐ A: `ro`
- ☐ B: `nouser`
- ☐ C: `noauto`
- ☐ D: `users`
- ☐ E: `user`

Frage 43:

Einige Einträge in der Datei */etc/fstab* sind mit der Option `defaults` konfiguriert. Welche der folgenden Optionen wird hierdurch nicht abgedeckt?

- ☐ A: `ro`
- ☐ B: `rw`
- ☐ C: `auto`
- ☐ D: `exec`
- ☐ E: `nouser`

Frage 44:

Sie wollen alle in der Datei */etc/fstab* definierten Swap-Dateisysteme aktivieren. Welches Kommando werden Sie verwenden? Geben Sie das Kommando und, wenn erforderlich, die Optionen in Kurzschreibweise an.

Frage 45:

Sie wollen den Automounter `autofs` verwenden, um Verbindungen zu Windows-Freigaben herzustellen. Welche der folgenden ist die Hauptkonfigurationsdatei von `autofs`?

- ☐ A: */etc/fstab*
- ☐ B: */etc/auto.master*
- ☐ C: */etc/mtab*
- ☐ D: */etc/auto.mist*
- ☐ E: */etc/auto.net*

Frage 46:

Sie verwenden ein Notebook, das täglich mehrmals heruntergefahren bzw. gestartet wird. Es passiert Ihnen deshalb häufig, dass während des Startvorgangs eine Dateisystemprüfung stattfindet. Sie wollen nun die Anzahl der Mount-Vorgänge zwischen den Überprüfungen erhöhen. Welches Programm können Sie hierfür verwenden?

- ☐ A: `mkisofs`
- ☐ B: `fsck`
- ☐ C: `debugfs`
- ☐ D: `tune2fs`
- ☐ E: `hdparm`

Frage 47:

Sie müssen die geladenen Module eines Kernels auflisten. Welches der folgenden Kommandos können Sie verwenden?

- ☐ A: `modprobe -l`
- ☐ B: `modprobe -r`
- ☐ C: `insmod`
- ☐ D: `lsmod`
- ☐ E: `rmmod`

Frage 48:

Ein Benutzer hat versehentlich eine wichtige Datei gelöscht. Leider gibt es keine Sicherung, in der diese Datei enthalten ist. Welches Tool können Sie verwenden, um die Datei wiederherzustellen?

- ☐ A: `mkfs`
- ☐ B: `tune2fs`
- ☐ C: `debugfs`
- ☐ D: `restore`
- ☐ E: `cp`

Frage 49:

Welches der folgenden Kommandos ist geeignet, den Master Boot Record einer Festplatte in eine Datei zu sichern?

- ☐ A: `dd if=/dev/hda of=mbr.backup count=1`
- ☐ B: `cp MBR mbr.backup`
- ☐ C: `dd if=/dev/hda1 of=mbr.backup ibs=512 count=1`
- ☐ D: `dd if=/dev/hda of=mbr.backup ibs=512 count=1`
- ☐ E: `dd if=mbr.backup of=/dev/hda ibs=512 count=1`

Frage 50:

Sie haben eine neue Festplatte an ein älteres System angeschlossen. Die Festplatte verwendet die Gerätedatei */dev/sdd*. Da Sie nicht sicher sind, welche Software auf dem Computer verfügbar ist, wollen Sie ein möglichst einfaches Programm zur Partitionierung der neuen Festplatte verwenden. Welches Kommando werden Sie zunächst eingeben?

Frage 51:

Sie haben `fdisk` verwendet, um eine Swap-Partition zu erstellen. Die Gerätedatei der Swap-Partition ist */dev/sdb1*. Welches Kommando werden Sie als nächstes verwenden?

- ☐ A: `swapon -v /dev/sdb1`
- ☐ B: `swapoff -v /dev/sdb1`
- ☐ C: `swapon -a`
- ☐ D: `vi /etc/fstab`
- ☐ E: `mkswap /dev/sdb1`

Frage 52:

Sie wollen den Inhalt des Verzeichnisses */home* eines Servers auf eine DVD brennen. Eine grafische Oberfläche steht auf dem System nicht zur Verfügung. Welches Programm müssen Sie zunächst verwenden, um eine ISO-9660-konforme Datei zu erhalten?

- ☐ A: `dd`
- ☐ B: `mkisofs`
- ☐ C: `cdrecord`
- ☐ D: `mkfs.ntfs`
- ☐ E: `fdisk`

Frage 53:

Zu Archivierungszwecken wollen Sie aus dem Inhalt einer CD eine ISO-Datei generieren. Die Gerätedatei des CD-Laufwerks ist */dev/hdc*. Welches der folgenden Kommandos erstellt die ISO-Datei?

- ☐ A: `cp /dev/hdc cdrom.iso`
- ☐ B: `cdrecord /dev/hdc`
- ☐ C: `dd if=cdrom.iso of=/dev/hdc`
- ☐ D: `dd if=/dev/hdc of=cdrom.iso`
- ☐ E: `cdrecord dev=/dev/hdc`

Frage 54:

Welche der folgenden Programme ermitteln Hardware, die an einen Computer angeschlossen (oder auch intern eingebaut) wurde? (Wählen Sie drei Antworten.)

- ☐ A: `lsusb`
- ☐ B: `lspci`
- ☐ C: `lsdev`
- ☐ D: `lsof`
- ☐ E: `ltrace`

Frage 55:

Sie ersetzen bei einem Debian-System eine defekte Netzwerkkarte, die den Alias `eth0` trägt. Nachdem Sie den Computer neu gestartet haben, stellen Sie fest, dass die neue Netzwerkkarte den Namen `eth1` verwendet. Um unnötigen Konfigurationsaufwand in bestehenden Skripten zu vermeiden, wollen Sie die Netzwerkkarte in `eth0` umbenennen. Welche Datei werden Sie bearbeiten?

- ☐ A: */etc/network/interfaces*
- ☐ B: */etc/init.d*
- ☐ C: */etc/udev/rules.d/70-persistent-net.rules*
- ☐ D: */etc/sysconfig/network*
- ☐ E: */etc/inittab*

Frage 56:

Sie vermuten Probleme mit der Elektronik einer USB-Festplatte. Um den Fehler einzugrenzen, wollen Sie live sehen, was in dem Moment geschieht, in dem Sie die Festplatte einstecken. Welche Programme können Sie hierfür verwenden? (Wählen Sie zwei Antworten.)

- ☐ A: `lsusb`
- ☐ B: `udevmonitor`
- ☐ C: `udevadm`
- ☐ D: `dmesg`
- ☐ E: `lspci`

Frage 57:

Sie planen die Konfiguration eines Software-RAID. Sie wollen vier Datenträger zu einem RAID-Verbund zusammenschalten, sodass möglichst viel Speicherkapazität bei optimaler Leistungsausnutzung verfügbar wird. Welchen RAID-Level werden Sie konfigurieren? (Antworten Sie z. B. mit RAID 7.)

Frage 58:

Sie planen die Konfiguration eines Software-RAID. Ihnen stehen lediglich zwei Festplatten zur Verfügung und Sie benötigen ein Maximum an Redundanz. Welchen RAID-Level werden Sie konfigurieren? (Antworten Sie z. B. mit RAID 7.)

Frage 59:

Sie müssen ein bestehendes RAID-5-System erweitern. Sie haben bereits eine neue Festplatte in den Computer eingebaut und partitioniert. Mit welchem Kommando fügen Sie die Partition dem RAID-Verbund hinzu?

- ☐ A: `fdisk`
- ☐ B: `cfdisk`
- ☐ C: `mdadm --detail /dev/md0`
- ☐ D: `mdadm --add /dev/md0 /dev/sdd1`
- ☐ E: `sfdisk`

Frage 60:

Sie verwalten einen Server, der mit einem Software-RAID-Array ausgestattet ist. Sie wollen eine Benachrichtigung per E-Mail erhalten, wenn ein Problem mit dem RAID-Array auftritt. In welcher Konfigurationsdatei werden Sie Ihre E-Mail-Adresse hinterlegen?

- ☐ A: */etc/mtab*
- ☐ B: */etc/fstab*
- ☐ C: */etc/mdadm/mdadm.conf*
- ☐ D: */etc/aliases*
- ☐ E: *~/.forward*

Frage 61:

Sie vermuten ein Problem mit einem RAID-Controller. Welches Kommando werden Sie ausführen, um Hinweise auf Probleme zu finden?

- ☐ A: `lspci`
- ☐ B: `dmesg`
- ☐ C: `lsusb`
- ☐ D: `lsmod`
- ☐ E: `uname`

Frage 62:

Während eines Systemstarts haben Sie eine Fehlermeldung gesehen, konnten den genauen Text aber aufgrund der hohen Geschwindigkeit des Bootvorgangs nicht lesen. In welcher Datei können Sie die Fehler nachlesen?

- ☐ A: */etc/syslog.conf*
- ☐ B: */proc/kmsg*
- ☐ C: */var/log/auth.log*
- ☐ D: */boot/grub/boot.img*
- ☐ E: */var/log/messages*

Frage 63:

Sie arbeiten an einem System, auf dem LVM installiert ist. Mit welchem Kommando können Sie feststellen, welche physikalischen Volumen auf dem System vorhanden sind?

- ☐ A: `pvscan -v`
- ☐ B: `vgscan -v`
- ☐ C: `lvscan -v`
- ☐ D: `vgcreate`
- ☐ E: `fdisk`

Frage 64:

Sie arbeiten an einem System, auf dem LVM installiert ist. Sie müssen mehrere physikalische Volumen auf neuen Festplatten erstellen. Welches Kommando kommt hier zum Einsatz?

- ☐ A: `fdisk`
- ☐ B: `vgscan`
- ☐ C: `pvscan`
- ☐ D: `vgcreate`
- ☐ E: `pvcreate`

Frage 65:

Sie arbeiten an einem neuen System, auf dem LVM installiert ist. Sie müssen sich einen Überblick über die vorhandenen Volumengruppen und physikalischen (LVM-) Volumen verschaffen. Welches Kommando werden Sie verwenden? (Nennen Sie nur das Kommando ohne Optionen oder Parameter.)

Frage 66:

Welche der folgenden Aussagen sind in Bezug auf LVM zutreffend? (Wählen Sie alle zutreffenden Antworten.)

- ☐ A: LVM wird ausschließlich für RAID verwendet.
- ☐ B: LVM und RAID sind kombinierbare Technologien.
- ☐ C: Volumen, die mit LVM verwaltet werden, sind nicht erweiterbar.
- ☐ D: Mit LVM verwaltete Volumen können um zusätzliche Festplatten ergänzt werden.
- ☐ E: LVM bietet Ausfallsicherheit durch Redundanz.

Frage 67:

Sie arbeiten an einem System, auf dem LVM installiert ist. Eines der logischen Volumen ist fast vollständig mit Daten gefüllt und soll nun mit einer neuen Festplatte erweitert werden. Sie erstellen auf der neuen Festplatte die nötige Partition und initialisieren diese anschließend mit dem Programm `pvcreate`. Welche beiden Tools werden Sie anschließend nacheinander einsetzen? (Wählen Sie zwei Antworten.)

- ☐ A: `lvcreate`
- ☐ B: `vgcreate`
- ☐ C: `lvextend`
- ☐ D: `resize2fs`
- ☐ E: `pvcreate`

Frage 68:

Welche Komponente von Linux abstrahiert für Programme die physikalischen Volumen einer Festplatte, sodass nur noch die logischen (LVM-) Volumen verwendet werden?

- ☐ A: `inetd`
- ☐ B: `xinetd`
- ☐ C: Device Mapper
- ☐ D: `lvcreate`
- ☐ E: `pvcreate`

Frage 69:

Sie überprüfen die Netzwerksicherheit eines Computers und benötigen eine Aufstellung der Hardwareadressen (MAC-Adressen) von Systemen, die kürzlich diesen Computer kontaktiert haben. Welches Kommando können Sie verwenden?

- ☐ A: `arp`
- ☐ B: `arp -d`
- ☐ C: `ifconfig`
- ☐ D: `route -n`
- ☐ E: `ifup`

Frage 70:

Sie müssen die Routing-Tabelle eines Computers überprüfen. Um die Ausgabe des Programms zu beschleunigen, soll hierbei die Namensauflösung der einzelnen Routing-Tabelleneinträge unterdrückt werden. Welches Kommando werden Sie verwenden?

- ☐ A: `route`
- ☐ B: `route -C`
- ☐ C: `route -v`
- ☐ D: `route -n`
- ☐ E: `traceroute`

Frage 71:

Sie müssen die IP-Adresse einer Netzwerkschnittstelle ermitteln. Welche der folgenden Programme sind hierfür am besten geeignet? (Wählen Sie zwei Antworten.)

- ☐ A: `ifconfig`
- ☐ B: `route`
- ☐ C: `arp`
- ☐ D: `ping`
- ☐ E: `ip`

Frage 72:

Sie müssen die Konfiguration der Drahtlosnetzwerkkarte *wlan0* überprüfen. Hierbei soll auch die Sendeleistung ermittelt werden. Welches der folgenden Kommandos werden Sie verwenden?

- ☐ A: `iwconfig wlan0 txpower on`
- ☐ B: `iwconfig wlan0`
- ☐ C: `ifconfig wlan0`
- ☐ D: `iwconfig wlan0 txpower off`
- ☐ E: `iwconfig wlan0 txpower 20mW`

Frage 73:

Sie müssen die Schnittstellenkonfiguration des Netzwerkadapters mit der Bezeichnung `eth1` überprüfen. Parameter anderer Schnittstellen sollen aus Gründen der Übersichtlichkeit nicht angezeigt werden. Geben Sie das für diesen Zweck einfachste Kommando ein.

Frage 74:

Sie müssen einer Routing-Tabelle einen neuen Eintrag hinzufügen. Das Routing-Ziel ist das Netzwerk 172.16.0.0. Der nächste Router auf dem Weg zu diesem Netzwerk verwendet die IP-Adresse 192.168.50.7. Welche der folgenden Kommandos können Sie für diese Aufgabe verwenden? (Wählen Sie zwei Antworten.)

- ☐ A: `route add -net 172.16.0.0 netmask 255.255.0.0 gw 192.168.50.7`
- ☐ B: `route add -net 192.168.50.7/24 gw 172.16.0.0`
- ☐ C: `route add -net 192.168.50.7 netmask 255.255.0.0 gw 172.16.0.0`
- ☐ D: `route add -net 172.16.0.0/24 gw 192.168.50.7`
- ☐ E: `route -n`

Frage 75:

Im Rahmen einer Sicherheitsprüfung müssen Sie auf einem entfernten System feststellen, welche TCP- und UDP-Ports geöffnet sind. Welches Werkzeug können Sie für diese Aufgabe verwenden?

- ☐ A: `iptables`
- ☐ B: `ifconfig`
- ☐ C: `lsof`
- ☐ D: `netstat`
- ☐ E: `nmap`

Frage 76:

Sie müssen eine USB-Festplatte von einem System entfernen. Bei dem Versuch, das Laufwerk auszuhängen, bekommen Sie eine Fehlermeldung, weil einige Dateien auf dem Gerät noch geöffnet sind. Mit welchem Programm können Sie feststellen, welche Dateien geöffnet sind?

Frage 77:

Die Netzwerkschnittstelle `eth0` eines Computers ist mit der IPv4-Adresse 192.168.50.1 konfiguriert. Sie wollen derselben Netzwerkschnittstelle eine weitere IPv4-Adresse hinzufügen. Welche der folgenden Kommandos führen eine solche Konfiguration durch? (Wählen Sie zwei Antworten.)

- ☐ A: `ifconfig eth0:1 192.168.7.5`
- ☐ B: `ifconfig eth0 192.168.7.5`
- ☐ C: `ip address show dev eth0`
- ☐ D: `ip address add 192.168.7.5/24 dev eth0`
- ☐ E: `ifconfig eth1:1 192.168.7.5`

Frage 78:

Sie haben einen neuen Apache Webserver installiert und wollen nun einen groben Funktionstest durchführen. Welche der folgenden Kommandos können Sie auf dem Host ausführen, um zu sehen, ob der Webserver bereit ist? (Wählen Sie zwei Antworten.)

- ☐ A: `netstat -an| grep :8080`
- ☐ B: `netstat -an| grep :443`
- ☐ C: `netstat -an| grep :80`
- ☐ D: `nmap localhost -p 443`
- ☐ E: `nmap localhost -p 80`

Frage 79:

Sie müssen den Netzwerkverkehr zu einem System überwachen. Zu diesem Zweck wollen Sie ein Programm verwenden, das eine grafische Ausgabe der erfassten Pakete ermöglicht. Welches Programm werden Sie verwenden?

- ☐ A: `wireshark`
- ☐ B: `netcat`
- ☐ C: `netstat`
- ☐ D: `nmap`
- ☐ E: `ifconfig`

Frage 80:

Sie wollen die Routing-Tabelle eines Computers einsehen, der als Netzwerk-Router fungiert. Um die Ausgabe möglichst schnell zu erhalten, sollen die Einträge der Routing-Tabelle nicht mittels DNS in Hostnamen aufgelöst werden. Welches Kommando werden Sie ausführen?

Frage 81:

Sie planen die Installation einer Netzwerkanwendung, die den TCP-Port 5112 verwenden wird. Vorher müssen Sie überprüfen, ob die Kommunikation über diesen Port in Ihrem Netzwerk eventuell durch eine Firewall blockiert wird. Welches Programm können Sie zu Hilfe nehmen, bis die Anwendung tatsächlich installiert ist?

- ☐ A: `nmap`
- ☐ B: `netstat`
- ☐ C: `wireshark`
- ☐ D: `netcat`
- ☐ E: `ifconfig`

Frage 82:

Sie wollen auf eine möglichst einfache Art feststellen, ob ein entfernter Computer eingeschaltet und mit dem Netzwerk verbunden ist. Bei der Überprüfung soll das Protokoll ICMP genutzt werden. Welches Programm setzen Sie ein?

Frage 83:

Sie benötigen genaue Informationen über den in einem Computer verwendeten Hauptprozessor. Wo erhalten Sie diese Informationen?

- ☐ A: */proc/cpuinfo*
- ☐ B: */proc/cpu*
- ☐ C: */var/log/messages*
- ☐ D: `lspci`
- ☐ E: `lsusb`

Frage 84:

Welche der folgenden Konfigurationsdateien dienen üblicherweise der Konfiguration des Namens für einen Host? (Wählen Sie zwei Antworten.)

- ☐ A: */etc/HOSTNAME*
- ☐ B: */etc/hosts*
- ☐ C: */etc/hostname*
- ☐ D: */etc/hosts.allow*
- ☐ E: */bin/hostname*

Frage 85:

Sie vermuten ein Problem mit einem USB-Controller und wollen deshalb die letzten 20 Zeilen des Kernel Ring Buffers einsehen. Welches Kommando werden Sie verwenden?

- ☐ A: `dmesg| tail`
- ☐ B: `dmesg| tail -n 20`
- ☐ C: `tail -n 20 /var/log/syslog`
- ☐ D: `tail -n 20 /var/log/messages`
- ☐ E: `dmesg`

Frage 86:

Von Ihrer Arbeitsstation aus müssen Sie probehalber regelmäßig einen Router anpingen. Der Router ist nicht in Ihrem DNS-System eingetragen, aber Sie wollen ihn zukünftig mit einem einfachen Namen ansprechen können. In welche lokale Konfigurationsdatei tragen Sie den Namen und die IP-Adresse des Routers ein?

- ☐ A: */etc/hosts.allow*
- ☐ B: */etc/bind/db.127*
- ☐ C: */etc/hostname*
- ☐ D: */etc/hosts*
- ☐ E: */etc/hosts.deny*

Frage 87:

Sie wollen ein System zur Verwendung eines bestimmten DNS-Servers konfigurieren. In welche der folgenden Konfigurationsdateien sollten Sie den DNS-Server eintragen?

- ☐ A: */etc/named.conf*
- ☐ B: */etc/hosts*
- ☐ C: */etc/resolv.conf*
- ☐ D: */etc/bind/named.conf*
- ☐ E: */etc/dnsclient.conf*

Frage 88:

Sie müssen überprüfen, welche Router ein TCP-Datenpaket durchläuft, wenn Sie den Server *deep.thought.adams.com* kontaktieren. Welches Kommando werden Sie verwenden? (Geben Sie das Kommando samt Parametern und ggf. Optionen an.)

Frage 89:

Sie konfigurieren die Sicherheitseinstellungen eines Systems, auf dem TCP-Wrapper verwendet werden. Es soll verhindert werden, dass überhaupt vom Netzwerk aus auf diesen Host zugegriffen werden kann. Welche Einträge werden Sie in welchen Konfigurationsdateien vornehmen? (Wählen Sie zwei Antworten. Beide Antworten sind Teil einer Lösung.)

- ☐ A: `ALL: ALL` in */etc/hosts.deny*
- ☐ B: */etc/hosts.allow* entleeren oder löschen
- ☐ C: `ALL: ALL` in */etc/hosts.allow*
- ☐ D: */etc/hosts.deny* entleeren oder löschen
- ☐ E: `DENY: ALL` in */etc/host.conf*

Frage 90:

Welche beiden der folgenden Einträge sind typisch für die Konfigurationsdatei *resolv.conf*? (Wählen Sie zwei Antworten.)

- ☐ A: `domain=lpic-1.de`
- ☐ B: `address 192.168.50.1`
- ☐ C: `dns-nameservers 192.168.50.1`
- ☐ D: `nameserver 192.168.50.1`
- ☐ E: `search lpic-2.de`

Frage 91:

Sie planen einen Server zu Wartungszwecken herunterzufahren. Sie wollen, dass alle Benutzer, die mit dem Server verbunden sind, über die Shell eine Benachrichtigung erhalten. Welche der folgenden Programme können Sie hierfür verwenden? (Wählen Sie zwei Antworten.)

- ☐ A: `issue`
- ☐ B: `shutdown`
- ☐ C: `wall`
- ☐ D: `issue.net`
- ☐ E: `info`

Frage 92:

Sie sollen dafür sorgen, dass alle Benutzer darüber informiert werden, dass die private Nutzung des Internets in Ihrem Unternehmen nicht mehr länger geduldet wird. Es soll eine Benachrichtigung erfolgen, sobald die Benutzer sich an ihren Systemen angemeldet haben. Welche Konfigurationsdatei werden Sie auf den Arbeitsstationen der User jeweils bearbeiten?

- ☐ A: *issue*
- ☐ B: *issue.net*
- ☐ C: *info*
- ☐ D: *motd*
- ☐ E: *resolv.conf*

Frage 93:

Sie versuchen, das USB-Medium */dev/sdc1* auszuwerfen, erhalten aber lediglich die folgende Fehlermeldung:

```
umount: /media/usb-disk1: device is busy.
```

Sie müssen den Prozess ermitteln, der auf das Medium zugreift, damit Sie diesen beenden können. Welche Kommandos können Sie hierfür verwenden? (Wählen Sie zwei Antworten.)

- ☐ A: `du /media/usb-disk1`
- ☐ B: `df /dev/sdc1`
- ☐ C: `lsof /media/usb-disk1`
- ☐ D: `lsof /dev/sdc1`
- ☐ E: `umount /dev/sdc1`

Frage 94:

Ein Bekannter bittet Sie, ihm bei einem Netzwerkproblem zu helfen. Die verwendete Linux-Distribution ist Ihnen nicht bekannt. Nach welchen Verzeichnissen werden Sie zunächst suchen, um die IP-Konfiguration zu überprüfen und ggf. anzupassen? (Wählen Sie zwei Antworten.)

- ☐ A: */etc/network/*
- ☐ B: */etc/init.d/*
- ☐ C: */proc/sys/net/ipv4*
- ☐ D: */etc/rc.d/*
- ☐ E: */etc/sysconfig/network-scripts/*

Frage 95:

Sie haben ein Programm in Form eines tar-Balls aus dem Internet heruntergeladen. Welches der folgenden Kommandos wird den tar-Ball entpacken?

- ☐ A: `./configure`
- ☐ B: `tar -xvzf tarball.tar.gz`
- ☐ C: `tar -xvfz tarball.tar.gz`
- ☐ D: `tar -tf tarball.tar.gz`
- ☐ E: `make`

Frage 96:

Auf der Suche nach einer bestimmten Datei müssen Sie den Inhalt der Archivdatei *archiv.tar* auflisten. Welches Kommando werden Sie verwenden? Geben Sie in Ihrer Antwort auch die Optionen und Parameter an.

Frage 97:

Sie haben ein Programm in Form eines tar-Balls aus dem Internet heruntergeladen. In welchem Verzeichnis sollten Sie den tar-Ball nach dem Filesystem Hierarchy Standard entpacken?

- ☐ A: */usr/src*
- ☐ B: */var/src*
- ☐ C: *~/src*
- ☐ D: */etc*
- ☐ E: */lib*

Frage 98:

Sie haben ein Programm in Form eines tar-Balls aus dem Internet heruntergeladen und ausgepackt. Als Nächstes soll das Programm für die Kompilierung vorbereitet werden. Welches Kommando werden Sie ausführen?

- ☐ A: `make`
- ☐ B: `make install`
- ☐ C: `tar -xvzf tarball.tar.gz`
- ☐ D: `CONFIGURE`
- ☐ E: `./configure`

Frage 99:

Welche der folgenden Programme dienen der Komprimierung von Dateien? (Wählen Sie zwei Antworten.)

- ☐ A: `bzip2`
- ☐ B: `tar`
- ☐ C: `cpio`
- ☐ D: `gzip`
- ☐ E: `gunzip`

Frage 100:

Sie haben ein Programm in Form eines tar-Balls aus dem Internet heruntergeladen, ausgepackt, konfiguriert und kompiliert. Jetzt sind Sie bereit, das Programm zu installieren. Welchen Befehl werden Sie eingeben?

Frage 101:

Sie erhalten eine Datei mit der Bezeichnung *archiv.gz*. Welche der folgenden Programme können Sie verwenden, um die Datei zu öffnen? (Wählen Sie drei Antworten.)

- ☐ A: `zcat`
- ☐ B: `bzip2`
- ☐ C: `gunzip`
- ☐ D: `gzip`
- ☐ E: `dpkg`

Frage 102:

Sie wollen die Kernel-Quellen passend zu Ihrem laufenden Kernel aus dem Internet herunterladen. Welches Kommando können Sie verwenden, um die momentan installierte Kernel-Version zu ermitteln? Es sollen keine nicht benötigten Informationen ausgegeben werden.

- ☐ A: `uname -a`
- ☐ B: `uname -s`
- ☐ C: `lsb_release -a`
- ☐ D: `uname -r`
- ☐ E: `lsb_release`

Frage 103:

Welches der folgenden Programme wurde ursprünglich hauptsächlich benutzt, um Backups für eine Bandsicherung vorzubereiten?

- ☐ A: `cpio`
- ☐ B: `tar`
- ☐ C: `dd`
- ☐ D: `rsync`
- ☐ E: `mt`

Frage 104:

Ein USB-Speichermedium verwendet die Gerätedatei */dev/sdb*. Sie müssen das komplette Dateisystem des Mediums inklusive Startumgebung (MBR) in die Datei *usbstick.img* sichern und wollen das Programm `dd` verwenden, um die Datensicherung im aktuellen Verzeichnis durchzuführen. Wie wird das komplette Kommando lauten? (Verwenden Sie nur die nötigsten Optionen und Parameter.)

Frage 105:

Welche der folgenden Verzeichnisse sollten bei einer Datensicherung unbedingt berücksichtigt werden? (Wählen Sie zwei Antworten.)

- ☐ A: */proc*
- ☐ B: */bin*
- ☐ C: */home*
- ☐ D: */var*
- ☐ E: */usr*

Frage 106:

Sie planen eine Datensicherung. Welche der folgenden Verzeichnisse sind bei einer regelmäßigen Sicherung am wenigsten wichtig? (Wählen Sie zwei Antworten.)

- ☐ A: */home*
- ☐ B: */var*
- ☐ C: */proc*
- ☐ D: */etc*
- ☐ E: */lib*

Frage 107:

Eine an ein System angeschlossene USB-Festplatte weist häufig Lesefehler auf. Das Ergebnis sind lange Wartezeiten beim Zugriff auf Dateien. Sie müssen herausfinden, ob es kernelbezogene Meldungen zu diesem Problem gibt. Welche Möglichkeiten stehen Ihnen zur Verfügung? (Wählen Sie zwei Antworten.)

- ☐ A: Geben Sie `dmesg` ein und suchen Sie nach Fehlern bezüglich USB und Dateisystem.
- ☐ B: Sichern Sie die Festplatte und führen Sie eine Neuformatierung durch. Stellen Sie die Daten anschließend wieder her.
- ☐ C: Durchsuchen Sie */proc/sys/kernel* nach Fehlermeldungen.
- ☐ D: Durchsuchen Sie */var/log/kernel* nach Fehlermeldungen.
- ☐ E: Suchen Sie nach `kernel` in der Datei */var/log/messages*.

Frage 108:

Sie müssen ein Modul zum laufenden Kernel hinzufügen. Sollte dieses Modul von anderen Modulen abhängen, dann sollen diese automatisch ebenfalls installiert werden. Welches Kommando können Sie hierfür verwenden?

- ☐ A: `lsmod`
- ☐ B: `insmod`
- ☐ C: `modprobe`
- ☐ D: `rmmod`
- ☐ E: `insmod /auto`

Frage 109:

Sie versuchen, ein Binärprogramm auf lesbare Zeichenketten hin zu durchsuchen, indem Sie den Pager `less` verwenden. Leider erhalten Sie keine lesbare Ausgabe. Welches Programm sollten Sie stattdessen verwenden?

- ☐ A: `vi`
- ☐ B: `strace`
- ☐ C: `strings`
- ☐ D: `ltrace`
- ☐ E: `lsof`

Frage 110:

Sie konfigurieren einen Router und wollen nun dauerhaft festlegen, dass der Computer IP-Pakete weiterleitet und ICMP-Anfragen ignoriert. Welche Konfigurationsdatei ist für diese Aufgabe vorgesehen?

- ☐ A: */etc/sysctl.conf*
- ☐ B: */etc/network/interfaces*
- ☐ C: */proc/sys/net/ipv4/ip_forward*
- ☐ D: */proc/sys/net/ipv4/icmp_echo_ignore_all*
- ☐ E: */etc/hosts.deny*

Frage 111:

Welche der folgenden Kommandos können Sie verwenden, damit ein Computer IP-Pakete weiterleitet? (Wählen Sie zwei Antworten.)

- ☐ A: `echo 1 > /proc/sys/net/ipv4/ip_forward`
- ☐ B: `sysctl -w net.ipv4.ip_forward=1`
- ☐ C: `echo 0 > /proc/sys/net/ipv4/ip_forward`
- ☐ D: `sysctl -w net.ipv4.ip_forward=0`
- ☐ E: `sysctl -n net.ipv4.ip_forward`

Frage 112:

Sie müssen einen Computer so konfigurieren, dass er auf ICMP-Echo-Anforderungen nicht mehr antwortet. Welche Konfigurationsdatei werden Sie modifizieren? (Geben Sie die Datei mitsamt Pfad an.)

Frage 113:

Sie bemerken einen Leistungsengpass auf einem Ihrer Systeme und wollen zunächst die Prozessorauslastung überprüfen. Welche der folgenden Programme helfen Ihnen hier weiter? (Wählen Sie drei Antworten.)

- ☐ A: `iostat`
- ☐ B: `ps`
- ☐ C: `top`
- ☐ D: `sar`
- ☐ E: `free`

Frage 114:

Sie wollen die aktuelle Prozessorbelastung eines Systems beobachten. Es sollen fünf Messwerte in einem Abstand von drei Sekunden angezeigt werden, so dass Sie diese miteinander vergleichen können. Welche der folgenden Programme können Sie hierfür verwenden? (Wählen Sie drei Antworten.)

- ☐ A: `vmstat`
- ☐ B: `ps`
- ☐ C: `top`
- ☐ D: `iostat`
- ☐ E: `sar`

Frage 115:

Sie verwenden ein Programm, das aufgrund eines Fehlers häufig abstürzt. Mit welchem der folgenden Kommandos können Sie nachsehen, ob das Programm zu einem Zombieprozess geworden ist?

- ☐ A: `sar`
- ☐ B: `iostat`
- ☐ C: `top`
- ☐ D: `vmstat`
- ☐ E: `ps -A`

Frage 116:

Sie vermuten, dass einer Ihrer Kollegen einen Server neu gestartet hat, ohne Sie darüber in Kenntnis zu setzen. Welche der folgenden Programme zeigen an, wie lange der Server bereits ohne Neustart läuft? (Wählen Sie drei Antworten.)

- ☐ A: `top`
- ☐ B: `when`
- ☐ C: `whatis`
- ☐ D: `uptime`
- ☐ E: `w`

Frage 117:

Beim Versuch, eine Partition auszuhängen, bekommen Sie eine Fehlermeldung, die besagt, dass auf dem fraglichen Dateisystem noch Dateien geöffnet sind. Mit welchem Programm bekommen Sie heraus, wer diese Dateien geöffnet hält?

Frage 118:

Bei welchen der folgenden Softwareprodukte handelt es sich um Monitoring-Lösungen, die für ihren Betrieb PHP und MySQL benötigen?

- ☐ A: collectd
- ☐ B: Cacti
- ☐ C: Nagios
- ☐ D: MRTG
- ☐ E: `vmstat`

Frage 119:

Zur Inbetriebnahmen eines Chipkarten-Lesegeräts muss der Dienst *pcscd* gestartet werden. Welchen Befehl werden Sie auf einem systemd-basierten System eingeben? (Geben Sie ggf. Optionen und Parameter mit an.)

Frage 120:

Welche der folgenden Programme erstellen eine initiale RAM-Disk?

- ☐ A: `update-grub`
- ☐ B: `mkinitrd`
- ☐ C: `lilo`
- ☐ D: `mkinitramfs`
- ☐ E: `ldconfig`

Antworten und Erklärungen zu den Prüfungsfragen

Hier finden Sie die Erläuterungen zu allen Fragen des ersten Teils. Sie sollten unbedingt auch die Kommentare zu den falschen Antworten lesen. Einige Fakten werden hier nicht zufällig mehrfach erwähnt, sondern weil wesentliche Prüfungsinhalte auf diese Weise besser in Ihrem Gedächtnis haften bleiben.

Frage 1:

B: */usr/src* ist das Verzeichnis, in dem üblicherweise die Kernel-Quellen gespeichert und verarbeitet werden.

zu A: */var/lib* beinhaltet diverse Programmbibliotheken.

zu C: */lib/modules* enthält die Unterverzeichnisse mit den jeweiligen Kernel-Modulen der installierten Kernel.

zu D: */boot* enthält u. a. den statischen Teil des fertigen Kernels.

zu E: */etc* enthält hauptsächlich Konfigurationsdateien.

Frage 2:

A: *bzImage* ist der Name des statischen Kernels, wenn dieser größer ist als 512kB. Das »b« steht hierbei für »big«.

zu B: *zImage* ist der Dateiname eines nach heutigen Maßstäben sehr kleinen Kernels (weniger als 512 kB).

zu C, D und E: *image.bz*, *image.gz* und *kernel.st* sind frei erfundene Dateinamen.

Frage 3:

C: `uname -r` zeigt das Release des laufenden Kernels an.

zu A: `uname -a` zeigt alle mittels `uname` abrufbaren Systeminformationen an. Es soll aber lediglich die Kernel-Version ermittelt werden.

zu B: `ls /lib/modules` listet das Verzeichnis auf, das die Kernel-Module enthält.

zu D: `dmesg` zeigt den Inhalt des Kernel Ring Buffers an.

zu E: `cat /proc/cpuinfo` zeigt die in der Pseudodatei *cpuinfo* enthaltenen Informationen über den Prozessor an.

Frage 4:

E: `make menuconfig` dient einer menügeführten Konfiguration des Kernels in einer Konsolensitzung.

zu A: `make config` fragt nacheinander alle (!) konfigurierbaren Kernel-Module ab. Sie können dann auswählen, ob eine Komponente statisch, modular oder gar nicht in den Kernel aufgenommen werden soll.

zu B: `make xconfig` dient der Kernel-Konfiguration in einer X Window-Umgebung.

zu C: `make gconfig` dient der Kernel-Konfiguration unter GNOME.

zu D: `make oldconfig` migriert die Konfiguration eines »alten« Kernels in eine neue Kernel-Konfiguration.

Frage 5:

D: `make oldconfig` migriert die Konfiguration eines »alten« Kernels in eine neue Kernel-Konfiguration.

zu A: `make config` siehe Frage 4.

zu B: `make xconfig` siehe Frage 4.

zu C: `make cloneconfig` ist eine Variante von `oldconfig`. Hierbei wird die Pseudodatei */proc/config.gz* als Grundlage verwendet, falls vorhanden.

zu E: `make menuconfig` siehe Frage 4.

Frage 6:

A: DKMS steht für Dynamic Kernel Module Support. Diese Technologie ist genau für diesen Zweck gedacht.

zu B: *dracut* wird zur Erstellung von initialen RAM-Disks verwendet.

zu C: *systemd* ist inzwischen für Vieles zuständig aber hierfür nicht.

zu D: */lib/modules* enthält zu jeder installierten Kernel-Version ein Unterverzeichnis mit den jeweiligen Kernel-Modulen.

zu E: */var/lib/modules* gibt es nicht.

Frage 7:

C: */urs/src/linux* wird typischerweise als Softlink erstellt, um den Zugriff auf den aktuell zu bearbeitenden Kernel zu erleichtern.

zu A: */etc/src* wäre kein sinnvoller Platz für diesen Link.

zu B: *~/src/linux* wäre ebenfalls kein sinnvoller Platz für diesen Link.

zu D: */usr/src/linux-headers-3.0.0-12* ist ein typisches Verzeichnis für die Kernel-Quellen.

zu E: */lib/modules/3.0.0-12-generic* enthält die Module eines installierten Kernels.

Frage 8:

B: `mkinitrd` oder D: `mkinitramfs` sind Programme, die zur Erstellung der initialen RAM-Disk dienen. Welches der beiden Programme auf einem System vorhanden ist, hängt von der verwendeten Linux-Distribution ab. C: *dracut* ist ein relativ neues Produkt, mit dem sie sehr bequem initiale RAM-Disks erstellen können.

zu A: `make` ist ein Frontend für Compiler und kommt z. B. beim Kompilieren des Kernels zum Einsatz.

zu E: `init` ist der Vater aller Prozesse und hat mit diesem Thema absolut nichts zu tun.

Frage 9:

C: `hdparm /dev/nvme1` ist die richtige Lösung. Das Programm `hdparm` kann hierfür verwendet werden und die Nummerierung von *NVMe*-Geräten beginnt mit `0`. Deshalb verweist */dev/nvme1* auf das zweite Gerät.

zu A: `hdparm /dev/nvme2` operiert mit dem dritten NVMe-Gerät falls vorhanden.

zu B: `tune2fs /dev/nvme1` würde versuchen Dateisystemparameter zu justieren, was bei einem physischen Gerät nicht sinnvoll ist.

zu D: `tune2fs /dev/nvme2` ähnelt Antwort B, aber hier wird zusätzlich das falsche Gerät angesprochen.

zu E: `fsck /dev/nvme1` ist falsch, weil eine Dateisystemprüfung auf einem physischen Gerät keinen Sinn ergibt.

Frage 10:

C: *device.map* ist die Datei, die solche Zuordnungen enthält.

zu A: *mtab* enthält Informationen zu den aktuell eingehängten Dateisystemen.

zu B: *fstab* enthält Konfigurationseinträge zu den in einem Computer vorhandenen Dateisystemen, Mount-Points und Dateisystemtypen.

zu D: *menu.lst* ist normalerweise ein Link auf *grub.conf*.

zu E: *grub.conf* ist die Hauptkonfigurationsdatei von GRUB. Sie enthält vor allem Konfigurationseinträge zu den Betriebssystemen, die auf einem Computer installiert sind.

Frage 11:

/usr/src ist die richtige Lösung. Sie können natürlich ein beliebiges anderes Verzeichnis auswählen, aber das ist nicht üblich.

Frage 12:

`ln -s /usr/src/linux-3.0.0-15 /usr/src/linux` ist das vollständige Kommando, um den gewünschten Softlink zu erstellen.

Frage 13:

E: *libncurses5-dev* ist eine Bibliothek, die für zeichenbasierte Benutzerschnittstellen verwendet wird. Sie wird aufgerufen, wenn Sie den Kernel mithilfe des Kommandos `make menuconfig` konfigurieren.

zu A: `make` wird ebenfalls benötigt, stellt aber selbst keine Menüführung bereit.

zu B: `gcc` ist der C und C++-Compiler des GNU-Projekts.

zu C: `mc` ist eine visuelle Shell, die auf *ncurses* basiert, hat aber nichts mit der Kernel-Konfiguration zu tun.

zu D: `dracut` ist ein Werkzeug zur Erstellung initialer RAM-Disks.

Frage 14:

`update-grub` ist die richtige Antwort.

Frage 15:

A: `lsmod` und D: `cat /proc/modules` zeigen die aktuell in den Kernel geladenen Module an.

zu B: `modprobe -l` zeigte in früheren Kernelversionen die für diesen Kernel verfügbaren Module an. Das sagte allerdings nichts darüber aus, ob diese Module geladen wurden.

zu C: `insmod` lädt ein Modul in den laufenden Kernel.

zu E: `cat /etc/modules` ist eine Konfigurationsdatei, in der Module eingetragen werden, die während des Systemstarts geladen werden sollen.

Frage 16:

B: `efibootmgr -B` ist, gefolgt von der zu entfernenden Bootnummer, das richtige Kommando.

zu A: `efivar` kann lediglich Variablen innerhalb von EFI ändern.

zu C: `fdisk` konfiguriert die Partitionierung von Datenträgern.

zu D: `vi` ist nicht geeignet, weil es sich bei dem Bootmenü nicht um eine Textdatei handelt.

zu E: `update-grub` aktualisiert lediglich das Bootmenü von GRUB, aber nicht das von EFI.

Frage 17:

A: `modprobe` erfüllt genau diese Aufgabe.

zu B: `insmod` ist ebenfalls geeignet, Kernel-Module zu laden, löst aber im Gegensatz zu `modprobe` Abhängigkeiten nicht automatisch auf.

zu C: `rmmod` entfernt ein Modul aus dem laufenden Kernel.

zu D und E: `init` und `start` haben mit diesem Thema nichts zu tun.

Frage 18:

C: `depmod -n` führt die gewünschte Operation durch.

zu A: `modprobe -a` lädt alle angegebenen Module (z. B. eines bestimmten Typs).

zu B: `depmod -A` führt die Erstellung der Datei *modules.dep* tatsächlich durch. Die Option `-A` sorgt für eine schnelle Verarbeitung, indem lediglich neue Module berücksichtigt werden.

zu D: `insmod > modules.dep` ist unsinnig und erzeugt eine Fehlermeldung.

zu E: `lsmod > modules.dep` erzeugt eine Liste der momentan in den Kernel geladenen Module und speichert diese unsinnigerweise in der Datei *modules.dep* ab.

Frage 19:

4 »echte« Partitionen können auf einer Festplatte angelegt werden.

Frage 20:

`depmod` ist das hierfür benötigte Programm.

Frage 21:

B: `init` mit PID 1 ist die richtige Antwort. Denken Sie in der Prüfung daran, dass der Name dieses Daemons nicht auf »d« endet.

Frage 22:

E: `lspci -vvv` ist hier richtig. Dreifach verbose ist wirklich schon sehr genau.

zu A: `lspci` ist zwar auch nicht falsch, aber nur Antwort E erfüllt die Anforderung der Genauigkeit.

zu B: `pciinfo` ist kein Programm, das auf einem Linux-Computer normalerweise vorkommt.

zu C: `ps` zeigt ohne Optionen lediglich die aktuellen Prozesse eines Benutzers in der aktuellen Shell an.

zu D: `rpm` ist der RPM Package Manager.

Frage 23:

B: `ldd` zeigt an, von welchen Bibliotheken ein Programm abhängig ist.

zu A: `ldconfig` konfiguriert den dynamischen Linker.

zu C: `ltrace` zeigt an, welche Bibliotheken ein Programm aufruft, während es läuft. Die Überprüfung soll jedoch vorher stattfinden. Fehlende Bibliotheken würden übrigens auch nicht angezeigt werden.

zu D: `strace` zeigt an, welche Systemaufrufe ein Programm zur Laufzeit tätigt.

zu E: `ld.so.conf` ist kein Programm, sondern eine Konfigurationsdatei für `ldconfig`. Hier befinden sich Informationen über Verzeichnisse, die Bibliotheken enthalten.

Frage 24:

`ldconfig` verlinkt die neuen Bibliotheken.

Frage 25:

B: */etc/init.d* und C: */etc/rc.d* sind die beiden Standardverzeichnisse für `init`-Skripte.

zu A: */etc/init* enthält Skripte für `upstart`.

zu D: */lib/modules* enthält Unterverzeichnisse für die Kernel-Module.

zu E: */boot* enthält hauptsächlich den Kernel und ggf. den Bootloader.

Frage 26:

B: *Der Computer befindet sich in Runlevel 3.* und E: *Der Computer befand sich zuvor in Runlevel 1.*

Die anderen Antworten sind entsprechend selbstverständlich falsch.

Frage 27:

E: `shutdown -rF now` startet den Computer neu und erzwingt die Ausführung von `fsck` beim Neustart.

zu A: `shutdown -r now` startet den Computer neu. `fsck` wird nur ausgeführt, wenn eine routinemäßige Überprüfung eines Dateisystems ohnehin angezeigt ist.

zu B: `init 6` startet den Computer ohne Ankündigung neu.

zu C: `init 0` fährt den Computer ohne weitere Ankündigung herunter.

zu D: `shutdown -rf now` startet den Computer neu und verhindert die routinemäßige Ausführung von `fsck` beim Neustart.

Frage 28:

A: *http://www.linuxbase.org*

B: *http://www.linuxfoundation.org*

Beide Webseiten gehören der Linux Foundation und befassen sich hauptsächlich mit der Standardisierung von Linux.

zu C: *http://www.lpi.org* ist die Webseite des LPI.

zu D: *http://www.kernel.org* stellt Linux-Kernel und Informationen über Linux-Kernel bereit.

zu E: *http://www.wikipedia.de* ist keine gute Anlaufstelle für dieses Thema.

Frage 29:

D: `dmesg|tail` zeigt die letzten zehn Zeilen des Kernel Ring Buffers an. Hier finden sich fast immer Meldungen, wenn es ein Problem mit einem Gerät gibt.

zu A: `fsck -t ext4 /dev/sda1` prüft ein ext4-Dateisystem.

zu B und zu C: `fdisk /dev/sda` und `cfdisk /dev/sda` sind jeweils geeignet, die Partitionierung des Gerätes */dev/sda* zu ändern.

zu E: `fsck -V /dev/sda1` prüft das Dateisystem auf */dev/sda1* und gibt ausführliche Meldungen aus.

Frage 30:

C: `lsof` listet geöffnete Dateien auf.

zu A: `df` erstellt eine Aufstellung zur Festplattennutzung.

zu B: `free` zeigt die Belegung des Arbeitsspeichers an.

zu D: `dmesg` zeigt den Inhalt des Kernel Ring Buffers an.

zu E: `du` zeigt die Belegung des Dateisystems an.

Frage 31:

B: `telinit-q` veranlasst den Daemon `init`, seine Konfiguration neu einzulesen. Das beinhaltet auch Konfigurationen in der Datei */etc/inittab*.

zu A und zu C: `init 6` und `reboot` starten jeweils das System neu. Hierbei wird zwar ebenfalls sichergestellt, dass die Konfiguration neu eingelesen wird, aber in der Frage wurde ausdrücklich erwähnt, dass die Verfügbarkeit des Systems nicht eingeschränkt werden soll.

zu D: `init -q` ist falsch, weil das Kommando `init` die Option `-q` nicht kennt.

zu E: `init 0` fährt lediglich das System herunter.

Frage 32:

`fsck` ist für diese Aufgabe das richtige Programm.

Frage 33:

A: `mount -o remount,rw /dev/sdb2` ist wahrscheinlich die geeignete Maßnahme. Da das Problem spontan aufgetreten ist, kann man annehmen, dass die Partitionen der Festplatte aufgrund eines Fehlers vom System read-only remounted wurden. Das ist bei modernen Linux-Distributionen nicht ungewöhnlich und kann durch den gerade genannten Befehl zumindest temporär rückgängig gemacht werden.

zu B: `mount -o remount,rw /dev/sdb1` ist grundsätzlich nicht falsch, steuert aber die falsche Partition an.

zu C: `mount /dev/sdb1 /temp` ist theoretisch denkbar, verwendet aber ebenfalls die falsche Partition.

zu D: `umask 022` führt hier zu nichts.

zu E: `chmod 777 / -R` ist falsch, weil die Beschreibung der Problematik nicht auf ein Berechtigungsproblem schließen lässt. Außerdem wäre diese Maßnahme auch stark übertrieben.

Frage 34:

E: Editieren Sie die Datei */boot/grub/grub.cfg*. Das ist der einzig richtige Weg. Änderungen der Kernel-Parameter während des Systemstarts innerhalb der GRUB-Shell sind nicht dauerhaft. Deshalb sind die Antworten A, B und C sowieso falsch.

zu D: Editieren Sie die Datei */etc/grub.cfg*. Die Datei *grub.cfg* befindet sich nicht unterhalb von */etc*, sondern in */boot/grub*.

Frage 35:

C: `fsck -a /dev/sdb3` repariert das Dateisystem automatisch ohne jede Nachfrage.

D: `fsck -y /dev/sdb3` beantwortet alle Fragen, die das Programm (auch bezüglich Reparaturen) stellt, automatisch mit »yes«.

zu A: `fsck -a /dev/sdb2` repariert die falsche Partition.

zu B: `fsck -y /dev/sdb2` repariert ebenfalls die falsche Partition.

zu E: `cfdisk` ist ein Partitionierungstool und nicht zur Reparatur von Dateisystemen geeignet.

Frage 36:

C: `fsck -t ext4 /dev/sdb1` überprüft ein ext4-Dateisystem auf dem Gerät */dev/sdb1*.

E: `fsck.ext4 /dev/sdb1` überprüft ebenfalls ein ext4-Dateisystem auf */dev/sdb1*.

zu A: `ext4.mkfs /dev/sdb1` gibt es nicht.

zu B: `mkfs -t ext4 /dev/sdb1` formatiert */dev/sdb1* mit dem ext4-Dateisystem.

zu D: `mkfs.ext4 /dev/sdb1` formatiert ebenfalls */dev/sdb1* mit dem ext4-Dateisystem.

Frage 37:

D: */etc/fstab* ist hierfür die richtige Konfigurationsdatei. Sie enthält statische Informationen über Dateisysteme.

zu A und zu B: */etc/mtab* und */proc/mounts* sind keine Konfigurationsdateien. Sie können in diesen Dateien nachsehen, welche Dateisysteme aktuell gemountet sind, aber Sie sollten diese Dateien nie bearbeiten.

zu C und zu E: */dev/sdb1* und */backup* können nicht bearbeitet werden.

Frage 38:

`sync` ist die richtige Antwort. Das Programm entleert den Dateisystempuffer.

Frage 39:

A: `mount`, B: `cat /proc/mounts` und E: `cat /etc/mtab` zeigen jeweils die aktuell eingehängten Dateisysteme an.

zu C: `sync` entleert die Dateisystempuffer und schreibt deren Inhalt auf die jeweiligen Datenträger.

zu D: `cat /etc/fstab` enthält zwar Informationen zu den Dateisystemen eines Computers, sagt aber nichts über den aktuellen Status aus.

Frage 40:

Bei den Antworten D und E sind jeweils die Angaben zu den Gerätedateien falsch. Die Angabe von *sda1* in Antwort D ist unzureichend und in Antwort E wird eine Gerätedatei unterhalb von */proc* verwendet.

Frage 41:

A: `mount /dev/sda3` und C: `mount /speicher` führen beide die gewünschte Operation durch.

E: `mount -a` hängt alle in der Datei */etc/fstab* enthaltenen Dateisysteme ein, bei denen nicht die Option `noauto` gesetzt ist.

zu B: `mount /speicher /dev/sda3 -t ext4` verursacht eine Fehlermeldung, weil das Gerät und der Mountpoint in der falschen Reihenfolge angegeben wurden.

zu D: `umount /dev/sda3 /speicher` hängt das genannte Dateisystem nicht ein, sondern aus.

Frage 42:

E: `user` bewirkt genau das gewünschte Verhalten.

zu A: `ro` ist die Option, die verwendet wird, um ein Dateisystem read-only zu mounten.

zu B: `nouser` verhindert, dass ein normaler Benutzer das Dateisystem einhängen kann.

zu C: `noauto` verhindert automatisches Einhängen während des Systemstarts bzw. durch das Kommando `mount -a`.

zu D: `users` sorgt dafür, dass ein beliebiger Benutzer ein Dateisystem einhängen darf. Allerdings kann auch ein beliebiger (auch ein anderer) Benutzer das Dateisystem wieder aushängen.

Frage 43:

A: `ro` wird durch `defaults` nicht abgedeckt. Es wäre auch nicht sinnvoll, wenn die Standardeinstellung die Read-only-Option beinhalten würde.

zu B: `rw` read-write ist enthalten.

zu C: `auto`: Die Möglichkeit, das Dateisystem mithilfe des Kommandos `mount -a` einzuhängen, ist ebenfalls enthalten.

zu D: `exec` beinhaltet die Möglichkeit, ausführbare Dateien zu verwenden, und ist enthalten.

zu E: `nouser` sorgt dafür, dass normale Benutzer das Dateisystem nicht einhängen können. Auch diese Option ist in `defaults` enthalten.

Frage 44:

`swapon -a` aktiviert alle in der Datei *fstab* definierten Swap-Dateisysteme, es sei denn, die Option `noauto` ist gesetzt.

Frage 45:

B: */etc/auto.master* ist die Hauptkonfigurationsdatei von `autofs`.

zu A: */etc/fstab* kann u. a. auch verwendet werden, um Dateisysteme beim Systemstart zu mounten, wird von `autofs` aber nicht verwendet.

zu C: */etc/mtab* enthält Informationen zu aktuell gemounteten Dateisystemen und wird zur Konfiguration von `autofs` nicht verwendet.

zu D und zu E: */etc/auto.mist* und */etc/auto.net* werden durch die Konfigurationsdatei *auto.master* referenziert, aber es handelt sich nicht um die Hauptkonfigurationsdateien.

Frage 46:

D: `tune2fs` dient der Anpassung von Dateisystemparametern. Dazu zählt auch die Steuerung der regelmäßigen Überprüfung.

zu A: `mkisofs` wird zur Erstellung von ISO-Dateisystemen verwendet.

zu B: `fsck` überprüft Dateisysteme. Dieses Programm selbst kommt während der regelmäßigen Dateisystemüberprüfungen zum Einsatz.

zu C: `debugfs` dient der interaktiven Untersuchung und Modifikation von Dateisystemen. Sie können mithilfe dieses Programms z. B. gelöschte Dateien wiederherstellen.

zu E: `hdparm` konfiguriert und analysiert Festplattenparameter. Das Programm ist u. a. für Festplattengeometrie, PIO-, DMA- und UDMA-Modes zuständig.

Frage 47:

D: `lsmod` listet die geladenen Module eines Kernels auf.

zu A: `modprobe -l` listete in früheren Kernelversionen die verfügbaren Module auf. Diese Module konnten bei Bedarf geladen werden.

zu B: `modprobe -r` entfernt ein laufendes Modul.

zu C: `insmod` installiert ein Modul.

zu E: `rmmod` entfernt (wie `modprobe -r`) ein laufendes Modul.

Frage 48:

C: `debugfs` kann verwendet werden, um eine versehentlich gelöschte Datei wiederherzustellen. Die Datei muss sich allerdings auf einem ext2-, ext3-, oder ext4-Dateisystem befunden haben.

zu A: `mkfs` erstellt neue Dateisysteme und ist deshalb für diese Aufgabe unbrauchbar.

zu B: `tune2fs` ändert die Parameter eines Dateisystems, kann aber nicht zur Wiederherstellung verwendet werden.

zu D und zu E: `restore` und `cp` können nicht verwendet werden, weil die Datei zuvor nicht gesichert wurde.

Frage 49:

D: `dd if=/dev/hda of=mbr.backup ibs=512 count=1` führt die Sicherung des MBR durch. Die Sicherungsdatei *mbr.backup* wird im aktuellen Verzeichnis abgelegt.

zu A: `dd if=/dev/hda of=mbr.backup count=1` sichert die komplette Festplatte mit der Gerätedatei */dev/hda*.

zu B: `cp MBR mbr.backup` ergibt eine Fehlermeldung, weil der MBR (jedenfalls normalerweise) keine Datei ist.

zu C: `dd if=/dev/hda1 of=mbr.backup ibs=512 count=1` sichert die erste Partition von */dev/hda*.

zu E: `dd if=mbr.backup of=/dev/hda ibs=512 count=1` wäre ein geeignetes Kommando zur Wiederherstellung eines zuvor gesicherten MBR.

Frage 50:

`fdisk /dev/sdd` ist die richtige Antwort. Es gibt zwar auch andere Programme für diese Aufgabe, aber `fdisk` ist als einziges Partitionierungswerkzeug immer als Bordmittel verfügbar. Vergessen Sie übrigens bitte nicht, dass es bei `fdisk` erforderlich ist, die Gerätedatei der zu partitionierenden Festplatte anzugeben.

Frage 51:

E: `mkswap /dev/sdb1` ist richtig. Bevor Sie die Swap-Partition, wie auch immer, in Betrieb nehmen, muss diese zunächst formatiert werden.

zu A: `swapon -v /dev/sdb1` können Sie zur Aktivierung der Swap-Partition ausführen, nachdem diese formatiert wurde.

zu B: `swapoff -v /dev/sdb1` würde die Verwendung der Swap-Partition deaktivieren.

zu C: `swapon -a` aktiviert alle Swap-Dateisysteme, die in der Datei */etc/fstab* aufgeführt sind, es sei denn, die Option `noauto` ist gesetzt.

zu D: `vi /etc/fstab` können Sie später verwenden, um die Swap-Partition z. B. gleich nach dem Start des Systems verfügbar zu machen.

Frage 52:

B: `mkisofs` erstellt Dateisysteme vom Typ ISO 9660.

zu A: `dd` konvertiert und kopiert Daten, kann aber keine Dateisysteme erzeugen.

zu C: `cdrecord` können Sie anschließend verwenden, um das entstandene ISO-Image auf die CD zu brennen.

zu D: `mkfs.ntfs` erstellt ein NTFS-Dateisystem.

zu E: `fdisk` dient der Partitionierung, unterstützt aber nicht ISO 9660.

Frage 53:

D: `dd if=/dev/hdc of=cdrom.iso` kopiert blockorientiert den Inhalt des Geräts */dev/hdc* in die Datei *cdrom.iso*. Das Ergebnis ist eine gültige ISO-Datei.

zu A: `cp /dev/hdc cdrom.iso` erzeugt kein ISO 9660-Format. Der Befehl `cp` erkennt */dev/hdc* als Datei und wird diese auch als solche auf die Datei *cdrom.iso* kopieren, aber das Ergebnis entspricht nicht einer ISO-9660-Datei.

zu B: `cdrecord /dev/hdc` dient der Erstellung von CDs.

zu C: `dd if=cdrom.iso of=/dev/hdc` funktioniert nicht, weil Quelle und Ziel vertauscht wurden.

zu E: `cdrecord dev=/dev/hdc` dient der Erstellung von CDs.

Frage 54:

A: `lsusb`, B: `lspci`, und C: `lsdev` führen diese Aufgabe für ihren jeweiligen Fachbereich aus.

zu D: `lsof` zeigt geöffnete Dateien an.

zu E: `ltrace` überprüft, welche Bibliotheken ein Programm während seiner Ausführung aufruft.

Frage 55:

C: */etc/udev/rules.d/70-persistent-net.rules* enthält u. a. Zuordnungen von MAC-Adressen der Netzwerkkarten zu deren verwendeten Namen. Hier können Sie den Eintrag für die alte Netzwerkkarte komplett löschen und gleichzeitig den Namen an die neue Netzwerkkarte binden.

zu A: */etc/network/interfaces* enthält die TCP/IP-Konfiguration. Die Namensvergabe von Netzwerkkarten wird an dieser Stelle schon vorausgesetzt.

zu B: */etc/init.d* enthält `init`-Skripte.

zu D: */etc/sysconfig/network* enthält bei Red Hat lediglich die Information, ob ein Netzwerk überhaupt verwendet wird, sowie den Hostnamen.

zu E: */etc/inittab* ist die initiale Konfigurationsdatei von `init`.

Frage 56:

B: `udevmonitor` wartet auf Kernel-Ereignisse, die mit `udev` in Zusammenhang stehen, und gibt diese live auf der Konsole aus.

C: `udevadm` ist in Kombination mit der Option `monitor` ein moderner Ersatz für den `udevmonitor`. Auch hier werden `udev`-bezogene Kernel-Ereignisse live ausgegeben.

zu A und zu D: `lsusb` und `dmesg` sind zur Diagnose des beschriebenen Problems grundsätzlich geeignet. In der Frage wurde aber ausdrücklich nach Programmen gefragt, die eine Live-Anzeige unterstützen.

zu E: `lspci` dient der Diagnose PCI-bezogener Probleme.

Frage 57:

RAID 0 ist die richtige Antwort. Bei Verwendung dieses RAID-Levels wird auf alle involvierten Datenträgern quasi gleichzeitig geschrieben bzw. gelesen, was eine optimale Leistungsausnutzung ermöglicht. Die Kapazitäten der (gleich großen) Einzeldatenträger addieren sich.

Frage 58:

RAID 1 ist hier die richtige Antwort. Bei Verwendung von RAID 1 werden zwei Datenträger gespiegelt. Dieses Vorgehen bietet ein Höchstmaß an Redundanz, allerdings keine nennenswerten Performancevorteile.

Frage 59:

D: `mdadm --add /dev/md0 /dev/sdd1` fügt einem bestehenden RAID-Array (*/dev/md0*) eine weitere Partition (*/dev/sdd1*) hinzu.

zu A, zu B und zu E: `fdisk`, `cfdisk` und `sfdisk` sind lediglich Partitionierungstools. Die Partitionierung ist aber bereits erfolgt.

zu C: `mdadm --detail /dev/md0` können Sie anschließend verwenden, um das Ergebnis des Eingriffs zu überprüfen.

Frage 60:

C: */etc/mdadm/mdadm.conf* ist hier die richtige Datei. Die Konfigurationsdateien aus den anderen Antworten (*/etc/mtab*, */etc/fstab*, */etc/aliases* und *~/.forward*) ergeben in diesem Zusammenhang überhaupt keinen Sinn.

Frage 61:

B: `dmesg` zeigt den Inhalt des Kernel-Ring-Buffers an. Wenn ein Problem mit einer Hardwarekomponente auftritt, werden Sie hier die ersten Hinweise finden.

zu A und C: `lspci` und `lsusb` zeigen lediglich die Hardwarekomponenten selbst, aber nicht deren Status an.

zu D: `lsmod` zeigt die geladenen Kernel-Module an.

zu E: `uname` zeigt nur grundlegende Systeminformationen an.

Frage 62:

E: */var/log/messages* enthält auch Meldungen, die während eines Systemstarts ausgegeben werden. */var/log/syslog* wäre ebenfalls eine richtige Antwort gewesen.

zu A: */etc/syslog.conf* dient der Konfiguration des `syslogd`, aber nicht der Protokollierung selbst.

zu B: */proc/kmsg* enthält den Inhalt des Kernel Ring Buffers. Sollten beim Systemstart Probleme aufgetreten sein, die nicht mit dem Kernel in Zusammenhang stehen, dann werden Sie hier nicht fündig.

zu C: */var/log/auth.log* enthält lediglich Protokollierungseinträge, die mit der Authentifizierung in Zusammenhang stehen.

zu D: */boot/grub/boot.img* enthält eine Sicherungskopie des MBR.

Frage 63:

A: `pvscan -v` durchsucht alle angeschlossenen Festplatten nach physikalischen Volumen.

zu B: `vgscan -v` sucht nach vorhandenen Volumengruppen.

zu C: `lvscan -v` sucht nach logischen Volumen.

zu D: `vgcreate` erstellt eine neue Volumengruppe.

zu E: `fdisk` ist ein Partitionierungstool, das u. a. zur Vorbereitung physikalischer Volumen verwendet werden kann.

Frage 64:

E: `pvcreate` wird verwendet, wenn neue physikalische Volumen für LVM erstellt werden müssen.

zu A: `fdisk` ist ein Partitionierungstool, das u. a. zur Vorbereitung physikalischer Volumen verwendet werden kann.

zu B: `vgscan` durchsucht angeschlossene Festplatten nach Volumengruppen und physikalischen Volumen.

zu C: `pvscan` durchsucht angeschlossene Festplatten nach physikalischen (LVM-) Volumen.

zu D: `vgcreate` erstellt neue LVM-Volumengruppen.

Frage 65:

`vgscan` ist hier die richtige Antwort. `vgscan` durchsucht angeschlossene Festplatten nach Volumengruppen und physikalischen Volumen.

Frage 66:

Die folgenden Aussagen sind richtig:

B: *LVM und RAID sind kombinierbare Technologien.*

D: *Mit LVM verwaltete Volumen können um zusätzliche Festplatten ergänzt werden.*

Bei den anderen drei Antworten ist eigentlich jeweils das genaue Gegenteil der Fall:

zu A: *LVM wird ausschließlich für RAID verwendet.*

Sie können LVM auch auf einer einzelnen Festplatte konfigurieren, während ein RAID immer aus mehreren physikalischen Festplatten besteht.

zu C: *Volumen, die mit LVM verwaltet werden, sind nicht erweiterbar.*

Doch, das sind sie. Genau genommen ist das einer der wichtigsten Vorteile von Volumen, die mit LVM verwaltet werden.

zu E: *LVM bietet Ausfallsicherheit durch Redundanz.*

LVM bietet absolut keine Redundanz. Auch diese Eigenschaft unterscheidet LVM von RAID.

Frage 67:

C: `lvextend` muss zuerst ausgeführt werden, um die Größe des logischen Volumens um das neue physikalische Volumen zu erweitern.

D: `resize2fs` wird anschließend verwendet, um das Dateisystem auf die neue Größe des logischen Volumens auszudehnen.

zu A: `lvcreate` erstellt logische Volumen.

zu B: `vgcreate` erstellt Volumengruppen.

zu E: `pvcreate` initialisiert physikalische Volumen. Dieser Vorgang ist aber schon durchgeführt worden.

Genau genommen fehlt hier noch ein Schritt. Das neue physikalische Volumen hätte nämlich zunächst mittels `vgextend` in die Volumengruppe aufgenommen werden müssen. Dieses Kommando ist aber in der Aufgabenstellung nicht enthalten.

Frage 68:

C: Device Mapper ist für diese Abstraktion zuständig.

zu A und B: `inetd` und `xinetd` sind jeweils Super-Daemons, die für den Start von Netzwerkdiensten zuständig sind.

zu D und E: `lvcreate` und `pvcreate` sind jeweils Tools zur Administration von LVM-basierten Volumen.

Frage 69:

A: `arp` zeigt den Inhalt des ARP-Caches an. Die Ausgabe des Kommandos beinhaltet eine Zuordnung von MAC-Adressen zu IP-Adressen von Computern desselben Netzwerksegments.

zu B: `arp -d` verursacht eine Fehlermeldung. Grundsätzlich kann mithilfe der Option `-d` ein bestimmter Eintrag aus dem ARP-Cache gelöscht werden. Dieser Eintrag muss dann als Parameter übergeben werden.

zu C: `ifconfig` wird ausschließlich zur Konfiguration von Netzwerkkarten verwendet.

zu D: `route -n` wird zur Anzeige und Modifikation von Routing-Tabellen benutzt.

zu E: `ifup` startet die angegebene Netzwerkschnittstelle.

Frage 70:

D: `route -n` sorgt für eine numerische Ausgabe der Routing-Tabelle. Eine Namensauflösung findet also nicht statt.

zu A: `route` zeigt den Inhalt der Routing-Tabelle an. Es wird versucht, die IP-Adressen in voll qualifizierte Hostnamen aufzulösen.

zu B: `route -C` zeigt die Routing-Tabelle des Kernels an.

zu C: `route -v` zeigt die Routing-Tabelle ausführlicher an.

zu E: `traceroute` ist für diese Aufgabe das falsche Programm. Dieses Tool würde die komplette Route eines Pakets zu einem angegeben Zielhost anzeigen.

Frage 71:

A: `ifconfig` und E: `ip` sind für diese Aufgabe optimal.

zu B: `route` zeigt die Routing-Tabelle an.

zu C: `arp` zeigt den Inhalt des ARP-Caches an.

zu D: `ping` sendet ICMP-Echoanforderungen.

Frage 72:

B: `iwconfig wlan0` zeigt die Konfiguration der Schnittstelle `wlan0` an. Die Ausgabe des Kommandos beinhaltet auch die Sendeleistung.

zu A: `iwconfig wlan0 txpower on` würde die Netzwerkkarte einschalten, wenn diese ausgeschaltet war.

zu C: `ifconfig wlan0` zeigt lediglich allgemeine Schnittstelleninformationen an, die es auch bei drahtgebundenen Netzwerkkarten gibt. Das beinhaltet also nicht die Sendeleistung.

zu D: `iwconfig wlan0 txpower off` schaltet die Drahtlosnetzwerkkarte ab.

zu E: `iwconfig wlan0 txpower 20mW` stellt die Sendeleistung der Drahtlosnetzwerkkarte auf 20 mW ein.

Frage 73:

`ifconfig eth1` liefert Ihnen das gewünschte Ergebnis.

`ip addr show eth1` wäre im Prinzip auch nicht falsch, aber es wurde ausdrücklich nach dem einfachsten Kommando gefragt.

Frage 74:

Die Antworten A und D erstellen den gewünschten Eintrag.

A: `route add -net 172.16.0.0 netmask 255.255.0.0 gw 192.168.50.7`

D: `route add -net 172.16.0.0/24 gw 192.168.50.7`

zu B: und C: Bei diesen beiden Antworten sind jeweils die Gateway-Adressen und die Zielnetzwerke vertauscht.

zu E: `route -n` zeigt lediglich den Inhalt der Routing-Tabelle an.

Frage 75:

E: `nmap` ist ein Portscanner und für diese Aufgabe optimal geeignet.

zu A: `iptables` dient der Konfiguration der `iptables`-Firewall.

zu B: `ifconfig` wird zur Schnittstellenkonfiguration verwendet.

zu C: `lsof` zeigt geöffnete Dateien an.

zu D: `netstat` kann verwendet werden, um geöffnete Ports auf einem lokalen System zu prüfen. In der Frage geht es aber ausdrücklich um ein entferntes System.

Frage 76:

`lsof` listet geöffnete Dateien auf. Für die hier beschriebene Situation ist `lsof` also wie geschaffen.

Frage 77:

Es gibt zwei Möglichkeiten, die hier in Betracht kommen. Wenn ein zusätzlicher Alias (hier `eth0:1`) nicht hinderlich ist, können Sie Antwort A wählen:

A: `ifconfig eth0:1 192.168.7.5`

Eleganter ist die Lösung durch Antwort D, weil hier kein zusätzlicher Alias erstellt wird:

D: `ip address add 192.168.7.5/24 dev eth0`

zu B: `ifconfig eth0 192.168.7.5` fügt keine zusätzliche Adresse hinzu, sondern ersetzt die vorhandene Adresse.

zu C: `ip address show dev eth0` zeigt lediglich die Konfiguration der Schnittstelle `eth0` an.

zu E: `ifconfig eth1:1 192.168.7.5` führt die gewünschte Konfiguration aus, jedoch auf der falschen Netzwerkkarte.

Frage 78:

C: `netstat -an| grep :80` ist richtig. Der erste Teil des Kommandos listet alle Sockets numerisch auf, einschließlich derer, die nur auf Verbindungen warten. Der zweite Teil sorgt dafür, dass nur Webserververbindungen (zu Port 80) angezeigt werden.

E: `nmap localhost -p 80` funktioniert ebenfalls. *Nmap* ist ein Portscanner, der hier die lokale Maschine nach dem Webserver-Port (80) abscannt.

zu A: `netstat -an| grep :8080` sucht nach dem falschen Port.

zu B: `netstat -an| grep :443` und

zu D: `nmap localhost -p 443` suchen jeweils nach dem Port für HTTPS. Da der Webserver neu installiert wurde, wird er wahrscheinlich noch nicht für SSL konfiguriert sein.

Frage 79:

A: `wireshark` ist richtig. Alle anderen genannten Programme werden als Kommandozeilentools verwendet und unterstützen keine grafischen Ausgaben.

Frage 80:

`route -n` zeigt die Routing-Tabelle eines Computers an. Die Option `-n` sorgt für eine numerische Ausgabe, also ohne Namensauflösung via DNS.

Frage 81:

D: `netcat` ist für diesen Test geeignet, weil mit diesem Programm ein Port-Listener erstellt werden kann, der den fraglichen Port dann abhört. Von einem anderen System aus können Sie diesen Port dann ebenfalls mithilfe von `netcat` ansteuern. In diesem Falle starten Sie den Listener mit:

```
nc -l p 5112
```

Die anderen genannten Programme können keine Port-Listener zur Verfügung stellen.

zu A: `nmap` ist ein Portscanner.

zu B: `netstat` zeigt u. a. Netzwerkverbindungen und Schnittstellenstatistiken an.

zu C: `wireshark` ist ein Protocol-Analyzer.

zu E: `ifconfig` dient der Konfiguration von Netzwerkkarten.

Frage 82:

`ping` ist hier die Lösung. Dieses Programm sendet ICMP-Echo-Anforderungen an Netzwerkcomputer.

Frage 83:

A: */proc/cpuinfo* ist eine Pseudodatei des */proc*-Dateisystems. Sie enthält genau die benötigten Informationen.

zu B: */proc/cpu* ist eine frei erfundene Datei.

zu C: */var/log/messages* enthält die Systemprotokollierung.

zu D: `lspci` zeigt Informationen zu PCI-Geräten an.

zu E: `lsusb` zeigt Informationen zu USB-Geräten an.

Frage 84:

A: */etc/HOSTNAME* und C: */etc/hostname* sind die hierfür zuständigen Konfigurationsdateien.

zu B: */etc/hosts* unterstützt den DNS-Client und dient lediglich der Namensauflösung.

zu D: */etc/hosts.allow* dient der Konfiguration von TCP-Wrappern.

zu E: `/bin/hostname` ist keine Konfigurationsdatei, sondern ein ausführbares Programm. Es zeigt oder konfiguriert den Hostnamen.

Frage 85:

B: `dmesg| tail -n 20` zeigt die letzten 20 Zeilen des Kernel Ring Buffers an.

zu A: `dmesg| tail` zeigt nur die letzten 10 Zeilen des Kernel Ring Buffers an.

zu C: `tail -n 20 /var/log/syslog` zeigt die letzten 20 Zeilen der Protokolldatei *syslog* an.

zu D: `tail -n 20 /var/log/messages` zeigt die letzten 20 Zeilen der Protokolldatei *messages* an.

zu E: `dmesg` gibt den gesamten Inhalt des Kernel Ring Buffers aus.

Frage 86:

D: */etc/hosts* ist eine statische Tabelle mit Zuordnungen von IP-Adressen zu Hostnamen. Sie unterstützt den DNS-Client bei der Namensauflösung. Der in der Frage geforderte Eintrag könnte z. B. so aussehen:

```
192.168.0.1 router
```

zu A und E: */etc/hosts.allow* und */etc/hosts.deny* sind Konfigurationsdateien für den TCP-Wrapper.

zu B: */etc/bind/db.127* ist eine Zonendatei des BIND-DNS-Servers, die zur Unterstützung des Loopback-Netzwerks dient.

zu C: */etc/hostname* enthält den Namen eines Computers.

Frage 87:

C: */etc/resolv.conf* ist die Konfigurationsdatei für den DNS-Client (Resolver). Hier sollte der DNS-Server eingetragen werden.

zu A: */etc/named.conf* ist die Hauptkonfigurationsdatei des BIND-DNS-Servers.

zu B: */etc/hosts* dient der Namensauflösung, nicht aber der Konfiguration des DNS-Clients.

zu D: */etc/bind/named.conf* ist die aktuelle Hauptkonfigurationsdatei des BIND-DNS-Servers.

zu E: */etc/dnsclient.conf* gibt es nicht. Der Name ist frei erfunden.

Frage 88:

`traceroute deep.thought.adams.com` ist die richtige Antwort.

Frage 89:

A: `ALL: ALL` in */etc/hosts.deny* und

B: */etc/hosts.allow* entleeren oder löschen.

Diese beiden Antworten gewährleisten in dieser Kombination, dass auf das System nicht von außen zugegriffen werden kann. Antwort A sorgt dafür, dass auf alle Dienste der Zugriff verwehrt wird, und zwar von allen Hosts aus. Mit Antwort B werden zuvor eventuell erteilte Erlaubnisse auf den Computer zuzugreifen entzogen. Beachten Sie, dass Einträge in der Datei *hosts.allow* Vorrang gegenüber Einträgen in der Datei *hosts.deny* haben.

zu C: `ALL: ALL` in */etc/hosts.allow*

und zu D: */etc/hosts.deny* entleeren oder löschen.

Die Antworten C und D in Kombination bewirken das Gegenteil von dem, was gewünscht wurde.

zu E: `DENY: ALL` in */etc/host.conf* ist ebenfalls falsch. Die hier genannte Datei ist eine Konfigurationsdatei des DNS-Clients (Resolver).

Frage 90:

D: `nameserver 192.168.50.1` ist ein gültiger Eintrag und er verweist auf den von diesem System zu verwendenden DNS-Server.

E: `search lpic-2.de` ist eine Anweisung, das angegebene Suffix bei Suchanfragen automatisch anzuhängen.

zu A: `domain=lpic-1.de` ist falsch. In der Datei *resolv.conf* werden keine Gleichheitszeichen verwendet.

zu B: `address 192.168.50.1` ist ein IP-Adress-Eintrag in der Datei */etc/network/interfaces*.

zu C: `dns-nameservers 192.168.50.1` ist ebenfalls ein typischer Eintrag in der Datei */etc/network/interfaces*.

Frage 91:

B: `shutdown` kann für Benachrichtigungen bezüglich des Herunterfahrens eines Systems verwendet werden. Wenn Sie die Option `-k` verwenden, werden lediglich die Benutzer benachrichtigt, das System fährt aber nicht herunter.

C: `wall` ist genau für solche Fälle gedacht. Das Programm sendet Nachrichten an die Shells aller angemeldeten Benutzer.

zu A und zu D: `issue` und `issue.net` sind keine Programme, sondern Dateien, die Informationen enthalten, die den Benutzern bei der Anmeldung gezeigt werden. In der Frage ging es außerdem um User, die bereits angemeldet sind.

zu E: `info` ist ein Programm zur Anzeige von Info-Dateien.

Frage 92:

D: *motd* ist hier genau richtig. Der Inhalt der Datei */etc/motd* wird den Benutzern erst nach einem erfolgreichen Login angezeigt.

zu A: *issue* wird bereits vor dem Login angezeigt.

zu B: *issue.net* wird nur angezeigt, wenn sich ein Benutzer über das Netzwerk (z. B. via SSH) anmeldet.

zu C: *info* ist ein Programm zur Anzeige von Info-Dateien.

zu E: *resolv.conf* ist eine Konfigurationsdatei des DNS-Clients (Resolver).

Frage 93:

C: `lsof /media/usb-disk1` und D: `lsof /dev/sdc1` stellen jeweils fest, welche Dateien durch welchen Prozess auf dem fraglichen Laufwerk geöffnet sind. Wenn die geöffneten Dateien als nicht problematisch einzustufen sind, können die zugreifenden Prozesse anschließend mithilfe von `kill` beendet werden.

zu A: `du /media/usb-disk1` stellt die Belegung der Verzeichnisse des eingehängten Dateisystems fest.

zu B: `df /dev/sdc1` zeigt die Belegung der eingehängten Partition an.

zu E: `umount /dev/sdc1` versucht, das Laufwerk auszuhängen. Das hat aber schon beim ersten Versuch nicht funktioniert und ist auch nicht das Thema der Aufgabe.

Frage 94:

A: */etc/network/* ist das Verzeichnis, in dem sich bei Debian und seinen Derivaten typischerweise die Datei *interfaces* befindet, die wiederum die IP-Konfiguration beinhaltet.

E: */etc/sysconfig/network-scripts/* wäre die erste Anlaufstelle, wenn es sich um ein Red Hat Derivat oder natürlich Red Hat selbst handelt.

zu B: */etc/init.d/* enthält lediglich die Skripte für `init`.

zu C: */proc/sys/net/ipv4* enthält Dateien, die tatsächlich einiges über den aktuellen Status der TCP/IP-Konfiguration aussagen. Eine Konfiguration direkt im */proc*-Dateisystem ergibt aber in diesem Falle keinen Sinn.

zu D: */etc/rc.d/* enthält ebenfalls die Skripte für `init`.

Frage 95:

B: `tar -xvzf tarball.tar.gz` entpackt den tar-Ball.

zu A: `./configure` werden Sie wahrscheinlich anschließend benötigen, um den Inhalt des tar-Balls zu konfigurieren.

zu C: `tar -xvfz tarball.tar.gz` enthält einen Fehler. Die Option `f` (file) muss als letzte Option angegeben werden. Die Reihenfolge der anderen Optionen ist beliebig.

zu D: `tar -tf tarball.tar.gz` listet lediglich die Dateien auf, die in diesem tar-Ball enthalten sind.

zu E: `make` wird in diesem Zusammenhang verwendet, um den Inhalt des tar-Balls zu kompilieren.

Frage 96:

`tar -tf archiv.tar` listet den Inhalt der Datei *archtiv.tar* auf. Die Option `-t` steht hier für »table« und die Option `-f` für »file«.

Frage 97:

A: */usr/src* ist laut Filesystem Hierarchy Standard (FHS) das vorgesehene Verzeichnis.

zu B: */var/src* ist kein übliches Verzeichnis.

zu C: *~/src* wird von Benutzern teilweise verwendet, ist aber auch nicht das laut Filesystem Hierarchy Standard (FHS) vorgesehene Verzeichnis.

zu D: */etc* sollte ausschließlich Konfigurationsdateien enthalten.

zu E: */lib* enthält Programmbibliotheken (libraries).

Frage 98:

E: `./configure` ist hier richtig. Normalerweise befindet sich in einem tar-Ball ein Skript mit der Bezeichnung `configure`. Da Linux weder bei der Ausführung von Programmen noch Skripts den aktuellen Pfad durchsucht, muss die Zeichenkombination `./` dem Skript vorangestellt werden.

zu A: `make` kompiliert das Programm. Es muss aber zunächst konfiguriert werden.

zu B: `make install` installiert das fertig konfigurierte und kompilierte Programm.

zu C: `tar -xvzf tarball.tar.gz` packt den tar-Ball aus. Das ist hier aber bereits durchgeführt worden.

zu D: `CONFIGURE` gibt es normalerweise weder als Skript, noch als Programm. Es handelt sich um eine freie Erfindung.

Frage 99:

A: `bzip2` und D: `gzip` dienen der Komprimierung von Dateien.

zu B: `tar` ist zwar auch fähig, Dateien zu komprimieren, dient aber primär der Archivierung.

zu C: `cpio` ist ebenfalls hauptsächlich für Archivierung gedacht.

zu E: `gunzip` dekomprimiert Dateien.

Frage 100:

`make install` wird das Programm installieren.

Frage 101:

A: `zcat`, C: `gunzip` und D: `gzip` können verwendet werden, um komprimierte Dateien mit der Erweiterung *gz* zu öffnen.

zu B: `bzip2` entpackt (unter Verwendung der Option `-d`) Dateien vom Typ *bz*, *bz2*, *tbz* und *tbz2*.

zu E: `dpkg` ist der Debian Packager.

Frage 102:

D: `uname -r` ist das richtige Kommando mit der richtigen Option. Der Schalter `-r` sorgt dafür, dass ausschließlich die Release-Informationen des laufenden Kernels angezeigt werden.

zu A: `uname -a` zeigt zwar die benötigten Informationen an, aber auch nicht benötigte Daten.

zu B: `uname -s` zeigt ausschließlich den Kernel-Namen an. Die Version ist hier nicht enthalten.

zu C und E: `lsb_release -a` und `lsb_release` zeigen Informationen zur verwendeten Distribution an, aber nicht die Version des laufenden Kernels.

Frage 103:

B: `tar` ist ein Kurzwort für Tape-Archiver, und es handelt sich hierbei um ein Archivierungsprogramm, das ursprünglich zur Vorbereitung von Datensicherungen auf Band entwickelt wurde.

zu A: `cpio` ist ebenfalls ein Archivierungstool. Es diente aber nie hauptsächlich zur Vorbereitung von Backups auf Band.

zu C: `dd` kopiert oder konvertiert Dateien. Es handelt sich hierbei um ein sehr universell einsetzbares Werkzeug. Bedenken Sie, dass Linux alles als Datei betrachtet, was kein Prozess ist.

zu D: `rsync` kopiert Dateien über das Netzwerk.

zu E: `mt` ist ein Programm zur Steuerung von Bandlaufwerken. Es bereitet das eigentliche Backup jedoch nicht vor.

Frage 104:

`dd if=/dev/sdb of=usbstick.img` ist die korrekte Lösung. Versuchen Sie nicht, in der Prüfung durch zusätzliche Optionen, wie `ibs`, `obs` oder `bs` zu glänzen. Das könnte zu einem Punkteabzug führen.

Frage 105:

C: */home* enthält die Daten der Benutzer. Das ist für eine Datensicherung meist das wichtigste Verzeichnis.

D: */var* enthält unter anderem Warteschlangen bei Mailservern und Webseiteninhalte, sollte also auch regelmäßig gesichert werden.

zu A: */proc* enthält das */proc*-Dateisystem. Da es sich hierbei um Informationen handelt, die physikalisch betrachtet im Arbeitsspeicher eines Computers liegen, wäre eine Datensicherung absolut sinnlos.

zu B: */bin* enthält ausführbare Programme. Dieses Verzeichnis ist normalerweise keinen großen Änderungen unterworfen und muss deshalb nur selten gesichert werden.

zu E: */usr* enthält ebenfalls hauptsächlich ausführbare Programme.

Frage 106:

C: */proc* enthält das */proc*-Dateisystem. Eine Sicherung ist, wie bereits in der vorangehenden Frage erläutert, sinnlos.

E: */lib* enthält Programmbibliotheken. Da diese Bibliotheken wahrscheinlich ohnehin noch auf weiteren Datenträgern vorhanden sind, ist eine Datensicherung vergleichsweise nicht so wichtig.

zu A und B: */home* und */var* enthalten, wie bereits erwähnt, meist Benutzerdaten und sollten deshalb unbedingt gesichert werden.

zu D: */etc* enthält eventuell viele manuell angepasste Konfigurationen. Dieses Verzeichnis sollte also unbedingt gesichert werden, damit der Aufwand bei der Wiederherstellung eines defekten Systems minimiert werden kann.

Frage 107:

A: Geben Sie `dmesg` ein und suchen Sie nach Fehlern bezüglich USB und Dateisystem.

E: Suchen Sie nach `kernel` in der Datei */var/log/messages*.

Beides sind gängige Methoden, um nach kernelbezogenen Fehlern zu suchen.

zu B: Sichern Sie die Festplatte und führen Sie eine Neuformatierung durch. Stellen Sie die Daten anschließend wieder her.

Das kann man zwar machen und möglicherweise läuft die Festplatte danach auch stabil, allerdings war hier lediglich eine Analyse gefordert und kein Reparaturvorschlag.

zu C und D: Die Dateien aus den Antworten C und D sind frei erfunden.

Frage 108:

C: `modprobe` bietet genau die gewünschte Funktionalität. Sie müssen nicht einmal eine Option angeben, wenn Abhängigkeiten automatisch aufgelöst werden sollen.

zu A: `lsmod` listet lediglich geladene Module auf.

zu B: `insmod` lädt ein angegebenes Modul, löst aber Abhängigkeiten nicht automatisch auf.

zu D: `rmmod` entfernt ein geladenes Modul.

zu E: `insmod /auto` funktioniert nicht. Es gibt diese Option nämlich für `insmod` nicht.

Frage 109:

C: `strings` gibt die druckbaren Zeichen eines Programms aus.

zu A: `vi` ist für diese Aufgabe ungeeignet.

zu B: `strace` zeigt an, welche Systemaufrufe ein Programm ausführt.

zu D: `ltrace` überprüft, welche Bibliotheken ein Programm aufruft.

zu E: `lsof` zeigt geöffnete Dateien an.

Frage 110:

A: */etc/sysctl.conf* ist die Konfigurationsdatei des Programms `sysctl`. Dieses Programm konfiguriert Kernel-Parameter.

zu B: */etc/network/interfaces* legt Schnittstellenparameter wie IP-Adressen und Netzwerkmasken fest.

zu C: */proc/sys/net/ipv4/ip_forward* ist eine Pseudodatei des */proc*-Dateisystems. Hier könnte nur eine der geforderten Einstellungen konfiguriert werden, und es handelt sich auch nicht um eine Konfigurationsdatei.

zu D: */proc/sys/net/ipv4/icmp_echo_ignore_all*: Für diese Datei gilt sinngemäß die Erläuterung zu Antwort C.

zu E: */etc/hosts.deny* steuert das Verhalten von TCP-Wrappern.

Frage 111:

A: `echo 1 > /proc/sys/net/ipv4/ip_forward`

B: `sysctl -w net.ipv4.ip_forward=1`

Beide Kommandos werden den entsprechenden Kernel-Parameter festlegen. Wenn Sie diese Einstellung dauerhaft konfigurieren wollen, sollten Sie die Datei */etc/sysctl.conf* bearbeiten.

zu C: `echo 0 > /proc/sys/net/ipv4/ip_forward` deaktiviert die Weiterleitung von IP-Paketen.

zu D: `sysctl -w net.ipv4.ip_forward=0` deaktiviert ebenfalls die Weiterleitung von IP-Paketen.

zu E: `sysctl -n net.ipv4.ip_forward` dient lediglich der Überprüfung des momentan eingestellten Wertes für diese Option.

Frage 112:

/etc/sysctl.conf ist hier die richtige Konfigurationsdatei. Erstellen Sie in dieser Datei folgenden Eintrag:

```
net.ipv4.icmp_echo_ignore_all = 1
```

Frage 113:

A: `iostat`, C: `top` und D: `sar` sind geeignet, die aktuelle Prozessoraktivität anzuzeigen.

zu B: `ps` zeigt zwar, welche Prozesse laufen und wie viel Systemzeit diese bereits verbraucht haben, nicht aber die aktuelle CPU-Last.

zu E: `free` zeigt die aktuelle Verwendung des Arbeitsspeichers und die Swap-Auslastung.

Frage 114:

A: `vmstat`, D: `iostat` und E: `sar` sind für diese Aufgabe optimal. Bei diesen Programmen können Sie sowohl die Anzahl der Messwerte als auch den Abstand zwischen diesen Werten angeben (z. B. `vmstat 3 5`).

zu B: `ps` zeigt lediglich einmal eine Liste der laufenden Prozesse mit den zu diesen Prozessen gehörenden Leistungsdaten an.

zu C: `top` zeigt fortwährend Leistungsdaten zu einzelnen Prozessen an.

Frage 115:

C: `top` zeigt unter anderem an, in welchem Status sich ein Programm befindet. Das beinhaltet auch den Status Zombie.

zu A: `sar`, B: `iostat` und D: `vmstat` zeigen allesamt keine Informationen zu einzelnen Prozesse an.

zu E: `ps -A` wäre eine gute Lösung, wenn zusätzlich die Option `-x` verwendet worden wäre. Ohne diese Option werden keine Statusinformationen zu Prozessen angezeigt.

Frage 116:

A: `top`, D: `uptime` und E: `w` zeigen unter anderem die Uptime an. So können Sie ermitteln, wann ein System zuletzt neu gestartet wurde.

zu B: `when` ist ein einfaches Kalenderprogramm.

zu C: `whatis` zeigt Kurzbeschreibungen aus Manpages an.

Frage 117:

`lsof` ist hierfür das optimale Werkzeug.

Frage 118:

B: Cacti, C: Nagios und D: MRTG werden mithilfe eines Webbrowsers bedient und benötigen serverseitig für ihren Betrieb PHP und MySQL.

zu A: *collectd* ist lediglich ein Daemon, der Leistungsdaten sammelt und weder mit PHP, noch mit MySQL etwas anfangen kann.

zu E: `vmstat` ist ein Konsolenprogramm zur Anzeige von Leistungsstatistiken und benötigt ebenfalls weder PHP noch MySQL.

Frage 119:

```
systemctl start pcscd
systemctl start pcscd.service
```

sind jeweils gültige Antworten auf diese Frage.

Frage 120:

B: `mkinitrd` erstellt eine initiale RAM-Disk unter Red Hat.

D: `mkinitramfs` erstellt eine initiale RAM-Disk unter Debian.

zu A: `update-grub` aktualisiert die Konfiguration von GRUB.

zu C: `lilo` ist der MAP-Installer des Bootloaders LILO.

zu E: `ldconfig` muss ausgeführt werden, wenn einem System neue Bibliotheken hinzugefügt wurden.

LPI 202

207 Domain Name Service (DNS)

Ohne DNS-Namensauflösung funktioniert in einem Netzwerk eigentlich fast gar nichts. Da sehr viele Störungen in Netzwerken auf DNS-Fehler zurückzuführen sind, sollten Sie sich mit diesem Thema besonders gut auskennen.

207.1 Grundlagen der DNS-Serverkonfiguration

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, BIND in der Funktion eines autoritativen und eines rekursiven Caching-only-DNS-Servers einzurichten. Dieses Lernziel umfasst die Verwaltung des laufenden Servers und die Konfiguration der Protokollierung.

Wichtigste Wissensgebiete:

- BIND 9.x-Konfigurationsdateien und Dienstprogramme
- Definition von BIND-Zonendateien in BIND-Konfigurationsdateien
- Nachladen von geänderten Konfigurations- und Zonendateien
- Kenntnis von alternativen DNS-Server *dnsmasq*, *djbdns* und *PowerDNS*

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/named.conf*
- */var/named/*
- `Rndc`
- `named-checkconf`
- `kill`
- `host`
- `dig`

Allgemeines

Zweifellos ist DNS einer der wichtigsten Dienste in TCP-/IP-Netzwerken überhaupt. DNS ist ein Akronym für Domain Name System und dient hauptsächlich der Auflö-

sung von Computernamen in IP-Adressen. Bevor es DNS gab, wurde die Namensauflösung lokal mithilfe der Datei */etc/hosts* durchgeführt. Die Pflege dieser Datei wurde jedoch immer aufwendiger, weil es im Laufe der Zeit immer mehr Computer gab. Wie Sie wissen, gibt es die Datei */etc/hosts* immer noch und sie kann auch nach wie vor verwendet werden.

Im November 1983 wurde DNS zum ersten Mal gleich in zwei verschiedenen RFCs beschrieben. Bei RFC 882 handelt es sich um eine Beschreibung der Konzepte und Einrichtungen für das Domain Name System, während RFC 883 die Spezifikationen für die Implementierung von DNS enthält.

Die Implementierung von DNS, die unter Linux verwendet wird, ist BIND. Dieses System ist gleichzeitig die Grundlage für die Namensauflösung im gesamten Internet. BIND wurde von Studenten an der Universität Berkeley entwickelt, woher er auch seinen Namen hat. BIND steht für Berkeley Internet Name Daemon, wobei neuere Quellen merkwürdigerweise von Berkeley Internet Name Domain sprechen. Die Zuständigkeit für BIND liegt heute beim Internet Systems Consortium (ehemals Internet Software Consortium), wo Sie übrigens immer die neueste Version des DNS-Servers herunterladen können:

http://www.isc.org/software/bind

Zur Prüfungsvorbereitung empfehle ich Ihnen allerdings ein fertiges Paket, weil ein selbst kompilierter DNS-Server noch etlicher Nacharbeiten bedarf, bis er läuft. Sie müssten die Konfigurationsdateien und Startskripte dann nämlich komplett von Hand erstellen. Da die fertigen Pakete unter den verschiedenen Linux-Distributionen zum Teil erheblich unterschiedliche Pfad- und Dateinamen verwenden, empfehle ich Ihnen in diesem Fall, ein Red Hat-Derivat wie Scientific, CentOS oder Fedora zu verwenden. Diese Systeme sind in Bezug auf BIND relativ konservativ gebaut und deshalb prüfungsnah.

Das DNS-System ist weltweit betrachtet hierarchisch aufgebaut. An oberster Stelle stehen hierbei die Root-Server, welche die Stammzone hosten. Von der Existenz dieser Zone bekommt ein normaler Benutzer gar nichts mit, weil es bei der Angabe eines voll qualifizierten Domänennamens (FQDN; Fully Qualified Domain Name) nicht nötig ist, die Stammzone mitanzugeben. Ein solcher FQDN sähe normalerweise so aus:

www.rheinwerk-verlag.de

Da diese Namen, der Hierarchie folgend, von rechts nach links gelesen werden, ist der Punkt nach »de« die höchstrangige Domäne überhaupt. Danach folgt die Top-Level-Domain »de«, die von der Organisation *denic* verwaltet wird, und »rheinwerk-verlag« ist in dieser Hierarchie das letzte Glied. »www« ist lediglich ein Alias für einen Computernamen.

Tipp

Sie finden viele hochinteressante, vor allem auch statistische Informationen über die Root-Server unter *http://www.root-servers.org*.

Cache-only-DNS-Server

Ein Cache-only-DNS-Server verfügt selbst über keinerlei Zonendaten. Wenn er eine Anfrage von einem Client erhält, muss er seinerseits eine Anfrage an andere DNS-Server richten. Das Ergebnis dieser Abfrage speichert er in seinem Cache, damit er bei einer wiederholten Abfrage desselben Namens schneller antworten kann. Es gibt zwei Möglichkeiten, woher ein Cache-only-DNS-Server seine Informationen beziehen kann. Sie können den DNS-Server konfigurieren, um Anfragen an einen anderen DNS-Server (z. B. den DNS-Server eines Internet Service Providers) weiterzuleiten (forward). Alternativ können Sie das einfach bleiben lassen. Dann wird sich der DNS-Server normalerweise direkt an die Root-Server wenden, was aber aus Gründen der Geschwindigkeit nicht empfehlenswert ist.

Eine Abfrage über die Root-Server dauert zwar nur geringfügig länger als die Abfrage eines durchschnittlichen Forwarders, aber bei einer Webseite, die unter Umständen Werbebanner von zwanzig verschiedenen Firmen enthält, addieren sich diese Verzögerungen. Da Sie beide Konfigurationen auf den nächsten Seiten kennenlernen werden, können Sie sich selbst ein Bild davon machen.

Sollten Sie sich für eine paketbasierte Installation entschieden haben, beginnen Sie z. B. unter Fedora mit:

```
[root@arch-fc /]# yum install bind
```

Wenn Sie Debian als Betriebssystem einsetzen, verwenden Sie:

```
root@arch-deb:/# apt-get install bind9
```

Um Fehler von vornherein zu vermeiden, sollten Sie dafür sorgen, dass Ihr Testsystem auf dem neuesten Stand ist, indem Sie `yum update` (Debian: `apt-get update` gefolgt von `apt-get upgrade`) ausführen.

Im weiteren Verlauf des Textes werde ich mich hauptsächlich auf Fedora beziehen, weil dieses System von den Pfad- und Dateinamen her prüfungsnäher ist. Sie werden aber zu Debian immer Informationen in Klammern vorfinden.

Die Hauptkonfigurationsdatei von BIND ist */etc/named.conf* (Debian: */etc/bind/named.conf*). Sie werden in Abhängigkeit von der verwendeten Linux-Distribution eventuell eine oder mehrere `include`-Anweisungen in dieser Datei finden, die auf weitere Konfigurationsdateien verweisen. Diese Dateien befinden sich normaler-

weise im selben Verzeichnis wie die Hauptkonfigurationsdatei und werden von BIND gehandhabt, als wären sie Bestandteil der Datei *named.conf* selbst. Beispiel:

```
include "/etc/named.rfc1918.zones";
include "/etc/named.root.key";
```

Mittels `include` eingebundene Konfigurationsdateien sind übrigens für die Prüfung nicht von Bedeutung, aber Sie sollten für die Praxis wissen, dass es sie gibt.

Sie können jetzt einen ersten Test des Servers durchführen. Falls die Paketverwaltung den Server nicht automatisch gestartet haben sollte, holen Sie das jetzt nach:

```
[root@arch-fc ~]# systemctl start named
```

Damit BIND beim nächsten Reboot automatisch startet, sollten Sie *systemd* anweisen sich darum zu kümmern:

```
[root@arch-fc ~]# systemctl enable named
```

Wie Sie den Kommandos entnehmen können, heißt der Daemon, der BIND repräsentiert, *named*. Das ausführbare Programm `named` befindet sich normalerweise im Pfad */usr/sbin*. Für den ersten Test können Sie das Programm `nslookup` verwenden. Es ist zu berücksichtigen, dass BIND in vielen Grundkonfigurationen aus Sicherheitsgründen nur das Loopback-Device abhorcht. Sie müssen das bei `nslookup` entsprechend angeben. Im folgenden Beispiel sind die fett gedruckten Zeichen die Eingaben durch den Benutzer:

```
[root@arch-fc ~]# nslookup
> server 127.0.0.1
Default server: 127.0.0.1
Address: 127.0.0.1#53
> www.lpi.org
Server:         127.0.0.1
Address:    127.0.0.1#53
Non-authoritative answer:
Name:    www.lpi.org
Address: 24.215.7.162
```

Der Server hat den Namen *www.lpi.org* ordnungsgemäß in die IPv4-Adresse 24.215.7.162 aufgelöst. Aus der Tatsache, dass noch keinerlei Konfigurationen (wie z. B. Angabe eines Forwarders) vorgenommen worden sind, kann man schließen, dass sich der Server an einen Root-Server gewendet haben muss. Informationen über die Root-Server und deren IP-Adressen bezieht BIND aus der Datei */var/named/named.ca* (Debian: */etc/bind/db.root*).

Damit der DNS-Server auch von anderen Computern aus erreicht werden kann, muss dafür gesorgt werden, dass er an einer Netzwerkschnittstelle horcht (dieser Schritt ist

bei Debian übrigens nicht nötig). Zu diesem Zweck müssen Sie die Konfigurationsdatei */etc/named.conf* modifizieren. Der Eintrag `listen-on` in der Sektion `options` sieht nämlich z. B. unter Fedora so aus:

```
listen-on port 53 { 127.0.0.1; };
```

Sie müssen den Eintrag um die IP-Adresse(n) erweitern, an denen der Server horchen soll. Der Port 53 ist der Standardport für DNS und sollte nicht geändert werden. Der erweiterte Eintrag könnte z. B. so aussehen:

```
listen-on port 53 { 192.168.50.12; 127.0.0.1; };
```

Prüfungs- und Praxishinweis

Sie müssen bei der Konfiguration exakt die Syntax einhalten. Das bezieht sich insbesondere auf Semikola und Leerzeichen. Das Beste wird sein, wenn Sie sich einige Einträge in der *named.conf*-Datei ansehen und verinnerlichen. Sie werden feststellen, dass sich BIND nicht mehr starten lässt, wenn Sie hier auch nur den kleinsten Fehler machen. In der Prüfung werden Sie unter Umständen mit Ausschnitten fehlerhafter Konfigurationen konfrontiert und müssen dann den Fehler finden.

Damit der Zugriff funktioniert, muss auf der Firewall noch Port 53 geöffnet werden. DNS verwendet hauptsächlich UDP. Sie sollten aber für alle Fälle auch den entsprechenden TCP-Port öffnen. Bei Fedora erledigen Sie das dauerhaft, indem Sie in die Datei */etc/sysconfig/iptables* die folgenden beiden Zeilen eintragen:

```
-A INPUT -p udp --dport 53 -j ACCEPT
-A INPUT -p tcp --dport 53 -j ACCEPT
```

Sie können natürlich auch die entsprechenden `iptables`-Kommandos in ein Startskript einbinden, wenn Sie eine andere Linux-Distribution verwenden, die per Voreinstellung über eine konfigurierte Firewall verfügt. Die passenden Kommandos sähen dann so aus:

```
iptables -t filter -A INPUT -p udp --dport 53 -j ACCEPT
iptables -t filter -A INPUT -p tcp --dport 53 -j ACCEPT
```

Das nächste Hindernis, auf das Sie stoßen könnten, befindet sich ebenfalls in der Datei *named.conf*. Sehen Sie sich den folgenden Eintrag an:

```
allow-query     { localhost; };
```

Der Eintrag erlaubt es der lokalen Maschine lediglich, selbst auf den DNS-Dienst zuzugreifen. Da es hier gerade nicht um Sicherheit geht, ändern Sie den Eintrag großzügig, sodass jeder den Server abfragen darf:

```
allow-query     { any; };
```

Anschließend muss der Server die Konfigurationsdateien neu einlesen. Das geht unabhängig von der verwendeten Linux-Distribution mit:

```
[root@arch-fc /]# rndc reload
server reload successful
```

Um den Cache-only-DNS-Server zu komplettieren (und schneller zu machen), sollte er einen Eintrag vom Typ `forwarders` erhalten. Dieser Eintrag gehört in die Sektion `options` der Datei *named.conf* und verweist meist auf einen oder mehrere DNS-Server eines Internet Providers.

```
forwarders { 24.215.7.99; 85.88.1.92; 141.1.1.1; };
```

Wenn Sie die `forwarders` eingerichtet haben, müssen Sie noch einmal das Kommando `rndc reload` ausführen und können anschließend mit einem anderen Computer mittels `nslookup` das Ergebnis Ihrer Bemühungen überprüfen. Die fett gedruckten Zeichen sind die Eingaben des Benutzers:

```
harald@arch-deb-book:~$ nslookup
> server 192.168.50.12
Default server: 192.168.50.12
Address: 192.168.50.12#53
> www.suse.de
Server:.........192.168.50.12
Address:       192.168.50.12#53
Non-authoritative answer:
www.suse.de        canonical name = turing.suse.de.
Name:   turing.suse.de
Address: 195.135.220.3
```

Bei der Abfrage eines einzelnen Hostnamens wie hier, macht es sich natürlich nicht bemerkbar, dass der Server jetzt nicht mehr die Root-Server abfragen muss, sondern direkt mit einem nahe gelegenen DNS-Server kommunizieren kann. Beim Besuch komplexer Webseiten, die Webcontent aus unterschiedlichen Domänen anfordern, wird der Geschwindigkeitszuwachs jedoch spürbar.

Dateien, Verzeichnisse und Kommandos

Sie haben während der Konfiguration des Cache-only-DNS-Servers bereits die wichtigsten Dateien, Verzeichnisse und Kommandos kennengelernt, die mit BIND im Zusammenhang stehen. Deshalb finden Sie hier nur noch ein paar Zusammenfassungen und Ergänzungen, die Sie für die Prüfung benötigen.

named, rndc und kill

Das ausführbare Programm von BIND finden Sie unter dem vollen Pfad */usr/sbin/named*. Normalerweise müssen Sie dieses Programm nicht von Hand ausführen. Die Startskripte sind in Abhängigkeit von der verwendeten Linux-Distribution unterschiedlich benannt. Gängig sind aktuell diese beiden Kommandos, wenn Sie BIND starten wollen:

```
[root@arch-fc /]# systemctl start named
root@arch-deb:/# systemctl start bind9
```

Wenn Sie Konfigurationsänderungen an BIND vorgenommen haben, müssen Sie `named` veranlassen, seine Konfigurationsdateien neu einzulesen. Es gibt verschiedene Möglichkeiten, um das zu tun:

```
[root@arch-fc /]# systemctl restart named
[root@arch-fc /]# kill -HUP `pidof named`
[root@arch-fc /]# rndc reload
```

Die ersten beiden Methoden würden bei vielen anderen Diensten ebenfalls bewirken, dass Konfigurationsdateien neu eingelesen werden. Das Programm `rndc` ist nur zur Steuerung von BIND geeignet und deshalb in der Prüfung natürlich die erste Wahl in diesem Zusammenhang. Das Programm `rndc` versteht diese wesentlichen Optionen:

- `reload` lädt die Konfiguration und Zonendateien neu.
- `halt` beendet `named` sofort. Eventuell laufende Schreibvorgänge und Zonentransfers werden abgebrochen.
- `stop` beendet `named` normal. Laufende Schreibvorgänge und Zonentransfers werden abgeschlossen.
- `freeze` friert Schreibvorgänge auf Zonendateien ein. Das ist sinnvoll, wenn Sie beabsichtigen Zonendateien manuell zu bearbeiten, in die `named` normalerweise dynamische Aktualisierungen von Clients schreibt.
- `thaw` taut Schreibvorgänge wieder auf, wenn Sie mit dem Bearbeiten der Zonendateien fertig sind. Sie sollten anschließend ein `reload` durchführen.

named-checkconf

Wenn Sie an der Konfiguration von BIND Änderungen vorgenommen haben, können Sie mithilfe des Programms `named-checkconf` überprüfen, ob die Konfigurationsdateien konsistent sind. Das ist keine schlechte Idee, denn wenn Sie den Server einfach veranlassen, die Konfiguration neu einzulesen oder Sie diesen neu starten, käme es zu einem Ausfall des DNS-Dienstes, wenn Sie einen Fehler gemacht haben. Wenn die Konfiguration in Ordnung ist, wird der Befehl `named-checkconf` keine Antwort liefern. Bei einem Fehler bekommen Sie einen entsprechenden Hinweis:

```
root@archangel:/ # named-checkconf
/etc/bind/named.conf:2: expected quoted string near '/'
```

Eine interessante Option ist `-z`. Wenn Sie diesen Schalter verwenden, werden zusätzlich alle Zonendateien geladen, die in der Konfiguration enthalten sind, so, wie der Server es selbst bei einem Neustart machen würde. Sie sehen dann sehr genau, ob alles in Ordnung ist:

```
root@archangel:/ # named-checkconf -z
zone localhost/IN: loaded serial 1
zone 127.in-addr.arpa/IN: loaded serial 1
zone 0.in-addr.arpa/IN: loaded serial 1
zone 255.in-addr.arpa/IN: loaded serial 1
zone schulung.de/IN: loaded serial 2016092701
zone datev.indisoft.de/IN: loaded serial 2016092701
zone datev-ms.indisoft.de/IN: loaded serial 2016092701
zone 10.in-addr.arpa/IN: loaded serial 1
zone 16.172.in-addr.arpa/IN: loaded serial 1
```

/var/named

Das Verzeichnis */var/named* enthält normalerweise die Zonendateien eines BIND-Servers und ggf. sonstige Verzeichnisse und Dateien, die zu `named` gehören. Es handelt sich hier allerdings lediglich um ein traditionell positioniertes Verzeichnis, das Sie für die Prüfung kennen müssen. Die tatsächliche Position dieses Verzeichnisses im Verzeichnisbaum ist allerdings Ihnen überlassen. Sie müssen die Position des Verzeichnisses in der Datei *named.conf* unter `options` eintragen. Bei Fedora sieht die Standardeinstellung so aus:

```
directory       "/var/named";
```

Die gleiche Anweisung steht bei Debian in der ausgelagerten Konfigurationsdatei */etc/bind/named.conf.options*, hat aber ein anderes Ziel:

```
directory       "/var/cache/bind";
```

Im Abschnitt 207.2, »Erstellen und Pflegen von DNS-Zonen«, werden Sie in diesem Verzeichnis Zonendateien erstellen. Vorher müssen Sie aber lernen, wie diese Zonendateien definiert werden.

/etc/named.conf

Im Laufe des Kapitels haben Sie schon einiges über die Hauptkonfigurationsdatei des BIND-Servers gelernt. Es gibt aber noch weitere Einstellungen in dieser Datei, die Sie kennen müssen.

BIND verwendet für die Protokollierung `syslogd`, wie andere Dienste auch, solange er aus der Datei *named.conf* keine anderen Anweisungen erhält. Debian und Ubuntu verzichten auf einen Eintrag dieses Typs, aber bei Fedora wird man fündig:

```
logging {
        channel default_debug {
                file "data/named.run";
                severity warning;
                category lame-servers { null; };
        };
};
```

Die Zeile `channel default_debug` ist lediglich eine Definition für den Eintragstyp `logging`. Es kann mehrere solcher Definitionen geben. Protokolliert wird in die Datei */var/named/data/named.run*. Es wird nämlich die Option `directory` auf den lediglich relativ angegebenen Pfad hinter der Option `file` angewendet. Mit `severity` wird das Logging-Level angegeben, ähnlich wie Sie es von der *syslog.conf*-Datei her kennen. Der separate Eintrag `category lame-servers { null; };` unterdrückt die Protokollierung fehlerhafter Delegierungen. Dieser Eintrag ist nicht standardmäßig vorhanden, wird aber bei Prüfungen manchmal abgefragt.

Forward-Lookup-Zonen

In der Datei *named.conf* werden auch die Zonendateien für den Server definiert. Zonen, die für die Auflösung von Hostnamen in IP-Adressen verwendet werden, sind Forward-Lookup-Zonen. Die Definition einer solchen Zone hat folgendes Format:

```
zone "homelinux.net" IN {
        type master;
        file "homelinux.net.zone";
};
```

Die erste Zeile gibt den Namen der Zone an. Die Bezeichnung `IN` ist inzwischen übrigens optional und kann auch weggelassen werden. In der zweiten Zeile wird der Typ der Zone festgelegt.

Prüfungstipp

Beachten Sie unbedingt, dass die Bezeichnung für den Zonentyp nicht »primary« ist, wie man vermuten könnte, sondern »master«.

Die dritte Zeile gibt den verwendeten Dateinamen der Zone an. Die Namenskonvention ist nicht fest vorgeschrieben und man kann deshalb seine eigenen Zonendateien benennen, wie man will. In Kombination mit dem Eintrag `directory` in der Sektion `options` ergibt sich also der Pfad:

/var/named/homelinux.net.zone

Die Definition einer Zone kann auch in einer fortlaufenden Zeile vorgenommen werden. Die Syntax ist aber ansonsten exakt identisch. Leider lässt sich ein solcher Eintrag in einem Buch (mangels Breite) nicht darstellen. Wenn Sie die Datei */etc/named.rfc1918.zones* betrachten (Debian: */etc/bind/zones.rfc1918*), dürfte der Zusammenhang klar werden. Der Aufbau der Zonendateien ist Thema des nächsten Abschnitts.

Reverse-Lookup-Zonen

Eine Reverse-Lookup-Zone dient der Auflösung von IP-Adressen in Hostnamen. Sie wird im Grunde genommen genauso deklariert wie eine Forward-Lookup-Zone. Zu beachten ist hier allerdings die Konvention für die Bezeichnung einer solchen Zone. Die Bezeichnung muss genau stimmen, damit `named` die Zone lädt. Das Beispiel zeigt eine Zone für das Netzwerk 192.168.50.0/24:

```
zone "50.168.192.in-addr.arpa" IN {
        type master;
        file "192.168.50.zone";
};
```

Informationen über den Aufbau dieser Zonen finden Sie ebenfalls im nächsten Abschnitt.

Root-Server

Hinweise auf die Root-Server findet BIND ebenfalls in einer Zonendefinition. Diese Definition ist vom Typ `hint`, woraus sich schließen lässt, dass der Server diese Zone nicht selbst hostet, sondern die Server, die in der entsprechenden Zone aufgelistet sind:

```
zone "." IN {
        type hint;
        file "named.ca";
};
```

Wenn ein BIND-Server selbst ein Root-Server sein soll, muss der Eintrag `type` von `hint` auf `master` geändert und eine neue Zonendatei erstellt werden.

/usr/bin/dig, /usr/bin/host und nslookup

Die Programme `dig`, `host` und `nslookup` dienen der Diagnose bei Problemen mit der Namensauflösung. Das erheblich ältere Programm `nslookup` soll irgendwann einmal vollständig von `dig` und `host` abgelöst werden. Die Verwendungszwecke sind dementsprechend ähnlich, wobei die beiden neueren Programme in der Summe den grö-

ßeren Funktionsumfang bieten. Eine wichtige Gemeinsamkeit aller drei Programme ist ihr Verhalten bei der Auflösung von Namen in IP-Adressen oder umgekehrt. Sie fragen nicht das lokal installierte Betriebssystem ab, das sich dann um die DNS-Abfragen kümmern muss, sondern wenden sich stattdessen direkt selbst an die Nameserver, sodass bei einer Serverdiagnose mit diesen Tools ein lokaler Client-Fehler von vornherein als Fehlerquelle ausgeschlossen werden kann.

Mit dem Kommando `host` sind sowohl einfache Namensauflösungsanfragen als auch Reverse-Lookup-Anfragen durchführbar. Das Programm liefert (solange man es nicht mit `-v` in den Verbose Mode schaltet) kurze und klare Antworten ohne jede Zusatzinformationen:

```
[root@fedora10 ~]# host www.rheinwerk-verlag.de
www.rheinwerk-verlag.de has address 85.88.3.146
```

Umgekehrt liefert eine Reverse-Abfrage:

```
[root@fedora10 ~]# host 85.88.3.146
146.3.88.85.in-addr.arpa domain name pointer rheinwerk-verlag.de.
```

Ähnlich wie `nslookup` können Sie dem Programm `host` den Typ der Abfrage als Option übergeben. Dazu geben Sie die Option `-t`, gefolgt vom Typ der Abfrage, mit auf der Kommandozeile ein, wie das folgende Beispiel anhand des MX-Eintrags demonstriert:

```
archangel:~ # host -t MX  efdah.com
efdah.com mail is handled by 2 efdah.com.s8a1.psmtp.com.
efdah.com mail is handled by 4 efdah.com.s8a2.psmtp.com.
efdah.com mail is handled by 6 efdah.com.s8b1.psmtp.com.
efdah.com mail is handled by 8 efdah.com.s8b2.psmtp.com.
```

Testen Sie bitte auch die Ausgabe mit der Option `-v`. Diese wurde hier aus Platzgründen nicht abgedruckt.

Mit dem Programm `dig` erhalten Sie per Voreinstellung erheblich umfangreichere Informationen. Außerdem können Sie schon direkt auf der Kommandozeile den abzufragenden DNS-Server angeben, während das Programm `host` den DNS-Server verwendet, der in der Datei */etc/resolv.conf* eingetragen ist. Das folgende Beispiel zeigt die (um ca. 20 Zeilen gekürzte) Ausgabe einer MX-Abfrage an den DNS-Server mit der IP-Adresse 198.151.35.5:

```
archangel:~ # dig @198.151.35.5 -t MX efdah.com
; <<>> DiG 9.3.2 <<>> @198.151.35.5 -t MX efdah.com
;; QUESTION SECTION:
;efdah.com.          IN   MX
;; ANSWER SECTION:
```

```
efdah.com.    600    IN   MX   8 efdah.com.s8b2.psmtp.com.
efdah.com.    600    IN   MX   2 efdah.com.s8a1.psmtp.com.
;; AUTHORITY SECTION:
efdah.com.   86400   IN   NS   ns1.dtc.cendant.com.
efdah.com.   86400   IN   NS   ns2.dtc.cendant.com.
;; ADDITIONAL SECTION:
efdah.com.s8b2.psmtp.com. 871 IN    A     64.18.7.14
efdah.com.s8a1.psmtp.com. 871 IN    A     64.18.7.10
ns1.dtc.cendant.com.      300 IN    A     198.151.35.5
ns2.dtc.cendant.com.      300 IN    A     198.151.36.1
```

Die Antwort, die `dig` erhält, beinhaltet nicht nur die abgefragten MX-Einträge, sondern auch die Namen der für diese Einträge zuständigen DNS-Server. Im unteren Abschnitt werden auch die IP-Adressen der zuständigen Mailserver und DNS-Server gelistet. Alles in allem ist `dig` ein sehr umfangreiches Diagnoseprogramm. Es lohnt sich in jedem Fall ein Blick auf `man dig`.

Last but not least wäre in dieser Gattung noch `nslookup` zu nennen. Sie sollten `nslookup` in jedem Fall kennen. Es ist bisher noch kein DNS-Problem bekannt, das man nicht mit `nslookup` hätte diagnostizieren können. Wenn Sie `nslookup` beherrschen, sind Sie dazu in der Lage, plattformübergreifend zu arbeiten, weil dieses Tool auch (sogar unter demselben Namen) auf Windows-Hosts existiert. Ein besonderes Merkmal von `nslookup` ist die interaktive Verwendbarkeit. Das ist besonders vorteilhaft, wenn Sie mehrere Abfragen hintereinander durchführen möchten. Die Ausgabe ist recht übersichtlich, und Sie haben auch hier (genau wie bei `dig`) die Möglichkeit, einen alternativen DNS-Server anzugeben, um genau diesen zu überprüfen.

Im folgenden Beispiel wird `nslookup` interaktiv gestartet. Anschließend wird der abzufragende Server auf die IP-Adresse 198.151.35.5 geändert. Danach wird der Abfragetyp auf MX eingestellt und zum Schluss wird die Abfrage nach den zuständigen Mailservern für *efdah.com* durchgeführt. Welche Eingaben innerhalb von `nslookup` durch den Benutzer ausgeführt worden sind, erkennen Sie am `nslookup`-Prompt `>` und am Fettdruck der Kommandos. Bei den anderen Zeilen handelt es sich jeweils um Antworten des Programms. Die Ausgaben von `nslookup` wurden diesmal lediglich um zwei Zeilen gekürzt:

```
archangel:~ # nslookup
> server 198.151.35.5
Default server: 198.151.35.5
Address: 198.151.35.5#53
> set type=mx
> efdah.com
Server:         198.151.35.5
Address:        198.151.35.5#53
```

```
efdah.com      mail exchanger = 8 efdah.com.s8b2.psmtp.com.
efdah.com      mail exchanger = 2 efdah.com.s8a1.psmtp.com
```

Alternative DNS-Server

dnsmasq

dnsmasq ist ein leicht konfigurierbarer DNS-Server und DHCP-Server. Sie können ihn in kleinen Firmenumgebungen als schlanke Alternative zu BIND und einem ausgewachsenen DHCP-Server einsetzen. Die Installation können Sie einfach über Ihr Paketmanagement durchführen. Das entsprechende Paket heißt einfach *dnsmasq*, unabhängig davon, welche Linux-Distribution Sie verwenden. Nach Abschluss der Installation nimmt *dnsmasq*, zumindest was DNS-Caching anbelangt, sofort seine Arbeit auf. Zu diesem Zweck verwendet er den DNS-Server als Forwarder, den er in der Datei *resolv.conf* vorfindet.

dnsmasq verwendet keine Zonendateien für die Namensauflösung. Stattdessen liest er in der clientseitigen Datei */etc/hosts*. Sie können die Einträge in der Datei *hosts* einfach nach deren normaler Syntax vornehmen:

```
80.67.16.8      compi1.lpic-1.de      compi1
80.67.16.9      compi2.lpic-2.de      compi2
```

Beachten Sie, dass auf einem BIND-Server zwei separate Zonen notwendig gewesen wären, um die Namen dieser beiden Computer zu veröffentlichen, weil diese Computer zwei verschiedenen Domänen angehören. *dnsmasq* genügen hierfür zwei Einträge in der *hosts*-Datei.

Um die DHCP-Funktionalität von *dnsmasq* zu nutzen, müssen Sie in der Datei */etc/dnsmasq.conf* die folgende Zeile ändern:

```
#dhcp-range=192.168.0.50,192.168.0.150,12h
```

Entfernen Sie hier einfach das Kommentarzeichen und passen Sie die beiden IP-Adressen, die den Anfang und das Ende des Bereichs zur Verteilung definieren, Ihren Bedürfnissen an. Der Wert `12h` gibt die Länge der Lease-Dauer an. Wenn Sie mehr als nur einen DHCP-Bereich benötigen, können Sie einfach mehrere dieser Zeilen in die Konfigurationsdatei einfügen.

djbdns

Ein weiterer DNS-Server ist *djbdns*. Der Name ist auf den Entwickler Daniel J. Bernstein zurückzuführen, der übrigens auch *qmail* programmiert hat. Das Besondere an diesem Server ist die Tatsache, dass er für jede Aufgabe, die ein DNS-Server ausführen muss, einen eigenen Daemon bereitstellt. So gibt es einen Daemon zur Beantwor-

tung von Anfragen, einen für das Caching und einen für Zonentransfers, um nur die Wichtigsten zu nennen.

Der Entwickler stuft seinen Server als erheblich sicherer ein als BIND und hat ein Preisgeld von 1.000 USD ausgesetzt, wenn jemand einen Sicherheitsmangel in *djbdns* nachweisen kann.

Aufgrund der geringen Prüfungsrelevanz und der hohen Komplexität der Konfiguration soll hier nicht auf weitere Einzelheiten eingegangen werden.

PowerDNS

PowerDNS ist ein DNS-Server, der auf verschiedenste Backends zugreifen kann. Er kann nicht nur unter Linux, sondern auch unter NetBSD, FreeBSD, OpenBSD, MAC OS X und Windows betrieben werden.

Sowohl bei Debian, als auch bei Red Hat-basierten Distributionen beginnen die Bezeichnungen der Pakete, die mit PowerDNS in Verbindung stehen, mit *pdns*. Der eigentliche Server ist im Paket *pdns-server* enthalten. Beachten Sie bitte, dass der passende Cache-DNS-Server in einem separaten Programmpaket kommt und nach der Installation ein separater Daemon ist. Der Name ist *pdns-recursor*. Bei den anderen Paketen handelt es sich hauptsächlich um Backends, auf die der Server zugreifen kann. Hier kommen, abgesehen von normalen BIND-Zonendateien, u. a. die folgenden Backends in Betracht:

- LDAP
- mySQL
- Oracle
- PostgreSQL
- SQLite
- OpenDBX
- IBM DB2

Die Verwaltung des Servers kann über ein Webfrontend mit dem Namen Poweradmin durchgeführt werden. Sie benötigen hierfür einen Webserver mit PHP5-Unterstützung. Das Frontend selbst erhalten Sie auf dieser Website:

https://www.poweradmin.org

Die genaue Konfiguration von PowerDNS ist (noch) nicht prüfungsrelevant, es finden sich aber bereits einige Anleitungen darüber auf den einschlägig bekannten Webseiten, wie Sie zu einem solchen Server kommen, wenn Sie jetzt (völlig zu Recht) neugierig geworden sind. Beachten Sie aber, dass PowerDNS keine dynamischen Aktualisierungen entgegennimmt, falls Sie diese Funktionalität benötigen.

207.2 Erstellen und Pflegen von DNS-Zonen

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Zonendateien für Forward- oder Reverse-Zonen oder einen Root-Level-Server zu erstellen. Dieses Lernziel beinhaltet das korrekte Setzen der Werte der Record-Einträge, das Hinzufügen von Hosts zu Zonen und von Zonen zum DNS. Ein Prüfling sollte weiterhin Zonen an andere DNS-Server delegieren können.

Wichtigste Wissensgebiete:

- BIND 9.x-Konfigurationsdateien, Begriffe und Dienstprogramme
- Dienstprogramme zur Informationsabfrage an DNS-Servern
- Aufbau, Inhalt und Speicherorte von BIND-Zonendateien
- verschiedene Methoden zum Hinzufügen von Hosts in die Zonendateien, Reverse-Lookup-Zonen eingeschlossen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */var/named/*
- Syntax der Zonendateien
- Datenformate der Ressourceneinträge
- `named-checkzone`
- `named-compilezone`
- *masterfile-format*
- `dig`
- `nslookup`
- `host`

Allgemeines

Auf den folgenden Seiten können Sie nachlesen, wie Sie Ihre(n) DNS-Server weiter ausbauen können. In jedem Falle sollten Sie eine primäre Forward-Lookup-Zone und eine Reverse-Lookup-Zone erstellen, um Ihr eigenes Netzwerk mit Namensauflösung zu bedienen. Wenn Sie über mindestens zwei Computer verfügen, können Sie auch sekundäre Zonen und Delegierungen erstellen. Wie Zonen in der Datei *named.conf* definiert werden, haben Sie bereits gelesen. Lesen Sie nun über den Inhalt der Zonendateien.

Inhalt von Zonendateien und Eintragstypen

Die für den Namensauflösungsprozess erforderlichen Daten sind in sogenannten Zonendateien gespeichert. Es gibt Forward-Lookup-Zonen und Reverse-Lookup-Zonen. Aus einer Forward-Lookup-Zone wird eine IP-Adresse für einen bekannten Hostnamen ermittelt. Eine Reverse-Lookup-Zone stellt Informationen zu FQDNs (FullyQualified Domain Names) zur Verfügung, wenn die IP-Adresse bekannt ist. Eine typische Forward-Lookup-Zone ist wie folgt aufgebaut:

```
homelinux.net.      IN SOA  dns01.homelinux.net. root.localhost. (
                            2011100107 ; serial
                            86400      ; refresh (1 day)
                            7200       ; retry (2 hours)
                            604800     ; expire (1 week)
                            172800     ; minimum (2 days)
                              )
```

Der erste Eintrag einer Zone ist immer der SOA-Eintrag (Start of a Zone of Authority). Er bezeichnet den Autoritätsursprung einer DNS-Zone. Mit diesem Eintrag wird vor allem festgelegt, welcher DNS-Server die Änderungen für eine Zone entgegennimmt. Alle weiteren Zeilen dieses Eintrags bestimmen das Verhalten der Server, die eine sekundäre Zone von dieser Zone beziehen. Ein Semikolon wird in dieser Datei als Kommentarzeichen gewertet. Alles, was rechts vom Semikolon steht, wird von BIND nicht interpretiert. Da ein DNS-Eintrag immer in einer einzigen Zeile stehen muss, werden in diesem Eintrag die eigentlich »falschen« Zeilenumbrüche durch das Setzen einer öffnenden Klammer am Ende der ersten und durch eine schließende Klammer in der letzten Zeile aufgehoben. Die anderen Zeilen steuern das Verhalten der Server mit sekundären Zonen:

- `serial` ist die Seriennummer der Datei. Wenn ein Zonentransfer zu einer sekundären Zone stattfinden soll, vergleichen die Server zunächst die Seriennummer. Ist diese auf beiden Servern identisch, wird der Zonentransfer nicht ausgeführt. Bei einer manuellen Änderung der primären Zone muss also auch die Seriennummer erhöht werden. Sonst werden die Änderungen nicht transferiert.
- `refresh` legt fest, in welchen Intervallen ein Server versucht, die Zone zu transferieren.
- `Retry` – Sollte ein Zonentransfer fehlschlagen, wird nach der hier festgelegten Zeit ein Wiederholungsversuch stattfinden. Dieser Wert ist normalerweise etwas kleiner als der Wert für `refresh`.
- `expire` ist die Ablaufzeit. Wenn bis dahin der Zonentransfer immer wieder fehlschlägt, gibt der Server, der die sekundäre Zone hostet, auf. Nach einer erfolgten Reparatur muss der erste Zonentransfer manuell veranlasst werden. Wenn eine

sekundäre Zone abgelaufen ist, beantwortet der DNS-Server keine Abfragen mehr, die sich auf diese Zone beziehen.

- `minimum` gibt die minimale Zeit an, die ein Eintrag auf einem Cacheserver oder auch in einem Clientcache zwischengespeichert werden soll.

Der nächste Eintrag ist meist ein NS-Eintrag. Hier sind die für die Zone zuständigen Nameserver verzeichnet. Das sind Server mit primären und sekundären Zonen für diesen Namespace. Stubserver und Weiterleiter gehören nicht dazu. Wenn nur ein Nameserver vorhanden ist, ist dieser mit der SOA identisch.

```
        NS      dns01.homelinux.net.
        NS      dns02.homelinux.net.
```

Denken Sie bitte daran: Alle Einträge enden immer mit einem Punkt, der die Root-Domäne (auch Stammzone, Punktzone) repräsentiert.

MX-Einträge (MX = Mail Exchanger) repräsentieren die zuständigen SMTP-Server einer Domäne. An die hier genannten Server senden alle externen SMTP-Server die E-Mails für diese Domäne:

```
    IN      MX      10   smtp01.homelinux.net.
    IN      MX      20   smtp02.homelinux.net.
```

Sollten zur Ausfallsicherheit mehrere Server als Mail Exchanger aufgeführt sein, wird zunächst versucht, eine E-Mail an den Server mit dem geringeren Wert für die Priorität (hier 10) zuzustellen. Nur wenn dieser Server nicht erreichbar ist, wird die E-Mail-Zustellung an den Server mit dem höheren Prioritätswert versucht.

Prüfungstipp

Sie sollten sich den gerade geschilderten Sachverhalt besonders deshalb gut merken, weil er so paradox klingt.

Die grundlegenden Einträge sind vom Typ Host (A). Sie enden als Einzige auf eine IP-Adresse und erledigen die eigentliche Auflösung eines FQDN in eine IPv4-Adresse:

```
ipcop                   A           192.168.0.56
fedora15                A           192.168.0.57
```

Für die Auflösung von Hostnamen in IPv6-Adressen gibt es einen eigenen Eintragstyp mit der Bezeichnung AAAA. Diese Einträge haben ein solches Format:

```
fedora15    AAAA    2a01:198:5dd:0:207c:6eff:febe:22fc
ipcop       AAAA    2a01:198:5dd:0:dc89:f072:2b73:32a5
```

Der *CNAME* (Canonical Name) ist ein Alias für einen bestehenden Computer. Sehr typisch sind hier `www`, `ftp`, `smtp`, `pop3` und ähnlich prominente Namen:

```
www                     CNAME     ubuntu-server.homelinux.net.
ftp                     CNAME     ubuntu-server.homelinux.net.
```

In den Reverse-Lookup-Zonen gibt es noch einen weiteren Eintragstyp. Der *PTR* (Pointer) zeigt mit dem letzten Oktett einer IP-Adresse auf einen FQDN. Der SOA-Eintrag einer Reverse-Lookup-Datei ist mit dem Eintrag einer Forward-Lookup-Zone identisch. Aus diesem Grund soll hier nur der PTR-Eintrag selbst beschrieben werden:

```
57                      PTR       fedora15.homelinux.net.
56                      PTR       ipcop.homelinux.net.
```

Es ist deshalb nur die Angabe des letzten Oktetts einer IP-Adresse erforderlich, weil der Rest dieser IP-Adresse schon implizit im Namen der Zonendatei enthalten ist. Das bedeutet, dass `named` den Rest schon »kennt«. Die Hostnamen einer Reverse-Lookup-Zone müssen deshalb voll qualifiziert notiert werden, weil es denkbar wäre, dass ein Host einer fremden Domäne ebenfalls eine IP-Adresse aus diesem Netzwerksegment verwendet. Es gibt noch weitere Eintragstypen, aber diese sind für die anstehende Prüfung nicht von Belang.

Erstellen von primären Zonen

Auf den vorangehenden Seiten haben Sie alles gelesen, was Sie über die Definition und den Inhalt von Zonendateien wissen sollten. Sie müssen diese Informationen jetzt nur noch zusammenfügen. Zur Erstellung einer primären Forward-Lookup-Zone definieren Sie diese zunächst, wie beschrieben, in der Datei */etc/named.conf*. Beispiel:

```
zone "domain.tld" IN {
        type master;
        file "domain.tld.zone";
};
```

Hinweis

Moderne Linux-Distributionen lagern die Definitionen der Zonendateien normalerweise in separate Konfigurationsdateien aus. Fedora verwendet z. B. die Datei *named.rfc1918.zones*, während Debian Zonendefinitionen nach *zones.rfc1918* auslagert.

Erstellen Sie anschließend im Verzeichnis */var/named* die Datei für die Zone, so wie Sie diese in der Datei *named.conf* definiert haben. Es ist erforderlich, dass `named`

zumindest Leserechte an dieser Datei erhält. Um Fehler zu vermeiden, können Sie `named` einfach zum Eigentümer dieser Datei machen:

```
[root@arch-fc /]# chown named:named /var/named/domain.tld.zone
```

Erstellen Sie in dieser Datei als Erstes einen Eintrag für den SOA. Als Zweites fügen Sie einen NS-Eintrag hinzu. Damit Sie die Zone testweise abfragen können, sollte mindestens ein Host-A-Eintrag vorhanden sein. Eine einfache erste Zone könnte so aussehen:

```
$TTL 3H
@       IN  SOA   dns01.domain.tld. root.domain.tld.  (
                                2012040001    ; serial
                                1D            ; refresh
                                1H            ; retry
                                1W            ; expire
                                3H )          ; minimum
@           NS    dns01.domain.tld.
dns01       A     192.168.50.1
```

Das hier verwendete `@`-Zeichen ist ein Platzhalter und referenziert auf den Namen der Zone selbst zurück. In diesem Fall steht das `@` also für die Domäne »domain.tld«. Laden Sie anschließend die Konfiguration und die Zonen erneut, indem Sie das bekannte Kommando verwenden:

```
[root@arch-fc /]# rndc reload
server reload successful
```

Wenn Sie Fehlermeldungen erhalten sollten, die Ihnen keine verwertbaren Hinweise auf eventuelle Fehler geben, versuchen Sie, BIND über das Startskript neu zu starten. Es werden durch das Startskript meist umfangreichere Fehlermeldungen generiert, die möglicherweise Aufschluss über Konfigurationsfehler geben. Wenn der Server fehlerfrei läuft, kann die Beispielkonfiguration mit einem einfachen Kommando (das in diesem Kapitel noch genauer betrachtet wird) getestet werden:

```
[root@arch-fc /]# dig dns01.domain.tld @localhost
```

Das Kommando fragt den einzigen existierenden Host-A-Eintrag der Beispielkonfiguration auf dem lokalen Computer ab. Sie sollten eine entsprechende Antwort erhalten.

Die Vorgehensweise bei der Erstellung einer Reverse-Lookup-Zone entspricht im Prinzip der bei der Erstellung einer Forward-Lookup-Zone. Sie sollten nur die Details für die entsprechende Definition im Abschnitt über die Datei */etc/named.conf* beachten.

Hinweis

Wenn bei Ihrem BIND-Server etwas nicht funktionieren sollte, überprüfen Sie nochmals die Datei *named.conf* und die Zonendateien auf eventuelle Syntaxfehler. Häufig fehlt nur ein Punkt, ein Semikolon oder ein Leerzeichen. Beliebte Fehlerquellen sind auch unzureichende Berechtigungen für `named` auf die Zonendateien.

Erstellen von sekundären Zonen

Sekundäre Zonen haben einerseits die Aufgabe, der Ausfallsicherheit zu dienen, andererseits können sie aber auch dazu verwendet werden, die Zweigstellen eines Unternehmens mit Namensauflösung zu versorgen. In jedem Falle muss das Beziehen einer sekundären Zone (Slave) auf dem Masterserver erlaubt werden. Diese Genehmigung erteilen Sie in der Datei *named.conf* des Masterservers, indem Sie die Definition der Zonendatei erweitern:

```
zone "domain.tld" IN {
        type master;
        file "domain.tld.zone";
        allow-transfer { 192.168.50.14; 192.168.50.15; };
};
```

Der `allow-transfer`-Eintrag erlaubt den Computern mit den angegebenen IP-Adressen die Zone zu transferieren. Beachten Sie auch hier wieder die genaue Syntax.

Auf dem oder den Server(n), welche die sekundäre Zone beziehen sollen, muss in der Datei *named.conf* eine Zonendefinition von Typ `slave` erstellt werden. Außerdem müssen Sie den Masterserver angeben:

```
zone "domain.tld" IN {
        type slave;
        file "domain.tld.zone";
        masters { 192.168.50.1; };
};
```

Es gibt diverse Sicherheitsmechanismen, die den Zonentransfer verhindern können und werden. Das gilt übrigens auch, wenn Sie BIND für dynamisches DNS konfigurieren wollen, damit Ihre Client-Computer sich automatisch im DNS registrieren können. Sie können einige Maßnahmen (unter der Einbuße von Sicherheit!) treffen, damit Sie zunächst einmal Ihre Übungen durchführen können. Übergeben Sie dem Benutzerkonto `named` (bei Debian `bind`) die Eigentumsrechte an seinem Datenverzeichnis und an den darin enthaltenen Unterverzeichnissen und Dateien:

```
[root@arch-fc /]# chown named:named /var/named/ -R
```

Bei Debian verwenden Sie entsprechend:

```
root@arch-deb:/# chown bind:bind /etc/bind -R
```

Bei den Red Hat-basierten Systemen verhindert SELinux möglicherweise den Schreibzugriff auf die Zonen. Erlauben Sie das mit:

```
[root@arch-fc /]# setsebool -P named_write_master_zones=1
```

Wenn Sie ein Debian-System verwenden, kommt Ihnen möglicherweise AppArmor in die Quere. Sie können dieses Sicherheitssystem vorübergehend deaktivieren:

```
root@arch-deb:/# /etc/init.d/apparmor stop
```

Nachdem Sie alle möglichen Hindernisse aus dem Weg geräumt haben, können Sie `named` neu starten. Die sekundäre Zone sollte dann automatisch erstellt werden. Wenn die Zone nicht erstellt wird, prüfen Sie die Syntax der Datei *named.conf* auf beiden Servern und konsultieren die Dateien */var/log/messages* bzw. *syslog*.

Um möglichen Problemen bei der ersten Zonenübertragung entgegenzuwirken, können Sie die primäre Zone mittels `scp` oder `rsync` zunächst auf den Server kopieren, der die sekundäre Zone hosten soll.

Bedingte Weiterleitung

Wenn zwei Unternehmen zusammen an einem Projekt arbeiten wollen, muss für den gegenseitigen Ressourcenzugriff ein Namensauflösungsmechanismus implementiert werden. Sie könnten hier natürlich gegenseitig sekundäre Zonen beziehen, aber es gibt auch die Möglichkeit, mit einer Weiterleitung zu arbeiten. Der Vorteil einer bedingten Weiterleitung ist, dass keine Datenübertragung durch Zonentransfers stattfindet und es nicht zu Latenzzeiten nach der Aktualisierung der primären Zone kommt. Eine bedingte Weiterleitung wird bei BIND genauso konfiguriert wie eine normale Zone. Bei einer Zusammenarbeit der Firmen *domain.tld* und *die-anderen.tld* würde die Weiterleitung so konfiguriert:

```
zone "die-anderen.tld" {
        type forward;
        forwarders { 192.168.222.1; };
};
```

192.168.222.1 ist hierbei die IP-Adresse des DNS-Servers, der die Zone *die-anderen.tld* hostet.

Delegieren von Zonen

Wenn ein Unternehmen über mehrere Standorte verfügt, kann es sinnvoll sein, diese Standorte in einer Domänenstruktur abzubilden. In Bezug auf das vorangehende Beispiel könnte es dann Domänen mit den Bezeichnungen *bonn.domain.tld* und *berlin.domain.tld* geben. Das bedeutet nicht zwingend, dass diese Domänen auch über eigene Zonen verfügen müssen. Damit der Sinn einer Delegierung klarer wird, soll zunächst gezeigt werden, wie solche Standort-Subdomänen auch ohne die Verwendung von eigenen Zonen und Delegierungen unterstützt werden können.

Beispiel: Am Standort Berlin gibt es einen Computer namens `linux1` mit der IPv4-Adresse 192.168.50.77. In Bonn gibt es den Computer `linux2` mit der IPv4-Adresse 192.168.70.44. Die FQDNs dieser beiden Computer sind dann:

```
linux1.berlin.domain.tld
linux2.bonn.domain.tld
```

Wenn diese Computer der Zonendatei *domain.tld* hinzugefügt werden sollen, ergibt sich folgendes Bild:

```
@       IN  SOA   dns01.domain.tld. root.domain.tld.  (
                                2012040001    ; serial
                                1D            ; refresh
                                1H            ; retry
                                1W            ; expire
                                3H )          ; minimum
@           NS    dns01.domain.tld.
dns01                 A     192.168.50.1
linux1.berlin         A     192.168.50.77
linux2.bonn           A     192.168.70.44
```

Da der Server *dns01.domain.tld* für den gesamten Namensraum der Domäne *domain.tld* autoritativ ist, liegt die Zuständigkeit für Subdomänen automatisch mit in seinem Verwaltungsbereich. Leider gibt es mehrere Nachteile zu beklagen, wenn man Computer aus Subdomänen in Stammdomänen mitverwaltet:

- Die Administration von DNS lässt sich nicht aufteilen.
- Eventuelle dynamische Updates durch Client-Computer müssen in der Primärzone des Hauptstandorts (unter Umständen über WAN-Verbindungen) durchgeführt werden.
- Die Zonendatei wird durch die große Anzahl von Einträgen unübersichtlich.

Um diese Nachteile aufzuheben, werden in den Standorten DNS-Server eingesetzt, die für ihre jeweiligen Standortdomänen selbst primäre Zonen hosten. Es kann dann pro Standort ein Administrator für DNS zuständig sein, und dynamische Updates

durch Client-Computer finden nur noch innerhalb der Standorte statt. Der DNS-Server des Hauptstandortes muss über die DNS-Server der Zweigstellen in Kenntnis gesetzt werden. Da der Server der Stammzone (in unserem Beispiel `dns01`) sich selbst für die Zone *domain.tld* allein zuständig hält, würde er Abfragen bezüglich Hostnamen in Subdomänen ansonsten mit negativen Antworten quittieren.

Das Beispiel für Delegierungseinträge geht von folgenden Voraussetzungen aus:

- In Berlin wurde ein DNS-Server mit dem Hostnamen *dns-bln01.berlin.domain.tld* installiert. Die IPv4-Adresse des Servers ist 192.168.50.5. Er hostet die primäre Zone *berlin.domain.tld*.
- In Bonn wurde ein DNS-Server mit dem Hostnamen *dns-bo01.bonn.domain.tld* installiert. Die IPv4-Adresse des Servers ist 192.168.70.1. Er hostet die primäre Zone *bonn.domain.tld*.
- Beide Server sind bereits in der Lage, die Namen ihrer jeweiligen Subdomains aufzulösen.

Die Stammzone *domain.tld* müsste dann wie folgt um die Delegierungseinträge ergänzt werden:

```
@                       NS      dns01.domain.tld.
dns01                   A       192.168.50.1
; Degierung für Berlin
berlin                  NS      dns-bln01.berlin.domain.tld.
dns-bln01.berlin        A       192.168.50.5
; Delegierung für Bonn
bonn                    NS      dns-bo01.bonn.domain.tld.
dns-bo01.bonn           A       192.168.70.1
```

Wenn jetzt bei *dns01.domain.tld* eine Anfrage eintrifft, die sich auf einen Hostnamen in einer der Subdomänen *berlin* oder *bonn* bezieht, kann *dns01* feststellen, welche DNS-Server für diese beiden Subdomänen zuständig sind. Auch die IPv4-Adressen der zuständigen DNS-Server sind in der Zonendatei enthalten, sodass die Anfrage weitergeleitet werden kann.

Achtung

Eine Delegierung sorgt nicht dafür, dass die Client-Computer der Zweigstellen die Namen der Stammdomäne oder der jeweils anderen Zweigstellen auflösen können. Wenn das erwünscht ist, müssen entsprechende Sekundärzonen oder Weiterleitungen eingerichtet werden. Die Delegierung funktioniert immer nur von oben nach unten. Client-Computer der Stammdomäne können also die Namen der Computer in den Subdomänen auflösen.

DNS-Diagnoseprogramme

Die DNS-Diagnoseprogramme `dig`, `host` und `nslookup` wurden bereits ausführlich behandelt und sollen hier nicht noch einmal besprochen werden. Da diese Tools aber in mehreren Topics der Prüfung genannt werden, sollten Sie sich entsprechend gut damit auskennen und vielleicht noch ein wenig damit experimentieren.

named-checkzone

Wenn Sie eine Zonendatei neu erstellt oder bearbeitet haben, sollten Sie diese auf Fehler prüfen, bevor Sie den Server veranlassen, diese Datei (neu) zu laden. Ein Fehler in einer solchen Datei kann dazu führen, dass BIND die entsprechende Zone gar nicht lädt. Zur Überprüfung können Sie das Werkzeug `named-checkzone` einsetzen. Für eine Überprüfung müssen Sie sowohl den Namen der Zone, als auch den Namen der Zonendatei angeben:

```
root@archangel:/etc/bind# named-checkzone schulung.de db.schulung.de
zone schulung.de/IN: loaded serial 2016100001
OK
```

Das Programm versteht einige Optionen und ich empfehle Ihnen, ein wenig damit zu experimentieren.

named-compilezone

Sehr ähnlich funktioniert das Programm `named-compilezone`. Der Hauptunterschied ist, dass `named-compilezone` zusätzlich zur Überprüfung eine neue verwendbare Zonendatei generieren kann. Im folgenden Beispiel wird aus einer bestehenden Zone die neue Datei *new.zone* generiert.

```
root@archangel:/etc/bind# named-compilezone -o new.zone schulung.de db.schulung.de
zone schulung.de/IN: loaded serial 2016100001
dump zone to test.txt...done
OK
```

`named-compilezone` kennt fast dieselben Schalter wie `named-checkzone`. Auch hier sind ein Blick in die Manpage und ein paar Experimente empfehlenswert.

masterfile-format

In der Vergangenheit lagen Zonendateien grundsätzlich im Textformat vor. Das hatte den Vorteil, dass sie zu Diagnosezwecken von Menschen gelesen werden konnten. Zur schnelleren Verarbeitung auf Maschinen wurde ab BIND 9.9 ein RAW-Format eingeführt. Zonendateien, die in diesem Format vorliegen, können nicht mehr mit einem Texteditor oder Pager geöffnet werden. An dieser Stelle kommt noch einmal

das Kommando `named-compilezone` ins Spiel. Sie können damit nämlich eine Zonendatei, die im RAW-Format vorliegt einlesen und im Textformat wieder ausgeben. Das Kommando wird dazu aufgebaut wie folgt:

```
root@archangel:/etc/bind# named-compilezone -f raw -F text \
-o beispiel.net.text beispiel.net beispiel.net.raw
```

In der Datei *named.conf* ist es übrigens möglich, das Format der Zonendateien anzugeben. Natürlich müssten Sie dann dafür sorgen, dass die Zonendatei in dem angegebenen Format vorliegt. Bei einer RAW-Zonendatei sähe ein entsprechender Eintrag so aus:

```
zone "beispiel.net" {
        type slave;
        masterfile-format raw;
        file "/var/named/beispiel.net.raw";
};
```

Wenn eine Zonendatei im Textformat vorliegt, wird das entsprechend so angegeben:

```
        masterfile-format text;
```

207.3 Absicherung eines DNS-Servers

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen DNS-Server mit anderen als root-Rechten zu starten und den Server in einer `chroot`-Umgebung zu betreiben. Dieses Lernziel umfasst den sicheren Datenaustausch zwischen DNS-Servern.

Wichtigste Wissensgebiete:

- BIND 9.x-Konfigurationsdateien
- Konfiguration, um BIND in einer `chroot`-Umgebung zu betreiben
- Aufteilung der BIND-Konfiguration durch Einfügen von include-Anweisungen
- Konfigurieren und Verwenden von Transaktionssignaturen (TSIG)
- Kenntnis von DNSSEC und zugehörigen Tools
- Kenntnis von DANE und zugehörigen Einträgen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/named.conf*
- */etc/passwd*

- DNSSEC
- `dnssec-keygen`
- `dnssec-signzone`

Allgemeines

Abschließend müssen Sie zum Thema BIND noch ein paar Sicherheitsmechanismen kennenlernen. In der Praxis kommen zur Absicherung auf Dateisystemebene SELinux (Red Hat) und AppArmor (Debian) zum Einsatz, wie Sie bereits ganz am Rande erfahren haben. Diese beiden Systeme machen `chroot`-Umgebungen eigentlich unnötig, aber Sie müssen `chroot` für die Prüfung trotzdem kennen. Wenn Sie Netzwerke administrieren, in denen auch noch ältere Systeme ihren Dienst verrichten, kann es auch nicht schaden, ältere Sicherheitsmechanismen administrieren zu können. Das Gegenteil ist übrigens in Bezug auf DNSSEC der Fall: Dieses Konzept befindet sich immer noch im Aufbau und hat sich bei Weitem noch nicht flächendeckend durchgesetzt.

Einschränkungen in named.conf

Es gibt unterschiedliche Möglichkeiten, um den Zugriff auf einen BIND-Server einzuschränken. Einige Varianten haben Sie bereits kennengelernt. Hier finden Sie noch einmal eine Zusammenfassung aus rein sicherheitstechnischer Sicht.

Abfragen einschränken

Damit Informationen über Ihr Unternehmensnetzwerk nicht vom Internet aus abgefragt werden können, kann festgelegt werden, von welchen Netzwerken oder Hosts aus ein DNS-Server abgefragt werden darf. Es handelt sich hierbei natürlich nur um eine zusätzliche Sicherheitsmaßnahme, weil ein Netzwerk ja ohnehin durch eine Firewall gesichert wird. Wenn beliebige Hosts den DNS-Server abfragen sollen, schreiben Sie in die Sektion `options` der Datei *named.conf* Folgendes:

```
allow-query     { any; };
```

Soll Zugriff nur durch den Host selbst und das Netzwerk 192.168.50.0/24 möglich sein, können Sie einen solchen Eintrag vornehmen:

```
allow-query     { localhost; 192.168.50.0/24; };
```

Dynamische Aktualisierungen einschränken

Dynamische Aktualisierungen in DNS-Zonen sind eine angenehme Einrichtung, wenn man in einem Netzwerk über viele Clients verfügt, die mit dynamischen IP-Adressen versehen sind, aber trotzdem jederzeit über DNS auflösbar sein müssen.

Leider birgt dieses Konzept auch Sicherheitsrisiken, wenn z. B. ein böswilliger Benutzer ein mitgebrachtes Notebook mit dem gleichen Namen konfiguriert, den bereits ein anderer Computer im Netzwerk verwendet, und dann dem DNS-Server entsprechend falsche dynamische Aktualisierungen sendet. Eine relativ sichere Methode besteht darin, nur dem DHCP-Server das Aktualisieren einer DNS-Zone zu erlauben und diesen für die Aktualisierung von DNS zu konfigurieren. Wenn der DHCP-Server selbst die IPv4-Adresse 192.168.50.1 hat, könnte der passende Eintrag in der Zonendatei so aussehen:

```
allow-update     { 192.168.50.1; };
```

Wenn Sie allen Client-Computern des Netzwerksegments 192.168.50.0/24 erlauben wollen, dynamische Updates durchzuführen, können Sie diesen Eintrag verwenden:

```
allow-update     { 192.168.50.0/24; };
```

Das ist aber, wie bereits erläutert, in Netzwerken, in denen die Benutzer selbst Computer konfigurieren oder mitbringen können, nicht ratsam.

Zonentransfer einschränken

Ein Zonentransfer sollte nur den Computern erlaubt werden, die eine sekundäre Zone für eine Domäne beziehen sollen. Die Verfahrensweise ist Ihnen ja durch das Thema »Sekundäre Zonen« schon bekannt:

```
allow-transfer { 192.168.50.14; 192.168.50.15; };
```

Rekursion einschränken

In der Standardeinstellung verwendet BIND bei Abfragen, die sich auf Domänen außerhalb seines Zuständigkeitsbereichs (also hauptsächlich Domänen im Internet) beziehen, Rekursion. Das bedeutet, dass BIND einen Service für den Client leistet, indem er die Antwort für den Client beschafft. Zu diesem Zweck wendet er sich an einen Forwarder oder fragt (je nach Konfiguration) zunächst die Root-Server ab. Im Gegensatz zur rekursiven Namensauflösung steht die iterative Namensauflösung. Bei der iterativen Auflösung verweist der DNS-Server den Client lediglich an einen anderen DNS-Server, der für die entsprechende Zone zuständig ist. Das ist für den Client leider zeitaufwendiger, entlastet aber den Server. Wenn ein Server sich rekursiv abfragen lässt, kann ein Angreifer das ausnutzen. Da nach rekursiven Abfragen die ermittelten Antworten im Cache des Servers abgelegt werden, kann ein böswilliger Benutzer gezielt fehlerhafte Antworten in den Cache einschleusen (z. B. *www.google.de = 127.0.0.1*). Dieser Vorgang wird als *Cache-Poisoning* bezeichnet. Um die rekursive Namensauflösung nur für Ihr eigenes Netzwerk zur Verfügung zu stellen, können Sie der Sektion `options` die folgende Zeile hinzufügen:

```
allow-recursion { 127.0.0.1; 192.168.50.0/24; };
```

Zugriffssteuerungslisten verwenden

Sie können für alle drei der gerade beschriebenen Einschränkungen auch Zugriffssteuerungslisten verwenden. Diese Access Control Lists (ACLs) bedeuten zwar bei der Ersteinrichtung etwas mehr Aufwand, sind in großen Umgebungen mit vielen Netzwerken aber übersichtlicher zu verwenden. Wenn Sie z. B. ACLs verwenden wollen, um den Zonentransfer zu beschränken, können Sie das so konfigurieren:

```
acl "dns-server" {
        192.168.50.14;
        192.168.50.15;
};
```

Sie können jetzt das Label `"dns-server"` anstatt der IP-Adressen in Statements verwenden:

```
allow-transfer { dns-server; };
```

Wie Sie sich vorstellen können, ist eine solche Konfiguration übersichtlicher und auch leichter zu erweitern. Noch deutlicher wird das in Kombination mit `allow-query`. Auch hierfür wird zuerst die ACL erstellt:

```
acl "firmennetze" {
        192.168.50.0/24;
        192.168.70.0/24;
};
```

Ein passendes Statement wäre in diesem Fall:

```
allow-query     { localhost; firmennetze; };
```

Wenn später ein weiteres Netzwerksegment im Unternehmen erstellt wird, kann es einfach der ACL hinzugefügt werden. Der administrative Aufwand wird durch die ACLs somit geringer und das Verfahren erhöht außerdem die Übersichtlichkeit in der Konfiguration. Davon abgesehen können Sie jetzt ganz einfach denselben Computern dynamische Aktualisierungen und Rekursion erlauben. Die ACL existiert bereits und Sie benötigen nur noch zwei kurze Statements:

```
allow-update     { firmennetze; };
allow-recursion { 127.0.0.1; firmennetze; };
```

named einschränken

Wenn eine böswillige Person den Daemon `named` unter seine Kontrolle gebracht hat, könnte diese Person im Sicherheitskontext von BIND Schadcode einschleusen und sogar ausführen. Um den potenziellen Schaden zu begrenzen, den diese Person

anrichten kann, sollte `named` nicht mit root-Rechten ausgestattet sein oder in einer gechrooteten Umgebung ausgeführt werden.

named ohne root-Rechte ausführen

Damit ein Angreifer, der den Daemon `named` unter seine Kontrolle gebracht hat, keinen zu großen Schaden anrichten kann, sollte der Daemon über ein eigenes Benutzerkonto verfügen, das nur mit den Rechten ausgestattet wird, die für den Betrieb eines Nameservers erforderlich sind. Wenn Sie nicht gerade einen uralten Linux-Rechner administrieren, ist das sowieso der Fall. Im Verlauf dieses Kapitels haben Sie schon gesehen, dass `named` unter Red Hat offensichtlich das Benutzerkonto `named` und unter Debian das Benutzerkonto `bind` verwendet.

Wenn BIND gerade nicht läuft, können Sie den Daemon mit folgendem Kommando starten, um das entsprechende Benutzerkonto zuzuordnen:

```
[root@arch-fc ~]# /usr/sbin/named -u named
```

Wenn Sie das Startskript */etc/init.d/named* (Debian: */etc/init.d/bind9*) untersuchen, werden Sie feststellen, dass dieses Skript irgendwo ein ähnliches Kommando ausführt. Die Verzeichnisse, auf die `named` Zugriff benötigt, müssen also auch mit den entsprechenden Berechtigungen konfiguriert sein.

named in einer chroot-Umgebung ausführen

Das kurze Wort »chroot« steht für »change root«. Bei der Anwendung dieses Verfahrens wird einem Server ein Unterverzeichnis im Dateisystem als Hauptverzeichnis vorgegaukelt. Wenn dann ein Angreifer den gechrooteten Serverdienst unter seine Kontrolle gebracht hat, kann er lediglich in diesem Unterverzeichnis Schaden anrichten. Eine solche Umgebung wird auch als *Jail* oder *Sandbox* bezeichnet. Die Vorbereitung des Hauptverzeichnisses für einen BIND-Server stellt einen erheblichen Aufwand dar. Sie benötigen zumindest die Unterverzeichnisse *dbfiles*, *dev*, *etc*, *lib*, *sbin* bzw. *usr/sbin*. Da BIND zur Laufzeit auf Gerätedateien zugreift, müssen entsprechende Knoten mithilfe des Werkzeugs `mknod` erstellt werden (*/dev/null* und */dev/random*). Anschließend müssen benötigte Konfigurationsdateien aus */etc* einkopiert und angepasst werden. Wenn Sie wissen wollen, welche Bibliotheken Sie entweder einkopieren oder verlinken müssten, können Sie das mit dem Kommando `ldd /usr/sbin/named` in Erfahrung bringen. Spätestens jetzt dürfte klar werden, dass es sehr aufwendig wäre, eine heutige Version von BIND zu chrooten. Moderne Linux-Distributionen sichern BIND ab, indem Mechanismen wie AppArmor oder SELinux zum Einsatz kommen. Für diese Technologien sprechen vor allem folgende Argumente:

- geringerer administrativer Aufwand
- zusätzlich Absicherung der Interprozesskommunikation (`chroot` sichert lediglich den Verzeichniszugriff)

- Automatische Sicherheitsupdates können verwendet werden. Updates können keine gechrootete Umgebung aktualisieren.

Sie müssen in der Prüfung nicht mit Detailfragen zu diesem Thema rechnen, sollten aber zumindest wissen, was eine gechrootete Umgebung ist und wozu man sie verwendet. Unter Debian 6.0 Squeeze können Sie die folgenden Kommandos nacheinander eingeben, um einen gechrooteten BIND-Server zu erhalten:

```
# laufenden BIND ggf. beenden
/etc/init.d/bind9 stop
# die Verzeichnisse erstellen
mkdir -p /var/sandbox/bind
mkdir /var/sandbox/bind/etc
mkdir /var/sandbox/bind/dev
mkdir -p /var/sandbox/bind/var/cache/bind
mkdir -p /var/sandbox/bind/run/bind/run
# ursprüngliche Konfiguration verschieben und verlinken
mv /etc/bind /var/sandbox/bind/etc
ln -s /var/sandbox/bind/etc/bind /etc/bind
# Knoten für benötigte Geräte anlegen
mknod /var/sandbox/bind/dev/null c 1 3
mknod /var/sandbox/bind/dev/random c 1 8
# BIND zum Besitzer seiner Dateien machen
chown -R bind:bind /var/sandbox/bind/var/*
# damit BIND den Syslog verwenden kann:
echo "$AddUnixListenSocket /var/sandbox/bind/dev/log" >
/etc/rsyslog.d/bind-chroot.conf
# Startoptionen für gechrooteten BIND festlegen
rm -f /etc/default/bind9
echo "RESOLVCONF=yes" > /etc/default/bind9
echo "OPTIONS=\"-u bind -t /var/sandbox/bind\"" >> /etc/default/bind9
# Daemonen starten
/etc/init.d/rsyslog restart
/etc/init.d/bind9 start
```

BIND sollte jetzt wieder laufen und Sie können ihn testen.

DNSSEC

DNSSEC ist ein Akronym für *DNS Security Extensions*. Es handelt sich hierbei um eine Technologie, die auf kryptografischen Schlüsseln basiert. Die übertragenen Daten sollen nicht mithilfe von DNSSEC verschlüsselt werden. Vielmehr sollen die Daten signiert werden, damit deren Integrität und Authentizität sichergestellt werden kann.

DNSSEC sichert nur die Kommunikation zwischen Servern, also Zonentransfers und rekursive Abfragen, bei denen z. B. der DNS-Server eines Unternehmensnetzwerks einen DNS-Server im Internet abfragt. Die clientseitigen Resolver sind für komplexe Abfragen, die mit DNSSEC gesichert sind, nicht geeignet. Ursprünglich wurde DNSSEC ohnehin lediglich als Schutz vor DNS-Cache-Poisoning entwickelt.

Damit DNSSEC genutzt werden kann, müssen zwei entsprechende Einträge in der Datei *named.conf* erstellt bzw. geändert werden:

```
dnssec-enable yes;
dnssec-validation yes;
```

Laden Sie die Konfiguration anschließend mittels `rndc reload` neu. Sie können jetzt schon feststellen, ob Ihr Server DNSSEC-Überprüfungen durchführt, indem Sie eine Abfrage mit `dig` ausführen, die DNSSEC ausdrücklich anfordert:

```
root@archangel:/ # dig +dnssec www.lpi.org |grep RRSIG
.                       16888   IN      RRSIG   NS 8 0 518400 20130818000000
20130810230000 49656 . WxcPm8ZBE7SeG1Fi39UvFjk/xYgZh5/ssEyd+iuyP//
QWRTnk4IIRj2B tir7qEqM57hhv/L/dy2d65ANNvZBIunpHUutduwgYilMpBxxDqRBcZjF
a/VXzC2coHv5cnCpYdeDB5Iqp+JxX93HlxCMn7+KjwegnO+myUMMffyg zvs=
```

Sollten Sie keine Antwort mit einer Signatur erhalten, unterstützt der DNS-Server, den Sie als Forwarder verwenden, vermutlich noch nicht die Verwendung von DNSSEC.

In der Prüfung müssen Sie, was DNSSEC betrifft, vor allem das Werkzeug `dnssec-keygen` bedienen können. Dieses Programm, das zum Generieren von Schlüsseln dient, verwendet drei wesentliche Optionen:

- `-a` gibt den zu verwendenden Algorithmus an. Möglich sind RSA, RSAMD5, DH, DSA, RSASHA1 und HMAC-MD5
- `-b` bestimmt die Schlüsselgröße. Die verwendbare Größe hängt vom Algorithmus ab.
- `-n` spezifiziert den Typ des Schlüssels. Mögliche Werte sind ZONE, HOST, ENTITY, USER, OTHER.

Es gibt noch wesentlich mehr Optionen, aber die genannten sind sowohl für die Praxis als auch für die Prüfung die wichtigsten. Das folgende Kommando generiert ein Schlüsselpaar für eine Zone:

```
root@archangel:/etc/bind# dnssec-keygen -a DSA -b 768 -n ZONE homelinux.net.
```

Die Erstellung des Schlüsselpaares kann auch auf schnellen Computern ein wenig dauern. Als Ergebnis erhält man zwei Dateien:

```
Khomelinux.net.+003+18062.key
Khomelinux.net.+003+18062.private
```

Dateien mit der Erweiterung *.key* enthalten den öffentlichen Schlüssel, während Dateien mit der Erweiterung *.private* den geheimen privaten Schlüssel enthalten. Der sendende DNS-Server kann den privaten Schlüssel verwenden, um damit seine Antworten zu signieren, während der empfangende Computer den öffentlichen Schlüssel verwenden kann, um diese Signatur zu überprüfen. Die beiden Dateien sind übrigens menschenlesbar und Sie sollten einmal mithilfe eines Pagers einen Blick darauf werfen.

Um den öffentlichen Schlüssel zu veröffentlichen, können Sie diesen einfach an die Zonendatei anhängen:

```
root@archangel:/etc/bind# cat Khomelinux.net.+003+18062.key >> db.homelinux.net
```

Alternativ können Sie auch eine `include`-Anweisung in die Zone schreiben, die auf den Namen der Datei zeigt, die den Schlüssel enthält. Einen solchen Eintrag schreibt man normalerweise direkt unter den SOA-Record. Die entsprechende Zeile würde so aussehen:

```
$include Khomelinux.net.+003+18062.key
```

Sie müssen die Zonendatei anschließend bearbeiten, indem Sie die Seriennummer erhöhen. Server mit sekundären Zonen würden die Zone ansonsten nicht transferieren. Laden Sie die Zone anschließend neu:

```
root@archangel:/etc/bind# rndc reload
```

Sie können das Signieren der Zone nun mit dem Tool `dnssec-signzone` durchführen:

```
root@archangel:/etc/bind# dnssec-signzone -t -g -o homelinux.net. \
db.homelinux.net /etc/bind/Khomelinux.net.+003+18062.private
```

Die Konfiguration der unterschiedlichen Funktionen von DNSSEC würde den Rahmen dieses Kapitels bei Weitem überschreiten. Sie sollten aber zumindest die DNS-Eintragstypen kennen, die mit DNSSEC im Zusammenhang stehen.

- `RRSIG` ist der Eintrag für eine Resource-Record-Signatur. Mit RRSIG werden einzelne Einträge innerhalb einer Zonendatei signiert und damit die Echtheit des Eintrags gegenüber dem Client garantiert.
- `SIG` ist der direkte Vorgänger von RRSIG und sollte nicht mehr verwendet werden.
- `DNSKEY` dient der Veröffentlichung der öffentlichen Schlüssel von DNSSEC.
- `KEY` ist der Vorläufer von DNSKEY und sollte nicht mehr verwendet werden.

- `NSEC` dient der Signatur von Resource Records in alphabetischer Reihenfolge. Mit der Abfrage der benachbarten Signaturen kann ein Client feststellen, ob es einen Host auch wirklich gibt. Wenn `"Compi-7"` und `"Compi-9"` aufgrund ihrer Signaturen als Nachbarn betrachtet werden müssen, kann es `"Compi-8"` folglich nicht geben.

TSIG

Unabhängig von DNSSEC ist die Sicherheitstechnologie *TSIG* (Transaction Signature) zu betrachten. Grundsätzlich soll auch TSIG die Authentizität und Integrität einer DNS-Kommunikation zwischen zwei Computern sicherstellen. Anders als bei DNSSEC kommt hier aber keine PKI (Public Key Infrastructure) zum Einsatz, sondern gemeinsame Geheimnisse, die zwischen zwei Hosts, die miteinander kommunizieren sollen, konfiguriert werden müssen.

Die benötigten Schlüssel können auch für TSIG mittels `dnssec-keygen` generiert werden. TSIG-Schlüssel können auch auf Client-Computern verwendet werden, um damit die dynamische Aktualisierung abzusichern. Diese Methode ist erheblich sicherer als die Genehmigung von Updates auf der Basis von IP-Adressen.

Das nun folgende Beispiel demonstriert die Absicherung des Zonentransfers zwischen zwei Servern. Auf einem Master-Nameserver generieren Sie zunächst ein Schlüsselpaar mit diesem Kommando:

```
root@morouter1:~# dnssec-keygen -a HMAC-MD5 -b 128 -n HOST rndc-key
Krndc-key.+157+29986
```

Sie erhalten dann diese beiden Dateien:

```
-rw------- 1 root root  52 Sep  8 19:49 Krndc-key.+157+29986.key
-rw------- 1 root root 165 Sep  8 19:49 Krndc-key.+157+29986.private
```

Der private Schlüssel hat folgenden Inhalt:

```
root@morouter1:~# cat Krndc-key.+157+29986.private
Private-key-format: v1.3
Algorithm: 157 (HMAC_MD5)
Key: dONYgH/5BUkNRa8PrYazAw==
Bits: AAA=
Created: 20160908174903
Publish: 20160908174903
Activate: 20160908174903
```

Den Inhalt der Zeile `Key: dONYgH/5BUkNRa8PrYazAw==` benötigen Sie während der Erstellung der Datei *tsig.key*. Diese Datei könnte etwa so aussehen:

```
key "TRANSFER" {
          algorithm hmac-md5;
          secret " dONYgH/5BUkNRa8PrYazAw==";
};
# Adresse des Slave-Servers
server 47.8.15.200 {
        keys {
                TRANSFER;
    };
};
```

Beachten Sie, dass das `secret` mit dem `Key` des zuvor generierten privaten Schlüssels übereinstimmt. Damit der Schlüssel verwendet wird, muss dieser in der Datei *named.conf* angegeben werden:

```
include "/etc/bind/tsig.key";
```

Auf dem Slave-Server ist im Prinzip dieselbe Konfiguration mit demselben Schlüssel erforderlich. Der einzige Unterschied ist, dass in der Datei *tsig.key* des Slave-Servers die IP-Adresse des Master-Servers angegeben werden muss.

Vergessen Sie bitte nicht, beiden Servern abschließend das Kommando `rndc reload` zu geben.

DANE

DANE steht für DNS-based Authentication of Named Entities. Diese Technologie dient der DNS-gestützten Überprüfung von SSL-Zertifikaten und stützt sich auf *DNSSEC*. Bisher prüfen Clients die Vertrauenswürdigkeit von Zertifikaten anhand einer Liste, die z. B. in Browsern in Form von vertrauenswürdigen Stamm-Zertifizierungsstellen hinterlegt sind. Da in der Vergangenheit häufig gefälschte Zertifikate von diesen vertrauenswürdigen CAs aufgetaucht sind, soll eine zusätzliche Prüfung mithilfe spezieller DNS-Records durchgeführt werden. Diese Records sind vom Typ *TLSA* und enthalten Zertifikate im PKIX-Format. Hierdurch hat ein Client die Möglichkeit, ein x.509-basiertes Zertifikat mit dem im TLSA-Record enthaltenen Hashwert zu vergleichen. Der Hashwert kann, je nach Verwendungszweck des Zertifikats, vom verwendeten Zertifikat oder von der ausstellenden Stammzertifizierungsstelle abgeleitet sein. Ein DANE-Record ähnelt vom Aufbau her einem SRV-Eintrag. Ein Eintrag wie dieser wäre z. B. für einen SMTP-Server erstellt worden:

```
_25._tcp.smtp.beispiel.net.      IN      TLSA      3 0 1 \
56afec72ad7506a053f4f183da534ca89b5eb17ef9d11468dd953d52acefd24e
```

Hier wird also als erstes der Zielport des zu erreichenden Servers, dann das Transportprotokoll und anschließend der FQDN des Servers angegeben. Die Werte `3`, `0` und

1 geben genaueren Aufschluss über den Inhalt und die Art des Eintrags, aber damit müssen sie sich zunächst nicht beschäftigen.

Aufteilung der BIND-Konfiguration

Split-Horizon-DNS, *Split-View-DNS* und *Split-Brain-DNS* sind drei verschiedene Fachbegriffe, die sich aber alle mit derselben Problematik beschäftigen, nämlich der unterschiedlichen Darstellung von DNS-Zoneninhalten für interne und externe Clients. Stellen Sie sich ein Unternehmen vor, das über eine interne Produktionsdomäne verfügt, die denselben DNS-Namen verwendet wie die Internetpräsenz des Unternehmens. Im Normalfall wären dann die folgenden Anforderungen bei der Namensauflösung zu erfüllen:

- Interne DNS-Clients sollen alle Hostnamen des internen Netzwerks auflösen können.
- Interne DNS-Clients sollen auch alle öffentlich zugänglichen Hosts des Unternehmensnetzwerks auflösen können (Webserver, Mailserver, FTP-Server usw.).
- Interne DNS-Clients sollen alle Hostnamen im Internet auflösen können.
- Externe Clients sollen öffentlich zugängliche Server des Unternehmens auflösen können (Webserver, Mailserver, FTP-Server usw.).
- Externe Clients dürfen keine Namen von internen Computern, wie Client-Rechner, Fileserver, Anwendungsserver u. ä. auflösen können.

Um diese Ziele zu erreichen, gibt es mehrere Lösungsansätze.

Praxistipp

In der Praxis ist es empfehlenswert, eine derartige Konfiguration von vornherein zu vermeiden. Wenn eine Firma im Internet mit der Domäne *beispiel.com* präsent ist, können Sie das Produktionsnetzwerk z. B. *prod.beispiel.com* nennen. Alternativ wären Standortdomänen gut geeignet, also hier *berlin.beispiel.com* usw.

Split-Horizon/Split-Brain

Um die fünf o. g. Anforderungen zu erfüllen, muss die Zone für die Domäne doppelt angelegt werden. Sie benötigen eine Version der Zone, auf die interne Computer zugreifen, und eine andere Version, die im öffentlichen DNS-System zugänglich ist. Die Zone, die den internen Clients präsentiert wird, enthält alle Einträge für die eigene Domäne, also sowohl interne als auch öffentlich zugängliche Hostnamen (Webserver, Mailserver usw.). Eine weitere Version der Zone mit erheblich reduziertem Inhalt wird der Öffentlichkeit zur Verfügung gestellt. Diese Zone enthält Einträge für Webserver, Mailserver, FTP-Server und MX-Records, damit E-Mails aus dem Internet ihren Weg in das Unternehmensnetzwerk finden.

Zwei physikalische Server

Sie können die beiden Versionen der Zonen nun z. B. auf zwei verschiedenen physikalischen Servern hosten. Die Version der Zone, die für interne Client-Computer gedacht ist, liefert ein DNS-Server aus, der sich im internen Netzwerk des Unternehmens befindet. Ein Bastionhost, der sich in der demilitarisierten Zone (DMZ) des Unternehmens befindet, könnte für die öffentliche Version derselben Zone zuständig gemacht werden.

Eine andere Variante wäre es, die öffentliche Version der Zone einfach von einem Internet Provider hosten zu lassen und die vertrauliche Version der Zone für das Produktionsnetzwerk auf einem internen Server selbst zu verwalten. Da sich die IP-Adressen der öffentlichen Server meist nicht allzu häufig ändern, ist das eine interessante Option.

Ein physikalischer Server

Wenn Sie beide Versionen der Zone selbst verwalten wollen, aber nur einen einzigen physikalischen Server zu Verfügung haben, können Sie zwei Instanzen von BIND auf derselben Maschine laufen lassen. Es ist dann aber erforderlich, mindestens eine Instanz des Servers in einer gechrooteten Umgebung zu verwenden.

Zugriffssteuerung

Egal für welche der genannten Methoden Sie sich entscheiden, bleibt die Konfiguration in den *named.conf*-Dateien doch immer gleich. Da Ihnen die Konfiguration dieser Datei bereits bekannt ist, sollen an dieser Stelle ein paar Hinweise für die Split-DNS-Konfiguration ausreichen.

Sie sollten den Zugriff auf den internen Server beschränken, sodass er nur von internen DNS-Clients abgefragt werden kann. Das Gleiche gilt natürlich auch für Zonentransfers und dynamische Aktualisierungen. Selbst der DNS-Server, der die öffentliche Version der Zone hostet, sollte am Zugriff auf den internen DNS-Server gehindert werden. Damit die Auflösung von Hostnamen im Internet reibungslos abläuft, sollten Sie von der Option `forwarders` Gebrauch machen. Für ein optimales Caching können Sie hier den externen DNS-Server angeben.

Der Zugriff auf den Server, der die öffentliche Version der Zone hostet, sollte so konfiguriert sein, dass er internen DNS-Clients nicht antwortet. Es wäre schließlich denkbar, dass beide Zonenversionen z. B. über unterschiedliche MX-Einträge verfügen. Ein falsch konfigurierter Client würde dann eventuell fehlerhafte Informationen empfangen, wenn eine E-Mail firmenintern versendet werden soll. Damit dieser Server nicht das Opfer einer Birthday-Attacke aus dem Internet wird, sollten Sie die Rekursion sicherheitshalber für externe Clients verhindern.

Split-View

Eine relativ neue und sehr elegante Art, zwei verschiedene Versionen derselben Zone auf einem einzigen Server zu hosten, sind Split-Views. Bei diesem Verfahren steuern Zugriffssteuerungslisten in Kombination mit `view`-Statements, welcher Client mit welchen Informationen versorgt wird. Innerhalb eines solchen Statements bestimmt der Eintrag `match-clients`, auf welche DNS-Clients das jeweilige `view`-Statement angewendet werden soll. Das folgende Beispiel demonstriert die Funktionsweise der Views. Es wird in diesem Beispiel davon ausgegangen, dass beide Versionen der Zonendatei unter dem gleichen Dateinamen, nämlich *beispiel.de.zone*, jeweils in den Verzeichnissen */var/named/intern* und */var/named/extern* abgelegt worden sind. Der relevante Teil der Datei *named.conf* könnte dann so aussehen:

```
options {
        directory "/var/named";
        listen-on { any; };
        allow-query { any; };
        recursion no;
};
acl "freunde" {
        192.168.50.0/24;
        192.168.20.0/24;
        127.0.0.1;
};
view "intern" {
        match-clients { freunde; };
        recursion yes;
       // Hinweise auf die Root-Server
        zone "." {
                type hint;
                file "db.root";
        };
        // Ein paar Standardeinträge:
        zone "localhost" {
                type master;
                file "db.local";
        };
        zone "127.in-addr.arpa" {
                type master;
                file "db.127";
        };
        zone "0.in-addr.arpa" {
                type master;
                file "db.0";
```

```
        };
        zone "255.in-addr.arpa" {
                type master;
                file "db.255";
        };
        // Die vertrauliche Version der DNS-Zone:
        zone "beispiel.de" {
                type master;
                allow-transfer { freunde; };
                file "intern/beispiel.de.zone";
        };
};
view "extern" {
        // Zugriff durch interne DNS-Clients verhindern :
        match-clients { !freunde; any; };
        recursion no;
        // Die öffentliche Version der DNS-Zone :
        zone "beispiel.de" {
                type master;
                file "extern/beispiel.de.zone";
        };
};
```

Die Kommentare innerhalb der Konfiguration erläutern die in diesem Buch noch nicht erwähnten Elemente der Datei *named.conf*.

208 HTTP-Dienste

Der Apache Webserver ist eines der erfolgreichsten Open-Source-Projekte überhaupt. Aufgrund seiner Sicherheit, Stabilität und Leistungsfähigkeit wurde sein Erfolg bis heute durch keinen anderen Webserver infrage gestellt.

208.1 Grundlegende Apache-Konfiguration

Wichtung: 4

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen Webserver zu installieren und einzurichten. Dieses Lernziel beinhaltet die Überwachung der Serverauslastung und -leistung, die Beschränkung von Benutzerzugriffen, die Einrichtung der Unterstützung von Skriptsprachenmodulen und die clientseitige Benutzerauthentifizierung. Des Weiteren ist auch die Konfiguration der Serveroptionen, die der Einschränkung der Ressourcennutzung dienen, mit eingeschlossen. Kandidaten sollten dazu in der Lage sein, virtuelle Hosts auf Webservern zu erstellen und den Dateizugriff anzupassen.

Wichtigste Wissensgebiete:

- Apache 2.4-Konfigurationsdateien, -Begriffe und -Dienstprogramme
- Konfiguration der Apache-Protokolldateien und deren Inhalte
- Methoden und Dateien zur Zugriffsbeschränkung
- Konfiguration von *mod_perl* und PHP
- Client-Benutzerauthentifizierungsdateien und -Dienstprogramme
- Einstellung der maximalen Anzahl an Anfragen sowie der minimalen und maximalen Anzahl an Serverprozessen und Clients
- Apache 2.4-Implementierung von virtuellen Hosts (mit und ohne fest zugeordneten IP-Adressen)
- Einsatz von Redirect-Anweisungen in Apache-Konfigurationsdateien, um Dateizugriffe zu individualisieren

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- Zugriffslogdateien (Access Logs) und Fehlerlogdateien (Error Logs)
- *.htaccess*

- *httpd.conf*
- *mod_auth_basic*, *mod_authz_host* und *mod_access_compat*
- `htpasswd`
- *AuthUserFile*, *AuthGroupFile*
- `apachectl`, `apache2ctl`
- *httpd*, *apache2*

Allgemeines

Der *Apache Webserver* ist nach wie vor der erfolgreichste Webserver im Internet. Die Ursprünge des Apache Webservers liegen in einem inzwischen ausgestorbenen Webserver, dem NCSA-Webserver *httpd*. Dieser wurde von der Organisation NCSA im Jahre 1995 eingestellt und nicht weiterentwickelt. Die NCSA ist übrigens dieselbe Organisation, die im Jahre 1993 den damals sensationellen Webbrowser *Mosaic* entwickelte. Das war der erste Browser, der auch von normalen Menschen bedient werden konnte, die nicht als absolute Computerfreaks bekannt waren.

Nachdem die NCSA ihren Webserver in der Version *httpd 1.3* eingestellt hatte, fingen diverse Entwickler, die sich später zur Apache Software Foundation zusammenschlossen, an, Sicherheitslücken in diesem Webserver zu beheben und ihm neue Funktionen hinzuzufügen. Es wurden immer mehr Patches zusammengetragen und den Aussagen von Zeitzeugen nach wurde daraus zunächst die Bezeichnung »A patchy Server«. Es gibt aber auch die Aussage, der Name wäre eine Hommage an die gleichnamigen nordamerikanischen Indianerstammesgruppen. Möglicherweise ist an beiden Geschichten etwas dran.

Installation von Apache

Es gibt verschiedene Möglichkeiten, wie man zu einem Apache Webserver kommen kann. In Abhängigkeit von der verwendeten Linux-Distribution können Sie natürlich `yum` oder `aptitude` verwenden, um den Webserver zu installieren. Damit Sie den Webserver in seiner reinsten Form kennenlernen, ist es im Rahmen der Prüfungsvorbereitung allerdings sinnvoller, Apache aus einem tar-Ball heraus zu installieren. Sie finden die entsprechenden Downloadlinks zu etlichen HTTP- und FTP-Mirror-Servern auf der Webseite *http://www.apache.org*.

Die folgende Beispielinstallation wurde auf einem Server unter CentOS ausgeführt. Führen Sie einfach die folgenden Kommandos aus, um einen funktionierenden Webserver zu installieren:

```
[root@arch-cent ~]# mkdir /usr/src/apache-2.2
[root@arch-cent ~]# cd /usr/src/apache-2.2/
```

Aus dem vorbereiteten Installationsverzeichnis heraus kann man mittels `wget` den tar-Ball gleich an Ort und Stelle herunterladen. Sie können natürlich auch eine andere Quelle und eine andere Version von Apache verwenden, wenn Sie das möchten.

```
[root@arch-cent apache-2.2]# wget \
http://ftp.halifax.rwth-aachen.de/apache/httpd/httpd-2.4.23.tar.gz
```

Packen Sie den tar-Ball aus:

```
[root@arch-cent apache-2.4]# tar xvzf httpd-2.4.23.tar.gz
```

Wechseln Sie in das Verzeichnis, das gerade durch die Extraktion des tar-Balls entstanden ist:

```
[root@arch-cent apache-2.4]# cd httpd-2.4.23
```

Damit Sie den Webserver konfigurieren und kompilieren können, benötigen Sie spätestens jetzt einen Compiler. Ich persönlich bevorzuge den GNU-C- und den GNU-C++-Compiler:

```
[root@arch-cent httpd-2.4.23]# yum install gcc
```

Bei der Konfiguration sollten Sie zumindest das Installationsverzeichnis mit `--prefix` übergeben. Das Verzeichnis */usr/local/apache2* ist ein typisches Ziel für den Webserver. Die anderen Optionen wählen die Funktionalitäten *http*, *https*, *cgi* und das automatische Einbinden von Modulen.

```
[root@arch-cent httpd-2.4.23]# ./configure --prefix=/usr/local/apache2 \
--enable-http --enable-https --enable-so --enable-cgi
```

Es gibt sehr viele Optionen, die Sie für die Konfiguration übergeben können. Sie erhalten eine komplette Liste der Optionen, indem Sie im Installationsverzeichnis das Kommando `./configure --help` ausführen.

Wenn die Konfiguration abgeschlossen ist, können Sie *httpd* mit `make` kompilieren:

```
[root@arch-cent httpd-2.4.23]# make
```

Wie bei der Installation vieler anderer tar-Balls folgt jetzt die eigentliche Installation des Webservers:

```
[root@arch-cent httpd-2.4.23]# make install
```

Wenn alles gut gegangen ist, können Sie den Apache Webserver jetzt zum ersten Mal starten:

```
[root@arch-cent /]# /usr/local/apache2/bin/apachectl start
httpd: Could not reliably determine the server's fully qualified domain name,
using 127.0.0.1 for ServerName
```

Es ist wahrscheinlich, dass Sie beim Start des Servers obige Fehlermeldung erhalten. Der Webserver läuft dann trotzdem, aber Sie sollten diesen Schönheitsfehler beheben, indem Sie der Datei */usr/local/apache2/conf/httpd.conf* (das ist übrigens die Hauptkonfigurationsdatei des Webservers) eine Zeile wie diese hinzufügen:

```
ServerName arch-cent.homelinux.net:80
```

Prüfen Sie nach, ob der Webserver läuft, indem Sie nach Instanzen des Daemons *httpd* suchen. Das ist der Daemon des Apache Webservers:

```
[root@arch-cent /]# ps -A|grep httpd
 9487 ?        00:00:00 httpd
 9488 ?        00:00:00 httpd
 9489 ?        00:00:00 httpd
 9490 ?        00:00:00 httpd
```

Den eigentlichen Test führen Sie dann mit einem Webbrowser aus. Wenn Sie lokal an derselben Maschine arbeiten, die den Webserver ausführt, rufen Sie einfach die URL *http://localhost* auf. Alternativ können Sie natürlich auch von einem anderen PC aus über das Netzwerk zugreifen. Der angezeigte Content befindet sich im Document-Root-Verzeichnis */usr/local/apache2/htdocs*, wenn Sie Apache von Hand installiert haben. Sie können hier Ihre eigenen Inhalte unterbringen.

Konfigurationsdateien

Die Konfigurationsdateien für Apache befinden sich üblicherweise in den Verzeichnissen */usr/local/apache2/conf* (wenn Sie den Server selbst kompiliert haben, befinden Sie sich genau da), */etc/apache*, */etc/apache2* oder */etc/httpd/conf*. In Abhängigkeit von der verwendeten Linux-Distribution können Sie sich aber auch noch in anderen Verzeichnissen befinden (hier hilft z. B. `find /etc -name httpd.conf`). Leider variiert auch der Inhalt des jeweiligen Konfigurationsverzeichnisses. Das liegt daran, dass in den meisten Linux-Distributionen Teile der Hauptkonfigurationsdatei *httpd.conf* in kleinere Dateien ausgelagert wurden, um die Übersichtlichkeit zu erhöhen. In der LPI-Prüfung wird häufig nach den folgenden Dateien gefragt:

- *httpd.conf* ist die Hauptkonfigurationsdatei. In dieser Datei stehen alle Parameter, die für den Betrieb des Webservers notwendig sind.
- *srm.conf* enthielt früher die `ResourceConfig`-Anweisungen. Da diese Anweisungen inzwischen mit in *httpd.conf* integriert worden sind, ist *srm.conf* heutzutage leer oder gar nicht mehr vorhanden.
- *access.conf* enthielt früher die Anweisungen zur Zugriffssteuerung. Auch diese Anweisungen wurden inzwischen mit in die Datei *httpd.conf* aufgenommen.

Wenn Sie Apache selbst konfiguriert und kompiliert haben, enthält dessen Konfigurationsverzeichnis */usr/local/apache2/conf* ein Unterverzeichnis namens *extra*. In diesem Unterverzeichnis finden Sie weitere Konfigurationsdateien, die mittels `include`-Anweisungen in die Datei *httpd.conf* integriert worden sind:

```
httpd-autoindex.conf
httpd-languages.conf
httpd-ssl.conf
httpd-dav.conf
httpd-manual.conf
httpd-userdir.conf
httpd-default.conf
httpd-mpm.conf
httpd-vhosts.conf
httpd-info.conf
httpd-multilang-errordoc.conf
```

Standardmäßig sind die zu diesen Dateien gehörenden `include`-Anweisungen auskommentiert. Wenn Sie eine der Konfigurationsdateien benötigen, müssen Sie lediglich in der Datei *httpd.conf* in der entsprechenden Zeile am Anfang die Raute entfernen. Der Verwendungszweck der einzelnen Dateien ist in der Hauptkonfigurationsdatei *httpd.conf* dokumentiert.

Tipp

Wenn Sie `httpd-manual.conf` aktivieren, indem Sie die vorangestellte Raute entfernen, haben Sie Zugriff auf das Manual von Apache. Sie müssen anschließend lediglich `apachectl restart` ausführen und können daraufhin auf die URL *http://localhost/manual* zugreifen.

Wichtige Einträge in der Datei httpd.conf

Unabhängig von deren tatsächlicher Wichtigkeit werden nun einige Einträge aus der Datei *httpd.conf* aufgezählt, die in der Prüfung häufig auftauchen. Lassen Sie sich nicht von Fragen erschrecken, die Optionen enthalten, die Sie gar nicht kennen. Oft kann man durch Logik und Ausschlussverfahren die richtigen Antworten ermitteln.

- `Port` ändert den Port, an dem Apache für eingehende Webanfragen lauscht. Der Standardport ist 80. Diese Direktive wurde in neueren Apache-Versionen durch `Listen` ersetzt.
- `Listen` ändert den Port und ggf. die IP-Adresse, an der Apache für eingehende Webanfragen lauscht. Standardmäßig ist hier Port 80 ohne IP-Adresse angegeben.

- `MinSpareServers` legt die Anzahl der Instanzen fest, die beim Starten von Apache in den Speicher geladen werden. Ein typischer Wert für kleine bis mittlere Serverauslastungen ist `10`.
- `ServerType` legt fest, ob Apache von `inetd` gestartet wird oder ob er selbstständig läuft. Mögliche Werte sind `standalone` oder `inetd`.
- `ServerRoot` ist das Verzeichnis, in dem Apache seine Protokolle, Serverkonfiguration, CGI-Skripte und Ähnliches findet.
- `DocumentRoot` ist das Hauptverzeichnis für Dokumente, zu dem die Öffentlichkeit Zugang haben soll.

Prüfungstipp

Diese Einträge sollten Sie unbedingt kennen. Achten Sie darauf, dass Sie nicht `Port` und `Listen` miteinander verwechseln. Die Anzahl der `MinSpareServers` für eine kleine bis mittlere Umgebung wird oft und in vielen Varianten gefragt. Merken Sie sich also hierbei unbedingt die Zahl 10!

Starten und stoppen

Es gibt mehrere Methoden, um einen Apache Webserver zu starten, zu beenden oder neu zu starten. Sie sollten die hierfür benötigten Kommandos kennen:

- Wie jeden anderen Daemon können Sie auch *httpd* mit dem Kommando `/etc/init.d/httpd start` starten. Sollten Sie den Server selbst kompiliert haben, gibt es kein Startskript. Sie können dann einfach ein vorhandenes Skript kopieren und entsprechend anpassen.
- ohne `init`-Skript: `/usr/local/apache2/bin/httpd -f /usr/local/apache2/conf/httpd.conf`
- `/usr/local/apache2/bin/apachectl` ist zur Laufzeit allerdings die eleganteste Lösung. Das Skript versteht die folgenden Optionen:
  - `start` startet den Server.
  - `stop` beendet den Server.
  - `restart` startet den Server neu.
  - `graceful` startet den Server neu, aber bestehende Verbindungen bleiben erhalten. So können etwaige Konfigurationsänderungen registriert werden.
  - `configtest` überprüft die Konfigurationsdateien auf Syntaxfehler.

Sie werden auf einigen Systemen `apache2ctl` anstatt `apachectl` antreffen. Die beiden Skripte unterscheiden sich allerdings nicht nennenswert voneinander.

Im Allgemeinen ist `apachectl` nicht sehr gesprächig und gibt lediglich beim `configtest` eine Erfolgsmeldung aus. In allen anderen Fällen sollten Sie sich bei einem Produktionsserver davon überzeugen, dass er die von Ihnen angeforderte Aktion auch wirklich durchgeführt hat.

```
[root@arch-cent bin]# ./apachectl configtest
Syntax OK
```

Die Richtigkeit der Konfigurationsänderungen wurde bestätigt.

Zugriffssteuerung

Zur Steuerung des Zugriffs auf eine Webseite gibt es eine ganze Reihe von Modulen, die entweder standardmäßig im Kern von Apache integriert sind oder optional geladen werden können. Drei ausdrücklich als Prüfungsthema angegebene Module sind:

- *mod_auth_basic* zur Authentifizierung in Klartext
- *mod_authz_host* zur Zugriffskontrolle über Hostnamen oder IP-Adressen (früher *mod_access*)
- *mod_access_compat* ist ein Vorläufer von *mod_authz_host*, der weniger Direktiven beinhaltet.

Diese drei Module sind im Kern des Servers enthalten und müssen nicht explizit geladen werden.

Wenn eine Webseite Informationen enthält, die nicht für die Öffentlichkeit bestimmt sind, diese Seite aber im Internet erreichbar ist, sollten Sie eine Authentifizierung von den Benutzern anfordern. Hier kommt das Modul *mod_auth_basic* zum Zuge. Die erforderlichen Benutzerkonten können mit dem Programm `htpasswd` erstellt und anschließend einer Webseite bzw. einem Webverzeichnis zugeordnet werden.

htpasswd verwenden

Die Befehlsfolge im folgenden Beispiel ist so gewählt, dass sie sich einerseits übersichtlich darstellen lässt und andererseits durch kurze Kommandos leicht nachzustellen ist:

```
[root@arch-cent /]# cd /usr/local/apache2/bin/
```

Das folgende Kommando erstellt eine Passwortdatei und legt im selben Arbeitsschritt den ersten Benutzer samt Passwort an. Die Passwortdatei befindet sich anschließend in *ServerRoot /usr/local/apache2*.

```
[root@arch-cent bin]# ./htpasswd -c ../password.list user1
New password:
Re-type new password:
Adding password for user user1
```

Wenn Sie weitere Benutzer für Ihren Webserver erstellen wollen, verwenden Sie ein ähnliches Kommando. Sie müssen dann lediglich die Option `-c` (das `c` steht für »create«) weglassen.

```
[root@arch-cent bin]# ./htpasswd ../password.list user2
New password:
Re-type new password:
Adding password for user user2
```

Wechseln Sie in eine Verzeichnisebene höher, um den Inhalt der Passwortdatei zu betrachten. Sie werden feststellen, dass die Passwörter verschlüsselt abgespeichert wurden:

```
[root@arch-cent bin]# cd ..
[root@arch-cent apache2]# cat password.list
user1:$apr1$MSADRkjD$RiOkhfPNrxeM3sFWuOvvf/
user2:$apr1$puQJAZbN$sgfv1MEZ2AnimKzqIY22DO
```

Um die Authentifizierung für die standardmäßig vorhandene Webseite anzufordern, müssen Sie die Konfigurationsdatei *httpd.conf* anpassen. Suchen Sie die Sektion `<Directory />` und fügen Sie die unten fett gedruckten Zeilen hinzu:

```
<Directory />
    Options FollowSymLinks
    AllowOverride None
    Order deny,allow
    Deny from all
    AuthName "Authentifizierung erforderlich"
    AuthType Basic
    AuthUserFile password.list
    require valid-user
</Directory>
```

Der Inhalt des Statements `AuthName` ist frei wählbar und wird dem Benutzer im Authentifizierungsdialog angezeigt. Mit `AuthType Basic` wird die Methode der Authentifizierung eingestellt. `AuthUserFile` zeigt auf die soeben erstellte Passwortdatei. Diese Datei befindet sich im vorliegenden Beispiel im Hauptverzeichnis des Webservers, weshalb hier auf eine Pfadangabe verzichtet werden kann. Das Statement `require valid-user` legt fest, dass sich hier nur gültige Benutzer anmelden können. Es ist übrigens auch möglich, konkreten Benutzern exklusiven Zugriff einzuräumen.

Sie können die Authentifizierung nun mithilfe eines Webbrowsers testen und anschließend die soeben vorgenommenen Änderungen aus der Datei *httpd.conf* löschen oder auskommentieren. Diese Maßnahme ist für die nächste Lektion nötig.

.htaccess

Eine andere Methode der Zugriffssteuerung ist die Verwendung einer verborgenen Datei mit der Bezeichnung *.htaccess*. Dieses Verfahren unterscheidet sich kaum von dem bereits beschriebenen Verfahren. Sie können sogar die zuvor erstellte Passwortdatei verwenden.

Damit Sie *.htaccess*-Dateien verwenden können, müssen Sie zuerst in der Datei *httpd.conf* die folgende Einstellung in der gewünschten `<Directory />`-Direktive konfigurieren:

```
AllowOverride All
```

Erstellen Sie anschließend eine Datei mit der Bezeichnung *.htaccess* und folgendem Inhalt im DocumentRoot des Webservers:

```
AuthName "Authentifizierung erforderlich"
AuthType Basic
AuthUserFile password.list
require valid-user
```

Wie Sie sehen, sind Inhalt und Syntax mit den Einträgen in der *httpd.conf*-Datei, die Sie vorher gemacht haben, absolut identisch. Der einzige Vorzug dieser Datei ist, dass sie auch ein gewöhnlicher Benutzer erstellen kann. Dazu benötigt der Benutzer keine Schreibrechte auf die Datei *httpd.conf*, sondern lediglich im jeweiligen Webverzeichnis. Auf diese Art kann die Verwaltung einzelner Webverzeichnisse an unterschiedliche Benutzer delegiert werden.

Testen Sie nun erneut den Zugriff auf den Webserver. Wenn Sie wiederholt Tests durchführen, sollten Sie nach jedem Test den Cache Ihres Browsers leeren. Ansonsten werden Ihre Tests zu unerwarteten Ergebnissen führen, weil Webseiten eventuell ohne Authentifizierung direkt aus dem Browsercache heraus angezeigt werden. Sie können die Datei *.htaccess* ganz einfach immer in die Verzeichnisse kopieren, auf die Sie den Zugriff beschränken wollen. Es sollte bei hochfrequentierten Servern aber berücksichtigt werden, dass die Verwendung dieser Methode die Performance des Servers negativ beeinflusst.

.htgroup

In größeren Umgebungen kann es sinnvoll sein, die Benutzer, die auf Webressourcen zugreifen dürfen, zu gruppieren. Im Prinzip können Sie die Datei zur Gruppierung von Benutzern nennen, wie Sie wollen, aber der Name *.htgroup* hat sich im Laufe der

Jahre eingebürgert. Diese Datei durch einen Punkt am Anfang zu verstecken ist grundsätzlich eine gute Idee, um die Datei vor neugierigen Augen zu verbergen, auch wenn sie keine Passwörter enthält. Um das Verfahren zu testen, erstellen Sie passend zu den bereits erstellten Benutzern in der Datei *password.list* die Datei *.htgroup*, ebenfalls im DocumentRoot-Verzeichnis des Servers. Die Datei *.htgroup* könnte diesen Inhalt haben:

```
all:user1,user2
agents:user1
```

Ändern Sie nun Ihre bestehende *.htaccess*-Datei ab, um die Gruppendatei zu verwenden:

```
AuthName "Zugriff nur für Agents"
AuthType Basic
AuthUserFile password.list
AuthGroupFile .htgroup
require group agents
```

Beachten Sie bitte, dass `AuthUserFile` angegeben werden muss, auch wenn die Zugangsvoraussetzung von der Gruppe abhängig ist. `AuthName` zeigt einem Benutzer schon beim ersten Zugriff, wer auf dieses Webverzeichnis zugreifen darf. Wenn Sie einen Test durchführen (nachdem Sie den Browsercache geleert haben), werden Sie feststellen, dass `user1` auf das Webverzeichnis zugreifen darf und `user2` nicht.

Hinweis

Es gibt wesentlich komplexere Methoden, um den Zugriff auf einen Webserver bzw. auf dessen Verzeichnisse zu steuern, wie Sie es sich wahrscheinlich schon gedacht haben. Für die Prüfung müssen Sie aber lediglich die bisher beschriebenen Mechanismen kennen.

Zugriff über Namen und IP-Adressen steuern

Wenn eine Webseite lediglich im lokalen Netz erreichbar ist, kann man möglicherweise auf die Authentifizierung der Benutzer verzichten. Das erspart sowohl dem Administrator Zeit, weil er nicht pro User den Zugriff einrichten muss, als auch dem Benutzer, weil er sich nicht mehr an der Webseite anmelden muss. In solchen Fällen kann eine Zugriffssteuerung über die IP-Adressen der Clientcomputer oder deren Host- bzw. Domänennamen erfolgen. Zuständig sind hierfür die Module *mod_authz_host* und *mod_access_compat*. Sie müssen diese Module nicht konfigurieren, denn sie werden standardmäßig geladen. Die Erlaubnis zum Zugriff auf eine Webseite erfolgt dann innerhalb deren Konfiguration mithilfe der Direktive `Allow`. Hier ein paar Beispiele:

```
Allow from beispiel.com
```

Erlaubt den Computern der Domäne beispiel.com den Zugriff. Es ist die Angabe kompletter FQDNs wie auch die Angabe von Domänen.

```
Allow from 192.168
Allow from 192.168.0.0/16
Allow from 192.168.0.0/255.255.0.0
```

Die vorangehenden drei Zeilen sind von der Aussage her identisch. In allen drei Fällen wird Clients, die private Adressen der C-Klasse verwenden, der Zugriff gestattet. Selbstverständlich können auch IPv6-Adressen herangezogen werden:

```
Allow from 2001:6F8:1D2D::/48
```

Module integrieren

Funktionen, die von den meisten Anwendern benötigt werden, sind im Kern von Apache fest integriert. Wenn weitere Funktionen benötigt werden, müssen diese als Module eingebunden werden. Die Integration von Perl und insbesondere PHP gehört allerdings immer noch zu den Standardaufgaben bei der Apache-Konfiguration.

Integration von mod_php

PHP ist eine Skriptsprache, mit der es möglich ist, HTML-Code dynamisch zu generieren. Sehr viele der heutigen Webseiten verwenden PHP auch, um Zugriffe auf MySQL-Datenbanken durchzuführen. Die folgende Anleitung wird Ihnen dabei helfen, PHP in Ihren Webserver zu integrieren. Erstellen Sie zunächst ein Installationsverzeichnis und wechseln Sie hinein:

```
[root@arch-cent /]# mkdir /usr/src/php
[root@arch-cent /]# cd /usr/src/php
```

Laden Sie anschließend den entsprechenden `tar`-Ball für PHP vom Webserver *php.net* herunter und packen Sie ihn aus:

```
[root@arch-cent php]# wget http://de.php.net/distributions/php-5.3.6.tar.bz2
[root@arch-cent php]# tar -xvjf php-5.3.6.tar.bz2
```

Beachten Sie bitte, dass es inzwischen eine neuere Version von PHP geben könnte. Besuchen Sie ggf. die *php.net*-Webseite um Näheres zu erfahren, falls der oben genannte Downloadlink nicht mehr funktionieren sollte. Wechseln Sie nun in das entstandene Installationsverzeichnis:

```
[root@arch-cent php]# cd php-5.3.6
```

Die Konfiguration von PHP setzt das Entwicklungspaket *libxml2-devel* voraus. Sie sollten das Paket also spätestens jetzt nachinstallieren:

```
[root@arch-cent php-5.3.6]# yum install libxml2-devel
```

Bei Systemen, die auf Debian basieren, heißt das entsprechende Paket übrigens *libxml2-dev*. Jetzt können Sie die Konfiguration durchführen. Sie müssen bei der Konfiguration zumindest das Verzeichnis angeben, in dem Apache seine Erweiterungen erwartet:

```
[root@arch-cent php-5.3.6]# ./configure --with-apxs2=/usr/local/apache2/bin/apxs
```

Wenn die Konfiguration sauber durchgelaufen ist, können Sie PHP kompilieren, testen und installieren. Der Test ist übrigens optional und dauert relativ lange. Wenn Sie also lediglich ein Testsystem zu Prüfungsvorbereitungszwecken konfigurieren, können Sie den Test getrost überspringen.

```
[root@arch-cent php-5.3.6]# make
[root@arch-cent php-5.3.6]# make test
[root@arch-cent php-5.3.6]# make install
```

Die Installationsroutine hat der Konfigurationsdatei *httpd.conf* eine Zeile hinzugefügt, die dafür sorgt, dass das PHP-Modul beim nächsten Neustart des Webservers geladen wird:

```
LoadModule php5_module        modules/libphp5.so
```

Damit Apache die Dateierweiterungen von PHP selbstständig erkennen kann, sollten Sie allerdings noch die folgende Zeile von Hand in die globale Sektion der Datei *httpd.conf* eintragen:

```
AddType application/x-httpd-php .php .phtml
```

Überprüfen Sie die Konfiguration des Webservers und starten Sie ihn anschließend neu, um die Konfigurationsänderungen zu übernehmen:

```
[root@arch-cent bin]# ./apachectl configtest
[root@arch-cent bin]# ./apachectl restart
```

PHP-Programmierung ist nicht Bestandteil Ihrer Prüfung, aber Sie finden im Internet leicht Beispielskripte um Ihre Konfiguration zu testen, wenn Sie das möchten.

Integration von mod_perl

Perl ist eine Skriptsprache, die ursprünglich nicht für Webseitenprogrammierung geschrieben worden ist. In den letzten Jahren wurde Perl, zumindest was Webseiten

anbelangt, mehr und mehr von PHP verdrängt. Die Installation des Perl-Moduls für Apache funktioniert etwas anders als die Integration von PHP. Konsequenterweise kommt bei der Installation ein Perl-Skript zum Einsatz, was natürlich voraussetzt, dass der Perl-Interpreter schon auf dem System vorhanden ist. Bei den meisten aktuellen Linux-Distributionen ist dieser Interpreter in einer Standardinstallation schon enthalten. Wenn Sie die folgenden Schritte durchführen, sollten Sie eine funktionierende Apache-Konfiguration mit Perl erhalten. Legen Sie zunächst wieder ein Installationsverzeichnis an, und wechseln Sie hinein:

```
[root@arch-cent /]# mkdir /usr/src/mod_perl
[root@arch-cent /]# cd /usr/src/mod_perl
```

Laden Sie anschließend das Perl-Modul als `tar`-Ball von der Apache-Webseite herunter:

```
[root@arch-cent mod_perl]# wget http://perl.apache.org/dist/mod_perl-2.0-
current.tar.gz
```

Es könnte jetzt natürlich schon neuere Versionen des Moduls geben. Passen Sie dann die Kommandos (bzw. hier die URL) entsprechend an. Packen Sie den `tar`-Ball aus:

```
[root@arch-cent mod_perl]# tar -xvzf mod_perl-2.0-current.tar.gz
```

Wechseln Sie anschließend in das gerade entstandene Installationsverzeichnis und führen Sie das Perl-Skript zur Erstellung des Makefiles aus wie angegeben. Die Variable `MP_APXS` gibt hierbei den absoluten Pfad zu den Apache-Erweiterungen an.

```
[root@arch-cent mod_perl]# cd mod_perl-2.0.5
[root@arch-cent mod_perl-2.0.5]# perl Makefile.PL MP_APXS=/usr/local/apache2/
bin/apxs
```

Ab hier unterscheidet sich die Installation nicht mehr von der Installation anderer Programme, die als `tar`-Ball vorliegen:

```
[root@arch-cent mod_perl-2.0.5]# make
[root@arch-cent mod_perl-2.0.5]# make install
```

Sie müssen jetzt nur noch dafür sorgen, dass Apache das Perl-Modul lädt. Fügen Sie der Konfigurationsdatei *httpd.conf* deshalb noch folgende Zeile hinzu:

```
LoadModule perl_module modules/mod_perl.so
```

Starten Sie Apache anschließend einmal neu, damit die Konfigurationsänderungen wirksam werden:

```
[root@arch-cent mod_perl-2.0.5]# /usr/local/apache2/bin/apachectl restart
```

Auch Perl-Programmierung ist nicht Bestandteil der LPI-Prüfungen. Deshalb muss ich auch hier bezüglich Perl-Programmen zu Testzwecken auf das Internet verweisen.

Protokollierungseinstellungen

Was die Protokollierung anbelangt, werden Sie bei den meisten Apache Webservern, die paketbasiert installiert worden sind, feststellen, dass die Protokolle der Webserver im Verzeichnis */var/log/apache2* liegen. Hier befinden sich im Normalfall die Dateien *access.log* und *error.log*. Da diese Dateien üblicherweise von *logrotate* rotiert werden, finden Sie hier auch noch archivierte, komprimierte Versionen dieser beiden Dateien.

In der Datei *access.log* wird jede einzelne URL, die abgerufen wird, protokolliert. Es ist deshalb möglich, sehr genau festzustellen, von welchen Computern aus auf welche Inhalte zugegriffen worden ist. Fehlgeschlagene Zugriffe werden hier ebenfalls protokolliert.

Die Datei *error.log* enthält keine Einträge, die auf Zugriffe durch Benutzer zurückzuführen sind, sondern vielmehr nur schwerwiegende Fehler. Hierbei kann es sich um Fehler durch Fehlkonfiguration des Servers handeln oder Module, die sich aus irgendeinem Grund nicht laden lassen. Sie sollten diese Datei also unbedingt konsultieren, wenn es zu Fehlfunktionen des Servers kommt.

Wenn Sie Ihren Webserver nach der Anleitung in diesem Buch konfiguriert haben, finden Sie die Protokolldateien in einem Unterverzeichnis des Webservers, nämlich in */usr/local/apache2/logs*. Die Dateinamen sind hier *access_log* und *error_log*.

Die Position und die Dateinamen der Protokolldateien werden in der Konfigurationsdatei *httpd.conf* festgelegt. Hierbei sind die Pfadangaben relativ ab dem Server-Root zu betrachten:

```
CustomLog "logs/access_log" common
ErrorLog "logs/error_log"
```

Leistungseinstellungen

In der Konfigurationsdatei *httpd.conf* finden Sie einige Direktiven, die sich auf die Leistung des Servers auswirken. Diese Direktiven müssen an die Umstände angepasst werden, unter denen der Server läuft. Wenn zu erwarten ist, dass ein Webserver ständig von sehr vielen Benutzern verwendet wird, sollten natürlich auch entsprechend viele Arbeitsprozesse zur Verfügung stehen, um diese Anfragen zu handhaben. Umgekehrt würden zu viele Arbeitsprozesse unnötigerweise die Systemressourcen verschwenden. Entsprechend den Anforderungen können Sie folgende Parameter konfigurieren:

- `StartServers`: Anzahl der Serverprozesse beim Start
- `MinSpareServers`: minimale Anzahl von Serverprozessen, die als Reserve zur Verfügung stehen müssen
- `MaxSpareServers`: maximale Anzahl von Serverprozessen, die als Reserve zur Verfügung stehen dürfen
- `ServerLimit`: maximale Anzahl von Serverprozessen, die zur Laufzeit des Servers ausgeführt werden dürfen
- `MaxClients`: maximale Anzahl von Serverprozessen, die gleichzeitig ausgeführt werden dürfen
- `MaxRequestsPerChild`: maximale Anzahl von Anfragen, die an einen Serverprozess gesendet werden dürfen

Konfiguration virtueller Hosts

Wenn auf einem Webserver mehrere Webseiten gehostet werden sollen, gibt es grundsätzlich drei verschieden Möglichkeiten, um diese Seiten voneinander zu unterscheiden. Sie könnten jeder Webseite eine eigene IP-Adresse zuordnen und die jeweiligen IP-Adressen in den entsprechenden DNS-Zonen eintragen (lassen). Da öffentliche IP-Adressen Geld kosten, ist das allerdings keine sehr gute Wahl. Eine weitere Möglichkeit besteht darin, unterschiedliche TCP-Ports für die verschiedenen Webseiten zu verwenden. Das ist allerdings bei öffentlich zugänglichen Webseiten schon aus kosmetischen Gründen abzulehnen, weil die Besucher der Webseite den entsprechenden Port dann in der URL-Zeile angeben müssten. Das sähe dann beispielsweise so aus: *http://www.beispiel.de:82*, also nicht gerade besonders professionell. Die dritte und im Normalfall beste Lösung ist die Unterscheidung von Webseiten über den Hostnamen. Die URL, die der Benutzer in seinen Browser eingegeben hat, wird innerhalb der Abfrage an den Webserver übermittelt. Diese URL kann der Webserver auswerten und einem virtuellen Host zuordnen.

Unabhängig davon, welche dieser drei Varianten konfiguriert werden soll, müssen in der Datei *httpd.conf* virtuelle Hosts erstellt werden. Hierbei handelt es sich jeweils um Gruppen von Direktiven, die am Anfang durch `<VirtualHost>` und am Ende mit `</VirtualHost>` eingeschlossen werden. Eine komplette Konfiguration für einen virtuellen Host könnte z. B. so aussehen:

```
<VirtualHost www.super-admin.org>
    ServerAdmin webmaster@super-admin.org
    DocumentRoot /var/www/super-admin
    ServerName www.super-admin.org
    ErrorLog logs/super-admin-error_log
    CustomLog logs/super-admin-access_log common
</VirtualHost>
```

Der Inhalt der Direktiven erklärt sich fast von selbst. Der wichtigste Eintrag ist jedoch `DocumentRoot`, weil diese Direktive dem Webserver sagt, in welchem Verzeichnis der zur angegebenen URL passende Content zu finden ist. Die einleitende Zeile fällt unterschiedlich aus, je nachdem, auf welche Art die Webseite identifiziert werden soll. Hier ein paar Beispiele:

- Identifikation durch die IPv4-Adresse:

```
<VirtualHost 24.215.7.162>
```

- Identifikation durch die IPv6-Adresse:

```
<VirtualHost [2a01:198:5dd:7a03:a00:27ff:fe2d:5987]>
```

- Identifikation durch den verwendeten TCP-Port:

```
<VirtualHost *:82>
```

- Identifikation durch den Hostnamen und einen TCP-Port:

```
<VirtualHost www.super-admin.org:82>
```

Beachten Sie, dass die Angabe eines TCP-Ports innerhalb eines virtuellen Hosts nicht verhindert, dass der Server weiterhin zusätzlich an Port 80 lauscht.

Die Redirect-Direktive

Redirects werden z. B. verwendet, wenn der Inhalt eines Webverzeichnisses an eine andere Stelle verschoben wurde. Der Inhalt des entsprechenden Verzeichnisses kann sich hierbei auch auf einem anderen Webserver befinden. In der Redirect-Direktive kann ein Statuscode angegeben werden. Ist das nicht der Fall, wird per Voreinstellung der Wert 302 (found) an den Client gesendet. Das lokale Verzeichnis wird relativ zum DocumentRoot angegeben, während das Ziel absolut und als URL notiert wird.

Im folgenden Beispiel wird das lokale Unterverzeichnis */pdf* an die URL *http://www.super-admins.org/dokumente* umgeleitet. An den Client wird der Statuscode 301 (moved permanently) ausgegeben. Diese Statusinformation ist besonders für Suchmaschinen interessant, weil eine dauerhafte Änderung (im Gegensatz zu einer temporären Änderung) im Index der Suchmaschinen berücksichtigt werden sollte.

```
Redirect permanent /pdf http://www.super-admins.org/dokumente
```

Für den Statuscode eines Redirects können folgende Argumente verwendet werden:

- `permanent`: Der Server sendet den Statuscode 301 (moved permanently), um anzuzeigen, dass die Ressource dauerhaft unter der neuen URL erreichbar ist.
- `temp`: Der Server gibt den Statuscode 302 (found) zurück. Dies ist die Standardeinstellung, wenn Sie kein Statusargument angeben.

- `seeother`: Der Server gibt den Statuscode 303 (see other) zurück. Das bedeutet, dass die ursprüngliche Ressource ersetzt wurde.
- `gone`: Der Server sendet den Statuscode 410 (gone). Die angeforderte Ressource ist auf dem Server dauerhaft nicht mehr erreichbar. Es ist keine neue URL bekannt.

Sie sollten übrigens nicht versuchen, ein Unterverzeichnis des ursprünglichen Webverzeichnisses als neues Ziel in einem Redirect zu verwenden. Diese Methode wird von der Redirect-Direktive nicht unterstützt und führt zu einer Fehlermeldung im Browser des Clients.

208.2 Apache für HTTPS konfigurieren

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen Webserver für die Nutzung von HTTPS zu konfigurieren.

Wichtigste Wissensgebiete:

- SSL-Konfigurationsdateien, -Begriffe und -Dienstprogramme
- Fähigkeit, einen privaten Serverschlüssel und einen Zertifikats-Request (CSR) für eine kommerzielle Zertifizierungsstelle zu erstellen
- Fähigkeit, ein selbst signiertes Zertifikat mithilfe einer privaten CA zu erstellen
- Fähigkeit, ein Zertifikat und einen Schlüssel zu installieren
- Konfiguration virtueller Hosts mithilfe von SNI
- Wissen über das Verhalten virtueller Hosts in Verbindung mit SSL
- Sicherheitsbelange bei der SSL-Nutzung, deaktivieren unsicherer Protokolle und Verschlüsselungen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- Apache-Konfigurationsdateien
- */etc/ssl/*, /etc/pki/**
- `openssl`,`CA.pl`
- `SSLEngine`, `SSLCertificateKeyFile`, `SSLCertificateFile`
- `SSLCACertificateFile`, `SSLCACertificatePath`
- `SSLProtocol`, `SSLCipherSuite`, `ServerTokens`, `ServerSignature`, `TraceEnable`

Allgemeines

Auf den letzten Seiten haben Sie eine Menge über die Grundkonfiguration eines Apache Webservers erfahren. In der Praxis werden Sie aber noch weitere Funktionen

eines Webservers benötigen. Webserver, die von Internet Service Providern betrieben werden, hosten normalerweise mehrere Webseiten. Es ist nicht ungewöhnlich, dass ein einziger Apache Webserver die Seiten von fünfzig und mehr Kunden hostet. Zu diesem Zweck werden virtuelle Hosts auf den Servern konfiguriert.

Eine weitere häufig benötigte Funktionalität ist SSL (Secure Socket Layer). Mit SSL wird die Authentizität einer Webseite garantiert, die Integrität gewahrt und der Inhalt während der Übertragung durch Verschlüsselung gesichert. Damit Sie HTTPS verwenden können, muss das entsprechende Modul SSL installiert sein.

Wenn eine Webseite eine Authentifizierung verlangt oder vertrauliche Daten übermittelt, sollte die Verbindung verschlüsselt erfolgen. Es wäre sonst zu befürchten, dass jemand die Verbindung abhorcht und dabei die Authentifizierungsdaten oder den vertraulichen Inhalt der Verbindung abfängt. In einer solchen Situation kommt HTTPS zum Einsatz. Im Grunde genommen handelt es sich hier immer noch um eine HTTP-Kommunikation, aber die Datenpakete werden in einer SSL-Verbindung gesichert übermittelt.

Das gesamte Konstrukt basiert auf einer Infrastruktur für öffentliche Schlüssel (Public Key Infrastructure, PKI). Da hier Zertifikate im Spiel sind, können Sie HTTPS nicht nur für die Verschlüsselung verwenden. Es ist auch eine gegenseitige Authentifizierung von Webserver und Client möglich, wobei in der Praxis meist nur die Authentifizierung des Servers gegenüber dem Client implementiert wird. Schließlich müssen Sie als Kunde wissen, ob Sie einer Webseite trauen können, wenn Sie z. B. Kreditkartentransaktionen durchführen wollen.

Wenn Sie eine mit SSL gesicherte Webseite konfigurieren, auf die vom Internet aus zugegriffen wird, sollten Sie über ein Zertifikat für diese Seite verfügen, das von einer öffentlichen Zertifizierungsstelle (z. B. GlobalSign, Thawte, Verisign) ausgestellt wurde. Ansonsten erhalten Benutzer beim Zugriff auf die Webseite eine Warnmeldung und meiden die Seite möglicherweise. Damit das Zertifikat eines Webservers von einem Client ohne Warnmeldungen akzeptiert wird, müssen drei Kriterien erfüllt sein:

- Das Zertifikat muss von einer vertrauten Zertifizierungsinstitution stammen.
- Das Zertifikat muss gültig sein (nicht abgelaufen oder zurückgezogen).
- Der Antragstellername im Zertifikat muss mit der URL übereinstimmen, die ein Benutzer in den Browser eingibt.

Das bedeutet, dass Sie problemlos zu Testzwecken ein Zertifikat verwenden können, das Sie selbst ausgestellt haben. Wenn Sie SSL verwenden, um so etwas wie eine Verwaltungswebseite (z. B. phpMyAdmin, CUPS, Webmin o. Ä.) abzusichern, wollen Sie wahrscheinlich lediglich sicherstellen, dass die Kommunikation verschlüsselt erfolgt und nicht die Echtheit des Zielservers überprüfen. Auch in solchen Fällen können Sie

ohne weiteres selbst signierte Zertifikate verwenden und entsprechende Warnmeldungen des Browsers ignorieren.

Konfiguration von SSL mittels openssl

Die Konfiguration auf den folgenden Seiten basiert diesmal auf einer Apache-Installation mittels `yum`. Auf diese Art lernen Sie gleichzeitig eine andere Variante von Apache kennen. Der wesentliche Unterschied zum manuell installierten Webserver sind die Positionen der Verzeichnisse des Servers im Dateisystem. Installieren Sie Apache mit:

```
[root@arch-cent ~]# yum install httpd
```

Wenn Sie CentOS verwenden, wird das Modul *mod_ssl* automatisch mitinstalliert. Bei einem System, auf dem Fedora ausgeführt wird, müssen Sie SSL zusätzlich installieren:

```
[root@arch-fc /]# yum install mod_ssl
```

Bei den Red Hat-basierten Systemen finden Sie die Konfigurationsdateien im Verzeichnis */etc/httpd*. Die Datei *httpd.conf* liegt jeweils im Verzeichnis */etc/httpd/conf*. Alle anderen Verzeichnisse sind durch Lesen der Hauptkonfigurationsdatei problemlos zu ermitteln (z. B. `ServerRoot` und `DocumentRoot`).

Wenn Sie mittels `yum` einen Apache Webserver installiert haben, verfügt dieser bereits über ein selbst signiertes Zertifikat. Sie sollten aber natürlich wissen, wie man ein solches Zertifikat mit `openssl` selbst erzeugt und an eine Webseite bindet. Sie finden hier wieder eine Schritt-für-Schritt-Anleitung. Denken Sie aber daran, dass diesem Zertifikat nicht öffentlich vertraut wird. Erstellen Sie zunächst ein Unterverzeichnis für das Zertifikat und den privaten Schlüssel und wechseln Sie anschließend hinein.

```
[root@arch-cent /]# mkdir /etc/httpd/ssl
[root@arch-cent /]# cd /etc/httpd/ssl
```

Verwenden Sie anschließend `openssl`, um das Zertifikat und den Schlüssel zu generieren. Die Benutzereingaben sind wieder fett gedruckt:

```
[root@arch-cent ssl]# openssl req -new -x509 -nodes -out
arch-cent.homelinux.net.crt -keyout arch-cent.homelinux.net.key
Generating a 1024 bit RSA private key
.................++++++
.....................++++++
writing new private key to 'arch-cent.homelinux.net.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) [GB]:DE
State or Province Name (full name) [Berkshire]:Germany
Locality Name (eg, city) [Newbury]:Berlin
Organization Name (eg, company) [My Company Ltd]:Maassen
Organizational Unit Name (eg, section) []:.
Common Name (eg, your name or your server's hostname)
   []:arch-cent.homelinux.net
Email Address []:harald@nwa-net.de
```

Es sollten nun zwei Dateien im aktuellen Verzeichnis liegen. Die Datei mit der Erweiterung *crt* enthält das Zertifikat mit dem öffentlichen Schlüssel, während die Datei mit der Erweiterung *key* den privaten Schlüssel enthält:

```
[root@arch-cent ssl]# ls -l
insgesamt 16
-rw-r--r-- 1 root root 1277 20. Jun 21:50 arch-cent.homelinux.net.crt
-rw-r--r-- 1 root root  887 20. Jun 21:50 arch-cent.homelinux.net.key
```

Damit der Webserver das Zertifikat und den Schlüssel auch verwendet, müssen Sie die entsprechenden Direktiven der Datei *httpd.conf* anpassen. Im Fall von CentOS bzw. Fedora wurde der hierfür zuständige Bereich per `include`-Anweisung in die Datei */etc/httpd/conf.d/ssl.conf* ausgelagert. Öffnen Sie diese Datei mit einem Editor und suchen Sie nach den Einträgen `SSLCertificateFile` und `SSLCertificateKeyFile`. Passen Sie die Dateien an Ihr eigenes Zertifikat und den privaten Schlüssel an:

```
SSLCertificateFile /etc/httpd/ssl/arch-cent.homelinux.net.crt
SSLCertificateKeyFile /etc/httpd/ssl/arch-cent.homelinux.net.key
```

Damit die Konfigurationseinstellungen sofort wirksam werden, sollten Sie Apache neu starten.

```
[root@arch-cent ssl]# apachectl restart
```

Greifen Sie nun mit einem beliebigen Webbrowser auf die abgesicherte Webseite zu, indem Sie das Präfix *https://* angeben. Sie sollten dann eine Sicherheitswarnung erhalten, weil das vom Webserver verwendete Zertifikat nicht von einer vertrauenswürdigen Zertifizierungsstelle ausgestellt worden ist. Die Warnung können Sie in diesem Fall natürlich ignorieren. Nur wenn Sie eine abgesicherte Webseite anderen Benutzern zur Verfügung stellen, sollten Sie ein Zertifikat von einer öffentlichen Zertifizierungsstelle erwerben.

Server Name Indication (SNI)

Vor Apache v2.2.12 mit OpenSSL v0.9.8 konnte pro IP-Adresse nur jeweils eine SSL-Webseite gebunden werden. Das bedeutete für öffentlich zugängliche HTTPS-Webseiten, dass diese nicht preisgünstig gehostet werden konnten, weil immer eine eigene öffentliche IP-Adresse nötig war. Dieses Manko behebt *Server Name Indication (SNI)*. Das Einzige, was Sie tun müssen, ist mehrere virtuelle Hosts zu erstellen, wie Sie es bereits von normalen HTTP-Seiten her kennen. Sie können zu diesem Zweck weitere Konfigurationsdateien erstellen oder eine bestehende Konfiguration modifizieren. Ein Beispiel für zwei HTTPS-Webseiten sehen Sie hier:

```
NameVirtualHost *:443

<VirtualHost *:443>
 ServerName www.beispiel.com
 DocumentRoot /var/www/beispiel
 SSLEngine on
 SSLCertificateFile /etc/ssl/www_beispiel_com.crt
 SSLCertificateKeyFile / etc/ssl/www_beispiel_com.key
 SSLCertificateChainFile / etc/ssl/MyCertCA.crt
</VirtualHost>

<VirtualHost *:443>
 ServerName www.exampel.org
 DocumentRoot /var/www/exampel
 SSLEngine on
 SSLCertificateFile / etc/ssl/www_exampel_org.crt
 SSLCertificateKeyFile / etc/ssl/www_exampel_org.key
 SSLCertificateChainFile / etc/ssl/MyCertCA.crt
</VirtualHost>
```

SSL-Zertifikate mittels CA.pl erstellen

Sie können die Zertifizierungsstelle und die Zertifikate, die Sie benötigen, auch mithilfe des Pearl-Skripts `CA.pl` erstellen. Hierbei handelt es sich um ein Frontend für das Kommando `openssl`. Sie werden sehen, dass bei der Verwendung von `CA.pl` kaum Optionen oder Parameter angegeben werden müssen. Vielmehr handelt es sich um ein interaktives Skript, das alle benötigten Angaben vom Benutzer erfragt und anschließend die daraus resultierenden `openssl`-Kommandos zusammenbaut.

Erstellen der CA-Hierarchie

Wo sich das Skript `CA.pl` befindet, ist von der verwendeten Distribution abhängig. Verwenden Sie `locate` oder `find`, um es zu finden. Sie können innerhalb des Skripts

Änderungen vornehmen, um es an Ihre Bedürfnisse anzupassen. Das gilt z. B. für die Gültigkeitsdauer von ausgestellten Zertifikaten. `CA.pl` arbeitet mit relativen Verzeichnisangaben. Das sollten Sie bei der Erstellung einer neuen CA berücksichtigen. Wenn Sie keine Änderungen an dem Skript vornehmen, erstellt `CA.pl` unterhalb des aktuellen Verzeichnisses eine Verzeichnisstruktur mit dem Namen *demoCA*. Um dieses Verhalten zu ändern, sollten Sie im Skript die folgende Zeile ändern:

von `$CATOP="./demoCA";` nach `$CATOP="/etc/pki";`

Dasselbe gilt für zwei Zeilen in der Konfigurationsdatei */etc/ssl/openssl.cnf*, die denselben Inhalt aufweisen:

aus `dir=./demoCA` wird `dir=/etc/pki`

So wird das Verzeichnis */etc/pki* gleich automatisch zu einem sinnvollen zentralen Speicherort für zertifikatsbezogene Daten.

Zum Erstellen der CA übergeben Sie einfach die Option `-newca`. Im folgenden Beispiel sind wieder der Übersichtlichkeit wegen alle Benutzereingaben fett gedruckt. Aus Platzgründen wurden einige für das Verständnis unwichtige Zeilen entfernt. Es müssen übrigens nicht alle Felder zwingend ausgefüllt werden:

```
root@archangel:/# /usr/lib/ssl/misc/CA.pl -newca
CA certificate filename (or enter to create)
Making CA certificate ...
Generating a 2048 bit RSA private key
writing new private key to './demoCA/private/cakey.pem'
Enter PEM pass phrase: ****** (wird nicht angezeigt)
Verifying - Enter PEM pass phrase: ****** (wird nicht angezeigt)
-----
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.
-----
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Germany
Locality Name (eg, city) []:Berlin
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Maassen
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:Harald Maassen
Email Address []:harald@lpic-2.de
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:sichtbares_passwort
An optional company name []:Maassen
Using configuration from /usr/lib/ssl/openssl.cnf
```

```
Enter pass phrase for /etc/pki/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
```

Die Zertifizierungsstelle ist jetzt einsatzbereit. Sehen Sie sich den Inhalt des Verzeichnisses */etc/pki* einmal genauer an. Hier finden Sie z. B. die Datei *cacert.pem*, die Sie benötigen, wenn ein Browser Ihrer Zertifizierungsstelle vertrauen soll. Importieren Sie die Datei in die vertrauenswürdigen Stammzertifizierungsstellen des Browsers.

Der nächste Schritt ist die Erstellung einer Zertifikatsanforderung. Beachten Sie bitte, dass `CA.pl` immer im aktuellen Verzeichnis operiert und auch hier die entsprechenden Dateien generiert. Da Sie diese Dateien jedoch ohnehin später an einen anderen Ort im Dateisystem verschieben werden, ist dieses Verhalten in Ordnung. Führen Sie den Request wie folgt durch:

```
root@archangel:/# /usr/lib/ssl/misc/CA.pl -newreq
Generating a 2048 bit RSA private key
writing new private key to 'newkey.pem'
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
Country Name (2 letter code) [AU]:DE
State or Province Name (full name) [Some-State]:Germany
Locality Name (eg, city) []:Berlin
Organization Name (eg, company) [Internet Widgits Pty Ltd]:Maassen
Organizational Unit Name (eg, section) []:
Common Name (e.g. server FQDN or YOUR name) []:www.lpic-2.de
Email Address []:harald@lpic-2.de
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:klartext challenge!
An optional company name []:Maassen
Request is in newreq.pem, private key is in newkey.pem
```

Auch diesmal wurden weniger wichtige Zeilen entfernt. Die letzte Zeile zeigt das Ergebnis des Kommandos. Es wurden ein privater Schlüssel (*newkey.pem*) und eine Anforderungsdatei (*newreq.pem*) generiert.

Hinweis

Wenn Sie ein Zertifikat von einer kommerziellen Zertifizierungsstelle anfordern müssen, benötigen Sie ebenfalls die Dateien *newkey.pem* und *newreq.pem*. Die folgenden Schritte würde dann der Betreiber dieser CA auf einem seiner Systeme durchführen und Ihnen das fertige Zertifikat übermitteln.

Lassen Sie die beiden Dateien zunächst da, wo sie jetzt sind, damit Sie den Request signieren können:

```
root@archangel:/# /usr/lib/ssl/misc/CA.pl -signreq
Using configuration from /usr/lib/ssl/openssl.cnf
Enter pass phrase for /etc/pki/private/cakey.pem:
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number:
            95:1a:27:fb:5c:3c:ce:a5
        Validity
            Not Before: Jul  9 14:13:56 2013 GMT
            Not After : Jul  9 14:13:56 2014 GMT
        Subject:
            countryName               = DE
            stateOrProvinceName       = Germany
            localityName              = Berlin
            organizationName          = Maassen
            commonName                = www.lpic-2.de
            emailAddress              = harald@lpic-2.de
Signed certificate is in newcert.pem
```

Auch hier wurden wieder unwesentliche Zeilen entfernt. Das neue Zertifikat (*newcert.pem*) finden Sie im aktuellen Verzeichnis.

Um das Zertifikat passwortgeschützt zu sichern, können Sie es mit PKCS12 (PKCS = Public Key Cryptography Standards) sichern. Sie können diese Datei später für den Import auch in fremde Betriebssysteme und Programme verwenden. Dabei werden Sie dann zur Eingabe des bei der Erstellung angegebenen Passworts aufgefordert. Erstellen Sie die PKSC12-Datei so:

```
root@archangel:/# /usr/lib/ssl/misc/CA.pl -pkcs12 "WebZerti"
Enter pass phrase for newkey.pem:
Enter Export Password:
Verifying - Enter Export Password:
PKCS #12 file is in newcert.p12
```

Wie Sie sehen, ist die Datei *newcert.p12* erstellt worden.

Die Benutzung des erstellten Zertifikats und des privaten Schlüssels funktioniert genauso, als hätten Sie das Kommando `openssl` zu deren Erstellung angewendet. Eine Wiederholung der entsprechenden Apache-Konfiguration soll deshalb an dieser Stelle nicht erfolgen. Sie sollten natürlich die generierten Dateien jeweils verschieben und aussagekräftig umbenennen, weil das Skript `CA.pl` immer dieselben Dateinamen erstellt.

Direktiven des Moduls mod_ssl und andere Sicherheitseinstellungen

Das Apache-Modul *mod_ssl* bringt viele zusätzliche Direktiven mit. Einige dieser Direktiven haben Sie bereits selbst angewendet, nämlich als Sie den Serverschlüssel und das Zertifikat in die Apache-Konfiguration eingebunden haben. Für die Prüfung sollten Sie aber weitere Direktiven aus diesem Umfeld kennen.

- `SSLEngine` aktiviert oder deaktiviert SSL. Typische Platzierung:

```
<VirtualHost _default_:443>
SSLEngine on
...
</VirtualHost>
```

- `SSLCertificateKeyFile` enthält den vollständigen Pfad zum privaten Serverschlüssel.
- `SSLCertificateFile` enthält den vollständigen Pfad zum Zertifikat des Servers.
- `SSLCertificateChainFile` enthält den Pfad zu einer optionalen Vertrauenskette. Sie können aus mehreren CA-Zertifikaten, von denen sich letztendlich Ihr Serverzertifikat ableitet, eine einzige Datei zusammenstellen und diese hier hinterlegen. Diese Datei enthält natürlich nur die öffentlichen Schlüssel der CAs.
- `SSLCACertificateFile` ähnelt `SSLCertificateChainFile`, enthält aber die Vertrauenskette der CAs, die die Zertifikate für die Clients ausgestellt haben.
- `SSLCACertificatePath` ähnelt `SSLCACertificateFile`, zeigt aber auf ein Verzeichnis, das die Zertifikate der CAs enthält und nicht auf eine Vertrauenskette in Form einer Datei.
- `SSLProtocol` legt fest, welche SSL-Protokolle erlaubt sind. Infrage kommen hier `SSLv2`, `SSLv3`, `TLSv1`, `TLSv1.1`, `TLSv1.2` oder `All`. Die Groß-/Kleinschreibweise der Protokolle muss berücksichtigt werden!
- `SSLCipherSuite` ist eine durch Doppelpunkte getrennte Auflistung der kryptographischen Algorithmen, die der Server verwenden darf. Je »strenger« die Kette ist, desto mehr Rechenzeit wird benötigt.

Die folgenden Direktiven gehören nicht zum Modul *mod_ssl*, enthalten aber wichtige Sicherheitseinstellungen:

- `ServerTokens` legt fest, welche Informationen Apache über sich selbst an den Client sendet. In der strengsten Einstellung (`Prod`) wird lediglich `Server: Apache` übermittelt. Wird diese Direktive mit `All` konfiguriert, informiert Apache den Client über seine genaue Version, das installierte Betriebssystem und die geladenen Module. Eine solche Konfiguration sollten Sie natürlich vermeiden.
- `ServerSignature` erlaubt es Ihnen automatisch, den Fehlermeldungen eine Fußnote hinzuzufügen. Das ist hilfreich, wenn Sie bei verketteten Konfigurationen herausfinden wollen, welcher Server eine Fehlermeldung generiert hat.

- `TraceEnable` sollte aus Sicherheitsgründen mit `off` konfiguriert werden. Dadurch wird unterbunden, dass potenzielle Angreifer zusätzliche Informationen über den Webserver erlangen können, z. B. ob ein Reverse-Proxy zwischengeschaltet ist oder ob der Server direkt antwortet.

Achtung

Bitte deinstallieren Sie Apache jetzt noch nicht, nur weil dieses Thema nun abgeschlossen ist. Sie können ihn später noch im Zusammenhang mit *nginx* verwenden!

208.3 Implementieren von Squid als Cache-Proxy

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen Proxy-Server zu installieren und zu konfigurieren, einschließlich der Zugriffsregeln, der Authentifizierung und der Ressourcennutzung.

Wichtigste Wissensgebiete:

- Squid 3.x-Konfigurationsdateien, -Begriffe und -Dienstprogramme
- Methoden zur Zugriffsbeschränkung
- Methoden zur Client-Benutzerauthentifikation
- Aufbau und Inhalt von ACLs in den Squid-Konfigurationsdateien

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- *squid.conf*
- `acl`
- `http_access`

Allgemeines

Ein Proxy-Server wird verwendet, um Webinhalte stellvertretend für Clients anzufordern, zwischenzuspeichern und anschließend an die Client-Computer auszuliefern. Ursprünglich wurden Proxy-Server in Unternehmensnetzwerken hauptsächlich eingesetzt, um die Bandbreite des Internetzugangs optimal nutzen zu können und Client-Anfragen mit zwischengespeicherten Webseiten schneller versorgen zu können. Die Möglichkeit, den Zugriff auf bestimmte Webseiten zu sperren oder den Zugriff auf das Web auf bestimmte Client-Computer zu beschränken, spielte zunächst eine untergeordnete Rolle.

Heutzutage hat sich die Aufgabe eines Proxys eher in Richtung Zugriffssteuerung verlagert. Da Webinhalte mehr und mehr dynamisch generiert und Webseiten immer schneller aktualisiert werden, ist die Zwischenspeicherung oftmals nicht sinnvoll. Aufgrund der inzwischen verfügbaren Bandbreiten heutiger Internetzugänge ist das Caching von Webseiten außerdem nicht mehr so wichtig wie früher. Trotzdem hat ein Proxy immer noch seine Daseinsberechtigung. Schließlich kann er verhindern, dass die Mitarbeiter eines Unternehmens ihre Arbeitszeit verschwenden und dass Kinder jugendgefährdendes Material aus dem Internet herunterladen.

Installation des Squid Proxy Servers

Da die fertigen Squid-Proxys, die als Pakete für die verschiedenen Linux-Distributionen vorliegen, sich nicht wesentlich voneinander unterscheiden, können Sie diesmal auch auf ein solches Paket zurückgreifen. Die Konfiguration eines von Hand installierten Squids ist allerdings übersichtlicher und deshalb für Proxy-Einsteiger leichter zu lesen.

Wenn Sie IPv6-Unterstützung für Internetzugriffe implementieren wollen, sollten Sie mindestens die Squid-Version 3.1 installieren. Die Webseite des Projekts finden Sie unter *http://www.squid-cache.org*. Hier finden Sie auch Informationen über Bezugsquellen der aktuellen Squid-Versionen. Die hier Schritt für Schritt dokumentierte Installation von Squid 3.2.0.9 wurde auf einem Debian 6.0-System durchgeführt.

Die ersten Arbeitsschritte sind Routinearbeiten: Arbeitsverzeichnis erstellen, Quellpaket herunterladen, dekomprimieren und entpacken.

```
root@arch-deb:~# mkdir /usr/src/squid
root@arch-deb:~# cd /usr/src/squid
root@arch-deb:/usr/src/squid# wget ftp://ftp.fu-berlin.de/unix/www/squid/
archive/3.2/squid-3.2.0.9.tar.bz2
root@arch-deb:/usr/src/squid# bunzip2 squid-3.2.0.9.tar.bz2
root@arch-deb:/usr/src/squid# tar -xvf squid-3.2.0.9.tar
root@arch-deb:/usr/src/squid# cd squid-3.2.0.9/
```

Das Konfigurationsskript benötigt für seine Ausführung, zumindest unter Debian, das Paket *build-essential*. Sie sollten dieses Paket also spätestens jetzt installieren:

```
root@arch-deb:/usr/src/squid/squid-3.2.0.9# apt-get install build-essential
```

Übergeben Sie dem Konfigurationsskript das Zielverzeichnis des Programms mit der Option `--prefix`.

```
root@arch-deb:/usr/src/squid/squid-3.2.0.9# ./configure --prefix=/usr/local/
squid
```

Wenn die Konfiguration fertig ist, können Sie wie gewohnt kompilieren und installieren:

```
root@arch-deb:/usr/src/squid/squid-3.2.0.9# make
root@arch-deb:/usr/src/squid/squid-3.2.0.9# make install
```

Nach der Installation finden Sie den kompletten Server inklusive Konfigurationsdateien und Verzeichnis für Logfiles unterhalb von */usr/local/squid*. Wenn Sie zunächst mit dieser rohen Konfiguration arbeiten wollen, müssen Sie das Unterverzeichnis *logs* für Squid beschreibbar machen. Da Squid per default unter dem Sicherheitskontext von nobody läuft, können Sie einfach dem User nobody die Eigentümerschaft an dem Verzeichnis für die Logdateien übertragen:

```
root@arch-deb:/# chown nobody /usr/local/squid/var/logs/ -R
```

Da es sich lediglich um eine Testumgebung handelt, in der Sicherheit keine große Rolle spielt, kann diese unsaubere, aber einfache Konfiguration verwendet werden.

Bevor Sie den Server in Betrieb nehmen können, muss der Cache einmalig initialisiert werden:

```
root@arch-deb:/# /usr/local/squid/sbin/squid -z
2011/06/25 19:25:35 kid1| Creating Swap Directories
```

Beachten Sie bitte, dass in der Standardeinstellung nur im Arbeitsspeicher zwischengespeichert wird. Um einen Festplattencache einzurichten, muss zunächst die Konfigurationsdatei *squid.conf* modifiziert werden. Sie finden auf den nächsten Seiten genaue Informationen über diese Konfigurationsdatei. Wenn der Cache fertig initialisiert worden ist, können Sie den Proxy testweise starten:

```
root@arch-deb:/# /usr/local/squid/sbin/squid
```

Es sind keine weiteren Optionen für den Start erforderlich. Squid läuft standardmäßig als Daemon. Warten Sie ein paar Sekunden und prüfen Sie dann, ob der Server nun läuft (`ps aux|grep squid` oder `pidof squid` sind hier hilfreich). Sollte es wider Erwarten zu Problemen kommen, konsultieren Sie die soeben erstellte Protokolldatei *cache.log*. Sollte diese Datei leer sein, gibt es ein Berechtigungsproblem mit dieser Datei.

Im Lieferumfang von Squid ist ein Testprogramm enthalten, mit dem Sie die Funktionsfähigkeit des Proxys testen können. Mit dem folgenden Kommando greifen Sie über den lokal installierten Proxy (localhost TCP-Port 3128) auf eine Webseite zu. Die Ausgabe des Kommandos sollte der HTML-Code der Webseite sein:

```
root@arch-deb:/# /usr/local/squid/bin/squidclient http://www.lpi.org
HTTP/1.1 200 OK
Date: Sun, 26 Jun 2011 14:08:06 GMT
```

```
Server: Apache/1.3.33 (Debian GNU/Linux) PHP/4.3.10-22 mod_ssl/2.8.22 OpenSSL/
0.9.7e mod_perl/1.29
X-Powered-By: eZ publish
Set-Cookie: eZSESSID=aa69db352a0ec2e3fb9ab1bbe347135b; path=/
Expires: Mon, 26 Jul 1997 05:00:00 GMT
Cache-Control: no-cache, must-revalidate
```

Konfiguration

Damit Sie die Ergebnisse Ihrer Konfigurationsarbeiten überprüfen können, sollten Sie zunächst einen Webbrowser Ihrer Wahl so konfigurieren, dass er durch den Squid-Proxy hindurch auf das Internet zugreift. Wenn Sie Firefox oder Iceweasel unter Linux verwenden, finden Sie die entsprechende Registerkarte unter BEARBEITEN • EINSTELLUNGEN • ERWEITERT • NETZWERK • EINSTELLUNGEN. Tragen Sie hier die Adresse des Proxy-Servers und den TCP-Port 3128 ein.

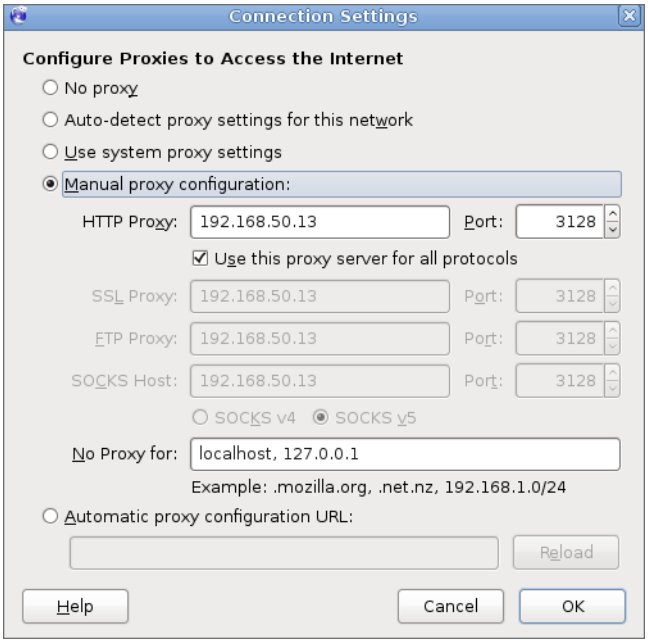

Abbildung 208.1 Tragen Sie die Adresse des Squid-Proxys und die zugehörige Portnummer hier ein.

Zur Fehlervermeidung sollten Sie überprüfen, ob der Squid Proxy Server selbst über eine funktionierende Internetverbindung verfügt. Wenn dennoch Fehler auftreten, lesen Sie bitte unbedingt die entsprechende Fehlermeldung im Browserfenster.

Die Hauptkonfigurationsdatei des Squid-Proxys ist die Datei *squid.conf*. Sie finden diese Konfigurationsdatei normalerweise unter */etc/squid.conf* oder */etc/squid3/*

squid.conf. Wenn Sie Squid aus einem tar-Ball heraus installiert haben, lautet der Dateipfad */usr/local/squid/etc/squid.conf*. Sie sollten bezüglich der Grundkonfiguration zumindest die folgenden Optionen in dieser Datei kennen:

- `http_port` legt fest, an welchem TCP-Port Squid lauschen soll. Der Standardwert ist 3128.
- `cache_mem` gibt an, wie viel Arbeitsspeicher für die Zwischenspeicherung verwendet werden darf (z. B. `cache_mem 512MB`). In der Standardeinstellung werden lediglich 8 MB verwendet.
- `cache_dir` konfiguriert den festplattenseitigen Cache des Proxys. Hierbei werden (in dieser Reihenfolge) der Typ des Caches, der Verzeichnisname, die Cachegröße in Megabytes, die Anzahl der Unterverzeichnisse erster Ebene und die Anzahl der Unterverzeichnisse zweiter Ebene festgelegt. Beispiel:

  ```
  cache_dir ufs /usr/local/squid/var/cache 1024 16 256
  ```

  In diesem Beispiel wird ein Cache vom Typ UFS verwendet. Das Hauptverzeichnis des Caches ist */usr/local/squid/var/cache* und wurde auf 1 GB beschränkt. In diesem Verzeichnis werden 16 Unterverzeichnisse und darin wiederum jeweils 256 Unterverzeichnisse erstellt.
- `reply_body_max_size` limitiert die Größe eines reply-body und hindert Benutzer daran, übergroße Dateien aus dem Internet herunterzuladen.
- `access_log` legt hauptsächlich den Pfad zu der Protokolldatei fest, in der Client-Zugriffe protokolliert werden.

Cache auf Festplatte einrichten

In der Standardkonfiguration verwendet Squid keinen Festplattencache, wenn Sie den Proxy aus einem tar-Ball heraus installiert haben. Damit der Cache initialisiert werden kann, benötigt Squid Schreibrechte auf das Verzeichnis, in dem die Verzeichnishierarchie für den Cache erstellt werden soll. Da in der Testumgebung Sicherheit keine Rolle spielt, können Sie einfach dieses Kommando verwenden:

```
root@arch-deb:/# chown nobody /usr/local/squid/var/cache/ -R
```

In der Datei *squid.conf* sollten Sie die Parameter für `cache_dir` Ihren persönlichen Bedürfnissen anpassen. Besonders der erste numerische Wert, der die Cachegröße in Megabytes festlegt, dürfte hier von Interesse sein. Die anderen Parameter von `cache_dir` wurden bereits im vorangehenden Abschnitt erläutert.

```
cache_dir ufs /usr/local/squid/var/cache 20000 16 256
```

Beenden Sie nun Squid, falls es noch laufen sollte, und führen Sie anschließend das folgende Kommando aus, um den Cache neu zu initialisieren:

```
root@arch-deb:/usr/local/squid/var# /usr/local/squid/sbin/squid -z
root@arch-deb:/usr/local/squid/var# 2011/06/26 15:24:44 kid1|
2011/06/26 15:24:44 kid1| Creating Swap Directories
2011/06/26 15:24:44 kid1| Making directories in /usr/local/squid/var/cache/00
2011/06/26 15:24:44 kid1| Making directories in /usr/local/squid/var/cache/01
2011/06/26 15:24:44 kid1| Making directories in /usr/local/squid/var/cache/02
...
2011/06/26 15:24:44 kid1| Making directories in /usr/local/squid/var/cache/0F
```

Die Initialisierung ist abgeschlossen und Sie können Squid nun wieder normal starten und verwenden.

Zugriffssteuerung mithilfe von ACLs

Die Zugriffssteuerung auf den Squid-Proxy wird über ACLs vorgenommen. In diesen ACLs können Sie zunächst einige Definitionen vornehmen. Im weiteren Verlauf der Konfiguration können Sie dann auf diese Definitionen (ACLs) zurückgreifen, um den Zugriff jeweils zu erlauben oder zu verweigern. Typischerweise enthalten ACLs Gruppen von Quell-IP-Adressen, Ziel-IP-Adressen, URL-Listen oder Ports. Namen von Zugriffssteuerungslisten können mehrfach verwendet werden. Wenn Sie Squid 3.x selbst kompiliert haben, gibt es z. B. drei ACL-Einträge in der Datei *squid.conf*, die für die Definition von privaten IPv4-Netzwerkadressen verwendet werden:

```
acl localnet src 10.0.0.0/8
acl localnet src 172.16.0.0/12
acl localnet src 192.168.0.0/16
```

Bis hierhin handelt es sich lediglich um eine Deklaration, in der alle privaten IPv4-Adressen als mögliche Quelladressen (`src`) der ACL `localnet` hinzugefügt werden. In einem ähnlichen Eintrag wird der lokale Computer selbst definiert:

```
acl localhost src 127.0.0.1/32 ::1
```

Ein paar Zeilen später wird dann für den lokalen Computer und die privaten IPv4-Netzwerke der Zugriff auf den Proxy erlaubt:

```
http_access allow localnet
http_access allow localhost
```

Hinweis

Wenn Sie Squid paketbasiert installieren, wird in den meisten Fällen per Voreinstellung überhaupt kein Zugriff auf den Proxy möglich sein. Sie müssen in diesem Fall die entsprechenden ACLs von Hand erstellen und den Zugriff erlauben.

Sie können den Zugriff auf Webseiten über reguläre Ausdrücke filtern. Dazu benötigen Sie zunächst einen geeigneten ACL-Eintrag in der Datei *squid.conf*:

```
acl sperrliste url_regex "/usr/local/squid/etc/sperrliste"
```

Den zugehörigen Berechtigungseintrag sollten Sie vor anderen Berechtigungseinträgen positionieren, weil vorangehende `allow`-Einträge sonst frühzeitig Zugriff gewähren, sodass der Filter gar nicht erst greift.

```
http_access deny sperrliste
http_access allow localnet
http_access allow localhost
```

Sie müssen jetzt nur noch die Datei */usr/local/squid/etc/sperrliste* anlegen und Schlagwörter (jeweils in einer eigenen Zeile) eingeben, die in den URLs nicht vorkommen dürfen. Nach einem Neustart von Squid können Sie die Konfiguration mit einem Browser testen.

Zum Abschluss der Zugriffssteuerung sollte immer der Zugriff für alle anderen Clients verweigert werden. Ansonsten könnten böswillige Benutzer vom Internet aus Ihren Proxy verwenden, um ihre Herkunft zu verschleiern und unter Ihrer Identität z. B. Webserver attackieren. Die letzte Regel sollte also immer diese sein:

```
http_access deny all
```

Praxistipp

Wenn Sie den Zugriff auf einen Proxy häufig umkonfigurieren müssen oder wenn eine recht komplexe Zugriffskonfiguration existiert, sollten Sie ein Frontend verwenden, um nicht den Überblick zu verlieren. Eine gute Wahl ist hierfür SquidGuard. Dieses Produkt ist aber nicht prüfungsrelevant.

Benutzerauthentifizierung

Sie können den Zugriff auf einen Proxy einschränken, indem Sie eine Authentifizierung konfigurieren. Die Benutzer werden dann beim Zugriff auf Webseiten aufgefordert, einen Benutzernamen und ein Passwort einzugeben. Squid verfügt über mehrere Authentifizierungsmodule, sodass Sie die Anmeldung z. B. über PAM, LDAP, Windows-Domänen oder, im einfachsten Fall, über eine eigene Passwortdatei abwickeln können. Die Verwendung einer Passwortdatei ähnelt der Konfiguration der Basisauthentifizierung von Apache. Führen Sie zunächst die folgenden Schritte durch, um die Authentifizierungskomponenten nachzuinstallieren:

```
root@arch-deb:/# cd /usr/src/squid/squid-3.2.0.9/helpers/basic_auth/NCSA
root@arch-deb:/usr/src/squid/squid-3.2.0.9/helpers/basic_auth/NCSA/# make
root@arch-deb:/usr/src/squid/squid-3.2.0.9/helpers/basic_auth/NCSA/#
 make install
```

Normalerweise kommt es bei der Installation zu keinerlei Komplikationen. Die installierten Authentifizierungsmodule finden Sie jetzt im Verzeichnis */usr/local/squid/libexec*.

Genau wie bei der Apache-Basisauthentifizierung erstellen Sie als Nächstes eine Passwortdatei mit Benutzernamen und Kennwörtern:

```
root@arch-deb:/# cd /usr/local/squid/etc
root@arch-deb:/usr/local/squid/etc# htpasswd -c passwd willi
New password:
Re-type new password:
```

Bei der Erstellung weiterer Benutzer lassen Sie die Option `-c` (create) einfach weg. Es würde sonst noch eine neue Datei erstellt und die bestehenden Benutzerkonten gingen verloren. Beispiel:

```
root@arch-deb:/usr/local/squid/etc# htpasswd passwd susi
New password:
Re-type new password:
```

Die Passwörter werden in der Passwortdatei verschlüsselt abgespeichert. Sie sollten auf einem Produktionssystem trotzdem sicherstellen, dass die Benutzer nicht das Recht haben, diese Datei zu lesen.

```
root@arch-deb:/usr/local/squid/etc# cat passwd
willi:9EvUN84SDrJy6
susi:5FMd5ieiINMf2
```

Hinweis

Sollten Sie Apache in der Zwischenzeit deinstalliert haben, verfügen Sie möglicherweise nicht mehr über das Programm `htpasswd`. Sie können das Programm aber bei Bedarf einzeln unter folgender Adresse aus dem Internet herunterladen:

http://www.squid-cache.org

Damit die Basisauthentifizierung funktioniert, sind einige Anpassungen an der Datei *squid.conf* erforderlich. Fügen Sie zunächst *in einer einzigen Zeile* den folgenden Eintrag hinzu, um das Authentifizierungsmodul zu laden:

```
auth_param basic program /usr/local/squid/libexec/basic_ncsa_auth /usr/local/
squid/etc/passwd
```

Hierbei zeigt */usr/local/squid/libexec/basic_ncsa_auth* den Pfad zum Authentifizierungsmodul und */usr/local/squid/etc/passwd* die zu verwendende Passwortdatei an. Es folgen ein paar Grundeinstellungen:

```
auth_param basic children 5
auth_param basic realm Authentifizierung erforderlich!
auth_param basic credentialsttl 8 hours
```

Der erste Parameter sorgt dafür, dass fünf Child-Prozesse gestartet werden, wie Sie wahrscheinlich schon selbst vermutet haben. Mit dem zweiten Eintrag können Sie eine Nachricht an den Benutzer übermitteln, wenn er sich anmeldet. Sie sehen das Ergebnis in Abbildung 208.2 weiter hinten. Der letzte Eintrag sorgt dafür, dass ein Benutzer sich innerhalb von acht Stunden nur einmal authentifizieren muss. Nach einem Browserneustart ist allerdings in jedem Fall eine neue Anmeldung fällig.

Als Nächstes müssen Sie der Datei *squid.conf* einen ACL-Eintrag hinzufügen, der die Verwendung einer Authentifizierung voraussetzt:

```
acl users proxy_auth REQUIRED
```

Bei der Positionierung des Statements, das den soeben erstellten ACL-Eintrag verwendet, ist Vorsicht geboten, was die Reihenfolge der Berechtigungsvergaben anbelangt. Wenn der Eintrag zu weit unten in der Konfiguration steht, könnte ein Benutzer schon Zugriff erlangen, bevor die Authentifizierung überhaupt greift. Beispiel:

```
http_access allow users
http_access allow localnet
http_access allow localhost
```

Falsch wäre etwa:

```
http_access allow localnet
http_access allow localhost
http_access allow users
```

Bei der falschen Konfiguration bekäme ein Benutzer bereits Zugriff aufgrund seiner Position in einem lokalen Netz.

Sie können die Authentifizierung jetzt mit einem Webbrowser testen.

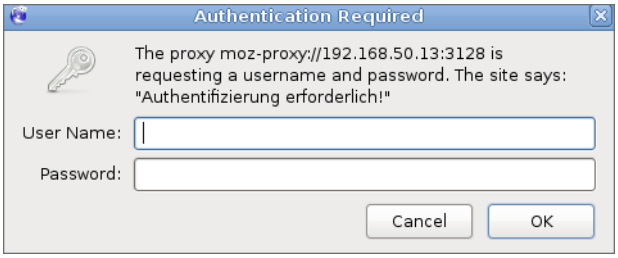

Abbildung 208.2 Authentifizierung bei einem Proxy-Server

Sollte der Authentifizierungsdialog nicht erscheinen und Sie können den Proxy ohne Authentifizierung verwenden, prüfen Sie noch einmal die Reihenfolge der Berechtigungseinträge in der Datei *squid.conf*. Schieben Sie den Eintrag `http_access allow users` im Zweifelsfall weiter nach oben.

208.4 Implementieren von nginx als Webserver und Reverse-Proxy

Wichtung: 2

Beschreibung: Die Kandidaten sollten dazu in der Lage sein, *nginx* als Reverse-Proxy zu konfigurieren. Auch Kenntnisse über die Grundkonfiguration als Webserver sind notwendig.

Wichtigste Wissensgebiete:

- *nginx*
- Reverse-Proxy
- Basiswebserver

Liste Wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/nginx*
- *nginx*

Allgemeines

nginx ist eine Kombination aus Webserver, Reverse-Proxy und E-Mail-Proxy für IMAP und POP3. Ausgesprochen wird es übrigens »Engine X«. Für die Prüfung ist die E-Mail-Proxy-Funktionalität nicht von Belang, aber Sie sollten zumindest eine Grundkonfiguration als Reverse-Proxy und als Webserver durchführen können. *nginx* ist plattformunabhängig, und die Windows-Version läuft sogar ohne eine Emulationsschicht direkt auf der Win32-API.

Reverse-Proxy

Da Sie vermutlich noch über einen komplett konfigurierten Apache Webserver verfügen, will ich hier mit der Konfiguration von *nginx* als Reverse-Proxy für einen Apache Server beginnen. In diesem Beispiel wird *nginx* auf derselben Maschine ausgeführt, wie der Apache Webserver selbst.

Zuerst müssen Sie *nginx* natürlich installieren. Unter Debian und seinen Derivaten können Sie das Paket ganz einfach mit `apt-get` installieren. Bei CentOS und Fedora müssen Sie zunächst die Paketquellen selbst nachinstallieren, weshalb hier der Einfachheit halber das Beispiel auf einem Ubuntu-Server präsentiert wird.

Am besten noch vor der Installation von *nginx* sollten Sie Apache auf einen anderen Port als den standardmäßigen Port umkonfigurieren. Je nachdem, wie Ihr Apache derzeit konfiguriert ist, müssen Sie in der Datei *httpd.conf* (bei vielen Distributionen in der Datei *ports.conf*) folgende Änderungen vornehmen:

```
NameVirtualHost *:8000
Listen 8000
```

Sie können natürlich einen anderen Port nehmen als den Port 8000; Hauptsache Sie verwenden keinen Port, der bereits von einem anderen Daemon verwendet wird. Wenn Sie eine Konfigurationsdatei für einen virtuellen Host verwenden, müssen Sie hier ebenfalls eine Änderung vornehmen:

```
<VirtualHost *:8000>
```

Bei einer Standardkonfiguration unter Debian und Ubuntu finden Sie die Konfigurationsdatei im Verzeichnis */etc/apache2/sites-available*. Starten Sie den Apache Webserver neu, damit er den geänderten Port verwendet:

```
root@archangel:/etc/apache2/sites-available# apachectl restart
```

Da Apache jetzt nicht mehr den TCP-Port 80 abhört, können Sie diesen für *nginx* verwenden. Sie können die Installation also jetzt gefahrlos durchführen:

```
root@archangel:/# apt-get install nginx
```

Nach der Installation ist *nginx* grob als Webserver (allerdings ohne existierendes Dokumentenverzeichnis) konfiguriert. Die Konfigurationsdateien befinden sich unterhalb von */etc/nginx*. Sie werden feststellen, dass es hier, genauso wie bei Apache, die Verzeichnisse *sites-available* und *sites-enable* gibt. Diese Verzeichnisse werden von *nginx* auch genauso verwendet wie von Apache. Um den standardmäßig konfigurierten Webserver zu deaktivieren, müssen Sie also lediglich den entsprechenden Link löschen:

```
root@archangel:/# rm /etc/nginx/sites-enabled/default
```

Die Hauptkonfigurationsdatei ist *nginx.conf*. Hier werden einige grundlegende Parameter definiert, wie die Anzahl der Arbeitsprozesse und die maximale Anzahl gleichzeitiger Zugriffe. Die anderen Dateien, die direkt in */etc/nginx* liegen, sind Beispieldateien, die Sie in Ihre eigene Konfiguration bei Bedarf einbinden können. Für eine einfache Reverse-Proxy-Konfiguration erstellen und bearbeiten Sie z. B. diese Datei:

```
root@archangel:/# vi /etc/nginx/sites-available/proxy
```

Erstellen Sie eine Serverdirektive mit mindestens diesem Inhalt:

```
server {
    listen 80 default;
    location / {
        proxy_pass http://127.0.0.1:8000;
    }
}
```

Der Reverse-Proxy lauscht an Port 80 und leitet Anfragen an Port 8000 des Loopback-Adapters weiter. Das ist der Port, auf den Sie vorhin Apache konfiguriert haben. An dieser Stelle bietet es sich optional an, den Inhalt der Beispieldatei */etc/nginx/proxy_params* einzubinden. Sie müssen jetzt nur noch einen Softlink anlegen, damit *nginx* diese Konfiguration auch lädt:

```
root@archangel:/#ln -s  /etc/nginx/sites-available/proxy \
/etc/nginx/sites-enabled/proxy
```

Jetzt ist der richtige Zeitpunkt für einen ersten Start:

```
root@archangel:/# /etc/init.d/nginx start
```

Wenn alles gut gegangen ist, können Sie jetzt mit einem Webbrowser ganz normal auf den TCP-Port 80 dieses Servers zugreifen und erhalten den Inhalt der Webseite, die von Apache gehostet wird.

Sie können ggf. die Syntax der Konfigurationsdateien von *nginx* überprüfen, indem Sie folgendes Kommando verwenden:

```
root@archangel:/# nginx -t
nginx: the configuration file /etc/nginx/nginx.conf syntax is ok
nginx: configuration file /etc/nginx/nginx.conf test is successful
```

Es scheint alles in Ordnung zu sein. Sie können *nginx* zur Laufzeit Signale übermitteln, indem Sie die Option `-s` mit einem der folgenden Parameter verwenden: `stop`, `quit`, `reopen`, `reload`. Wenn Sie eine Konfigurationsänderung durchgeführt haben, können Sie diese also mit folgendem Kommando sofort wirksam werden lassen:

```
root@archangel:/# nginx -s reload
```

nginx als Webserver

Wenn Sie einen Apache Webserver konfigurieren können, wird Ihnen die Einrichtung eines *nginx*-Servers auch keine Schwierigkeiten bereiten. Sie können einfach mehrere virtuelle Webserver bereitstellen, indem Sie im Verzeichnis */etc/nginx/sites-available* die jeweiligen Konfigurationsdateien erstellen und bei Bedarf in */etc/nginx/sites-enabled* verlinken. Eine solche Konfigurationsdatei könnte z. B. so aussehen:

```
server {
    listen          80;
    server_name     www.lpic-2.de;
    index           index.html;
    root            /var/www/lpic-2.de;
}
```

Was Sie hier sehen, ist ein Serverblock. Bei Apache würde man dieses Konstrukt als einen virtuellen Server bezeichnen. Die meisten Einstellungen sind selbsterklärend, aber zur Sicherheit sollen diese hier dennoch erläutert werden.

Die Direktive `listen` legt fest, an welchem Port der Server lauschen soll. Mit der Einstellung `server_name` wird der Hostheadername festgelegt, falls auf dem physischen Server mehrere Webseiten unter derselben IP-Adresse und demselben Port erreichbar sein müssen. Diese Webseite wird also nur dann ausgeliefert, wenn Sie in den Browser *www.lpic-2.de* eingeben. Vergessen Sie nicht, die entsprechende DNS-Konfiguration durchzuführen, falls Sie eine Konfiguration mit mehreren Webseiten ausprobieren wollen. Der Eintrag `index` sagt dem Webserver, welche Datei er herausgeben soll, wenn keine Datei ausdrücklich auf der URL angegeben wurde. Die Direktive `root` entspricht dem DocumentRoot bei Apache. Es handelt sich also um das Hauptverzeichnis des virtuellen Servers.

Bei Servern, die mehrere Webseiten hosten, bietet sich als sauberer Abschluss eine Konfiguration an, welche jene Client-Anfragen bedient, bei denen der Hostheaderwert mit keiner gültigen Konfiguration übereinstimmt, oder falls der Benutzer eine IP-Adresse in das URL-Feld des Browsers eingegeben hat:

```
server {
    listen          80 default_server;
     index           index.html;
    root            /var/www/default;
}
```

Dieser Eintrag bedient alle verbleibenden Anfragen an TCP-Port 80 des Servers über all seine erreichbaren IP-Adressen.

209 Freigabe von Dateien

Samba ermöglicht den problemlosen Austausch von Dateien zwischen Windows-Clients und Linux- bzw. Unix-Servern. Das primäre Ziel des Samba-Projekts ist der Abbau von Barrieren und die Gewährleistung der Interoperabilität zwischen diesen beiden Rechnerwelten.

209.1 Konfiguration eines Samba-Servers

Wichtung: 5

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen Samba-Server für verschiedene Clients zu konfigurieren. Dieses Lernziel beinhaltet die Konfiguration eines Samba-Servers als allein stehender Server oder als Mitglied einer Active-Directory-Domäne. Auch die Einrichtung von CIFS- und Druckerfreigaben ist in diesem Lernziel enthalten. Das Konfigurieren eines Linux-Clients zur Benutzung eines Samba-Servers ist ebenfalls Prüfungsinhalt. Weiterhin wird die Fehlerbehebung in Installationen geprüft.

Wichtigste Wissensgebiete:

- Samba-4-Dokumentation
- Samba-4-Konfigurationsdateien
- Samba-4-Werkzeuge und -Daemons
- CIFS-Freigaben unter Linux einbinden
- Windows-Benutzernamen auf Linux-Benutzernamen abbilden
- User-Level- und Share-Level- und AD-Sicherheit

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- *smbd, nmbd, winbindd*
- `smbcontrol`, `smbstatus`, `testparm`, `smbpasswd`, `nmblookup`
- `samba-tool`
- `net`
- `smbclient`
- `mount.cifs`
- */etc/samba/*
- */var/log/samba/*

Allgemeines

Die Entwicklungsgeschichte des Samba-Servers begann im Jahr 1992. Der Australier Andrew Tridgell benötigte von einem DOS-Computer aus Zugriff auf die Festplatten eines Unix-Systems. Das war zunächst kein Problem, weil es bereits einen NFS-Client für DOS gab, über den ein Dateiaustausch mit Unix ohne Probleme möglich war. Zu seinem Pech (und unserem Glück!) besaß er aber auch ein Programm, das NetBIOS-Zugriff benötigte und deshalb nicht mit NFS zusammenlief.

Mit einem Paket-Sniffer analysierte Tridgell den Netzwerkverkehr des von Microsoft-Servern verwendeten *SMB-Protokolls* (SMB = Server Message Block), inklusive NetBIOS, und programmierte das Protokoll nach. Er installierte seine neue Software auf einem Unix-System und war jetzt auch in der Lage, seine NetBIOS-Anwendung über das Netzwerk zu verwenden. Etwas später versuchte er, mit dem Windows-PC seiner Frau auf den Server zuzugreifen und auch das funktionierte. Etwa zwei Jahre später suchte Tridgell in einer Dictionary-Datei nach Worten, welche die Zeichen »smb« enthalten, weil er einen Namen für sein Programm benötigte. Er wählte Samba aus.

smbd, nmbd und winbindd

Der Samba-Server besteht aus drei Hauptkomponenten, nämlich den Daemons *smbd*, *nmbd* und *winbindd*. Diese Dienste stellen die folgenden Funktionen bereit:

- Datei- und Druckdienste
- Authentifizierung und Autorisierung
- Namensauflösung
- Dienstankündigung
- Schnittstellen zu Windows Domänen

Der Daemon *smbd* ist für die eigentliche Hauptaufgabe des Samba-Servers zuständig, nämlich die Bereitstellung von Datei- und Druckdiensten. Deshalb führt dieser Daemon bei allein stehenden Servern auch gleichzeitig die Authentifizierung und Autorisierung der Benutzer durch.

Die Namensauflösung über NetBIOS (das ist der hauseigene Namensauflösungsmechanismus von Microsoft) und die Dienstankündigungen führt der Daemon *nmbd* durch. Er ermöglicht es den Windows-Benutzern, über die Netzwerkumgebung Ressourcen leicht aufzufinden. Sie können *nmbd* auch als vollwertigen Ersatz für einen *WINS-Server* in einem Windows-Umfeld verwenden.

Bei der Integration von Samba-Servern in eine Windows Domäne hilft *winbindd*. Dieser Daemon kann nicht nur Benutzer- und Gruppeninformationen einer Active-Directory-Domäne verwalten, sondern bietet auch passende Authentifizierungsmodule für PAM.

Samba-Konfigurationsdateien

Die Konfigurationsdateien eines Samba-Servers befinden sich normalerweise im Verzeichnis */etc/smb/* oder */etc/samba*. Sie können fast alle Einstellungen in der Datei *smb.conf* vornehmen. Zuordnungen von Windows-Benutzerkonten zu Linux-Benutzerkonten konfigurieren Sie in der Datei *smbusers*.

/etc/samba/smb.conf

In der Datei *smb.conf* gibt es eine globale Sektion, deren Einstellungen sich auf alle erstellten Shares auswirken. Weiterhin werden hier Optionen eingestellt, die sich nicht auf Shares beziehen. Die Verwendung eines WINS-Servers oder gar die Bereitstellung von WINS-Diensten auf einem Linux-Computer fällt unter diese Kategorie. Die folgende Beispielsektion enthält einige wichtige Optionen:

```
[global]
        workgroup = rheinwerkcomputing
```

Das ist der Name der Arbeitsgruppe oder der NetBIOS-Domäne, für die der Samba-Server Dateien und Druckdienste bereitstellt.

```
        wins-support = no
```

Dieser Server stellt selbst keine WINS-Server-Dienste bereit.

```
        wins-server = 192.168.57.1
```

Der Samba-Server verwendet aber (als WINS-Client) den WINS-Server mit der IP-Adresse 192.168.57.1.

```
        log file = /var/log/samba-log.%m
```

Hier legt der Samba-Server seine Protokolldateien ab.

```
        lock directory = /var/lock/samba
```

In diesem Verzeichnis erstellt Samba seine Lockfiles, um ggf. feststellen zu können, ob schon andere Instanzen von Samba laufen.

```
        security = user
```

Dieser Eintrag ist die Standardeinstellung für allein stehende Server oder wenn Samba selbst als Domaincontroller fungiert.

```
guest account = nobody
```

Nicht authentifizierten Benutzern können Sie mit dieser Option die UID des Kontos `nobody` zuweisen.

```
hide files = /*.tmp/*.bak/*.old/
```

Diese Zeile sorgt dafür, dass Dateien mit den Erweiterungen *tmp*, *bak* und *old* nicht angezeigt werden. Wenn ein Benutzer eine Datei direkt auswählt, die eine der genannten Dateierweiterungen aufweist, wird die Datei jedoch geöffnet. Wenn eine Datei nicht angezeigt werden soll und der direkte Zugriff verhindert werden muss, können Sie stattdessen die folgende Zeile verwenden:

```
        veto files = /*.tmp/*.bak/*.old/
```

Alle weiteren Sektionen innerhalb der Datei *smb.conf* beziehen sich auf einzelne Shares oder bereitgestellte Drucker.

```
[printers]
        comment = All Printers
        path = /var/tmp
        create mask = 0600
        printable = Yes
        browseable = Yes
```

Das ist ein weitverbreiteter Standardeintrag, der alle an den Rechner angeschlossenen Drucker gegenüber Windows-Computern darstellt. Diese Freigabe sorgt nur dafür, dass in einer Windows-Netzwerkumgebung der Knoten Drucker und Fax-Geräte zu sehen ist. Die einzelnen Drucker benötigen trotzdem eine eigene Sektion in der Datei *smb.conf*.

```
[lp]
        path = /var/tmp
        read only = No
        create mask = 0600
        guest ok = Yes
        printable = Yes
        printer name = lp
        browseable = Yes
```

Hier handelt es sich um einen funktionierenden Eintrag für einen gewöhnlichen Drucker. Da Samba nicht selbst zwischen Druckern und Dateisystemen unterscheidet, sind einige Einträge erforderlich, die man vielleicht merkwürdig finden könnte. Schließlich ist es klar, dass ein Drucker `printable` sein muss und ein `read only`-Drucker wäre auch nicht sonderlich hilfreich. Mit `guest ok = yes` wird bewirkt, dass auch nicht authentifizierte Benutzer drucken dürfen, auch wenn in der globalen Sektion `security = user` gesetzt ist.

```
[public]
        path = /public
        read only = Yes
        browseable = Yes
        guest ok = Yes
```

In dieser Sektion wird das Verzeichnis */public* auf dem Samba-Server freigegeben. Da es sich um ein Verzeichnis handelt, das öffentlich zugänglich sein soll, wird `read only = yes` und `guest ok = yes` gesetzt. Mit dem Eintrag `browseable = yes` erreicht man, dass diese Freigabe in der Netzwerkumgebung eines Windows-Computers angezeigt und somit auch Anfängern zugänglich gemacht wird. Wenn man `browseable = no` setzt, wird diese Share in der Netzwerkumgebung eines Windows-Computers nicht angezeigt.

/etc/samba/smbusers

Die Datei *smbusers* wird verwendet, um eine Zuordnung von Windows-Benutzerkonten zu den entsprechenden Linux-Benutzerkonten herzustellen. Zu diesem Zweck werden in der Datei *smbusers* die jeweiligen Konten durch ein Gleichzeichen miteinander verknüpft. Links stehen immer die Linux-Konten und rechts die entsprechenden Windows-Benutzerkonten. Damit diese Sicherheitsüberprüfung stattfinden kann, muss in der Datei *smb.conf* die Option `security = user` gesetzt sein. Eine typische *smbusers*-Datei könnte also so aussehen:

```
# See section 'username map' in the manual page of smb.conf
# for more information.
root = administrator
harald = harald
martha = martha
dominik = dominik
```

Mitgliedschaft in einer Active-Directory-Domäne

Wenn ein Samba-Server Mitglied einer *Active-Directory-Domäne* werden soll, sind einige Punkte zu beachten. Zunächst ist es unbedingt erforderlich, dass die AD-Domäne, der ein Server beitreten soll, über DNS auflösbar ist. Das gilt übrigens auch für Windows-Computer, die einer AD-Domäne beitreten sollen. Damit dies gewährleistet ist, wird üblicherweise mindestens ein für die AD-Domäne autoritativer DNS-Server als Namensserver in die Datei *resolv.conf* des Samba-Servers eingetragen. Außerdem sollte für die Domäne ein Sucheintrag existieren, damit auch Computernamen, bei denen der Domänen-Suffix nicht mit angegeben wurde, auffindbar sind.

Achtung

Ab hier wird davon ausgegangen, dass der Samba-Server den Namen **sf7** trägt. Die Active-Directory-Domäne heißt **nwa-net.de** und der verwendete Domänencontroller heißt **dc1.nwa-net.de**. Dieser Domänencontroller ist unter der IPv4-Adresse **10.16.0.10** erreichbar. Die IPv6-Adresse des Domain-Controllers ist **2001:6f8:1d2d::10**.

Die Datei */etc/resolv.conf* könnte für dieses Beispiel folgenden Inhalt haben:

```
domain nwa-net.de
search nwa-net.de
nameserver 16.16.0.10
nameserver 2001:6f8:1d2d::10
```

Sie sollten vor dem versuchten Domänenbeitritt mittels `dig`, `host` oder `nslookup` die Namensauflösung testen:

```
[root@sf7 ~]# host -t A dc1.nwa-net.de
dc1.nwa-net.de has address 10.16.0.10
```

Da eine Authentifizierung mit Kerberos keine größeren Zeitabweichungen zwischen dem Domain-Controller und dem AD-Client erlaubt (in der Regel weniger als 5 Minuten), sollte eine Zeitsynchronisation mittels NTP durchgeführt werden.

Um die lokale Namensauflösung während des Beitritts zur Domäne sicher zu stellen, sollten Sie einen entsprechenden Eintrag in der Datei */etc/hosts* vornehmen:

```
10.16.0.99      sf7.nwa-net.de  sf7
```

Nachdem alle Vorbereitungen getroffen sind, können Sie nun die globale Sektion der Datei */etc/samba/smb.conf* bearbeiten. Das sind die für dieses Beispiel relevanten Einträge:

```
[global]
workgroup = NWA-NET
netbios name = SF7
security = ADS
password server = dc1.nwa-net.de
realm = NWA-NET.DE
```

Wenn Sie die Konfiguration durchgeführt haben, sollten Sie die Dienste des Samba-Servers neu starten. Danach steht dem Domänenbeitritt nichts mehr im Wege:

```
[root@sf7 ~]# net ads join -U administrator
Enter administrator's password: ********
```

```
Using short domain name -  NWA-NET
Joined 'sf7' to dns domain 'nwa-net.de'
```

Damit der Samba-Server Zugriff auf die Benutzer- und Gruppenkonten der Active-Directory-Domäne erhält, muss *winbind* konfiguriert werden. Das ist mit ein paar kleinen Einträgen in der Datei */etc/nsswitch.conf* erledigt:

```
passwd:     files winbind
shadow:     files winbind
group:      files winbind
```

Der Daemon *winbindd* ist übrigens nicht standardmäßig installiert. Sie sollten die Installation spätestens jetzt nachholen. Nachdem *winbindd* (neu) gestartet wurde, können Sie die Konfiguration testen. Ein gutes Werkzeug ist hierfür `wbinfo`. Sie können damit z. B. eine Liste der Benutzer vom Verzeichnisdienst beziehen:

```
[root@sf7 ~]# wbinfo -u
NWA-NET\administrator
NWA-NET\gast
NWA-NET\antoinette
NWA-NET\carolin
NWA-NET\claudia
NWA-NET\ines
NWA-NET\lisa
NWA-NET\markus
NWA-NET\matthias
NWA-NET\michael
NWA-NET\torsten
```

Praxistipp

Weitere Konfigurationsschritte im AD-Umfeld liegen außerhalb der Prüfungsthemen, aber wenn Sie sich mehr mit Samba beschäftigen wollen, ist hier eine gute Anlaufstelle:

https://wiki.samba.org

Werkzeuge und Dienstprogramme für Samba

Es gibt einige Programme, die Sie zur Verwaltung und Einrichtung des Samba-Servers einsetzen können. Sie sollten sich für die Prüfung mit den Tools, die auf den folgenden Seiten erläutert werden, vertraut machen.

smbcontrol

Mit `smbcontrol` können Sie den Daemons *smbd*, *nmbd* und *winbindd* jeweils Kommandos senden. Sie können damit z. B. einzelne Daemons veranlassen, die Konfiguration neu einzulesen:

```
[root@sf7 ~]# smbcontrol smbd reload-config
```

Wenn alle drei Daemons die Konfiguration neu einlesen sollen, verwenden Sie dieses Kommando:

```
[root@sf7 ~]# smbcontrol all reload-config
```

Weitere Kommandos finden Sie in der entsprechenden Manpage.

samba-tool

Das `samba-tool` dient der Verwaltung der Domänenfunktionen von Samba. Es hat einen enormen Funktionsumfang, wie Sie der Manpage entnehmen können. Da sich die Prüfung auf alleinstehende und Mitgliedserver beschränkt, ist nicht mit Detailfragen zu diesem Werkzeug zu rechnen.

smbstatus

Um einen Überblick über den aktuellen Zustand eines Samba-Servers zu erhalten, können Sie das Programm `smbstatus` verwenden:

```
archangel:~ # smbstatus
Samba version 3.0.13
Service      pid     machine       Connected at
-------------------------------------------------------
IPC$         14617   connor        Sat Oct 29 16:55:43 2011
daten        14617   connor        Sat Oct 29 22:51:49 2011
Locked files:
Pid    DenyMode   Access      R/W        Oplock           Name
--------------------------------------------------------------
14617  DENY_NONE  0x20089     RDONLY     EXCLUSIVE+BATCH  /storage/musik/
Hendrix - Hey Joe.mp3  Sat Oct 29 23:54:30 2011
```

Man kann der Ausgabe des Kommandos entnehmen, dass gerade ein Benutzer mit dem Namen `connor` von einer Arbeitsstation aus auf die Share mit der Bezeichnung `daten` zugreift. Offensichtlich hört er gerade Musik (die letzte Zeile verrät ihn), was aber Samstagnachts auch nicht den Produktionsbetrieb stören sollte.

smbpasswd

Mit diesem Kommando wird das Kennwort eines Samba-Users geändert. Die Änderung selbst findet dann in */etc/samba/smbpasswd* statt. Der Aufbau der Datei *smbpasswd* ähnelt dem Aufbau der Datei *passwd* für normale Linux-Benutzer. Genauere Kenntnisse über diese Datei sind in der Prüfung nicht erforderlich. Sie sollten allerdings wissen, dass sowohl die Datei als auch das Kommando `smbpasswd` existieren.

testparm

Mit dem Programm `testparm` können Sie die Syntax der Datei *smb.conf* überprüfen lassen:

```
root@ubuntu-server:~# testparm
Load smb config files from /etc/samba/smb.conf
Processing section "[printers]"
Processing section "[print$]"
Processing section "[lp]"
Processing section "[public]"
Loaded services file OK.
Server role: ROLE_STANDALONE
```

In diesem Fall scheint kein Fehler vorzuliegen. Sie sollten die Ausgabe des Kommandos immer sorgfältig lesen, weil eventuelle Fehlermeldungen leicht übersehen werden können.

nmblookup

Bei `nmblookup` handelt es sich im Prinzip um einen Netzwerk-Client, der NetBIOS über TCP/IP verwendet, um NetBIOS-Namen bzw. deren zugehörige IP-Adressen zu ermitteln. Bei der einfachsten Verwendung wird lediglich der NetBIOS-Name des zu ermittelnden Computers (hier `CH01`) angegeben:

```
root@ch-1-125:~# nmblookup CH01
querying CH01 on 192.168.1.255
192.168.1.198 CH01<00>
192.168.1.82 CH01<00>
```

Der zweiten Zeile kann man entnehmen, dass bei der Namensauflösung die Broadcast-Adresse des Netzwerks verwendet wird. Die Zeilen drei und vier enthalten das Ergebnis der Abfrage. Der Computer `CH01` verfügt offensichtlich über zwei IP-Adressen im gleichen Netzwerksegment.

Wenn Sie dem Kommando die Option `-S` anhängen, wird im Anschluss an die Auflösung des Namens in eine IP-Adresse eine Statusabfrage des Zielcomputers durchge-

führt. Das Ergebnis der Abfrage enthält z. B. Informationen über die Arbeitsgruppenzugehörigkeit eines Systems und die MAC-Adresse der entsprechenden Netzwerkkarte. Beispiel:

```
root@ch-1-125:~# nmblookup CH-1-124 -S
querying CH-1-124 on 192.168.1.255
192.168.1.124 CH-1-124<00>
Looking up status of 192.168.1.124
.......CH-1-124        <00> -         M <ACTIVE>
.......INDISOFT        <00> - <GROUP> M <ACTIVE>
.......CH-1-124        <20> -         M <ACTIVE>
.......INDISOFT        <1e> - <GROUP> M <ACTIVE>

.......MAC Address = 00-C0-26-A1-98-1D
```

Alternativ können Sie die Option `-A` verwenden, wenn Sie nur die IP-Adresse des abzufragenden Systems kennen, nicht aber den NetBIOS-Namen. Es wird dann direkt die Statusabfrage an die Ziel-IP gesendet:

```
root@ch-1-125:~# nmblookup 192.168.1.124 -A
```

Wenn Sie bei einer Abfrage auf einen WINS-Server zurückgreifen wollen, benötigen Sie die Optionen `-R` (recursive) und `-U` (unicast). Der Option `-U` übergeben Sie dann die IP-Adresse des WINS-Servers als Parameter. Beispiel:

```
root@ch-1-125:~# nmblookup NAS -R -U 192.168.1.198
querying NAS on 192.168.1.198
192.168.1.252 NAS<00>
```

net

Das Tool `net` ist ein weiteres Werkzeug der Samba-Suite. Es dient der Administration von Samba und entfernten CIFS-Servern (CIFS = Common Internet File System). Sie können innerhalb eines Befehls sowohl Optionen als auch Kommandos verwenden. Um etwa die Uhrzeit von einem bestimmten Server zu erfahren, verwenden Sie die Option `-S` und das Kommando `TIME`, um die Zeit zu erfragen:

```
root@ch-1-125:~# net -S CH01 TIME
Tue Sep 20 14:10:32 2011
```

Die verwendbaren Optionen sind nicht allzu zahlreich und die wichtigste Option (`-S`) haben Sie bereits kennengelernt. Weitere wichtige Optionen sind:

- `-I` gibt die IP-Adresse des Zielservers an.
- `-S` gibt den NetBIOS-Namen des Zielservers an.

- `-U` gibt den zu verwendenden Benutzernamen an.
- `-h` zeigt eine umfangreiche Hilfe zum Programm an.

Das Programm `net` versteht zwar nur wenige Optionen, dafür aber umso mehr Kommandos. Sie bekommen schnell einen Überblick, indem Sie das Kommando `net` einfach einmal ohne Optionen eingeben. Hier ein Auszug:

```
net rpc             Run functions using RPC transport
net rap             Run functions using RAP transport
net ads             Run functions using ADS transport
net file            Functions on remote opened files
net share           Functions on shares
net session         Manage sessions
net server          List servers in workgroup
net domain          List domains/workgroups on network
net printq          Modify printer queue
net user            Manage users
net group           Manage groups
net groupmap        Manage group mappings
net sam             Functions on the SAM database
net validate        Validate username and password
net groupmember     Modify group memberships
net admin           Execute remote command on a remote OS/2 server
net service         List/modify running services
net password        Change user password on target server
net changetrustpw   Change the trust password
net changesecretpw  Change the secret password
net setauthuser     Set the winbind auth user
net getauthuser     Get the winbind auth user settings
net time            Show/set time
net lookup          Look up host names/IP addresses
net g_lock          Manipulate the global lock table
net join            Join a domain/AD
net dom             Join/unjoin (remote) machines to/from a domain/AD
net cache           Operate on the cache tdb file
net getlocalsid     Get the SID for the local domain
net setlocalsid     Set the SID for the local domain
net setdomainsid    Set domain SID on member servers
net getdomainsid    Get domain SID on member servers
net maxrid          Display the maximul RID currently used
net idmap           IDmap functions
net status          Display server status
net usershare       Manage user-modifiable shares
net usersidlist     Display list of all users with SID
```

```
net conf            Manage Samba registry based configuration
net registry        Manage the Samba registry
net eventlog        Process Win32 *.evt eventlog files
net help            Print usage information
```

Sie müssen für die Prüfung natürlich nicht alle Kommandos kennen, die man mit `net` ausführen kann. Viele Kommandos in der obigen Auflistung beziehen sich z. B. auf die Mitgliedschaft in einer Domäne, während es in der Prüfung lediglich um die Fileserver-Eigenschaften eines Samba-Servers geht. Den Status der Freigaben eines Fileservers können Sie z. B. so abfragen:

```
root@archangel:/home/harald# net status shares
Service      pid     machine        Connected at
-------------------------------------------------------
daten        16282   192.168.50.10  Tue Sep 20 14:09:33 2011
daten        16284   192.168.50.17  Tue Sep 20 14:14:15 2011
daten        16288   192.168.50.14  Tue Sep 20 14:44:53 2011
```

Sie können mittels `net` NetBIOS-Namen in IP-Adressen auflösen:

```
root@ch-1-125:~# net lookup ch01
192.168.1.198
```

Hinweis

Sie sollten die Kommandozeilentools zur Verwaltung von Samba für die Prüfung kennen. Die Erwähnung von SWAT folgt hier lediglich der Vollständigkeit halber.

SWAT

SWAT ist ein Akronym für *Samba Web Administration Tool*. Mit SWAT können Sie eigentlich alle oben beschriebenen administrativen Eingriffe an einem Samba-Server mithilfe eines Webbrowsers durchführen. Bei den meisten Distributionen gehört SWAT zur Standardausrüstung und muss in der Regel nur noch über `inetd` bzw. `xinetd` aktiviert werden. Danach wartet SWAT am TCP-Port 901 auf Verbindungen. Geben Sie also in die URL-Zeile Ihres Webbrowsers einfach den Pfad *http://meincomputername:901* ein und SWAT steht Ihnen zur Verfügung. Die Angabe des Präfixes *http://* ist in diesem Fall notwendig, weil der Browser anhand der Portnummer 901 nicht ermitteln kann, welches Protokoll verwendet wird.

SWAT ist, wie gesagt, nicht prüfungsrelevant und soll deshalb hier nicht näher beschrieben werden. Es gibt übrigens noch weitere Frontends zur Konfiguration von Samba, die Sie mithilfe einer Suchmaschine leicht finden können.

Samba 4-Dokumentation

Wenn Sie nach umfassenden Informationen zum Samba-Server suchen, werden Sie in jedem Fall auf der Samba-Website fündig:

http://www.samba.org

Sie finden auf dieser Seite nicht nur Downloadlinks zu den neuen Versionen von Samba und dessen Konfigurationstools, sondern auch HOWTOs in verschiedenen Sprachen.

In der Sektion 5 der Manpages können Sie sich ausführlich über einzelne Parameter der Konfigurationsdatei *smb.conf* belesen. Die wichtigsten Einstellungen sind in der Datei *smb.conf* allerdings auch direkt kommentiert, sodass Sie direkt am lebenden Objekt lernen können.

Einen Überblick über die komplette Samba-Suite finden Sie in der Sektion 7 der Manpages. Führen Sie einfach das Kommando `man samba` aus, wenn Sie eine Übersicht der mit Samba in Zusammenhang stehenden Programme benötigen.

Bei der Installation von Samba werden unterhalb von */usr/share/doc* mehrere Unterverzeichnisse angelegt, die weitere Informationen über Samba enthalten. Bei aktuellen Versionen von Samba ist als wesentliche Anlaufstelle */usr/share/doc/samba-doc* zu nennen. Hier gibt es ein Unterverzeichnis namens *htmldocs*, das eine browsertaugliche Version der Dokumentation enthält. Außerdem finden Sie an dieser Stelle das Unterverzeichnis *examples*, das Beispiele zu allen erdenklichen Szenarien enthält, die im Zusammenhang mit einem Samba-Server möglich sind.

Samba-Freigaben unter Linux einbinden

Wenn Sie von einem Linux-Client aus auf eine Samba-Share oder eine Windows-Freigabe zugreifen wollen, muss zuvor das Paket *smbclient* installiert worden sein. Anschließend können Sie die Shares ein- und aushängen wie lokal angeschlossene Datenträger. Um eine Verbindung zu einer Freigabe herzustellen, sind die folgenden beiden Kommandos absolut äquivalent:

```
root@ubuntu-server:~# mount -t smbfs //archangel/daten /mnt
root@ubuntu-server:~# smbmount //archangel/daten /mnt
```

Sollte bei dem Zielsystem eine Authentifizierung erforderlich sein, müssen Sie das Kommando erweitern, indem Sie die Optionen `username` und `password` übergeben. Die Syntax des Befehls würde dann so aussehen:

```
root@ubuntu-server:~# mount -t smbfs -o username=harald,password=
meinpasswort //archangel/daten /mnt
```

Beachten Sie, dass zwischen den Optionen keine Leerzeichen verwendet werden. Der Übersichtlichkeit halber sehen Sie hier die reine Authentifizierung noch einmal einzeln:

```
-o username=harald,password=meinpasswort
```

Wenn Sie eine Samba-Share bzw. Windows-Freigabe schon während des Systemstarts einbinden wollen, können Sie einen entsprechenden Eintrag in der Datei */etc/fstab* vornehmen. Bezogen auf das vorangehende Beispiel könnte dieser Eintrag wie folgt aussehen:

```
//archangel/daten    /mnt    cifs       rw,auto,username=harald,password=
                                        meinpasswort  0  0
```

Wenn kein Zugriff auf die Shares mehr benötigt wird, können Sie die Dateisysteme wahlweise mit einem der folgenden Kommandos wieder aushängen:

```
root@ubuntu-server:~# umount /mnt
root@ubuntu-server:~# smbumount /mnt
```

Es ist natürlich grundsätzlich eine gute Idee, die Mount-Vorgänge durch einen Automounter erledigen zu lassen. Die Verwendung der Datei */etc/fstab* setzt nämlich voraus, dass das Netzwerk bereits läuft. Wenn Sie in einem Notebook einen WLAN-Adapter verwenden, wird das Netzwerk in der Regel erst nach der Benutzeranmeldung gestartet. Das ist für den Einsatz der Datei */etc/fstab* einfach viel zu spät.

209.2 Konfiguration eines NFS-Servers

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Dateisysteme über NFS zu exportieren. Dieses Lernziel beinhaltet Zugriffsbeschränkungen, das Einbinden eines NFS-Dateisystems auf einem Client und das Absichern von NFS.

Wichtigste Wissensgebiete:

- Konfigurationsdateien von NFS Version 3
- Werkzeuge und Dienstprogramme von NFS
- Zugriffsbeschränkungen für bestimmte Hosts und/oder Subnetze
- Mount-Optionen für Server und Client
- TCP-Wrapper
- Kenntnis von NFS Version 4

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/exports*
- `exportfs`
- `showmount`
- `nfsstat`
- */proc/mounts*
- */etc/fstab*
- `rpcinfo`
- `mountd`
- `portmapper`

Allgemeines

NFS (Network File System) ist ein Protokoll der Anwendungsschicht, das von Sun Microsystems entwickelt worden ist, um über das Netzwerk auf Dateien zuzugreifen. Ältere Versionen von NFS basierten auf dem Transportprotokoll UDP. Die von der IETF (Internet Engineering Task Force) weiterentwickelten Versionen verwenden heutzutage TCP. Weitere Verbesserungen in NTFS Version 4 sind höhere Stabilität, die Authentifizierung von Benutzern und die Möglichkeit, Daten verschlüsselt über das Netzwerk zu transportieren. Bis zur Version 3 authentifizierte NFS immer nur auf Basis des Computers.

Prüfungstipp

Die LPI-Prüfungen befassen sich thematisch üblicherweise mit der Authentifizierung auf Computerebene.

Serverseitige Konfiguration

Damit Sie NFS verwenden können, benötigen Sie einen NFS-Server. Dieser ist normalerweise in allen Linux-Distributionen enthalten. Voraussetzung für das Einrichten einer NFS-Sitzung ist allerdings der *RPC-Portmapper*. Starten Sie also diese beiden Dienste:

```
archangel:~ # /etc/init.d/portmap start
archangel:~ # /etc/init.d/nfsserver start
```

Sollten die Serverkomponenten auf Ihrem System nicht vorhanden sein, können Sie diese wie gewohnt nachinstallieren. Verwenden Sie dann wahlweise (in Abhängigkeit von der installierten Distribution) eines der folgenden beiden Kommandos:

```
archangel:~ # apt-get install portmap nfs-server
archangel:~ # yum install portmap nfs-server
```

Prüfungstipp

Da der RPC-Portmapper oftmals vergessen wird, ist er auch ein beliebtes Prüfungsthema.

Das Programm `rpc.mountd` implementiert das NFS-Mount-Protokoll. Wenn `mountd` eine Mount-Anforderung von einem Client empfängt, prüft das Programm, ob das durch den Client angeforderte Dateisystem freigegeben ist und ob der Client dazu berechtigt ist, auf dieses Dateisystem zuzugreifen.

Bei der Bereitstellung von Dateien via NFS spricht man vom Exportieren der Dateisysteme (Verzeichnisse). Diese Exporte werden in der Konfigurationsdatei */etc/exports* vorgenommen. In der folgenden Beispieldatei werden die wichtigsten Optionen dargestellt:

```
/               admincomp(rw,no_root_squash)
/public         *(ro,all_squash)
/geheim         agent*.mydomain.com(rw)
```

In diesem Beispiel sind die wichtigsten Elemente für eine */etc/exports*-Datei enthalten. In der ersten Zeile wird ausschließlich für den Computer `admincomp` der Lese- und Schreibzugriff (`rw`) auf das Hauptverzeichnis erteilt. Standardmäßig wird der Benutzer `root` von NFS als User `nobody` behandelt, weil es sich bei `root` um ein allgemein bekanntes und obendrein gefährliches Konto handelt. Um dieses Verhalten außer Kraft zu setzen, wird die Option `no_root_squash` verwendet. Auf das Verzeichnis */public* soll von jedem beliebigen (*) Computer aus lesend (`ro`) zugegriffen werden können. Die Option `all_squash` sorgt dafür, dass jeder Benutzer aus Sicherheitsgründen wie `nobody` behandelt wird. Auf das Verzeichnis */geheim* soll exklusiv von Computern aus zugegriffen werden, deren Namen mit `agent` beginnen und die Mitglieder der Domäne `mydomain.com` sind. Es soll Lese- und Schreibzugriff (`rw`) erteilt werden.

Damit der Export bei laufendem NFS-Server stattfinden kann, benötigen Sie das Kommando `exportfs`:

```
archangel:~ # exportfs -a
/               admincomp
/public         <world>
/geheim         agent*.mydomain.com
```

Danach sollten die Verzeichnisse von den angegebenen Systemen aus zugreifbar sein. Sie können das Kommando `exportfs` auch verwenden, um Dateisysteme direkt

mit einem Kommando zu exportieren. Genau genommen ist das sogar das Standardverhalten von `exportfs`. Mit der Option `-a` aufgerufen, liest `exportfs` die Datei */etc/exports* ein, exportiert die darin gefundenen Dateisysteme und zeigt das Ergebnis anschließend an. Beim manuellen Export von Dateisystemen gehen Sie so vor:

```
[root@fedora16 /]# exportfs -o rw ubuntu-desktop:/home/dominik
[root@fedora16 /]# exportfs
/home/dominik   ubuntu-desktop.homelinux.net
```

In der ersten Zeile wird das Verzeichnis */home/dominik* mit Lese- und Schreibzugriff (`rw`) so exportiert, dass nur von der Arbeitsstation `ubuntu-desktop` aus zugegriffen werden kann. Mit dem Kommando `exportfs` (ohne Optionen) wird anschließend das Ergebnis überprüft.

Es sollte immer berücksichtigt werden, dass durch das Netzwerk ein Zugriff auf das lokale Dateisystem stattfindet. Hier wird natürlich (genauso wie bei einem lokalen Dateizugriff) eine entsprechende Sicherheitsfilterung durchgeführt. Bei unterschiedlichen Systemen wird es oft vorkommen, dass die Benutzer nicht auf allen Systemen die gleiche UID haben. Die Zugriffskontrolle basiert aber auf diesen UIDs, wie die folgenden beiden Ausgaben des `ls`-Befehls zeigen:

```
archangel:/ # ls -l /public
total 8
drwxrwxrwx   2 root   root  4096 Oct 19 18:42 .
drwxr-xr-x  25 root   root  4096 Oct 19 18:32 ..
-rw-r--r--   1 harald users    0 Oct 19 18:42 testfile
```

Die normale Ausgabe von `ls` zeigt den Besitzer und die Besitzergruppe für *testfile* an. Wenn man die numerische Ausgabe von `ls` verwendet, sieht man, welche UID sich hinter diesen Besitzern verbirgt:

```
archangel:/ # ls -ln /public
total 8
drwxrwxrwx   2    0   0 4096 Oct 19 18:42 .
drwxr-xr-x  25    0   0 4096 Oct 19 18:32 ..
-rw-r--r--   1 1000 1000   0 Oct 19 18:42 testfile
```

Die UID des Besitzers ist 1000 und die GID der Gruppe ist ebenfalls 1000. Wenn nun über das Netzwerk ein anderer User, der ebenfalls die UID 1000 hat, auf diese Datei zugreift, bekommt er dieselben Zugriffsrechte wie der User `harald` auf dem hiesigen System. Das ist auch der Grund, wieso bei kritischen Daten mit `all_squash` die Netzwerkbenutzer zu `nobody` degradiert werden. Diese User haben dann dieselben Rechte wie `others`.

NFS-Client-Konfiguration

Um eine Verbindung mit einem NFS-Server herzustellen, braucht man einen Mountpoint, gerade so, als würde man ein blockorientiertes Gerät wie eine Festplattenpartition oder eine CD-ROM mounten. Im folgenden Beispiel wird eine Verbindung zu einem in der vorangehenden Lektion exportierten Dateisystem hergestellt:

```
[root@fedora16 /]# mkdir /arch-pub
[root@fedora16 /]# mount -t nfs archangel:/public /arch-pub
[root@fedora16 /]# cd /arch-pub/
[root@fedora16 arch-pub]# ls -l
insgesamt 0
-rw-r--r-- 1 1000 users 0 19. Okt 18:42 testfile
```

Zuerst wurde mit `mkdir` ein Verzeichnis erstellt, das als Mountpoint dienen soll. Der `mount`-Befehl im zweiten Schritt verbindet das exportierte Verzeichnis */public* des Hosts `archangel` mit dem lokalen Verzeichnis */arch-pub*. Die Option `-t` (type) gibt dem `mount`-Befehl die Information, dass es sich um ein NFS-Dateisystem handelt. Anschließend wird das entfernte Dateisystem mit der Syntax `<hostname>:/<Verzeichnis>` übergeben. Die letzte notwendige Angabe ist der Mountpoint. Ein Blick in das frisch verbundene Verzeichnis zeigt, dass es hier zu dem berüchtigten Problem gekommen ist, dass es auf dem System `fedora16` keinen Benutzer mit der UID 1000 gibt. Der Besitzer wird deshalb einfach numerisch angezeigt.

Sie sollten unbedingt darauf achten, dass Sie die Syntax des Befehls `exportfs` nicht mit der Syntax des `mount`-Befehls verwechseln. Die Trennung von Host und Verzeichnis durch einen Doppelpunkt hat jeweils eine völlig andere Bedeutung. Vergleichen Sie die beiden Abschnitte noch einmal miteinander, wenn Sie sich nicht sicher sind, wo genau der Unterschied liegt.

Wenn ein Dateisystem wieder ausgehängt werden soll, verwenden Sie, wie von normalen Dateisystemen gewohnt:

```
[root@fedora16 /]# umount /arch-pub
```

Wenn ein exportiertes Dateisystem eines anderen Computers immer beim Systemstart automatisch verbunden werden soll, können Sie es einfach in die Datei */etc/fstab* eintragen. Sollte dann allerdings ein NFS-Server während des Bootvorgangs nicht erreichbar sein, wird sich der Startprozess dieses Computers erheblich verzögern. Ein entsprechender Eintrag in einer *fstab*-Datei könnte so aussehen:

```
archangel:/public       /arc-pub    nfs    defaults    0  0
```

Die Optionen der Datei */etc/fstab* wurden bereits im ersten Teil des Buchs ausführlich behandelt. Die Syntax ist auf NFS eins zu eins übertragbar und soll hier deshalb

nicht mehr weiter ausgeführt werden. Im Übrigen kann man sagen, dass es auch bei Zugriffen auf NFS-Server sinnvoll ist, einen Automounter einzusetzen. Das gilt insbesondere für Notebook-User, bei denen das WLAN normalerweise erst nach der Anmeldung am System gestartet wird.

Tools für NFS

Es gibt einige nützliche Diagnoseprogramme, die Sie im Zusammenhang mit NFS-Problemen einsetzen können. Die auf den folgenden Seiten beschriebenen Tools werden als Themen für die Prüfung ausdrücklich genannt.

showmount

Mit `showmount` können Sie sich einen Überblick darüber verschaffen, von welchen Computern aus NFS-Client-Verbindungen zu einem NFS-Server bestehen. Zu diesem Zweck fragt das Programm `showmount` den Daemon `mountd` entweder lokal oder, wenn Sie wollen, auch auf einem entfernten System ab.

Wenn Sie `showmount` einfach ohne Optionen auf einem NFS-Server eingeben, erhalten Sie eine Liste der per NFS verbundenen Client-Computer bzw. deren IP-Adressen:

```
root@archangel:/# showmount
Hosts on archangel:
192.168.50.134
192.168.50.147
192.168.50.166
```

Wenn Sie die Abfrage von einem entfernten Computer aus durchführen wollen, hängen Sie an das Kommando einfach den Namen des Zielcomputers an. Im folgenden Beispiel wird auf einem entfernten Computer abgefragt, welche Verzeichnisse per NFS von den Benutzern verwendet werden.

```
root@arch-deb-book:/# showmount archangel -d
Directories on archangel:
/home/dominik
/home/harald
/home/martha
/storage
```

Sie können auch feststellen, welche Verzeichnisse ein NFS-Server grundsätzlich exportiert, indem Sie den Schalter `-e` verwenden:

```
root@arch-deb-book:/# showmount archangel -e
Export list for archangel:
```

```
/home/dominik
/home/harald
/home/martha
/storage     *
/            *
```

Um eine Zuordnung von Client-Computern zu den gemounteten Verzeichnissen zu bekommen, verwenden Sie den Schalter `-a`. Die Ausgabe des Kommandos wurde hier gekürzt:

```
root@arch-deb-book:/# showmount archangel -a
All mount points on archangel:
192.168.50.134:/home/harald
192.168.50.134:/storage
```

Es ist gut möglich, dass in der Prüfung die ausgeschriebenen Versionen für die Optionen verwendet werden. Das sind:

- `--all` für `-a`
- `--directories` für `-d`
- `--exports` für `-e`
- `--help` für `-h`
- `--version` für `-v`

rpcinfo

Wie Sie bereits wissen, basiert der Verbindungsaufbau zu den NFS-Servern auf RPC (Remote Procedure Call). Deshalb musste auch zusammen mit dem NFS-Server gleich der RPC-Portmapper installiert werden. Sie können das Programm `rpcinfo` verwenden, um (auch ggf. wieder auf entfernten Hosts) festzustellen, welche Dienste ein Host per RPC zur Verfügung stellt. Eine einfache Abfrage könnte so aussehen:

```
root@arch-deb-book:/# rpcinfo -p archangel
   Program Vers Proto   Port
    100000    2   tcp    111  portmapper
    100000    2   udp    111  portmapper
    100024    1   udp  48583  status
    100024    1   tcp  35940  status
    100021    1   udp  59602  nlockmgr
    100021    3   udp  59602  nlockmgr
    100021    3   tcp  40333  nlockmgr
    100021    4   tcp  40333  nlockmgr
    100003    2   udp   2049  nfs
```

```
100003    3   udp   2049  nfs
100003    4   udp   2049  nfs
100003    2   tcp   2049  nfs
100003    3   tcp   2049  nfs
100003    4   tcp   2049  nfs
100005    1   udp  34055  mountd
100005    1   tcp  35193  mountd
100005    3   udp  34055  mountd
100005    3   tcp  35193  mountd
```

Aus dem Ergebnis der Abfrage kann man erkennen, dass es sich bei dem Zielsystem tatsächlich um einen NFS-Server handelt. Der Server unterstützt sowohl TCP- als auch UDP-Verbindungen.

Das Programm verwendet lediglich Kurzformen für Optionen. Das sind:

- `-p` sondiert (probe) den hier angegebenen Host.
- `-u` führt einen RPC-Aufruf per UDP zur Prozedur 0 des angegebenen Programms auf dem angegebenen Host aus.
- `-t` führt einen RPC-Aufruf per TCP zur Prozedur 0 des angegebenen Programms auf dem angegebenen Host aus.
- `-n` gibt den zu verwendenden Port in Kombination mit den Optionen `-u` oder `-t` an.
- `-b` führt einen RPC-Broadcast aus.
- `-d` löscht (delete) die Registrierung eines RPC-Dienstes.

nfsstat

Mit dem Programm `nfsstat` können Sie statistische Informationen sowohl über NFS-Server- als auch NFS-Client-Aktivitäten anzeigen. Da die Ausgabe des einfachen Kommandos recht umfangreich ausfällt, gibt es mehrere Optionen zur Filterung der Anzeige. Diese Optionen sind:

- `-s` bzw. `--server` zeigt nur serverseitige Informationen.
- `-c` bzw. `--client` zeigt nur clientseitige Statistiken.
- `-n` bzw. `--nfs` zeigt nur NFS-Statistiken.
- `-r` bzw. `--rpc` zeigt nur RPC-Informationen.
- `-2`, `-3` oder `-4` zeigt nur Statistiken zur angegebenen Version von NFS.
- `-m` bzw. `--mounts` zeigt Statistiken zu den jeweils gemounteten Dateisystemen.
- `-o` zeigt nur Informationen zur jeweils angegebenen Funktion (z. B. `rpc`, `nfs`, `net`).

Wenn Sie etwa nur die RPC-Aufrufe an einen Server analysieren wollen, können Sie dieses Kommando verwenden:

```
root@archangel:~# nfsstat --server --rpc
Server rpc stats:
calls      badcalls   badauth    badclnt    xdrcall
97843      0          0          0          0
```

Wie Sie sehen, hat es auf diesem Server in der letzten Zeit keine Probleme mit Remote-Prozeduraufrufen gegeben.

Um nur die NFS-Server-Informationen der Version 3 zu erhalten, kann man diese Statistik verwenden:

```
root@archangel:~# nfsstat --server -o nfs -3
Server nfs v3:
null       getattr    setattr    lookup     access     readlink
156     1% 4493   40% 320     2% 837     7% 2099   18% 69      0%
read       write      create     mkdir      symlink    mknod
807     7% 591     5% 101     0% 4       0% 1       0% 0     0%
remove     rmdir      rename     link       readdir    readdirplus
75      0% 0       0% 14      0% 0     0% 0       0% 799       7%
fsstat        fsinfo        pathconf      commit
239        2% 199        1% 98         0% 146        1%
```

Die Ausgabe des Kommandos ist etwas unübersichtlich, aber für eine erste Orientierung helfen vielleicht ein paar konkrete Informationen. Es gehören hier immer zwei Zeilen zusammen. Eine Zeile enthält die Bezeichnungen der jeweiligen Parameter und direkt darunter befindet sich der jeweils dazugehörende Wert. Unter dem Parameter `create` finden Sie den Wert 101. Es sind also offensichtlich 101 Dateien erstellt worden. Gleich daneben steht der Parameter `mkdir`. Die darunter stehende 4 besagt, dass vier Verzeichnisse über NFS erstellt wurden. Es sind 14 Dateien umbenannt worden, wie Sie am Parameter `rename` ablesen können.

Zugriffsbeschränkungen

Wie Sie wissen, können Sie in der Datei */etc/exports* festlegen, von welchen Hosts aus auf ein bestimmtes exportiertes Dateisystem zugegriffen werden darf. Wenn Sie den Zugriff auf den ganzen NFS-Server steuern wollen, können Sie TCP-Wrapper einsetzen. Bearbeiten Sie zu diesem Zweck die Datei */etc/hosts.allow*. Wenn auf den NFS-Server z. B. nur vom Netzwerk 192.168.50.0/24 aus zugegriffen werden soll, fügen Sie dieser Datei die folgende Zeile hinzu:

```
portmap : 192.168.50.0/24
```

Beachten Sie, dass zur Zugriffssteuerung an dieser Stelle nur IP-Adressen oder das Schlüsselwort `ALL` verwendet werden dürfen. Host- oder Domänennamen sind beim Schutz des Portmappers nicht erlaubt. Die Verwendung von TCP-Wrappern wird in einem späteren Kapitel noch einmal genauer durchleuchtet. Es soll an dieser Stelle bei diesem einfachen Beispiel bleiben.

Auch der Einsatz von `iptables` ist beim Schutz von NFS-Servern sinnvoll, aber auch dies ist Thema eines späteren Kapitels.

210 Verwaltung von Netzwerk-Clients

In mittleren und großen Netzwerken sollte die Administration einiger Funktionen zentralisiert werden. Dazu zählt die zentrale Vergabe von IP-Adressen mittels DHCP genauso wie eine zentrale Authentifizierung.

210.1 DHCP-Konfiguration

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen DHCP-Server zu konfigurieren. Dieses Lernziel umfasst das Setzen von Standard- und clientspezifischen Optionen sowie das Hinzufügen von statischen und BOOTP-Hosts. Ebenfalls dazu gehören die Handhabung eines DHCP-Relay-Agenten und die Pflege des DHCP-Servers.

Wichtigste Wissensgebiete:

- DHCP-Konfigurationsdateien, -Begriffe und -Dienstprogramme
- Einrichtung von Optionen in Subnetz- und dynamisch zugewiesenen Bereichen
- Kenntnis von DHCPv6 und IPv6-Routerankündigungen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- *dhcpd.conf*
- *dhcpd.leases*
- DHCP-Protokollierung in *syslog* oder *systemd*-Journal
- *arp*
- *dhcpd*
- *radvd*
- *radvd.conf*

Allgemeines

Zur automatischen Konfiguration von Netzwerkschnittstellen verwendet man das *Dynamic Host Configuration Protocol* (DHCP). Mit diesem Protokoll ist es nicht nur

möglich, IP-Adressen zu verteilen, sondern es kann auch die Konfiguration vieler anderer Netzwerkparameter wie Netzwerkmaske, Standard-Gateway, DNS-Server, NIS-Server usw. mit DHCP vorgenommen werden. Die Übermittlung einer DHCP-Konfiguration ist in vier Phasen unterteilt. Diese werden in der folgenden Reihenfolge abgearbeitet.

- **DHCPDISCOVER** – Der Client macht eine Rundsendung über die Broadcast-Adresse 255.255.255.255, in der er nach einem DHCP-Server sucht. In dieser Nachricht ist die MAC-Adresse des Clients enthalten, damit eine eindeutige Zuordnung der Konfiguration auch dann möglich ist, wenn mehrere Clients gleichzeitig einen DHCP-Server kontaktieren. Die Anfrage wird an Port 67 des DHCP-Servers gerichtet. Der Quellport ist hier ausnahmsweise nicht dynamisch, sondern auf Port 68 festgelegt.
- **DHCPOFFER** – Der DHCP-Server bietet dem Client eine Konfiguration an. Dieses Angebot wird an die Adresse 255.255.255.255 gesendet, weil der Client ja noch keine eigene IP-Adresse besitzt. Für die eindeutige Zuordnung ist die MAC-Adresse des Clients im Angebot enthalten.
- **DHCPREQUEST** – Der Client fordert die soeben angebotene Konfiguration explizit an. Dieses Verhalten ist sinnvoller, als es erscheint. Wenn ein Client eine Adresse erneut anfordert, werden nämlich die ersten beiden Phasen des Lease-Vorgangs weggelassen.
- **DHCPACK/DHCPNACK** – Mit DHCPACK bestätigt der DHCP-Server die Konfiguration, und der Client verwendet anschließend die angebotene Konfiguration. Ein DHCPNACK kommt selten vor. Es handelt sich um die Ablehnung einer Konfiguration. Das kann passieren, wenn ein Client, der ausgeschaltet war, eine Konfiguration anfordert, die inzwischen an einen anderen Computer vergeben wurde. Der Lease-Vorgang muss dann von Anfang an neu ausgeführt werden.

DHCP-Clients

Die Konfiguration der DHCP-Clients ist eher ein Thema der LPIC-1-Prüfungen. Trotzdem soll der Vollständigkeit halber an dieser Stelle kurz auf die verschiedenen DHCP-Client-Programme eingegangen werden.

- `dhcpcd` steht für DHCP-Client-Daemon. Er ist der empfohlene Client des Internet Software Consortium (ISC) und wohl der stabilste und selbstständigste DHCP-Client für Linux. Sie können das Softwarepaket von *http://www.isc.org* herunterladen, wenn es nicht in Ihrer Distribution enthalten sein sollte.
- `dhclient` versucht im Gegensatz zu `dhcpcd` nicht einfach, jeder Schnittstelle eine dynamische Konfiguration zuzuweisen, sondern liest zunächst in der Datei */etc/dhclient.conf*, welche Schnittstellen er konfigurieren soll.

- `pump` arbeitet ähnlich wie der `dhclient`. Auch `pump` wird über eine Konfigurationsdatei, nämlich */etc/pump.conf* gesteuert.

Die Clients unterscheiden sich vor allem auch darin, welche DHCP-Optionen sie verstehen. Die alltäglichen DHCP-Optionen wie DNS, Standard-Gateway und NIS werden von allen drei Programmen verstanden. Eine weitere Gemeinsamkeit ist, dass alle drei Programme als Daemon arbeiten und zur gesamten Laufzeit des Systems verfügbar bleiben.

DHCP-Server

Die Installation eines DHCP-Servers ist, unabhängig von der verwendeten Linux-Distribution, relativ einfach. Sie können wie gewohnt das jeweilige Paketmanagement verwenden. Führen Sie also unter Debian bzw. Ubuntu folgendes Kommando aus, um den DHCP-Server zu installieren:

```
root@archangel:/# apt-get install isc-dhcp3-server
```

Es wird hier lediglich der DHCP-Server installiert. Der entsprechende Client und auch der DHCP-Relay-Agent sind sinnvollerweise in eigenen Paketen enthalten. Es ist eigentlich nie sinnvoll, auf einem Computer sowohl den DHCP-Server, den DHCP-Client als auch den DHCP-Relay-Agent gleichzeitig auszuführen.

Verwenden Sie bei den Red Hat-Derivaten `yum`:

```
[root@arch-cent ~]# yum install dhcp
```

Das Paket enthält sowohl den DHCP-Server als auch den DHCP-Relay-Agent.

Auf der Webseite des Internet Systems Consortium wird bereits eine 4er-Version der DHCP-Suite angeboten. Die Neuerungen dieser Version sind noch nicht prüfungsrelevant, aber Sie können den Server natürlich trotzdem zur Prüfungsvorbereitung nutzen, wenn Sie das wollen. Das Internet Systems Consortium finden Sie hier:

http://www.isc.org.

Die Konfigurationsdatei dhcpd.conf

Die Konfigurationsdatei des DHCP-Servers heißt *dhcpd.conf*. Die Position dieser Datei variiert in Abhängigkeit von der verwendeten Linux-Distribution. Debian und Ubuntu erstellen für die Konfiguration des DHCP-Servers das Verzeichnis */etc/dhcp3*, während Red Hat traditionell vorgeht und die Datei *dhcpd.conf* direkt im Verzeichnis */etc* erstellt.

Sollten Sie den DHCP-Server direkt vom Internet Systems Consortium bezogen und selbst kompiliert und installiert haben, wird zunächst gar keine Konfigurationsdatei

angelegt. Sie müssen das selbst nachholen. Der DHCP-Server erwartet seine Konfigurationsdatei dann direkt in */etc*.

Die Konfiguration der DHCP-Bereiche (Scopes) und auch der globalen DHCP-Optionen ist verhältnismäßig einfach. Sie finden hier eine Beispielkonfiguration mit den gängigsten Optionen und Anmerkungen.

Wenn Sie den DHCP-Server verwenden wollen, um dynamische Updates in DNS stellvertretend für Clients durchzuführen, verwenden Sie die folgende Option:

```
ddns-updates on;
```

Es ist dann natürlich zusätzlich erforderlich, den DNS-Server oder zumindest die entsprechenden DNS-Zonen für dynamische Updates zu konfigurieren.

Um das primäre DNS-Suffix der Clients zu konfigurieren, benötigen Sie eine solche Option:

```
option domain-name "homelinux.net";
```

Legen Sie fest, welche(n) DNS-Server die Client-Computer verwenden sollen:

```
option domain-name-servers 192.168.50.1;
option dhcp6.name-servers 2001:6f8:1d2d::10;
```

Einige (insbesondere ältere) Windows-Clients benötigen für die NetBIOS-Namensauflösung einen WINS-Server. Der *nmbd* des Samba-Servers ist natürlich eine gute Alternative. Diesen legen Sie so fest:

```
option netbios-name-servers 192.168.50.1;
```

Wenn ein WINS-Server (oder *nmbd*) verwendet wird, ist es sinnvoll, den NetBIOS-Knotentyp für die Clients mit *Hybrid (8)* festzulegen:

```
option netbios-node-type 8;
```

Auch in Microsoft-Netzwerken verliert NetBIOS immer mehr an Bedeutung, und Sie brauchen sich, insbesondere für die LPIC-Prüfungen, nicht mit NetBIOS auseinanderzusetzen.

Die vier gerade genannten Optionen können auch innerhalb eines bestimmten DHCP-Bereichs verwendet werden. Da es sich hierbei aber um Optionen handelt, die nicht mit der Position des Clients in einem bestimmten Netzwerksegment zusammenhängen (wie etwa das Standard-Gateway), können diese hier global angegeben werden.

Wenn ein DHCP-Server eine DHCP-Konfiguration vergibt, ist diese nur für einen festgelegten Zeitraum (die Lease-Zeit) gültig.

```
default-lease-time 86400;
max-lease-time 864000;
```

Die `default-lease-time` ist die Lease-Zeit, die Clients zugewiesen wird, die nicht ausdrücklich nach einer bestimmten Lease-Dauer fragen, während der Wert für `max-lease-time` den höchstmöglichen Wert für die Lease-Dauer darstellt. Nach dem Ablauf der Lease-Zeit darf ein Client die IP-Konfiguration, die durch den DHCP-Server zugewiesen wurde, nicht mehr verwenden.

```
log-facility local7;
```

Mit `log-facility` wird festgelegt, welche Protokollierungseinrichtung der DHCP-Server verwenden soll. Es kann ggf. notwendig sein, einen entsprechenden Eintrag in der Datei */etc/syslog.conf* (bei neueren Systemen auch */etc/rsyslog.conf*) vorzunehmen.

Die Kernaufgabe eines DHCP-Servers ist die Vergabe von IP-Adressen. Sie müssen entsprechend jedem Netzwerksegment, das der Server bedienen soll, einen Bereich (Scope) anlegen. Ein solcher Bereich besteht zunächst aus der Definition des Netzwerks (`subnet`) selbst. Zusätzlich muss die Bereichsgröße für die Vergabe der Adressen (`range`) festgelegt werden. Die in diesem Beispiel verwendete Option (`option routers`) für den Standard-Gateway ist optional, wird aber in fast allen Fällen verwendet, wenn Clients auch mit Computern außerhalb ihres eigenen Netzwerksegments kommunizieren können sollen:

```
subnet 192.168.50.0 netmask 255.255.255.0 {
range 192.168.50.101 192.168.50.199;
option routers 192.168.50.3;
}
```

Grundsätzlich können alle Optionen, die weiter oben schon global festgelegt wurden (z. B. `option domain-name-servers`), auch innerhalb einer Bereichsdefinition verwendet werden. Das kann nötig werden, wenn etwa nur Computer eines bestimmten Bereichs in Ihrem Netzwerk aus besonderen Gründen einen anderen DNS-Server verwenden sollen. Ein Netzwerk für Gäste (z. B. Hotspot) soll eventuell Namen im Internet auflösen können, nicht aber die Computer Ihres Produktionsnetzwerks.

Reservierungen werden verwendet, wenn ein DHCP-Client immer dieselbe IP-Konfiguration erhalten soll. Das hat gegenüber der statischen Konfiguration des Clients den Vorteil, dass Sie von zentraler Stelle aus weiterhin IP-Optionen verteilen können. Wenn Sie z. B. die IP-Adresse eines DNS-Servers ändern müssen, können Sie das diesen quasistatisch konfigurierten Computern trotzdem zentral mitteilen. Außerdem können Sie bei Bedarf die IP-Adresse eines DHCP-Clients ändern, ohne diesen aufsuchen zu müssen. Es ist üblich, DHCP-aktivierten Netzwerkdruckern IP-Adressen zu reservieren. Das geschieht einfach über die Zuordnung der reservierten IP-Adresse zur MAC-Adresse des Clients:

```
host arch-deb-book
{
hardware ethernet 00:50:aa:c5:e1:8e;
fixed-address 192.168.50.7;
}
```

Beachten Sie, dass sich eine reservierte IP-Adresse nicht innerhalb eines Bereichs befinden sollte, den der DHCP-Server zur Verteilung verwendet.

Ein DHCP-Bereich für ein IPv6 Subnetz hat diesen Aufbau:

```
subnet6 2001:6f8:1d2d::/64 {
        range6 2001:6f8:1d2d::129 2001:6f8:1d2d::254;
        range6 2001:6f8:1d2d::/64 temporary;
}
```

Der Aufbau ähnelt also im Prinzip dem von IPv4-Bereichen.

Prüfungstipp

Alle Optionen, die mit dem Schlüsselwort `option` beginnen, werden den DHCP-Clients übergeben, während alle anderen Optionen das Verhalten des Servers beeinflussen.

Überwachung und Diagnose des DHCP-Servers

Zur Diagnose eines DHCP-Servers können Sie bei neueren Linux-Distributionen in der Regel `journalctl` verwenden. In älteren Distribution findet die Protokollierung entweder in der Datei */var/log/messages* oder in */var/log/syslog* statt. Sie finden hier Einträge zu jedem einzelnen Lease-Vorgang und ggf. Fehlermeldungen. Im Falle eines Fehlers wird meist zusätzlich ein Eintrag in der Protokolldatei */var/log/daemon.log* vorgenommen. Wie sich ein DHCP-Server in Bezug auf die Protokollierung verhält, kann in der Datei *dhcpd.conf* mit dem Statement `log-facility` festgelegt werden. Sie können hier auch eine eigene Datei für die Protokollierung angeben.

Wenn ein DHCP-Server eine IP-Konfiguration vergibt, dann wird dieser Lease-Vorgang in einer Datei mit dem Namen *dhcpd.leases* festgehalten. Auf diese Art kann der Server feststellen, welche IP-Adressen bereits vergeben wurden. Außerdem kann der Server so überprüfen, ob ein wiederkehrender Client tatsächlich schon einmal eine Lease von ihm erhalten hat und ob diese noch gültig ist. Die Position der Datei im Dateisystem ist von der verwendeten Linux-Distribution abhängig. Typische Pfade sind:

- */var/lib/dhcpd/dhcpd.leases* (Red Hat)
- */var/lib/dhcp3/dhcpd.leases* (Debian)
- */var/db/dhcpd.leases* (Internet Systems Consortium)

Ein Eintrag in dieser Datei enthält die IP-Adresse, die Start- und die Endzeit der Lease und die MAC-Adresse des Client-Computers. Wenn der DHCP-Server den Namen des Client-Computers ermitteln kann, dann wird auch dieser hier protokolliert:

```
lease 192.168.16.39 {
  starts 5 2013/04/08 07:12:28;
  ends 5 2013/04/08 07:14:28;
  tstp 5 2013/04/08 07:14:28;
  cltt 5 2013/04/08 07:12:28;
  binding state free;
  hardware ethernet 00:00:1c:da:46:17;
  uid "\001\000\000\034\332F\027";
  client-hostname "R2PC2";
}
```

DHCP-Relay-Agent

Der komplette Lease-Vorgang, also die Kommunikation zwischen DHCP-Server und DHCP-Client, findet über Broadcast-Nachrichten statt. Deshalb kann ein DHCP-Server normalerweise auch nur für die Verteilung von IP-Adressen innerhalb eines Netzwerksegments verwendet werden. Router stellen also eine Grenze für die DHCP-Kommunikation dar. Um diese Einschränkung aufzuheben, können Sie DHCP-Relay-Agents einsetzen. Das Funktionsprinzip ist denkbar einfach. Sie platzieren in Netzwerksegmenten, die über keinen eigenen DHCP-Server verfügen, jeweils einen Relay-Agent (diese Aufgabe kann z. B. ein Router übernehmen). Dieser nimmt dann stellvertretend die Anfragen von DHCP-Clients (`DHCPDISCOVER` und `DHCPREQUEST`) entgegen und leitet diese als Unicasts an einen DHCP-Server weiter. Der DHCP-Server wird dann dem Relay-Agent ebenfalls via Unicast antworten. Der Relay-Agent gibt die Antwort wiederum per Broadcast an den Client zurück. Einfach ausgedrückt übersetzt ein DHCP-Relay-Agent ankommende Broadcast-Sendungen in Unicast-Nachrichten, damit diese Pakete von Routern weitergeleitet werden können.

Die Konfiguration eines DHCP-Relay-Agents findet aufgrund der Einfachheit des Programms direkt auf der Kommandozeile statt. Eine Konfigurationsdatei ist nicht erforderlich:

```
[root@arch-gw ~]# /usr/sbin/dhcrelay -i eth0 192.168.50.1
```

Mit diesem Kommando wird der Agent gestartet. Er wartet dann auf DHCP-Anfragen, die seine Schnittstelle `eth0` erreichen und leitet diese Anfragen an den DHCP-Server mit der IP-Adresse 192.168.50.1 weiter. Beachten Sie bitte, dass das Programm `dhcrelay` heißt (ohne `p`!).

Achtung

Vergessen Sie nicht, dass der DHCP-Server über die entsprechenden Bereiche verfügen muss, um Clients in den jeweiligen Netzwerksegmenten zu versorgen, in denen Sie die Relay-Agents platziert haben.

Router Advertisement

Eine weitere Methode, Clientcomputer automatisch mit IPv6-Adressen zu konfigurieren, sind *Router Advertisements*. Hierbei kann man z. B. auf einem ohnehin vorhandenen Router, der vielleicht auch als Gateway zum Internet fungiert, einen Daemon installieren, der regelmäßig seinen eigenen IPv6-Präfix im Netzwerk ankündigt. Das benötigte Paket heißt bei allen mir bekannten Linux-Distributionen *radvd*. Die Konfiguration geschieht in der Datei */etc/radvd.conf*. Es ist zwar für die Prüfung kein Detailwissen zu dieser Datei erforderlich, aber da im Internet kaum funktionierende Beispiele auffindbar sind, die auch die Übergabe von DNS-Optionen beinhalten, sehen Sie hier eine Konfiguration, die eine real existierende AD-Domäne versorgt:

```
interface eth0 {
        AdvSendAdvert on;
        MinRtrAdvInterval 3;
        AdvOtherConfigFlag on;
        MaxRtrAdvInterval 10;
        prefix 2001:6f8:1d2d:0::/64 {
                AdvOnLink on;
                AdvAutonomous on;
                AdvRouterAddr on;
        };
RDNSS   2001:6f8:1d2d::10 { };
RDNSS   2001:6f8:1d2d::11 { };
DNSSL nwa-net.de { };
};
```

210.2 Konfiguration eines OpenLDAP-Servers

Hinweis

Gegenüber der Aufstellung des LPI sind dieses und die beiden darauf folgenden Themen (LDAP-Client-Konfiguration und PAM-Authentifizierung) getauscht. So sind die Inhalte für Sie besser nachvollziehbar. Die ungünstige Reihenfolge der Topics beim LPI ist lediglich historisch begründet und hat keinen technischen Hintergrund.

Wichtung: 4

Beschreibung: Die Kandidaten sollten dazu in der Lage sein, die Grundkonfiguration eines *OpenLDAP-Servers* durchzuführen und über das LDIF-Format sowie die grundlegende Zugriffssteuerung Bescheid wissen.

Wichtige Wissensgebiete:

- OpenLDAP
- Verzeichnisbasierte Konfiguration
- Zugriffssteuerung
- Distinguished Names
- Changetype-Operationen
- Schemata und WhitePages
- Verzeichnisse
- Object-IDs, Attribute und Klassen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- *slapd*
- *LDIF*
- `slapcat`
- */var/lib/ldap/*
- `slapd-config`
- `slapadd`
- `slapindex`
- Loglevel

Allgemeines

OpenLDAP ist die Open-Source-Variante des *Lightweigt Directory Access Protokolls* (LDAP). Hierbei handelt es sich eigentlich nur um ein Protokoll zur Abfrage eines Verzeichnisdienstes, der in einer Datenbank gehostet wird. Der Aufbau von LDAP ist im X.500-Standard definiert. Andere bekannte Implementierungen von LDAP sind:

- Active Directory (Microsoft)
- eDirectory (Novell)
- Open Directory (Apple)
- 389 Directory Server (Red Hat)

Die meisten dieser Implementierungen von LDAP dienen hauptsächlich dazu, Benutzer zu authentifizieren. Zu diesem Zweck sind in den entsprechenden Produkten gleichzeitig Authentifizierungsmechanismen implementiert, die auf einen Verzeichnisdienst zugreifen können. Der bekannteste Mechanismus in diesem Zusammenhang dürfte wohl Kerberos sein. Dennoch sollte man im Hinterkopf behalten, dass in einem Verzeichnisdienst nicht ausschließlich Benutzerkonten zu finden sind.

Wenn man über OpenLDAP spricht, ist im Normalfall nicht nur das Abfrageprotokoll gemeint, sondern auch die dazugehörige Datenbank. Der Aufbau der Objekte, die in der Datenbank gespeichert werden können, ist im sogenannten *Schema* definiert.

LDAP-Schema

Im *Schema* definieren die Objektklassen, welche Objekttypen erstellt werden können. Das wären zum Beispiel Organisationen, Domänen, Organisationseinheiten, Benutzer, Gruppen und natürlich Computer. Zu den Objektklassen gibt es jeweils Sätze von Attributen. So gehören zur Objektklasse `Person` z. B. die Attribute `givenname`, `userpassword` und `telephonenumber`. Man könnte also vereinfachend sagen, dass es sich beim Schema um eine komplexe Vorlage zur Erstellung von Verzeichnisdienstobjekten handelt. Da das Schema mit Modulen erweiterbar ist, können Sie weitere Objektklassen und Attribute hinzufügen. Das können auch speziell angepasste eigene Klassen und Attribute sein. Hier sehen Sie zwei prominente Attribute aus dem *core*-Schema:

```
attributetype ( 2.5.4.42 NAME ( 'givenName' 'gn' )
DESC 'RFC2256: first name(s) for which the entity is known by'
SUP name )
attributetype ( 2.5.4.4 NAME ( 'sn' 'surname' )
DESC 'RFC2256: last (family) name(s) for which the entity is known by'
SUP name )
```

Die erste Zeile gibt jeweils den Typ des Attributs an. An erster Stelle stehen hierbei die eindeutigen Object-IDs (OIDs). Diese IDs gibt es sowohl für Klassen als auch für Attribute. Für deren eindeutige Vergabe ist die IANA zuständig (*www.iana.org*). Es folgt der Eintrag `NAME` mit einem oder mehreren Einträgen (hier: `givenName`, `gn`, `sn`, und `surname`), die, technisch gesehen, lediglich Aliasse für die OID darstellen. Die zweite Zeile enthält hinter `DESC` eine Beschreibung (Description). `SUP` dient zur Angabe der Elternentität. Bei Objektklassen gibt es zusätzlich die Angaben `MUST` und `MAY`, die festlegen, ob ein Attribut erforderlich oder lediglich zugelassen ist.

Die Objekte werden in der LDAP-Datenbank in einer Baumhierarchie abgelegt. Dieser Baum wird als *Directory Information Tree* (DIT) bezeichnet. In der Hierarchie steht die Organisation an erster Stelle. An zweiter Stelle befinden sich Domänen und darunter eventuell vorhandene Subdomänen. Innerhalb der Domänen befinden sich ggf. Organisationseinheiten, die wiederum die sogenannten Blattobjekte (Benutzer, Gruppen, Computer u. Ä.) enthalten.

Der Distinguished Name (kurz DN) gibt hierbei die genaue Position eines Objekts im Verzeichnisbaum an. Wenn ein Benutzer mit dem Namen `john` in der Organisations-

einheit `users` unterhalb der Domäne `homelinux.net` angelegt wurde, sieht der entsprechende DN so aus:

```
dn: uid=john,ou=users,dc=homelinux,dc=net
```

Installation des OpenLDAP-Servers

Sie können die Installation des OpenLDAP-Servers wie gewohnt mit der zu Ihrem System gehörenden Paketverwaltung durchführen. Die folgende Konfiguration bezieht sich überwiegend auf Debian 8, aber Sie sollten keine Probleme haben, wenn Sie eine andere Distribution Ihrer Wahl verwenden. Beginnen Sie die Installation mit:

```
root@archangel:~# apt-get install ldap-server
```

slapd ist der LDAP-Server-Daemon. Er horcht in der Voreinstellung an TCP-Port 389 und wartet auf Anfragen von LDAP-Clients. Er ist außerdem für sämtliche Datenbankoperationen zuständig. Sie können zur Installation von OpenLDAP auch *slapd* als zu installierendes Paket angeben, also:

```
root@archangel:~# apt-get install slapd
```

Ein Installationsassistent wird Sie nach einem Administratorpasswort für Ihren Verzeichnisdienst fragen. Je nach Version des Assistenten wird der DNS-Name des Systems ermittelt und verwendet, um den Verzeichnisdienst zu konfigurieren. Wenn Sie den Namen manuell konfigurieren wollen, den Datenbanktyp selbstauswählen und aus Gründen der Stabilität die Verwendung des veralteten LDAPv2-Protokolls deaktivieren wollen, führen Sie dieses Kommando aus:

```
root@archangel:~# dpkg-reconfigure -plow slapd
```

Ein Assistent wird Sie dann nach den oben genannten Parametern Ihrer Wahl fragen. Stellen Sie nun sicher, dass der Server läuft:

```
root@archangel:~# ps aux |grep slapd
root@archangel:~# netstat -anp |grep slapd
```

Wenn Sie hier positive Ergebnisse sehen, wird eine Zweckentfremdung des Kommandos `slapcat` mehr Aufschluss bringen:

```
root@archangel:~# slapcat
dn: dc=homelinux,dc=net
objectClass: top
objectClass: dcObject
objectClass: organization
```

```
o: homelinux.net
dc: homelinux
structuralObjectClass: organization
entryUUID: 6f7f8afe-0c59-1036-8648-b3edbe6bec5d
creatorsName: cn=admin,dc=homelinux,dc=net
createTimestamp: 20160911105127Z
entryCSN: 20160911105127.962661Z#000000#000#000000
modifiersName: cn=admin,dc=homelinux,dc=net
modifyTimestamp: 20160911105127Z
```

Dateien und Verzeichnisse

Die Konfigurationsdateien für den OpenLDAP-Server finden Sie im Regelfall unterhalb von */etc/openldap*. Bei Debian 8 ist der Pfad allerdings */etc/ldap*. Sie finden hier hauptsächlich LDIF-Dateien, die als Basis für die Konfiguration des Servers dienen. Diese Dateien sollten allerdings nicht von Hand geändert werden. Stattdessen werden Änderungen an der Konfiguration immer über LDAP-Operationen ausgeführt, also mithilfe der Werkzeuge `ldapadd`, `ldapdelete` oder `ldapmodify`. Mehr Informationen darüber finden Sie in der Sektion 5 der Manpages unter *slapd-config*. Die Konfigurationsmethode *slapd-config* erlaubt Konfigurationsänderungen im laufenden Betrieb. Ein Neustart von *slapd* ist also nicht mehr erforderlich.

Ältere Versionen von OpenLDAP (vor Version 2.3) verwendeten zur Konfiguration des Servers die Datei *slapd.conf*. Diese finden Sie immer noch in einschlägigen Internetforen. Lassen Sie sich hierdurch nicht verwirren.

Die Datendateien der eigentlichen LDAP-Datenbank finden Sie im Verzeichnis */var/lib/ldap*. Sie müssen dieses Verzeichnis normalerweise nur aufsuchen, um eine Offlinesicherung der Datenbanken durchzuführen oder versehentlich geänderte Zugriffsberechtigungen zu korrigieren. Wenn Sie eine Offlinesicherung durchführen wollen, müssen Sie *slapd* vorher stoppen.

Bei der Auswahl der zu verwendenden Datenbank können Sie zwischen BDB (Berkeley Database) und HDB (Hybrid Database) wählen. Beide Datenbanken stammen aus Berkeley, wobei HDB die neuere Variante ist.

Exkurs

Um den LDAP-Server in einem realistischen Szenario zu testen, sollten Sie jetzt einen Abstecher zum Abschnitt 210.3, »LDAP-Client-Konfiguration«, machen. Sie können den Server dann mithilfe der entsprechenden Werkzeuge mit Daten füllen, Abfragen ausführen und andere administrative Aufgaben erledigen. Kehren Sie anschließend wieder hierher zurück.

slapd-Kommandos

Im Prinzip kann man einen OpenLDAP-Server komplett mithilfe der clientseitigen Werkzeuge administrieren. Es gibt aber auch einige Tools, die zum Server gehören, und diese müssen Sie für die Prüfung kennen.

slapadd

Mit dem Kommando `slapadd` können Sie dem OpenLDAP-Server Einträge hinzufügen. Da Sie diese Aufgabe bereits während Ihres Exkurses zu LDAP-Client-Konfiguration durchgeführt haben, sollen hier keine vollständigen Beispiele mehr abgedruckt werden. Das Programm `slapadd` ist kein gewöhnlicher LDAP-Client. Vielmehr kann dieses Programm selbst die LDAP-Datenbank öffnen. Deshalb müssen Sie den Daemon *slapd* beenden, damit dieser die Datenbank schließt, bevor Sie mit `slapadd` arbeiten können.

Ein weiterer wichtiger Punkt bei der Verwendung von `slapadd` ist, dass der ausführende Benutzer der Besitzer der Datenbank wird, die `slapadd` erstellt. Deshalb muss die Eigentümerschaft der Dateien ggf. angepasst werden, weil der Daemon sonst keinen Zugriff auf die Datenbanken erhält.

Genau wie `ldapadd` können Sie `slapadd` mit LDIF-Dateien füttern. Die Syntax des Befehls ist allerdings etwas einfacher:

```
root@archangel:/etc/ldap/ldif# slapadd -l user.ldif
```

Die Option `-l` sagt dem Kommando, dass es nicht vom Standardeingabekanal, sondern aus der angegebenen LDIF-Datei lesen soll. Weitere Schalter finden Sie in der entsprechenden Manpage.

Praxistipp

Um Problemen mit Datenbankinkonsistenzen aus dem Wege zu gehen, sollten Sie zumindest bei bereits existierenden Datenbanken vorzugsweise das Programm `ldapadd` verwenden.

slapcat

Wenn Sie den Inhalt einer OpenLDAP-Datenbank in das LDIF-Format exportieren wollen, können Sie `slapcat` verwenden. Auch hier ist die Syntax gegenüber dem entsprechenden LDAP-Client-Werkzeug erheblich einfacher. Ohne Angabe von Optionen schreibt `slapcat` in den Standardausgabekanal. Sie können die Option `-l` verwenden, um die Ausgabe in eine LDIF-Datei umzulenken:

```
root@archangel:/etc/ldap/ldif# slapcat -l alles.ldif
```

Bei der Verwendung von `slapcat` ist es übrigens nicht nötig, *slapd* zu beenden. Das ausgegebene Format ist darauf optimiert, mit `slapadd` wieder eingelesen zu werden. Sie können eine mit `slapcat` generierte LDIF-Datei nicht an `ldapadd` übergeben, ohne zuvor einige Modifikationen vorgenommen zu haben.

slapindex

Mit `slapindex` können Sie die Indizes einer LDAP-Datenbank regenerieren. Als Basis für den zu erstellenden Index wird der aktuelle Inhalt der Datenbank verwendet. Für dieses Programm gelten ähnliche Regeln wie für `slapadd`. Sie müssen *slapd* beenden, wenn Sie die Datenbank neu indexieren wollen. Außerdem sollten Sie mit demselben Benutzerkonto angemeldet sein, das *slapd* zur Laufzeit verwendet, um Probleme mit der Datenbankkonsistenz aufgrund falsch gesetzter Eigentumsrechte zu vermeiden. Eine entsprechende Warnung wird ausgegeben, wenn Sie hier unvorsichtig sind:

```
root@archangel:/etc/ldap/ldif# slapindex
WARNING!
Runnig as root!
There's a fair chance slapd will fail to start.
Check file permissions!
```

Im Notfall werden die Eigentümerschaften einfach wieder korrigiert:

```
root@archangel:/# chown openldap:openldap /var/lib/ldap/*
```

Prüfungstipp

Erfreulicherweise verstehen sowohl die Kommandos `slapadd`, `slapcat` als auch `slapindex` fast dieselben Optionen. Wenn Sie die jeweiligen Manpages miteinander vergleichen, werden Sie sehen, dass es deshalb kein Problem ist, sich die wichtigsten Schalter zu merken.

Loglevel konfigurieren

Um das *Loglevel* von OpenLDAP festzulegen, muss in der Datei *slapd.conf* der Eintrag `loglevel` erstellt bzw. geändert werden. Der Wert für das Loglevel kann dezimal, hexadezimal oder mit den zu protokollierenden Parametern angegeben werden. Mögliche Werte sind:

- `1` (`0x1 trace`) trace function calls
- `2` (`0x2 packets`) debug packet handling
- `4` (`0x4 args`) heavy trace debugging (function args)
- `8` (`0x8 conns`) connection management

- 16 (0×10 BER) print out packets sent and received
- 32 (0×20 filter) search filter processing
- 64 (0×40 config) configuration file processing
- 128 (0×80 ACL) access control list processing
- 256 (0×100 stats) connections, LDAP operations, results (recommended)
- 512 (0×200 stats2) stats log entries sent
- 1024 (0×400 shell) print communication with shell backends
- 2048 (0×800 parse) entry parsing
- 16384 (0×4000 sync) LDAPSync replication
- 32768 (0×8000 none) only messages that get logged whatever log level is set

Ich habe die Tabelle in der englischen Sprache belassen, weil die Parameter für die Option Loglevel zum Teil an die englische Sprache angelehnt sind. Sie sehen hier einige Konfigurationsbeispiele aus der Manpage von *slapd.conf*, die alle zu demselben Ergebnis führen:

```
loglevel 129
loglevel 0x81
loglevel 128 1
loglevel 0x80 0x1
loglevel acl trace
```

Wie Sie sehen, können hier mehrere zu überwachende Parameter, durch Leerzeichen voneinander getrennt, angegeben werden. Wenn Sie die erste und die dritte Zeile oder die zweite und die vierte Zeile miteinander vergleichen, dann sehen Sie, dass die Werte für das Logging auch einfach addiert werden können. Alle fünf Einträge führen deshalb zum selben Ergebnis.

Zugriffssteuerung

Wenn Sie die Zugriffssteuerung nicht konfigurieren, haben selbst anonyme Benutzer per Voreinstellung Leserechte auf Ihr Verzeichnis. Nur der *rootDN* hat hier auch Schreibrechte. Sie können global Zugriffsrechte in der Datei *slapd.conf* konfigurieren und anschließend spezifische Rechte in der Datenbank vergeben. Die im Verzeichnis gesetzten Berechtigungen setzen die in der Datei *slapd.conf* konfigurierten Rechte außer Kraft. Die Zugriffssteuerung funktioniert immer nach diesem Muster:

```
access to <was> by <wer> <Zugriffsart>
```

<was> ist das Objekt, auf das die Zugriffsteuerung konfiguriert wird. Hierbei kann der vollständige DN oder auch ein regulärer Ausdruck verwendet werden. Da *slapd* die

erste Regel, die er findet, sofort anwendet, sollten spezifische Regeln weit oben im Regelwerk stehen und allgemeinere Regeln eher unten.

`<wer>` ist die Entität, der ein Recht gewährt werden soll. Hierbei kommen folgende Einträge infrage:

- `*` sind alle Benutzer ohne Ausnahme.
- `anonymous` sind nicht authentifizierte Benutzer.
- `users` sind authentifizierte Benutzer.
- `self` ist die mit einem Eintrag verbundene Entität selbst.
- `dn.regex=<regex>` sind Benutzer, auf die der reguläre Ausdruck zutrifft.

`<Zugriffsart>` bestimmt, auf welche Art zugegriffen werden darf. Hier können folgende Berechtigungen vergeben werden:

- `none` – kein Zugriff.
- `disclose` wird zur Informationsausgabe bei Fehlern benötigt.
- `auth` wird zur Authentifizierung benötigt.
- `compare` wird für Vergleichsoperationen benötigt.
- `search` gibt das Recht, Suchfilter zu verwenden.
- `read` gibt das Recht, Suchergebnisse anzuzeigen.
- `write` gibt das Recht zum Ändern und Umbenennen.
- `manage` gibt vollständige Verwaltungsrechte.

Die folgenden einfachen Beispiele zeigen Ihnen, wie Sie die Zugriffssteuerung verwenden können:

```
access to * by * read
```

Dieser Eintrag gibt allen Benutzern das Recht, alle Einträge zu lesen. Das ist gleichzeitig die Standardeinstellung.

```
access to *
        by self write
        by anonymous auth
        by * read
```

Hier werden mehrere globale Rechte auf einmal vergeben. Für alle Objekte im Verzeichnis gilt:

Die Entität, der ein Eintrag zugeordnet ist (`self`), hat für diesen auch Schreibrechte. Anonyme Benutzer können Einträge zur Authentifizierung verwenden. Danach sind sie übrigens, wenn kein Fehler auftritt, authentifizierte Benutzer. Zum Schluss werden Leserechte für alle Benutzer vergeben.

Bei manchen Einträgen kommt es auf die Reihenfolge an, wie das folgende Beispiel verdeutlicht:

```
access to dn.children="dc=homelinux,dc=net"
        by * search
access to dn.children="dc=net"
        by * read
```

Hier wurden Leserechte an alle Entitäten unterhalb von `dc=net` vergeben und Suchrechte lediglich an Entitäten unterhalb von `dc=homelinux,dc=net`.

210.3 LDAP-Client-Konfiguration

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Daten von einem LDAP-Server abzufragen und zu ändern. Ebenfalls eingeschlossen ist das Importieren und Hinzufügen von Elementen sowie das Hinzufügen und Verwalten von Benutzern.

Wichtigste Wissensgebiete:

- LDAP-Dienstprogramme zur Datenverwaltung und Abfrage
- Ändern von Benutzerpasswörtern
- Abfragen von Informationen aus einem LDAP-Verzeichnis

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `ldapsearch`
- `ldappasswd`
- `ldapadd`
- `ldapdelete`

Allgemeines

Da es sich bei LDAP um ein (nach X.500) standardisiertes Protokoll handelt, können Sie Ihren Server mit einem beliebigen Client ansprechen. Im Internet werden Sie sogar leicht kostenlose Clients finden, die Sie auf einem Windows-Computer verwenden können, um Ihr OpenLDAP abzufragen oder auch zu ändern. Für die Beispiele auf dieser und den folgenden Seiten habe ich Debian verwendet. Es sollte Ihnen aber keine Probleme bereiten, die entsprechenden Schritte auf jedem anderen System nachzuvollziehen.

Installation und Verwendung des LDAP-Clients

Die Installation des LDAP-Clients geschieht unter Debian und seinen Derivaten ganz einfach mit `apt-get`:

```
root@arch-deb-book:/# apt-get install ldap-client
```

Anschließend müssen in der Datei */etc/ldap/ldap.conf* der zu verwendende LDAP-Server und die Basis des Verzeichnisses angegeben werden. Tragen Sie sinngemäß die folgenden Zeilen ein:

```
BASE    dc=homelinux,dc=net
URI     ldap://archangel.homelinux.net:389
```

Wenn Sie den Server, so wie hier, namentlich angeben, dann muss dieser natürlich namentlich (via DNS oder *hosts*-Datei) auflösbar sein.

Achtung

Sie müssen hinter `BASE` den DN (Distinguished Name) angeben, der mit dem `Suffix` in der Datei *slapd.conf* des LDAP-Servers übereinstimmt. Hinter *ldap://* muss der Name des LDAP-Servers angegeben werden. Das ist in der Regel der FQDN.

Hinzufügen von Objekten zum Verzeichnisdienst

Wenn einem LDAP-Verzeichnisdienst Daten hinzugefügt werden sollen, dann können hierfür LDIF-Dateien verwendet werden. LDIF ist ein Akronym für das *LDAP Data Interchange Format*. Wie der Name bereits erahnen lässt, ist dieses Format auch zum Datenaustausch (im Sinne von Migration) zwischen unterschiedlichen Implementierungen von LDAP geeignet. Außerdem können diese Dateien verwendet werden, um bestehende Objekte zu ändern, wie Sie später noch sehen werden. Auf den nächsten Seiten lernen Sie anhand von Beispielen, wie Sie LDIF-Dateien erstellen und anschließend importieren können. Sie können die Dateien natürlich Ihren eigenen Wünschen entsprechend anpassen.

Erstellen Sie zunächst als Speicherort für Ihre LDIF-Dateien ein neues Verzeichnis. Bei den Kommandos in den folgenden Beispielen gehe ich von der Existenz des Pfades */etc/ldap/ldif* aus.

Um zunächst auf höchster Ebene ein Objekt für eine Domäne bereitzustellen (die Organisation lasse ich hier aus Platzgründen einmal aus), können Sie mit einem beliebigen Texteditor eine solche Datei erstellen:

```
dn:dc=homelinux,dc=net
changetype: add
objectClass: top
```

```
objectClass: domain
dc:homelinux
```

Speichern Sie die Datei unter dem Pfad */etc/ldap/ldif/domain.ldif* ab. Die erste Zeile (beginnend mit `dn:`) enthält den *Distinguished Name*. Er gibt den kompletten Namen eines Objekts innerhalb des Directory Information Tree an. Sie können anhand des *Distinguished Name* Rückschlüsse auf die Position eines Objekts (z. B. Zugehörigkeit zu einer Domäne oder Organisationseinheit) innerhalb des Verzeichnisbaums ziehen. Der angegebene `changetype` weist Programme, die mit dieser LDIF-Datei arbeiten, an, wie mit diesem Objekt zu verfahren ist. In diesem Fall soll das Objekt hinzugefügt werden (`add`). Die `ObjectClass`-Einträge referenzieren auf das Schema und geben hauptsächlich an, um was für einen Objekttyp es sich handelt (hier um eine Domäne). Anschließend folgt der Name des eigentlichen Objekts, hier vom Typ Domain Component (`dc:`). Wenn eine Organisation mehrere Domänen innerhalb der Top-Level-Domain `net` verwendet, ist es angebracht, zunächst auf höchster Ebene ein Objekt zu erstellen, das lediglich die Bezeichnung `dc=net` als *Distinguished Name* verwendet. Somit entsteht ein Stamm, der die unterschiedlichen net-Domains beinhaltet. Entsprechend ergibt die Verwendung der Verwaltungseinheit `Organisation` erst dann einen Sinn, wenn ein Unternehmen Domänen innerhalb mehrerer Top-Level-Domains enthält. Wenn Sie einen zukunftssicheren Baum für ein Unternehmen erstellen wollen, ist es deshalb in jedem Falle sinnvoll, eine Organisation und eine Domäne im Sinn einer TLD (Top-Level-Domain) zu erstellen, bevor Sie die eigentlichen Produktionsdomänen implementieren.

Hinweis

Da in den folgenden Beispielen teilweise mehrzeilige Kommandos verwendet werden, ist der jeweilige Befehl fett gedruckt. Auf diese Art können Sie besser sehen, was zum Kommando gehört und was Ausgaben des Programms sind.

Sie können die Domäne nun in Ihr LDAP-Verzeichnis importieren, indem Sie folgendes Kommando verwenden:

```
harald@arch-deb-book:/etc/ldap/ldif$ ldapadd -x -D
"cn=admin,dc=homelinux,dc=net" -W -f domain.ldif
Enter LDAP Password:
adding new entry "dc=homelinux,dc=net"
```

Wenn Ihre Konfiguration aus dem Abschnitt »Installation des OpenLDAP-Servers« vollständig nachvollzogen wurde, erhalten Sie hier allerdings diese Meldung, weil die Domäne bereits existiert:

```
adding new entry "dc=homelinux,dc=net"
ldap_add: Already exists (68)
```

Das ist ein Zeichen dafür, dass Sie bis jetzt alles richtig gemacht haben. Beim Kommando `ldapadd` handelt es sich eigentlich nur um einen Softlink, der auf das Programm `ldapmodify` zeigt. Sie hätten die Domäne deshalb auch genauso gut mithilfe des Kommandos `ldapmodify -a` erzeugen können. Die im vorangehenden Beispiel verwendeten Optionen von `ldapadd` bedeuten Folgendes:

- `-x` verwendet normale Authentifizierung anstatt SASL.
- `-D` kündigt den Distinguished Name für die Authentifizierung am LDAP-Server an. Der anschließend angegebene Distinguished Name wurde in der Datei *slapd.conf* des Servers konfiguriert.
- `-W` sorgt dafür, dass ein Passwort erst nach der Kommandoeingabe abgefragt wird. Es müsste ansonsten im Klartext (sichtbar!) auf der Kommandozeile eingegeben werden.
- `-f` (file) kündigt die LDIF-Datei an.

Die nächste LDIF-Datei soll dazu dienen, zwei Organisationseinheiten höchster Ebene zu erstellen. Die eine Organisationseinheit soll dann später normale Benutzer und die andere administrative Benutzerkonten aufnehmen. Erstellen Sie also zunächst mit einem beliebigen Editor eine Datei mit der Bezeichnung *ou.ldif* und folgendem Inhalt:

```
dn: ou=managers,dc=homelinux,dc=net
changetype: add
objectClass: top
objectClass: organizationalUnit
ou: managers

dn: ou=users,dc=homelinux,dc=net
changetype: add
objectClass: top
objectClass: organizationalUnit
ou: users
```

Der Aufbau beider Einträge innerhalb der Datei ist identisch. Die erste Zeile enthält, wie vorgeschrieben, jeweils den Distinguished Name des zu erstellenden Objekts. Die optionalen Zeilen des `changetype` besagen wiederum, dass die Objekte jeweils hinzugefügt werden sollen. Es handelt sich bei beiden Objekten um Organisationseinheiten auf Top-Level-Ebene, was durch die Einträge `objectClass` jeweils definiert wird. Die letzten Zeilen enthalten noch einmal den reinen Namen der jeweiligen OU (Organizational Unit).

Der Import der LDIF-Datei wird genauso durchgeführt wie bei der Erstellung der Domäne. Wie Sie sehen, können Sie mit einer einzigen LDIF-Datei ohne Probleme

mehrere Objekte in einem Arbeitsschritt anlegen. Die enthaltenen Objekte müssen dazu übrigens noch nicht einmal vom gleichen Typ sein. Verwenden Sie also wieder diesen Befehl:

```
harald@arch-deb-book:/etc/ldap/ldif$ ldapadd -x -D
"cn=admin,dc=homelinux,dc=net" -W -f ou.ldif
Enter LDAP Password:
adding new entry "ou=managers,dc=homelinux,dc=net"
adding new entry "ou=users,dc=homelinux,dc=net"
```

Die nächste LDIF-Datei soll verwendet werden, um zwei Benutzerkonten zu erstellen. Das erste Konto ist das des Admin, damit dieser in Zukunft (wenn die PAM-Konfiguration entsprechend angepasst worden ist) via LDAP authentifiziert werden kann. Das zweite Konto soll als normales Benutzerkonto verwendet werden. Erstellen Sie eine Datei mit der Bezeichnung *user.ldif* und folgendem Inhalt:

```
dn: cn=admin,ou=managers,dc=homelinux,dc=net
changetype: add
objectClass: top
objectclass: organizationalRole
ou: managers
cn: admin
description: LDAP-Admin

dn: uid=harald,ou=users,dc=homelinux,dc=net
changetype: add
objectclass: top
objectclass: person
objectclass: organizationalPerson
objectclass: inetOrgPerson
cn: Harald Maassen
sn: Maassen
givenname: Harald
uid: harald
ou: users
description: Benutzerkonto
```

Wie Sie sehen, stecken in dieser LDIF-Datei zwei verschiedene Objektklassen, denen auch unterschiedliche Attribute zugeordnet sind. Sie können den Import aber auch diesmal mit dem bereits bekannten Kommando durchführen:

```
harald@arch-deb-book:/etc/ldap/ldif$ ldapadd -x -D
"cn=admin,dc=homelinux,dc=net" -W -f user.ldif
Enter LDAP Password:
```

```
adding new entry "uid=harald,ou=users,dc=homelinux,dc=net"
adding new entry "cn=admin,ou=managers,dc=homelinux,dc=net"
```

Es wird Zeit, eine Datenbankabfrage durchzuführen, damit Sie überprüfen können, ob OpenLDAP Ihre Importvorgänge auch wirklich durchgeführt und die Objekte gespeichert hat. Eine sehr einfache Abfrage mit einer Klartextauthentifizierung (`-x`) und der gesamten Domäne als Basis (`-b`) für die Suche könnte so aussehen:

```
harald@arch-deb-book:~$ ldapsearch -x -b dc=homelinux,dc=net
# extended LDIF
#
# LDAPv3
# base <dc=homelinux,dc=net> with scope subtree
# filter: (objectclass=*)
# requesting: ALL
#

# homelinux.net
dn: dc=homelinux,dc=net
objectClass: top
objectClass: domain
dc: homelinux

# managers, homelinux.net
dn: ou=managers,dc=homelinux,dc=net
objectClass: top
objectClass: organizationalUnit
ou: managers

# users, homelinux.net
dn: ou=users,dc=homelinux,dc=net
objectClass: top
objectClass: organizationalUnit
ou: users

# admin, managers, homelinux.net
dn: cn=admin,ou=managers,dc=homelinux,dc=net
objectClass: top
objectClass: organizationalRole
ou: managers
cn: admin
description: LDAP-Admin
```

```
# harald, users, homelinux.net
dn: uid=harald,ou=users,dc=homelinux,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Harald Maassen
sn: Maassen
givenName: Harald
uid: harald
ou: users
description: Benutzerkonto
mail: harald@homelinux.net

# search result
search: 2
result: 0 Success

# numResponses: 6
# numEntries: 5
```

Sie können `ldapsearch` natürlich auch einsetzen, wenn Sie nach einem bestimmten Objekt suchen. Eine typische Suche nach einem Benutzer könnte dann so aussehen:

```
harald@arch-deb-book:~$ ldapsearch -x -LLL uid=harald
dn: uid=harald,ou=users,dc=homelinux,dc=net
objectClass: top
objectClass: person
objectClass: organizationalPerson
objectClass: inetOrgPerson
cn: Harald Maassen
sn: Maassen
givenName: Harald
uid: harald
ou: users
description: Benutzerkonto
```

Wie Sie sehen, wurde diesmal keine Basis für die Suche (`-b`) angegeben. Stattdessen wurde das gesamte Verzeichnis nach `uid=harald` durchsucht. Die Option `-LLL` reduziert die Ausgabe um einige zusätzliche Zeilen. Das erste `L` beschränkt die Ausgabe des Kommandos auf die LDIF-Version 1. Das zweite `L` deaktiviert die Ausgabe von Kommentaren und das dritte `L` unterdrückt die Ausgabe der LDIF-Version.

Als Suchkriterium kommt übrigens jedes beliebige Attribut eines Objekts infrage. Sie können also z. B. auch nach Vornamen (`givenName`) oder Nachnamen (`sn`) suchen.

LDAP-Passwörter ändern

Wenn OpenLDAP dazu verwendet wird, Benutzer zu authentifizieren, benötigen Sie natürlich auch eine Möglichkeit, um die Passwörter der Benutzer zu ändern. Für diese Aufgabe können Sie das Programm `ldappasswd` verwenden. Die Syntax bezüglich der Authentifizierung gegenüber dem LDAP-Server (mit `-x`, `-D` und `-W`) ist mit der Authentifizierung von `ldapadd` bzw. `ldapmodify` identisch. Es folgt die Angabe des Distinguished Name des Benutzers, dessen Passwort geändert werden soll und hinter der Option `-s` wird als Parameter das neue Passwort übergeben:

```
harald@arch-deb-book:~$ ldappasswd -x -D "cn=admin,dc=homelinux,dc=net" -W
"uid=harald,ou=users,dc=homelinux,dc=net" -s NeuesPasswort
Enter LDAP Password:
```

Sie können auch ein neues Passwort automatisch generieren lassen, indem Sie einfach die Option `-s` weglassen. Das neue Passwort wird dann am Bildschirm ausgegeben:

```
harald@arch-deb-book:/$ ldappasswd -x -D "cn=admin,dc=homelinux,dc=net" -W
"uid=harald,ou=users,dc=homelinux,dc=net"
Enter LDAP Password:
New password: elmEWgvp
```

LDAP-Einträge ändern

Wenn Sie Einträge in der LDAP-Datenbank ändern wollen, können Sie ebenfalls LDIF-Dateien verwenden. Sie müssen dann allerdings die `changetype`-Zeile mit `modify` festlegen. Sie können dann nicht nur bestehende Attribute eines Objekts ändern, sondern auch zusätzliche Attribute zu einem Objekt hinzufügen. Um einem bestehenden Benutzer eine E-Mail-Adresse als Attribut hinzuzufügen, wird im folgenden Beispiel eine LDIF-Datei mit dem Namen *userh.ldif* erstellt:

```
dn: uid=harald,ou=users,dc=homelinux,dc=net
changetype: modify
add: mail
mail: harald@homelinux.net
```

Sie können die LDIF-Datei anschließend mit dem Programm `ldapmodify` importieren. Da Sie schon mehrfach mit `ldapadd` in Berührung gekommen sind und `ldapadd` ja lediglich ein Link auf `ldapmodify` ist, sollten Sie keine Probleme damit haben, die Syntax dieses Kommandos nachzuvollziehen:

```
harald@arch-deb-book:/etc/ldap/ldif$ ldapmodify -x -D
"cn=admin,dc=homelinux,dc=net" -W -f userh.ldif
Enter LDAP Password:
modifying entry "uid=harald,ou=users,dc=homelinux,dc=net"
```

Sie sollten anschließend `ldapsearch` verwenden, um das Ergebnis der Änderung zu überprüfen:

```
harald@arch-deb-book:~$ ldapsearch -LLL -x -b
uid=harald,ou=users,dc=homelinux,dc=net mail
dn: uid=harald,ou=users,dc=homelinux,dc=net
mail: harald@homelinux.net
```

LDAP-Einträge löschen

Um Einträge aus dem Verzeichnisdienst zu löschen, wird das Programm `ldapdelete` verwendet. Die Authentifizierung gegenüber dem LDAP-Server geschieht wieder genauso, wie Sie es bereits von `ldapadd` und `ldapmodify` her kennen, mit den Optionen `-D` und `-W`. Bei der Löschung wird dann lediglich der Distinguished Name des zu löschenden Objekts angegeben:

```
harald@arch-deb-book:/$ ldapdelete -D "cn=admin,dc=homelinux,dc=net" -W
"uid=harald,ou=users,dc=homelinux,dc=net"
Enter LDAP Password:
```

Sie können auch hier eine LDIF-Datei verwenden, wenn Sie mehrere oder sogar viele Objekte löschen müssen.

Zusammenfassung der Optionen

Wie Ihnen mit Sicherheit aufgefallen ist, verwenden die Kommandozeilentools für OpenLDAP jeweils ähnliche Optionen. Das ist insofern hilfreich, als dass Sie für die Prüfung weniger Optionen auswendig kennen müssen.

Es folgt deshalb an dieser Stelle eine Aufstellung der wichtigsten Optionen der LDAP-Kommandos:

- `-D` – Der hinter dieser Option angegebene Distinguished Name wird für die Authentifizierung gegenüber dem LDAP-Server verwendet. Man spricht hier von einer Bindung zum Server.
- `-w` – Hinter dieser Option kann das Passwort für die Authentifizierung direkt auf der Kommandozeile angegeben werden. Da das Passwort im Klartext auf der Konsole angezeigt wird, ist das keine gute Option, wenn Ihnen gerade jemand bei der Arbeit zusieht.

- `-W` sorgt dafür, dass ein Passwort erst nach der Betätigung der Eingabetaste vom Server abgefragt wird. Das Passwort wird bei der Verwendung dieser Option nicht auf der Konsole angezeigt.
- `-n` bewirkt, dass die Ausführung eines Kommandos lediglich simuliert wird. Das ist eine nützliche Option, wenn Sie etwas testen müssen.
- `-a` bewirkt bei `ldapmodify`, dass ein Eintrag nicht modifiziert, sondern hinzugefügt wird (add). Die ausgeführte Operation entspricht dann der des Befehls `ldapadd`.
- `-f` wird der ggf. zu verwendenden LDIF-Datei (file) vorangestellt.
- `-x` sorgt für einfache Authentifizierung ohne SASL.

210.4 PAM-Authentifizierung

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, PAM für verschiedene Authentifizierungsmethoden zu konfigurieren. Das beinhaltet auch SSSD-Authentifizierung.

Wichtigste Wissensgebiete:

- PAM-Konfigurationsdateien, -Begriffe und -Dienstprogramme
- Passwörter in *passwd* und *shadow*
- Verwendung von SSSD zur LDAP-Authentifizierung

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/pam.d*
- *pam.conf*
- *nsswitch.conf*
- *pam_unix*, *pam_cracklib*, *pam_limits*, *pam_listfile*. *pam_sss*
- *sssd.conf*

Allgemeines

PAM ist das Akronym für *Pluggable Authentication Modules*. Es handelt sich hierbei um ein System, das aus Softwarebibliotheken besteht und Schnittstellen zwischen Authentifizierungsmechanismen und -diensten bereitstellt. Aufseiten der Authentifizierungsmechanismen kämen z. B. Dateien (*/etc/passwd*), Kerberos, LDAP oder Smartcards in Betracht. Diese Methoden können dann von Diensten wie Login, SSH, Webserver oder FTP verwendet werden, ohne dass die Mechanismen zur Authentifi-

zierung innerhalb der jeweiligen Anwendungen einzeln implementiert werden müssen. Die eigentlichen Module befinden sich normalerweise im Verzeichnis */lib/security*. Man kann bereits an den Dateierweiterungen der Module (**.so*) erkennen, dass es sich hierbei um Shared Objects im Sinne von Bibliotheken handelt. Einige typische Module sind:

- *pam_env.so*
- *pam_deny.so*
- *pam_lastlog.so*
- *pam_motd.so*
- *pam_sss.so*
- *pam_permit.so*
- *pam_unix.so*
- *pam_mail.so*
- *pam_limits.so*

Wenn weitere Module benötigt werden, können diese dem Verzeichnis */lib/security* einfach hinzugefügt und anschließend in die entsprechenden Konfigurationsdateien eingebunden werden. In den meisten Fällen ist z. B. kein Modul für die Authentifizierung via LDAP vorhanden. Dieses Modul lässt sich aber ggf. über die jeweilige Paketverwaltung ohne Probleme nachinstallieren, wie Sie noch sehen werden.

PAM-Konfiguration

Traditionell wird die Konfiguration von PAM in der Datei */etc/pam.conf* vorgenommen. Heutzutage verwenden Anwendungen aber fast nur noch Dateien unterhalb des Verzeichnisses */etc/pam.d* zur Konfiguration ihrer PAM-Module. Diese Konfigurationsdateien sind zum Teil mithilfe von `include`-Anweisungen miteinander verknüpft. Da die Reihenfolge bei der Verarbeitung der Regeln in dieser Datei von Bedeutung ist, müssen Sie bei der Positionierung neuer Regeln vorsichtig sein.

Achtung

Wenn das Verzeichnis */etc/pam.d* auf einem System existiert, wird die Datei */etc/pam.conf* ignoriert.

Es gibt bei der Verwendung der Konfigurationsdatei */etc/pam.conf* einen geringfügigen Unterschied in der Syntax der Regeln gegenüber der Verwendung von einzelnen Dateien unterhalb von */etc/pam.d*. In der Datei *pam.conf* hat eine Regel diesen Aufbau:

```
Service    Type     Control     Modulepath     Moduleargumente
```

Eine Beispielregel für die Authentifizierung gegenüber `sshd` könnte innerhalb der Datei */etc/pam.conf* so aussehen:

```
sshd       auth       sufficient       pam_unix.so       nullok
```

Diese Zeile ist zu interpretieren wie folgt:

- `sshd` ist der `Service`, auf den diese Regel angewendet werden soll.
- `auth` ist der `Type`, also die Verwaltungsgruppe von PAM. In diesem Fall dient er der Authentifizierung des Benutzers.
- `sufficient` ist hier das `Control`, welches das Verhalten von PAM steuert. Bei dieser Regel ist eine normale Authentifizierung also ausreichend. `pam_unix.so` steht in dieser Regel für den `Modulepath`. Wie Sie sehen, ist keine absolute Pfadangabe erforderlich, wenn sich das Modul in dem dafür vorgesehenen Standardverzeichnis (*/lib/security*) befindet.
- `nullok` ist ein relativ gefährliches `Moduleargument`. Es erlaubt die Anmeldung von Benutzern, die über kein Passwort verfügen.

Bei der Verwendung einzelner Konfigurationsdateien pro Service (unterhalb von */etc/pam.d*) ist die Angabe der ersten Spalte nicht nötig, weil der Dienst durch den Dateinamen der Datei selbst repräsentiert wird. In Bezug auf das vorangehende Beispiel bedeutet das die Existenz der Datei */etc/pam.d/sshd* mit zumindest diesem Inhalt:

```
auth       sufficient      pam_unix.so      nullok
```

Beispiele für Services, die typischerweise PAM zur Authentifizierung verwenden, sind `login`, `sshd`, `sudo`, `crond`, `dovecot`, `squid` und `samba`, um nur ein paar wichtige Dienste zu nennen.

PAM-Verwaltungsgruppen (Type)

Es gibt vier Typen, also Verwaltungsgruppen, die Sie bei der Konfiguration von PAM verwenden können. Das sind:

- `auth` hat zwei Aufgaben. Einerseits wird die Authentizität eines Benutzers geprüft, indem die jeweilige Anwendung (z. B. `sshd`) aufgefordert wird, von dem Benutzer ein Passwort (oder auch andere Authentifizierungsmechanismen, wie Smartcards usw.) zu erfragen. Andererseits ist `auth` für die Gewährung von Gruppenmitgliedschaften zuständig.
- `account` führt Managementaufgaben aus, die nicht auf Authentifizierung basieren. Das wären z. B. das Erteilen oder Verweigern von Ressourcenzugriffen, basierend auf der Tageszeit oder dem Systemzustand (Auslastung usw.).
- `password` befasst sich mit den Eigenschaften des Benutzerpasswortes. Es ist z. B. möglich, in Kombination mit dem Modul *pam_cracklib.so* einen Vergleich eines vom Benutzer gewünschten Passworts mit einem Wörterbuch durchzuführen. So können leicht zu erratende Passwörter vermieden werden.

- `session` ist hauptsächlich für die Protokollierung zuständig. Das beinhaltet auch die Protokollierung von Zugriffen auf Ressourcen durch den Benutzer.

PAM-Verhalten (Control)

Bei einem einzigen Authentifizierungsprozess sind normalerweise mehrere PAM-Module involviert. Das bedeutet aber nicht, dass ein Benutzer sich nicht anmelden kann, wenn nur ein einziges PAM-Modul während der Authentifizierung fehlschlägt. Es könnte z. B. möglich sein, dass ein System mehrere Authentifizierungsmethoden unterstützt (z. B. Kennwortauthentifizierung und Smartcard-Authentifizierung). In einem solchen Fall kann über die Control-Felder von PAM konfiguriert werden, ob ein User sowohl die Bedingung der Kennwortauthentifizierung als auch die Smartcard-Authentifizierung bestehen muss oder nur eine der beiden Bedingungen.

Traditionell werden im Control-Feld die folgenden Werte verwendet:

- `required`: Wenn ein Fehler in diesem Modul auftritt, erhält die Anwendung einen Fehlerstatus, sobald alle verbleibenden Module dieses Typs für diese Anwendung verarbeitet worden sind. Es wird dann kein Zugriff gewährt.
- `requisite` ähnelt `required`, gibt aber die Kontrolle sofort an die Anwendung zurück. Auch hier wird dem Benutzer kein Zugriff erteilt.
- `sufficient`: Wenn die Voraussetzungen dieses Moduls erfüllt sind, wird die Authentifizierung bestätigt. Das gilt auch, wenn andere Module einen Fehlerstatus erzeugen würden. Ein Fehlschlagen dieses Moduls alleine verhindert aber nicht die Authentifizierung.
- `optional`: Wie der Name es erahnen lässt, werden hier optionale Module geladen, deren Erfolg oder Fehler nicht entscheidend für die Authentifizierung sind.
- `include` bindet den Inhalt weiterer Konfigurationsdateien für einen angegebenen Typen an dieser Stelle ein.

Für komplexere Konfigurationen gibt es eine erweiterte Syntax in der Form `[wert1=aktion1 wert2=aktion2…]`. Auf diese Art können für verschiedene Fälle unterschiedliche Aktionen konfiguriert werden. Da diese Syntax bisher nicht in Prüfungen thematisiert wird, soll an dieser Stelle auch nicht weiter darauf eingegangen werden.

/etc/nsswitch.conf

Die Konfigurationsdatei */etc/nsswitch.conf* kennen Sie bereits im Zusammenhang mit der DNS-Client-Konfiguration. Hier wurde festgelegt, ob ein Client bei der DNS-Namensauflösung zuerst die lokale Datei */etc/hosts* oder einen DNS-Server verwenden soll. Die Verwendung dieser Datei im Zusammenhang mit PAM ist ganz ähnlich. Wenn Sie zum Beispiel zur Authentifizierung LDAP einsetzen, dann sollten Sie die Verwendung Ihrer lokal vorhandenen Unix-Konten nicht endgültig verhin-

dern. Sie könnten sich dann nämlich nicht anmelden, wenn der LDAP-Server nicht erreichbar ist. Eine gute Idee ist es, zunächst auf lokale Konten und erst dann auf LDAP zurückzugreifen, wenn ein Konto nicht auf der lokalen Maschine existiert. Die relevanten Zeilen in der Datei *nsswitch.conf* müssten in diesem Fall wie folgt konfiguriert werden:

```
passwd:     files ldap
shadow:     files ldap
group:      files ldap
```

In dieser Konfiguration werden zur Authentifizierung zuerst lokale Dateien (`files`) herangezogen. Sie kennen diese Dateien bereits, weshalb hier keine detaillierte Beschreibung, sondern lediglich eine kurze Aufzählung zu Ihrer Erinnerung folgt:

- */etc/passwd* zur Ermittlung des Benutzerkontos
- */etc/shadow* zur Ermittlung des Passworts
- */etc/group* zur Zuordnung der Benutzergruppen

Wird PAM in den lokalen Dateien nicht fündig, dann wird zur Authentifizierung LDAP herangezogen. Voraussetzung hierfür ist natürlich eine entsprechende PAM-Konfiguration und die Existenz des für die Authentifizierung benötigten Moduls *pam_ldap.so*.

PAM-Module

Es gibt vier PAM-Module, die vom LPI ausdrücklich als Prüfungsthemen genannt werden. Das sind die Module *pam_unix*, *pam_cracklib*, *pam_limits* und *pam_listfile*. Es lohnt sich also, diese genauer zu betrachten.

pam_unix

pam_unix ist das Modul, das für die traditionelle Passwortauthentifizierung verwendet wird. Es liest während der Benutzeranmeldung die Dateien */etc/passwd* und */etc/shadow*. Wenn keine Optionen für dieses Modul konfiguriert werden, akzeptiert *pam_unix* keine leeren Passwörter. Sie können diesem Modul eine ganze Reihe von Optionen übergeben. Sehen Sie bei Bedarf in der entsprechenden Manpage nach.

pam_cracklib

pam_cracklib wird normalerweise nur in Regeln vom Typ *password* verwendet. Wenn ein Benutzer sein Passwort ändern will, kann *pam_cracklib* diverse Überprüfungen durchführen, um die Sicherheit des neu gewählten Passworts zu gewährleisten. Das gewünschte Passwort wird gegen ein Wörterbuch geprüft, um real existierende Wörter als Passwort auszuschließen. Auf diese Art wird ein Wörterbuch-

angriff auf das Benutzerkonto nahezu ausgeschlossen. Weiterhin prüft *pam_cracklib*, ob die Komplexitätsvoraussetzungen für ein sicheres Kennwort gegeben sind. Es werden auch Vergleiche mit den vorherigen Passwörtern eines Benutzers durchgeführt. Alte Passwörter speichert das System zu diesem Zweck in der Datei */etc/security/opasswd*. Es wird nicht nur geprüft, ob der User ein Passwort schon einmal verwendet hat, sondern auch, ob das neue gewünschte Passwort lediglich ein Palindrom, eine Rotation oder nur eine Änderung der Groß-/Kleinschreibung eines vorherigen Passworts darstellt.

pam_limits

Dieses Modul wird verwendet, um den Zugriff eines Benutzers auf Ressourcen zu begrenzen. Bei einem Computer, der von mehreren Benutzern gleichzeitig verwendet wird, kann es z. B. erforderlich werden, die CPU-Zeit, die Anzahl gleichzeitig laufender Prozesse oder auch die Nutzung von Arbeitsspeicher zu limitieren. Typischerweise wird dieses Modul in Regeln vom Typ `auth` als Letztes verwendet. Die Einstellungen werden (soweit nicht ausdrücklich anders festgelegt) in Dateien mit der Erweiterung *conf* im Verzeichnis */etc/security/limits.d/* festgelegt. Ursprünglich gab es nur die Konfigurationsdatei */etc/security/limits.conf*. Da der Inhalt dieser Dateien nicht prüfungsrelevant ist und deren Optionen, was den Praxiseinsatz anbelangt, sehr gut kommentiert sind, soll hier auf weitere Einzelheiten verzichtet werden.

pam_listfile

Wenn Sie für bestimmte Dienste den Zugriff auf einer Liste basierend steuern müssen, dann können Sie das Modul *pam_listfile* verwenden. Hierbei können diese Listen verwendet werden, um den Zugriff auf einen Dienst zu gestatten (`allow`) oder zu verweigern (`deny`).

1. Beispiel: Wenn Sie bestimmten Benutzern den Zugriff via SSH verweigern möchten, erstellen Sie eine Liste mit den entsprechenden Benutzernamen und speichern diese unter dem Namen */etc/nosshuser* ab. Fügen Sie anschließend der Datei */etc/pam.d/sshd* eine Zeile mit folgendem Inhalt hinzu:

```
auth required pam_listfile.so item=user sense=deny file=/etc/nosshuser
```

Die Regel ist fast selbsterklärend. Sie wird während der Authentifizierung (`auth`) verwendet, und das erfolgreiche Bestehen der Prüfung ist Voraussetzung (`required`). Das Modul *pam_listfile.so* kommt zum Einsatz.

Die verwendete Liste (`file=/etc/nosshuser`) enthält Benutzer (`item=user`), denen der Zugriff verweigert (`sense=deny`) werden soll.

2. Beispiel: Sie können den Zugriff auch auf einer Liste basierend gewähren. Legen Sie in diesem Fall eine Liste mit Benutzern an, denen der Zugriff erlaubt werden soll (z. B. */etc/sshuser*). Diesmal muss der Eintrag in der Datei */etc/pam.d/sshd* inhaltlich so aussehen:

```
auth required pam_listfile.so item=user sense=allow file=/etc/sshuser
```

Unterschiede bestehen lediglich in der verwendeten Datei (*/etc/sshuser*) und darin, dass der Wert für `sense` diesmal auf `allow` gesetzt wurde.

PAM-Authentifizierung mit LDAP

Wenn Sie PAM verwenden wollen, um eine Authentifizierung via LDAP durchzuführen, dann sind verschiedene Punkte zu berücksichtigen.

1. Zunächst einmal benötigen Sie ein entsprechendes PAM-Modul. Bei Debian und seinen Ablegern werden Sie fündig, wenn Sie wahlweise die Pakete *libpam-ldap* oder *libpam-ldapd* installieren. In beiden Fällen erhalten Sie das benötigte Modul *pam_ldap.so*. Wenn Sie Red Hat oder ein verwandtes System verwenden, installieren Sie das Paket *nss_ldap*, um zu demselben Ergebnis zu kommen.
2. Passen Sie die Datei */etc/nsswitch.conf* an, wie Sie es auf den vorangehenden Seiten gelernt haben. Die zur Authentifizierung relevanten Zeilen sollten anschließend etwa so aussehen:

   ```
   passwd:     files ldap
   shadow:     files ldap
   group:      files ldap
   ```

   Für den Fall, dass etwas schief läuft, werden also weiterhin Ihre lokalen Benutzerkonten in der Datei */etc/passwd* verwendet.
3. Jetzt muss das PAM-Modul in die Konfiguration aufgenommen werden. Das ist insofern schwierig, weil sich an dieser Stelle die verschiedenen Linux-Distributionen stark voneinander unterscheiden. Bei einer aktuellen Version von Red Hat (Stand: Dezember 2016) können Sie gut die Datei *system-auth-ac* im Verzeichnis */etc/pam.d* verwenden. Diese Datei wirkt sich bei der Authentifizierung gegenüber allen Diensten aus. Fügen Sie die Zeile für das Modul *pam_ldap.so* ein, sodass der Anfang der Datei so aussieht:

   ```
   auth     required      pam_env.so
   auth     sufficient    pam_ldap.so
   auth     sufficient    pam_unix.so nullok try_first_pass
   auth     requisite     pam_succeed_if.so uid >= 500 quiet
   auth     required      pam_deny.so
   ```

Bei Debian und seinen Verwandten können Sie einen entsprechenden Eintrag in der Datei *common-auth* ebenfalls unterhalb von */etc/pam.d* vornehmen.

4. Erstellen Sie geeignete Benutzerkonten in Ihrem LDAP-Verzeichnis. Die Konten aus den Übungen im Abschnitt über LDAP sind für eine Anmeldung unzureichend. Ein Unix-Konto sollte z. B. über eine UID und ein geeignetes Home Directory verfügen. Sie sollten die Werte für die UIDs ziemlich hoch ansiedeln, um Konflikte mit lokal angelegten Benutzerkonten zu vermeiden. Werte ab 1.200 aufwärts sollten in den meisten Fällen ausreichend sein. Erstellen Sie auch das Home Directory und versehen Sie es mit entsprechenden Rechten, damit es bei der Anmeldung nicht zu Fehlermeldungen kommt. Das folgende Beispielkonto enthält alle Attribute, die für eine Anmeldung erforderlich sind. Sie können es einfach in eine LDIF-Datei übernehmen und Ihren Gegebenheiten anpassen:

```
dn: uid=willi,ou=users,dc=homelinux,dc=net
objectClass: person
objectClass: inetOrgPerson
objectClass: organizationalPerson
objectClass: posixAccount
uid: willi
userpassword: WillisPasswort
facsimiletelephonenumber: +49 30 4711 0815
givenname: Willi
cn: Willi Wichtig
telephonenumber: +49 30 4711 0814
sn: Wichtig
roomnumber: 0
homeDirectory: /home/willi
mail: willi@homelinux.net
l: Berlin
ou: users
uidNumber: 1211
gidNumber: 1211
```

Importieren Sie die fertige Datei in Ihren Verzeichnisdienst, wie Sie es im Abschnitt über LDAP gelernt haben:

```
root@arch-deb-book:/etc/ldap/ldif# ldapadd -x \
-D "cn=admin,dc=homelinux,dc=net" -W -f willi.ldif
```

Wenn Sie das Konto zur Anmeldung verwenden, müssen Sie lediglich den reinen Benutzernamen angeben. Die Angabe einer Domäne ist nicht erforderlich.

PAM-Authentifizierung mit SSSD

Der *System Security Services Daemon* (*SSSD*) ist ein Dienst, der Unterstützung für unterschiedliche Authentifizierungsanbieter bereitstellt. Er kann eine Authentifizierung direkt an einer nativen LDAP-Domäne, aber auch einer Domäne mit Kerberos Unterstützung vornehmen. Hierbei ist auch ein Caching der Authentifizierung implementiert, sodass ein Benutzer auch dann authentifiziert werden kann, wenn vorübergehend kein LDAP-Server erreichbar ist. Es gibt außerdem ein passendes PAM-Modul, nämlich *pam_sss.so* und es sind in Zukunft weitere Authentifizierungsquellen denkbar.

Unabhängig von der verwendeten Linux-Distribution muss zunächst das Paket *sssd* installiert werden.

Auch bei der Verwendung von *sssd* muss zunächst die Datei *nsswitch.conf* angepasst werden, wenn das von der Paketverwaltung nicht automatisch erledigt wurde. Das könnte z. B. so aussehen:

```
passwd:         compat sss
group:          compat sss
shadow:         compat sss
```

Wenn Sie die Paketverwaltung verwendet haben, dann wird wahrscheinlich auch PAM schon angepasst worden sein. Der »primary block« enthält z. B. auf einem Debian-System diese Zeilen:

```
Password        requisite                       pam_pwquality.so retry=3
password        [success=2 default=ignore]      pam_unix.so obscure use_
authtok try_first_pass sha512
password        sufficient                      pam_sss.so use_authtok
```

Die Konfigurationsdatei *sssd.conf* ist im Normalfall nicht vorhanden und muss erst erzeugt werden. Die folgende Beispieldatei geht davon, dass zur Authentifizierung die Domäne verwendet wird, die im Zusammenhang mit Samba und Active Directory verwendet wurde, und dass die entsprechende Samba-Konfiguration noch vorliegt.

```
[sssd]
services = nss, pam
config_file_version = 2
domains = nwa-net.de
[domain/NWA-NET.DE]
id_provider = ad
access_provider = ad
pam_mkhomedir.so
override_homedir = /home/%d/%u
```

211 E-Mail-Dienste

Als Ray Tomlinson im Jahre 1971 die erste E-Mail der Welt verschickte, verwendete er bereits das @-Zeichen als Trennung zwischen dem Benutzernamen und dem Zielcomputer. Inzwischen löst die E-Mail weitestgehend den Brief und die Postkarte ab.

211.1 Betreiben von E-Mail-Servern

Wichtung: 4

Beschreibung: Die Prüflinge sollten in der Lage sein, einen E-Mail-Server zu betreiben. Dies umfasst die Einrichtung von E-Mail-Aliassen, E-Mail-Quotas und virtuellen E-Mail-Domains. Dieses Lernziel beinhaltet auch die Konfiguration von internen E-Mail-Relays und die Überwachung des E-Mail-Servers.

Wichtigste Wissensgebiete:

- Konfigurationsdateien für *Postfix*
- Grundlagen der TLS-Konfiguration für *Postfix*
- Grundlagen des SMTP-Protokolls
- Kenntnis von Sendmail und *Exim*

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- Konfigurationsdateien und Befehle für *Postfix*
- */etc/postfix/**
- */var/spool/postfix*
- `sendmail`-Befehle der Emulationsschicht
- */etc/aliases*
- Mailbezogene Logdateien unter */var/log/*

Allgemeines

Die Verarbeitung und Übermittlung von E-Mail-Nachrichten wird in der Hauptsache von drei Softwarekomponenten durchgeführt:

- **MUA**
  Der Mail User Agent ist ein Mail-Client-Programm, mit dem E-Mails geschrieben, gelesen und versendet werden, z. B. KMail, Evolution, Thunderbird, Outlook usw.
- **MDA**
  Der Mail Delivery Agent verarbeitet E-Mails auf einem Server. Er entscheidet, wie mit den E-Mails weiter zu verfahren ist. Handelt es sich um lokale E-Mail-Adressen, stellt er die E-Mail im entsprechenden E-Mail-Verzeichnis des Adressaten zu. Ansonsten übergibt er die Nachricht an den MTA, der dann für deren Weiterleitung sorgt. (MDAs sind z. B. *Procmail*, *Maildrop* und *Cyrus*.)
- **MTA**
  Der Mail Transfer Agent nimmt die E-Mail vom Client entgegen. Er ist für die Zustellung einer Nachricht an den richtigen Zielserver verantwortlich. Hierbei wird normalerweise das Protokoll SMTP verwendet. Auf dem Zielserver übernimmt dann wieder der MDA die Nachrichten und verteilt sie in die entsprechenden E-Mail-Verzeichnisse. (MTAs sind z. B. *Sendmail*, *Postfix*, *Qmail* und *Exim*.)

Für die endgültige Auslieferung einer E-Mail-Nachricht an ein Client-Programm benötigen Sie letztendlich noch mindestens einen weiteren Serverdienst wie IMAP oder POP. Diese werden im dritten Teil dieses Kapitels ebenfalls erläutert. Was die MTAs anbelangt (umgangssprachlich auch als SMTP-Server bezeichnet), sollten Sie sich für die Prüfung insbesondere mit *Postfix*, *Sendmail* und *Exim* befassen. Es gibt Grundeinstellungen, die bei der Konfiguration jedes MTA vorgenommen werden müssen. Dazu zählen z. B. die Konfiguration der eigenen Domänen und Relay-Einstellungen sowie die Angabe eines Smarthosts. Zumindest diese Einstellungen sollten Sie bei einem beliebigen MTA vornehmen können.

Sendmail

Sendmail ist ein MTA, der von dem Programmierer Eric Allman entwickelt wurde. Die Basis hierfür wurde schon in den späten siebziger Jahren des letzten Jahrhunderts mit dem Programm *Delivermail* gelegt.

Konfiguration

Sie finden alle Konfigurationsdateien von *Sendmail* im Verzeichnis */etc/mail*. Dieses Verzeichnis hat typischerweise den folgenden Inhalt:

```
[root@arch-cent6 mail]# ls -l
insgesamt 252
-rw-r--r--. 1 root root   490 29. Okt 19:30 access
-rw-r-----. 1 root root 12288 29. Okt 19:30 access.db
-rw-r--r--. 1 root root     0 29. Okt 14:02 aliasesdb-stamp
-rw-r--r--. 1 root root   233 12. Apr 2007  domaintable
```

```
-rw-r-----. 1 root root 12288 29. Okt 14:02 domaintable.db
-rw-r--r--. 1 root root  5584 11. Nov 2010  helpfile
-rw-r--r--. 1 root root    64 12. Apr 2007  local-host-names
-rw-r--r--. 1 root root   997 12. Apr 2007  mailertable
-rw-r-----. 1 root root 12288 29. Okt 14:02 mailertable.db
-rwxr-xr-x. 1 root root  2700 20. Mai 2009  make
-rw-r--r--. 1 root root    92 20. Mai 2009  Makefile
-rw-r--r--. 1 root root  7251 29. Okt 19:13 sendmail.mc
-rw-r--r--. 1 root root 41521 11. Nov 2010  submit.cf
-rw-r--r--. 1 root root   941 11. Nov 2010  submit.mc
-rw-r--r--. 1 root root   127 12. Apr 2007  trusted-users
-rw-r--r--. 1 root root  1847 12. Apr 2007  virtusertable
-rw-r-----. 1 root root 12288 29. Okt 14:02 virtusertable.db
```

Die Konfiguration ist letztendlich einfacher, als es auf den ersten Blick aussieht. Die Hauptkonfigurationsdatei ist die Datei *sendmail.cf*. Es wird offiziell empfohlen, diese Datei nicht von Hand zu editieren. Stattdessen sollten Sie die nötigen Anpassungen in der Datei *sendmail.mc* vornehmen. Anschließend wird dann mithilfe des Programms `m4` (dem *M4-Makro-Präprozessor*) die Datei *sendmail.cf* aus der Datei *sendmail.mc* generiert, indem Sie dieses Kommando eingeben:

```
[root@arch-cent6 mail]# m4 sendmail.mc > sendmail.cf
```

Prüfungstipp

Die Konfiguration von *Sendmail* ist heutzutage nicht mehr relevant für die LPI-Prüfungen, weil *Sendmail* kaum noch Verwendung findet. Konzentrieren Sie sich deshalb primär auf die Konfiguration von *Postfix*.

Protokollierung

Aktuelle Versionen von *Sendmail* (Version 8.15.x) verwenden nur eine einzige Protokolldatei, nämlich die Datei */var/log/mail.log*. Wie Sie es von anderen Protokollen unter Linux gewohnt sind, ist diese Datei sehr hilfreich bei der Diagnose. Sie sollten sie im Fehlerfall immer heranziehen.

Postfix

Postfix wurde 1998 als Alternative zu *Sendmail* vorgestellt. Der niederländische Entwickler Wietse Zweitze Venema hatte hierbei insbesondere die gegenüber *Sendmail* einfachere Administrierbarkeit, aber auch die absolute Kompatibilität zu den Aufrufen von *Sendmail* zum Ziel. Die vereinfachte Administration dürfte wohl der Hauptgrund für die schnelle Verbreitung von *Postfix* gewesen sein.

Konfiguration

Die Konfigurationsdateien von *Postfix* befinden sich im Verzeichnis */etc/postfix*. Im Vergleich zu *Sendmail* finden Sie hier wesentlich weniger Konfigurationsdateien:

```
root@mailsrv:/etc/postfix# ls -l
total 72
-rw-r--r-- 1 root root   318 2016-11-05 16:52 dynamicmaps.cf
-rw-r--r-- 1 root root  1289 2016-06-03 11:08 main.cf
-rw-r--r-- 1 root root  4300 2016-11-05 16:52 master.cf
-rw-r--r-- 1 root root 18231 2016-09-10 01:55 postfix-files
-rwxr-xr-x 1 root root  7421 2016-09-10 01:55 postfix-script
-rwxr-xr-x 1 root root 22774 2016-09-10 01:55 post-install
drwxr-xr-x 2 root root  4096 2016-09-10 01:55 sasl
```

Die beiden Hauptkonfigurationsdateien heißen *main.cf* und *master.cf*. Sie müssen nach einer Konfigurationsänderung keinen Präprozessor oder anderen Konverter aufrufen. Stattdessen wird eine geänderte Konfiguration einfach neu eingelesen. Führen Sie dazu dieses Kommando aus:

```
root@mailsrv:/etc/postfix# postfix reload
```

Grundlegende Einstellungen, die typischerweise in der Prüfung abgefragt werden, sind hauptsächlich in der Datei *main.cf* zu finden. Das wären im Einzelnen die folgenden Optionen:

myorigin

Mit `myorigin` wird festgelegt, welches Suffix an eine E-Mail angehängt wird, wenn diese direkt vom System aus verschickt wird. Üblicherweise geben Sie dieses Suffix nicht direkt an, sondern legen es in der Datei */etc/mailname* fest. Deshalb wird der Option `myorigin` diese Datei meist einfach als Parameter übergeben:

```
myorigin = /etc/mailname
```

Wenn die Datei */etc/mailname* z. B. lediglich den Eintrag *lpic-2.de* enthält, und der User `root` versendet eine E-Mail, dann wird der Absender automatisch zu *root@lpic-2.de* ergänzt.

mydestination

Die Option `mydestination` sagt dem Server, für welche Ziele er selbst zuständig ist. E-Mails, die an Adressen mit den hier konfigurierten Suffixen gerichtet sind, muss der Mailserver also selbst lokal zustellen. Ein solcher Eintrag könnte z. B. so aussehen:

```
mydestination = lpic-1.de, lpic-2.de, localhost
```

mynetworks

Sie können *Postfix* mitteilen, für welche Client-Computer er als Relay-Server fungieren soll, indem Sie die Option `mynetworks` verwenden. Um den Zugriff nur für das Netzwerk 192.168.50.0/24 und für die lokale Maschine selbst zu erlauben, können Sie diesen Eintrag verwenden:

```
mynetworks = 127.0.0.0/8 192.168.50.0/24
```

Sie sollten in Produktionsnetzwerken allerdings unbedingt eine Authentifizierung für den Zugriff auf SMTP-Server konfigurieren, wenn diese auch E-Mails ins Internet senden können. Böswillige Benutzer könnten den Server sonst missbrauchen, was Ihnen eine Menge Ärger mit Ihrem Internet Service Provider einhandeln könnte.

relayhost

Wenn Sie für die externe E-Mail-Zustellung einen Smarthost im Internet verwenden müssen, dann definieren Sie die Option `relayhost`. Der Verwendungszweck ist identisch mit der Option Smarthost bei *Sendmail*, weshalb hier nicht noch einmal im Detail darauf eingegangen werden soll. Unter der Annahme, dass der SMTP-Relay-Server Ihres Internet Providers unter dem Namen *smtp01.example.net* erreichbar ist, würde der entsprechende Eintrag in der Datei *main.cf* so aussehen:

```
relayhost =  smtp01.example.net
```

TLS-Unterstützung

Heutzutage ist es üblich, die Kommunikation zwischen Mailclients und Mailservern abzusichern. Einerseits sollen Nachrichten verschlüsselt durch das Internet übertragen werden, andererseits soll ein Mailserver dem Client seine Authentizität nachweisen. Der umgekehrte Fall ist ebenfalls konfigurierbar, wird aber wesentlich seltener angewendet. Sie sollten Ihre Benutzer aber dafür sensibilisieren, dass deren E-Mails hierbei lediglich bei der Übertragung geschützt werden, aber weiterhin unverschlüsselt auf den Festplatten der Mailserver gelagert werden.

Für einen ersten Test können Sie TLS ohne Zertifikat verwenden. Die Konfiguration hierfür benötigt lediglich zwei zusätzliche Zeilen in der Datei *main.cf*:

```
smtpd_tls_cert_file=none
smtpd_use_tls=yes
```

In der Datei *master.cf* muss zusätzlich das Kommentarzeichen vor dieser Zeile entfernt werden:

```
smtps     inet  n       -       -       -       -       smtpd
```

Starten Sie *Postfix* anschließend neu. Wenn alles gut gegangen ist, sollte eine Überprüfung zeigen, dass der Computer jetzt den Port 465 abhört:

```
root@debian:/etc/postfix# ss -an | grep :465
LISTEN     0      100                        *:465                      *:*
```

Wenn *TLS* (*Transport Layer Security*) ausschließlich für Verschlüsselung, aber nicht zur Überprüfung der Authentizität eines Servers verwendet wird, benötigen Sie kein Zertifikat von einer öffentlich vertrauten Authentifizierungsstelle. Sie können dann die Schlüssel und Zertifikate verwenden, die mit Ihrer Linux-Distribution ausgeliefert werden. Eine Standardkonfiguration würde dann in der Datei *main.cf* so aussehen:

```
smtpd_tls_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls=yes
smtpd_tls_session_cache_database = btree:${data_directory}/smtpd_scache
smtp_tls_session_cache_database = btree:${data_directory}/smtp_scache
```

In dieser Konfiguration, die z. B. bei Debian und Ubuntu Standard ist, wird ein selbst signiertes Zertifikat verwendet. Für Testzwecke und hausinterne Mail ist das ausreichend. Wenn Mailserver mit Clients über das Internet kommunizieren, sollten allerdings Zertifikate verwendet werden, denen die Clientcomputer vertrauen.

Wie Sie der ersten Zeile der Konfiguration entnehmen können, wird dem Server zunächst ein Zertifikat zugewiesen. Dieses kann er verwenden um sich gegenüber den Clients auszuweisen. In der zweiten Zeile steht der Pfad zum privaten Schlüssel des Servers. Diesen benötigt der Server zur Entschlüsselung der von den Clientcomputern gesendeten Nachrichten. Rechte auf diese Datei sollte natürlich lediglich root und der Mailserver selbst (über eine geeignete Gruppenmitgliedschaft) erhalten.

Um dem Server ein Zertifikat von einer öffentlichen Zertifizierungsstelle zuzuordnen, müssen Sie ein solches Zertifikat zunächst beschaffen. Eine gute Anlaufstelle hierfür ist:

http://www.cacert.org/

Wenn Sie das Zertifikat haben, sollten Sie mithilfe des Kommandos `cat` das ursprüngliche Serverzertifikat, das neue Zertifikat und das Stammzertifikat der ausstellenden CA in eine einzige Datei zusammenführen. Die hier genannte Reihenfolge ist dabei einzuhalten. Beispiel:

```
root@debian:/etc/ssl/certs#  cat ssl-cert-
snakeoil.pem meinZertifikat.pem root.pem > ssl-server.pem
```

Unter der Annahme, dass Sie die Zieldatei *ssl-server.pem* genannt haben, muss die erste Zeile der TLS-Konfiguration wie folgt geändert werden:

```
smtpd_tls_cert_file=/etc/ssl/certs/ssl-server.pem
```

Starten Sie *Postfix* neu, damit die Konfiguration übernommen wird.

Protokollierung

Die Logfiles für *Postfix* werden von *Syslog* verwaltet und bei den gängigen Linux-Distributionen in mehrere Dateien aufgeteilt. Gängig sind hierbei die Dateien *mail.info*, *mail.warn* und *mail.err*, wobei die Dateinamenserweiterungen die jeweiligen Loglevel repräsentieren. Außerdem gibt es die Datei *mail.log*, in der alle E-Mail-Ereignisse zusätzlich aufgezeichnet werden.

Exim

Der MTA *Exim* wurde 1995 von Philip Hazel entwickelt. Genau wie *Postfix* ist *Exim* aufrufkompatibel zu *Sendmail*. *Exim* ist übrigens ein Kürzel für *Experimental Internet Mailer*. Der Name ist heute nicht mehr ganz zutreffend, wenn man bedenkt, wie weit *Exim* inzwischen verbreitet ist.

Konfiguration

Wenn Sie *Exim4* unter Debian einsetzen, dann ist eine Grundkonfiguration für *Exim* dank `dpkg-reconfigure` leicht zu erstellen. Führen Sie dazu folgendes Kommando aus:

```
root@arch-deb-book:/# dpkg-reconfigure exim4-config
```

Ein Assistent führt Sie durch die wichtigsten Grundeinstellungen, die Sie für einen MTA festlegen sollten. Debian verwendet *Exim4* standardmäßig, sodass Sie diese Methode bei Debian-Systemen auch problemlos anwenden können. Die Konfigurationsdateien von *Exim4* finden Sie im Verzeichnis */etc/exim4*:

```
root@arch-deb-book:/# ls -l /etc/exim4/
insgesamt 80
drwxr-xr-x 9 root root         1024  9. Apr 2011  conf.d
-rw-r--r-- 1 root root        76284 31. Jan 2011  exim4.conf.template
-rw-r----- 1 root Debian-exim   205 24. Okt 23:20 passwd.client
-rw-r--r-- 1 root root         1084  9. Mär 22:36 update-exim4.conf.conf
```

Tipp

Auch die Konfiguration von *Exim* wird in der Prüfung nicht mehr im Detail abgefragt. Deshalb wird diesem MTA auch an dieser Stelle nicht mehr Platz eingeräumt als nötig.

Protokollierung

Exim4 verwendet für die Protokollierung ein eigenes Verzeichnis, nämlich */var/log/exim4*. Sie finden hier (mal abgesehen von den üblichen Protokollarchiven) drei Logfiles, nämlich *mainlog*, *paniclog* und *rejectlog*.

Gemeinsamkeiten der MTAs

Da es sich bei SMTP um ein genormtes Protokoll handelt (definiert in RFC 5321), gibt es natürlich auch etliche Gemeinsamkeiten zwischen den oben beschriebenen MTAs. Die (insbesondere für die Prüfung) wichtigsten werden auf den folgenden Seiten näher beschrieben.

Servertest mittels telnet oder netcat

Um die Funktionsweise des SMTP-Datenverkehrs nachvollziehen zu können, ist es möglich eine E-Mail mittels `netcat` oder `telnet` zu versenden. Die Vorgehensweise ist jeweils gleich, egal welches der beiden Programme Sie verwenden. Im Folgenden werden die Benutzereingaben wieder fett hervorgehoben, damit sie von den Antworten des SMTP-Servers unterscheidbar sind. Für das Beispiel wurde *Exim4* als MTA verwendet, Sie können aber auch einen anderen SMTP-Server verwenden.

```
harald@arch-deb-book:~$ netcat localhost 25
220 arch-deb-book ESMTP Exim 4.72 Sun, 04 Mar 2012 10:38:02 +0100
EHLO ausgedachter SMTP.example.net
250-arch-deb-book Hello localhost [127.0.0.1]
250-SIZE 52428800
250-PIPELINING
250 HELP
MAIL FROM: harald@lpic-2.de
250 OK
RCPT TO: harald@lpic-1.de
250 Accepted
DATA
354 Enter message, ending with "." on a line by itself
Subject: Testmail mit Netcat

Das hier ist der Textkoerper. Nachdem der Betreff (Subject:) eingegeben wurde,
 muss eine Leerzeile folgen. Um die Mail abzusenden muessen Sie eine Zeile ers
tellen, die nur einen Punkt enthaelt.
.
250 OK id=1S47tq-0001Oc-T7
```

Hinweis

Sie müssen in der Prüfung die korrekte Syntax von SMTP-Nachrichten zumindest wiedererkennen können.

E-Mail-Aliasse einrichten

Wenn ein Benutzer mehrere E-Mail-Adressen benötigt, könnten Sie auf einem Mailserver natürlich einfach auch mehrere Benutzerkonten für diesen User anlegen. Das hätte dann aber für den Benutzer den Nachteil, dass er von jedem Account seine E-Mails einzeln abholen muss. Das würde die Übersichtlichkeit für den Benutzer erheblich verschlechtern. Deshalb sollten Sie in solchen Fällen Aliasse für den Benutzer einrichten. Hierdurch werden mehrere E-Mail-Adressen mit einem einzigen Benutzerkonto assoziiert, was die Verwaltung, sowohl für den Administrator als auch für den Benutzer, erheblich vereinfacht. Um Aliasse hinzuzufügen, bearbeiten Sie die Datei */etc/aliases*. Wenn Sie z. B. für den Benutzer Maximilian Mustermann einen Alias anlegen wollen, damit er auch unter seinem Spitznamen Max erreichbar ist, fügen Sie diese Zeile hinzu:

```
max:        maximilian
```

Anschließend müssen Sie das Kommando `newaliases` verwenden, damit die Änderung in der Datei */etc/aliases* übernommen wird. Sie können auch E-Mails von einem Benutzerkonto an ein anderes Benutzerkonto weiterleiten. Ein typisches Beispiel hierfür ist die Weiterleitung von E-Mails, die an den Benutzer `root` gesendet werden, an ein normales Benutzerkonto. E-Mails für den Benutzer `root` werden für gewöhnlich selten oder sogar nie abgeholt und man würde wichtige Benachrichtigungen, die das System sendet, verpassen. Bei meinen Systemen sehen diese Einträge einfach so aus:

```
root:       harald
```

Die Verwendung der Datei */etc/aliases* bietet aber noch weitere Funktionen. Wenn z. B. ein Programm Statusmeldungen regelmäßig an eine bestimmte E-Mail-Adresse schickt, dann können Sie diese in eine Datei umleiten. Es muss in diesem Fall noch nicht einmal ein Benutzerkonto als Ziel für diese Mails existieren, sondern lediglich die besagte Datei. Erstellen Sie dann einfach einen Eintrag wie diesen:

```
protokoll:      "/var/log/protokolldatei"
```

Nach der Eingabe von `newaliases` können Sie eine solche Konfiguration einfach testen, indem Sie eine E-Mail an die gerade erstellte Adresse *protokoll@localhost* senden.

Ein weiterer Mechanismus, den Sie in der Datei */etc/aliases* einsetzen können, sind `include`-Einträge. Sie können damit auf einfache Weise unter einer einzigen E-Mail-Adresse mehrere Personen gleichzeitig erreichen. Erstellen Sie z. B. die Textdatei */var/adressliste* und tragen Sie dort jeweils in einer eigenen Zeile die E-Mail-Adressen der zu erreichenden Personen ein. Anschließend erstellen Sie den folgenden Alias:

```
adressliste:        :include:/var/adressliste
```

Nachdem Sie wieder das Kommando `newaliases` angewendet haben, sind alle Personen, die Sie in die Liste eingetragen haben, unter der angelegten Adresse erreichbar.

Ein weiterer Eintragstyp sorgt dafür, dass eingehende E-Mails an ein Programm übergeben werden. Als Verwendungszweck kommen hier hauptsächlich Mailinglisten infrage. *Majordomo* ist ein solches Mailinglistenprogramm. Der Haupteintrag, den man für *Majordomo* erstellt, sieht typischerweise so aus:

```
majordomo:          "|/usr/lib/majordomo/wrapper majordomo"
```

Dieser Eintrag übergibt eingehende E-Mails, die an die Adresse `majordomo` gerichtet sind, an das Programm `wrapper` im Verzeichnis */usr/lib/majordomo*. Der Unterschied zur E-Mail-Weiterleitung in eine Datei ist das Pipe-Zeichen zu Beginn des Umleitungspfades. *Majordomo* verwendet übrigens noch weitere Einträge in der Datei */etc/aliases*. Die hier gezeigte Zeile soll als Beispiel aber ausreichen.

Hinweis

Vergessen Sie nicht die Eingabe des Kommandos `newaliases`, wenn Sie Änderungen an der Datei */etc/aliases* vorgenommen haben.

Weiterleitung

Ein Benutzer kann in seinem eigenen Heimatverzeichnis eine versteckte Datei mit dem Namen *.forward* anlegen bzw. bei Bedarf modifizieren. Diese Datei kann er dazu verwenden, seine eingehenden E-Mail-Nachrichten an eine andere E-Mail-Adresse weiterleiten zu lassen. Zu diesem Zweck muss ausschließlich die Zieladresse in diese Datei eingetragen werden. Das Besondere an dieser Vorgehensweise ist, dass in diesem Fall kein Eingriff eines Administrators in die Datei */etc/aliases* erforderlich wird.

Gemeinsame Kommandos

Sendmail war für viele Jahre *der* MTA, der im Internet fast ausschließlich eingesetzt wurde. Aus Kompatibilitätsgründen und um Administratoren den Umstieg zu erleichtern, wurden die wichtigsten Kommandos, die innerhalb von *Sendmail* verwendet wurden, auf die neuen MTAs portiert. Hierzu gehört z. B. auch das Kommando `newaliases`, das Sie ja bereits kennen.

In der Prüfung werden häufig die Befehle `mailq` und `sendmail-bp` thematisiert, die übrigens beide denselben Verwendungszweck haben, nämlich das Anzeigen der Mailqueue. Da die Verarbeitung von Mails sehr schnell vonstattengeht, werden Sie auf einem Server, der nicht sehr viele Nachrichten verarbeiten muss, immer nur eine leere Queue zu sehen bekommen. Sie können aber durch eine Firewall-Regel den ausgehenden Zugriff auf den TCP-Port 25 sperren. Geben Sie dazu auf Ihrem Testserver folgendes Kommando ein:

```
iptables -t filter -I OUTPUT -p tcp --dport 25 -j DROP
```

Die genaue Verwendung von `iptables` erlernen Sie im nächsten Kapitel. Es versteht sich von selbst, dass Sie ein solches Experiment lieber nicht auf einem Produktionssystem durchführen sollten. Versenden Sie nun ein paar Testnachrichten. Sie können hierbei ebenfalls die Kommandozeile in Anspruch nehmen, indem Sie z. B. ein solches Kommando verwenden:

```
echo test | mail -s "Testmail" info@example.org
```

Sie können natürlich auch mit `telnet` oder `netcat` experimentieren. Geben Sie anschließend den Inhalt der Mailqueue aus:

```
root@arch-deb-book:/# mailq
.Om   1.4K 1S3t7H-0001Od-18 <> *** frozen ***
          test@example.org

 Om    338 1S4AL1-0001Zq-KR <root@arch-deb-book>
          info@example.org

 Om    339 1S4ALE-0001Zv-A7 <root@arch-deb-book>
          info@example.net
```

Sie werden feststellen, dass das Kommando `sendmail -bp` exakt dasselbe Ergebnis an den Tag fördert. Im vorliegenden Fall befindet sich eine verdächtige E-Mail in der Queue, die keinen Absender enthält und deshalb von *Exim* eingefroren wurde. Die anderen beiden Mails sind unverdächtig und warten auf ihre Zustellung.

Vergessen Sie nicht, nach Abschluss Ihres Experiments die Blockierung des SMTP-Ports 25 wieder aufzuheben:

```
iptables -t filter -D OUTPUT -p tcp --dport 25 -j DROP
```

Warteschlangen (Queues)

Alle MTAs verwenden Warteschlangen, weil Mailserver nach dem Store-and-Forward-Prinzip arbeiten. Diese Warteschlangen werden als *Mailqueues* bezeichnet. In diesen Queues werden Nachrichten abgespeichert, die auf ihre Auslieferung oder

Weiterleitung warten. Sie sollten die entsprechenden Verzeichnisse der Warteschlangen für die drei prüfungsrelevanten MTAs kennen:

- Postfix: */var/spool/postfix/**
- Sendmail: */var/spool/mqueue*
- Exim4: */var/spool/exim4/**

In den genannten Mailqueues finden Sie ausschließlich Daten zu Nachrichten, die noch nicht ausgeliefert wurden. Zugestellte E-Mails werden, unabhängig vom verwendeten MTA, üblicherweise in Dateien abgelegt, die dem jeweiligen Benutzernamen entsprechen, für den eine E-Mail bestimmt ist. Sie finden diese Dateien in */var/spool/mail*. Es ist üblich, dieses Verzeichnis nach */var/mail* zu verlinken.

211.2 Verwalten der Mailauslieferung

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Software zur Filterung, Sortierung und Überwachung des eingehenden E-Mail-Verkehrs zu implementieren.

Wichtigste Wissensgebiete:

- Verständnis der *Sieve*-Funktionalität,-Syntax und -Operatoren
- *Sieve* unter Berücksichtigung von Absender, Empfänger, Mail-Header und Größe zur Sortierung und zur Filterung von Mail verwenden
- Kenntnis von *procmail*

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- Bedingungen und Vergleichsoperatoren
- `keep`, `fileinto`, `redirect`, `reject`, `discard`, `stop`
- Dovecot Vacation-Erweiterung

Allgemeines

Für die lokale E-Mail-Zustellung benötigen Sie einen Mail Delivery Agent (MDA). Hierfür kommen *Sieve* oder *Procmail* in Frage. Der MDA entscheidet darüber, wie mit einer E-Mail-Nachricht weiter zu verfahren ist. Wenn eine Ziel-Adresse als nicht lokal identifiziert wird, übergibt der MDA die E-Mail an den MTA, der sich um die weitere Verarbeitung kümmert. Es ist aber auch möglich, dass ein im Urlaub befindlicher Benutzer eine Weiterleitung zu einer anderen E-Mail-Adresse konfiguriert hat. Auch dann wird die E-Mail mit der neuen Adresse an den MTA übergeben. Weitere Aktionen, die *Sieve* oder *Procmail* durchführen können, sind:

- Löschen von Nachrichten (z. B. zwecks Spam-Bekämpfung)
- Speichern von Nachrichten in Dateien
- Versenden von automatischen Antworten (z. B. Benachrichtigungen über Abwesenheit)

Tipp

Dem Umstand, dass *Sieve* gegenüber *Procmail* inzwischen wesentlich weiter verbreitet ist, wird auch in der Prüfung Rechnung getragen. Deshalb sind genaue Kenntnisse über *Procmail* nicht mehr erforderlich.

Konfiguration von Sieve

Wenn Sie zu Testzwecken bereits einen Mailserver betreiben, auf dem *Dovecot* läuft, können Sie die Konfiguration von *Sieve* wie hier beschrieben nachvollziehen. Sollte dem nicht so sein, machen Sie bitte einen Exkurs zum Abschnitt »Verwalten des Zugriffs auf Mail«. Dieses Buch hält sich bezüglich der Themenabfolge an die Vorgaben der LPI-Topics, was an dieser Stelle ausnahmsweise ungünstig ist.

Davon ausgehend, dass *Dovecot* bereits läuft, können Sie nun die für *Sieve* benötigten Pakete installieren. Je nach verwendeter Distribution können Sie dafür eines der folgenden Kommandos verwenden:

```
apt-get install dovecot-sieve
yum install dovecot-pigeonhole
pacman -S  pigeonhole
```

Hierbei werden an Ihrer *Dovecot*-Konfiguration erste Änderungen vorgenommen. Eine Überprüfung zeigt Folgendes:

```
root@archangel:/# dovecot -n
# 2.2.13: /etc/dovecot/dovecot.conf
# OS: Linux 4.6.3 x86_64 Debian 8.5
plugin {
  sieve = ~/.dovecot.sieve
  sieve_dir = ~/sieve
}
protocols = " imap pop3"
ssl = yes
userdb {
  driver = passwd
}
```

Die meisten nicht relevanten Zeilen wurden aus Platzgründen entfernt. Sie sehen, dass *Sieve* als Plugin in *Dovecot* integriert wurde. Außerdem sind zwei relative Pfade angegeben.

- `sieve = ~/.dovecot.sieve` ist ein Verweis auf eine Skript-Datei, die unterhalb jedes Heimatverzeichnisses eines Benutzers ausgewertet wird, falls vorhanden. Es ist die Hauptskriptdatei des Users.
- `sieve_dir = ~/sieve` ist ein Verweis auf ein Verzeichnis, in dem ein Benutzer eigene Skripte ablegen kann. Hoffentlich benutzerfreundliche Frontends (*ManageSieve*) sind bereits in der Entwicklung und auch schon verfügbar.

Sie können auch einen zentralen Speicherort definieren, damit Skripte für alle Benutzer zentral gepflegt werden können. Solche Einstellungen werden in der Datei *90-sieve.conf* konfiguriert.

Damit *Sieve* überhaupt etwas macht, sind noch einige Änderungen erforderlich. Welche das sind, hängt von der verwendeten Linux-Distribution und -Version ab. Eine gute Dokumentation finden sie hier:

http://wiki.dovecot.org/Pigeonhole

Gängige Gründe für Nichtfunktion sind:

In der Datei *15-lda.conf* fehlt der Eintrag für das Plugin. Das sollte so aussehen:

```
protocol lda {
mail_plugins = $mail_plugins sieve
}
```

Sinngemäß müsste ein solcher Eintrag ggf. in der Datei *20-lmtp.conf* stehen, falls dieses Protokoll zum Einsatz kommt:

```
protocol lmtp {
mail_plugins = $mail_plugins sieve
}
```

In der Postfixkonfiguration wird oft auf *Procmail* gesetzt. Deshalb muss in der Datei *main.cf* ebenfalls eine Anpassung vorgenommen werden:

```
mailbox_command = /usr/lib/dovecot/deliver
```

Sieve-Skripte

Um die für die Prüfung interessanten Skriptbeispiele ausprobieren zu können reicht es aus, wenn Sie Ihre eigene `~/.dovecot.sieve`-Datei bearbeiten. Sie können diese Datei einfach direkt in Ihrem Heimatverzeichnis erstellen und bearbeiten. Änderungen an dieser Datei sind sofort wirksam. Sie müssen also keinen Dienst neu starten

oder ähnliches. Ich empfehle Ihnen immer `journalctl -f` auf einer Konsole laufen zu lassen, damit Sie sehen können, was gerade geschieht. Die folgenden Beispiele werden Ihnen helfen, den Aufbau der Skripte zu verstehen.

```
require "fileinto";
if header :contains "X-Spam-Flag" "YES"  {
  fileinto "INBOX/Spam";
}
```

Die erste Zeile dieses Skripts teilt dem Interpreter mit, dass das Kommando `fileinto` in diesem Skript verwendet wird. In der zweiten Zeile wird eine Bedingung angegeben, unter der das eigentliche Kommando, das in der dritten Zeile steht, ausgeführt wird. Die Bedingung ist also, dass das `X-Spam-Flag` im Header einer Mail auf `Yes` gesetzt ist. Das wird üblicherweise durch SpamAssassin erledigt. Das ausgeführte Kommando verschiebt eine Mail, auf die diese Bedingung zutrifft, in den IMAP-Ordner `INBOX/Spam`.

Hinweis

Der Zielordner wird hier so angegeben, wie ein IMAP-Client ihn »sieht«. Sie werden im Internet Beispiele finden, bei denen die Syntax anders aussieht, aber bei einem aktuellen *Dovecot* sollte es so funktionieren, wie hier beschrieben.

Das folgende Skript leitet E-Mails, deren Betreff das Wort »Bewerbung« enthalten, an eine andere Mailadresse weiter:

```
if header :contains "subject" ["Bewerbung", "application"] {
  redirect "personalverwaltung@mycompany.com";
}
```

Anstatt Spam-Mails in einen speziellen Ordner zu sortieren, können solche Mails auch gleich verworfen werden. Dieses Vorgehen ist natürlich mit Vorsicht zu genießen, weil manchmal Mails fälschlicherweise als Spam klassifiziert werden. Wenn Sie es dennoch so machen wollen, verwenden Sie diese Zeilen:

```
if header :contains "X-Spam-Level" "**********" {
  discard;
  stop;
}
```

Die Nachricht wird dann gelöscht und der Absender erhält keine Benachrichtigung. Die weitere Bearbeitung der Nachricht wird gestoppt. Wenn Sie möchten, dass der Absender einen Nachricht darüber informiert wird, dass seine Mail ausgesiebt wurde, dann verwenden Sie anstatt `discard` das Kommando `reject`.

Sie können in Skripten auch das Kommando keep verwenden. Eine Mail, auf die die gestellte Bedingung zutrifft, würde dann in der Inbox behalten werden. Am Ende einer Regelkette steht immer implizit keep.

Tipp

Sie können mit *Sieve* sehr komplexe Regeln aufstellen. Für die Prüfung sollten Sie zumindest die in den Beispielen aufgeführten Kommandos kennen.

Vacation-Erweiterung für Dovecot

Die *Vacation-Erweiterung* bietet eine Möglichkeit, eingehende E-Mails automatisch zu beantworten. Damit Absender, die Ihnen häufiger schreiben nicht jedes Mal eine Abwesenheitsbenachrichtigung erhalten, können Sie Intervalle festlegen, wann frühestens eine weitere Benachrichtigung an denselben Absender erfolgen soll. Hier sehen Sie eine Beispielkonfiguration:

```
require ["vacation"];
vacation
  # Antworte einem Absender maximal einmal täglich
  :days 1
  :subject "Grüße aus dem Urlaub"
"Leider bin ich gerade im Urlaub. Wenden Sie sich bitte in der Zwischenzeit an
 meine Kollegen.
Harald Maaßen";
```

211.3 Verwalten des Zugriffs auf Mail

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, POP- und IMAP-Daemons zu installieren und zu konfigurieren.

Wichtigste Wissensgebiete:

- *Dovecot IMAP* und *POP3* konfigurieren und administrieren
- *TLS* Grundkonfiguration für *Dovecot*
- Kenntnis von Courier

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/dovecot/*
- *dovecot.conf*

- `doveconf`
- `doveadm`

Allgemeines

Auf den letzten Seiten ging es bisher nur darum, E-Mails von einem Computer zu einem anderen zu transferieren, zu filtern und zu sortieren. Nun fehlt nur noch ein Dienst, der es den Benutzern ermöglicht, ihre E-Mails abzuholen. Für diese Aufgabe kommen heutzutage eigentlich nur der POP3- oder IMAP4-Server infrage. *POP* ist das *Post Office Protocol* und es verwendet den TCP-Port 110. *IMAP* ist das *Internet Message Access Protocol* und verwendet den TCP-Port 143. Wenn ein Client E-Mails von einem Mailserver via POP abholt, dann wird die E-Mail im Normalfall vom Server gelöscht. IMAP-Server behalten Kopien der E-Mail-Nachrichten, sodass beim Ausfall von Client-Computern kein Verlust von E-Mails zu befürchten ist. Weitere Vorteile von IMAP sind zum einen die Möglichkeit, Verzeichnisse auf dem Mailserver anzulegen, und so eine serverseitige Sortierung zu ermöglichen; zum anderen werden sogar gesendete Nachrichten auf den IMAP-Mailserver hochgeladen, sodass auch diese zentral gesichert werden können. Es gibt mehrere Softwareprodukte, die sowohl POP3 als auch IMAP unterstützen. Für Linux und Unix sind das:

- *UW-IMAP* gilt als die Referenzimplementierung von IMAP und wurde an der University of Washington entwickelt, wie der Name schon vermuten lässt. Unterstützt werden POP, IMAP, NNTP und SMTP.
- *Dovecot* ist ein sehr flexibler Server für IMAP und POP. Er unterstützt Mbox und Maildir, verfügt aber auch zusätzlich über ein eigenes Format, nämlich Dbox. Dovecot ist vollständig kompatibel zu UW-IMAP und Courier.
- *Courier* ist primär ein MTA. Die Implementierungen von POP und IMAP unterstützen die Formate Maildir und Maildir++. Courier bringt einen eigenen MDA, nämlich Maildrop mit. Die Administration von Courier ist auch über eine Webschnittstelle möglich.
- *Cyrus* ist ein reiner IMAP- und POP-Server, der an der Carnegie Mellon University entwickelt wurde.

Auf den bedeutendsten Mailservern im Sinne von Postfachservern finden Sie heutzutage *Courier* oder *Dovecot* vor.

Tipp

Auch *Courier* verliert gegenüber *Dovecot* immer mehr an Bedeutung und dem wird in der Prüfung Rechnung getragen. Detaillierte Kenntnisse sind deshalb nur noch für *Dovecot* erforderlich.

Dovecot-Mailserver

Konfiguration

Wenn Sie eine paketbasierte Installation von *Dovecot* durchgeführt haben, verfügt Ihr Server meist über eine brauchbare Grundkonfiguration. Diese finden Sie im Verzeichnis */etc/dovecot* wieder. Die Hauptkonfigurationsdatei heißt *dovecot.conf*. Sie können die aktuelle Konfiguration des Servers ohne Kommentarzeilen anzeigen, indem Sie einfach folgendes Kommando ausführen:

```
root@mailsrv:~# doveroot@archangel:/# dovecot -n
# 2.2.13: /etc/dovecot/dovecot.conf
# OS: Linux 4.6.3 x86_64 Debian 8.5
mail_location = mbox:~/mail:INBOX=/var/mail/%u
namespace inbox {
  inbox = yes
  location =
  mailbox Drafts {
    special_use = \Drafts
  }
  mailbox Junk {
    special_use = \Junk
  }
  mailbox Sent {
    special_use = \Sent
  }
  mailbox "Sent Messages" {
    special_use = \Sent
  }
  mailbox Trash {
    special_use = \Trash
  }
  prefix =
}
passdb {
  driver = pam
}
plugin {
  sieve = ~/.dovecot.sieve
  sieve_dir = ~/sieve
}
protocols = " imap pop3"
ssl_cert = </etc/ssl/certs/dovecot.pem
ssl_key = </etc/ssl/private/dovecot.pem
```

```
userdb {
  driver = passwd
}
```

Diese Funktion ist sehr nützlich, wenn man sich einen Überblick über die Konfiguration verschaffen will, ohne die Datei *dovecot.conf* zunächst von Kommentarzeilen befreien zu müssen. Bei den ausgegebenen Zeilen handelt es sich nur um die Optionen, die nicht per Default festgelegt sind. Wenn Sie die komplette Konfiguration inklusive Default-Einstellungen sehen wollen, verwenden Sie stattdessen das Kommando `dovecot -a`.

Der obigen Konfiguration kann man entnehmen, dass dieser Server sowohl POP- als auch IMAP-Dienste zur Verfügung stellt. Welche Protokolle verwendet werden sollen, konfigurieren Sie in dieser Zeile:

```
protocols = " imap pop3"
```

Voraussetzung ist natürlich, dass die entsprechenden Pakete installiert sind. Das wären z. B. auf Debian-basierten Systemen *dovecot-common*, *dovecot-imapd* und *dovecot-pop3d*.

Wichtig sind natürlich auch die Einstellungen bezüglich der Authentifizierung. In diesem Fall ist eine Anmeldung auch mit unverschlüsseltem Benutzernamen und Passwort möglich:

```
disable_plaintext_auth: no
```

Hierbei handelt es sich übrigens um die Standardeinstellung. Im unteren Bereich der Konfigurationsdatei wird festgelegt, welche Authentifizierungsmechanismen *Dovecot* verwendet. In der Standardeinstellung verwendet *Dovecot PAM* mit lokalen Benutzerkonten, die in der Datei */etc/passwd* abgelegt sind.

```
auth default:
  passdb:
    driver: pam
  userdb:
    driver: passwd
```

Alternativ können Sie Benutzerkonten in einer Textdatei hinterlegen oder auf eine LDAP-Datenbank zugreifen.

Die Konfiguration von *Dovecot* findet sich nicht vollständig in der Hauptkonfigurationsdatei *dovecot.conf* wieder. Viele Einstellungen werden in spezialisierten Konfigurationsdateien vorgenommen, die sich im Verzeichnis */etc/dovecot/conf.d* befinden. Ein in jedem Fall prüfungsrelevantes Beispiel ist der Inhalt der Datei *10-ssl.conf*.

Wenn Sie einen Server mit SSL/TLS absichern wollen, müssen hier entsprechend Einträge vorgenommen werden:

```
ssl = yes
```

Um die Verwendung von SSL zu erzwingen, können Sie anstatt `yes` das Argument `required` verwenden, weil SSL ansonsten nur optional verwendbar wäre. Eine Alternative finden Sie in der Datei *10-auth.conf*. In dieser Datei können Sie Klartextauthentifizierung durch folgenden Eintrag einfach ausschalten:

```
disable_plaintext_auth = yes
```

Da für die gesicherte Übertragung ein Zertifikat und ein privater Schlüssel erforderlich sind, müssen diese ebenfalls angegeben werden. Das könnte im einfachsten Fall so aussehen:

```
ssl_cert = </etc/ssl/certs/dovecot.pem
ssl_key = </etc/ssl/private/dovecot.pem
```

In vielen Distributionen sind hier in der Voreinstellung falsche Pfade angegeben worden. Hierbei unterstelle ich allerdings gut gemeinte Absicht, denn auch dieser Sicherheitsmechanismus ist wertlos, wenn hier lediglich selbst signierte Zertifikate zum Einsatz kommen. Diese Problematik ist Ihnen aber bereits von diversen anderen Serverdiensten her bekannt.

Auch bei *Dovecot* können Sie die Verwendung von Zertifikaten auf Seite der Clients verlangen. Eine solche Konfiguration ist aber nicht Gegenstand der LPI-Zertifizierung.

Die Werkzeuge doveconf und doveadm

Bei `doveconf` handelt es sich um ein Werkzeug, mit dem Sie die Konfiguration von *Dovecot* prüfen können. Es gibt einige Parallelen zum Kommando `dovecot`. So liefern beide Kommandos, wenn Sie jeweils die Option `-n` oder `-a` verwenden, dasselbe Ergebnis. Der Unterschied liegt allerdings darin, dass Sie das Kommando `dovecot` auch zur Steuerung des Servers verwenden können, während `doveconf` lediglich die Konfiguration analysiert. Wenn eine Konfiguration fehlerhaft ist, sind folgende Optionen von `doveconf` besonders hilfreich:

- `-d` zeigt an, wie die Standardeinstellungen wären.
- `-n` zeigt nur Einstellungen an, die nicht Standard sind.
- `-N` zeigt Einstellungen an, die nicht Standard sind, und explizit eingestellte Werte.

Hieraus ergeben sich gute Vergleiche zur Fehlerdiagnose.

Das Werkzeug `doveadm` dient der Administration des Servers. Da es direkt mit dem Prozess `dovecot` kommuniziert, können Sie den Server mithilfe dieses Tools zwar die Konfiguration neu einlesen lassen (`reload`) und ihn stoppen (`stop`) aber nicht starten, während er nicht läuft. Die Möglichkeiten, die dieses Werkzeug bietet, sind umfangreich und würden den Rahmen dieses Kapitels sprengen. Es ist aber wie immer eine gute Idee, einen Blick in die entsprechende Manpage zu werfen und ein paar Kommandos auszuprobieren.

Courier-Mailserver

Courier ist ein vollwertiger Mailserver inklusive MTA. Sie können aber auch *Postfix* als MTA einsetzen und von *Courier* lediglich die POP- und IMAP-Komponenten verwenden. Was den MDA anbelangt, können Sie wahlweise *Maildrop* oder *Procmail* verwenden. Es gibt allerdings eine speziell angepasste *Maildrop*-Version, die für *Courier* geschrieben wurde. Wenn Sie die entsprechende Komponente installieren, können Sie den Server auch über ein Webfrontend konfigurieren.

Tipp

Die Konfiguration von *Courier* ist für die LPI-Prüfung nicht mehr von Belang. Konzentrieren Sie sich an dieser Stelle unbedingt auf *Dovecot*.

212 Systemsicherheit

Sicherheit ist ein wichtiger Aspekt bei der Konfiguration von Netzwerken. Hierbei geht es vor allem um die Vertraulichkeit, Integrität und Authentizität von Daten, während sie das Netzwerk passieren, und um das Fernhalten von Eindringlingen.

212.1 Router-Konfiguration

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, ein System zur Weiterleitung von IP-Paketen und für Network Address Translation (NAT, IP-Masquerading) zu konfigurieren und dessen Bedeutung für die Netzwerkabsicherung kennen. Dieses Lernziel umfasst die Portweiterleitung, das Aufstellen von Netzwerkfilterregeln und die Abwehr von Angriffen.

Wichtigste Wissensgebiete:

- `iptables`- und `ip6tables`-Konfigurationsdateien, -Begriffe und -Dienstprogramme
- Werkzeuge, Befehle und Dienstprogramme zur Verwaltung von Routing-Tabellen
- private IP-Adressbereiche (IPv4), *Unique Local Addresses* und *Link Local Addresses* (IPv6)
- Port- und IP-Weiterleitung
- Auflisten und Erstellen von Filterregeln, die Datenpakete akzeptieren oder blockieren, abhängig von Protokoll, Quell- oder Zielport und -adresse
- Sichern und Wiederherstellen von Filterkonfigurationen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */proc/sys/net/ipv4*
- */proc/sys/net/ipv6*
- */etc/services*
- `iptables`
- `ip6tables`

Allgemeines

Ein Router hat die Aufgabe, mehrere Netzwerksegmente miteinander zu verbinden. Da er auf Layer 3 des Osi-Modells operiert, ist er für die Zustellung von Paketen auf der Basis von IP-Adressen zuständig. Das beinhaltet natürlich auch die Weiterleitung von Daten aus dem oder in das Internet. Als Router kommen nicht nur hostbasierte Systeme infrage, auf denen Linux oder ein anderes Betriebssystem läuft, sondern in der Praxis kommen hier eher spezialisierte Geräte zum Einsatz. Bei einem hostbasierten Router spricht man auch von einem Software-Router. Die Geräte, die ausschließlich für Routing-Zwecke eingesetzt werden können, bezeichnet man als Hardware-Router. Ob Sie einen Linux-PC oder eher einen Hardware-Router einsetzen sollten, hängt von den jeweiligen Anforderungen ab. Hier ein paar Überlegungen:

- Hardware-Router können keine Aufgaben durchführen, für die sie nicht entwickelt wurden.
- Software-Router verbrauchen bei gleichem Aufgabengebiet meist erheblich mehr Strom.
- Software-Router sind anfälliger für Hardwaredefekte, weil es hier Komponenten gibt, die ein Router nicht zwingend benötigt (Grafikkarte, Soundkarte, Festplatte).
- Software-Router können bei plötzlichem Bedarf leicht für zusätzliche Aufgaben verwendet werden (z. B. als DNS-Server oder Fileserver).

Die Bearbeitung von Routing-Tabellen wurde in Kapitel 20S, »Netzwerkkonfiguration«, bereits beschrieben. Auf den folgenden Seiten geht es nun um die Absicherung eines hostbasierten Routers.

/proc/sys/net/ipv4

Bevor es an die Firewall-Konfiguration mittels `iptables` geht, sollten Sie noch zwei Einstellungen am */proc*-Dateisystem vornehmen. Damit der Router Pakete weiterleitet, muss ihm mitgeteilt werden, dass er das überhaupt soll. Zu diesem Zweck können Sie die entsprechende Datei im */proc*-Device wie folgt ändern:

```
root@arch-deb-book:/# echo 1 > /proc/sys/net/ipv4/ip_forward
```

Standardmäßig steht in dieser Datei eine Null, sodass ein Computer nicht als Router fungiert. Da es sich bei einem Router um ein wichtiges Element einer Netzwerkinfrastruktur handelt, ist er ein beliebtes Ziel für Angriffe. Um sehr einfache Portscanner, die zunächst lediglich ein ICMP-Paket senden, in dem Glauben zu lassen, der Router würde gar nicht existieren, können Sie die Beantwortung von ICMP-Echo-Anforderungen ausschalten, indem Sie in die folgende Pseudodatei des */proc*-Dateisystems ebenfalls eine Eins schreiben: */proc/sys/net/ipv4/icmp_echo_ignore_all*

Es sollte Ihnen aber klar sein, dass Sie mit dieser Maßnahme bestenfalls Skriptkiddies fernhalten und außerdem eine Diagnosemöglichkeit abschalten.

Um die oben genannten Konfigurationen dauerhaft festzulegen, sollten Sie entsprechende Einträge in der Datei */etc/sysctl.conf* vornehmen. Diese Einträge sehen so aus:

```
net.ipv4.ip_forward = 1
net.ipv4.icmp_echo_ignore_all = 1
```

/proc/sys/net/ipv6

Auch für IPv6 gibt es eine eigene Sektion im */proc*-Dateisystem. Im Verzeichnis */proc/sys/net/ipv6* finden Sie zunächst einige allgemeine Einstellungen, die sich auf alle Schnittstellen gleichermaßen beziehen. Etwas spezifischer wird es, wenn man sich die Inhalte der Unterverzeichnisse von *ipv6/conf* ansieht. Unterhalb von *conf* gibt es verschiedene Unterverzeichnisse. Für ein System mit nur einer Netzwerkschnittstelle könnten z. B. diese Verzeichnisse existieren:

- `default` für Voreinstellungen, die alle Netzwerkschnittstellen betreffen
- `all` zur Konfiguration aller Netzwerkschnittstellen
- `eth0` zur Konfiguration der Schnittstelle eth0
- `lo` zur Konfiguration des Loopbackadapters

Innerhalb dieser Verzeichnisse finden Sie identische Konfigurationsdateien für den jeweiligen Geltungsbereich, z. B.:

- `autoconf` aktiviert die automatische Konfiguration durch *Routeradvertising*
- `forwarding` legt das Routingverhalten fest
- `mtu` maximum tranfer unit
- `disable_ipv6` deaktiviert das Protokoll IPv6

/etc/services

Die Datei */etc/services* hat auf den ersten Blick gar nichts mit der Absicherung eines Routers zu tun. Sie enthält lediglich eine Zuordnung von Netzwerkdiensten zu den jeweils zugehörigen Protokollen und Ports. Sie können allerdings, basierend auf den Zuordnungen in dieser Datei, Firewall-Regeln mit `iptables` erstellen. Die folgenden beiden Kommandos entsprechen sich deshalb:

```
iptables -A INPUT -p tcp --dport ssh -j DROP
iptables -A INPUT -p tcp --dport 22 -j DROP
```

In beiden Fällen werden eingehende Verbindungen per SSH verworfen. Das erste Kommando bezieht sich hierbei auf den Netzwerkdienst, der in der Datei */etc/services* definiert ist. Der zweite Befehl verwendet einfach den entsprechenden Port 22. Die genaue Verwendung von `iptables` wird auf den folgenden Seiten beschrieben. Sehen Sie hier einen kurzen Auszug aus der Datei */etc/services*. Die Zeilen sind gezielt ausgewählt worden und enthalten Zuordnungen, die jeder Administrator unbedingt kennen sollte:

```
ftp-data    20/tcp
ftp         21/tcp
fsp         21/udp  fspd
ssh         22/tcp          # SSH Remote Login Protocol
ssh         22/udp
telnet      23/tcp
smtp        25/tcp  mail
domain      53/tcp          # name-domain server
domain      53/udp
bootps      67/tcp          # BOOTP server
bootps      67/udp
bootpc      68/tcp          # BOOTP client
bootpc      68/udp
www         80/tcp   http   # WorldWideWeb HTTP
www         80/udp          # HyperText Transfer Protocol
pop3        110/tcp  pop-3  # POP version 3
pop3        110/udp  pop-3
imap2       143/tcp  imap   # Interim Mail Access P 2 and 4
imap2       143/udp  imap
https       443/tcp         # http protocol over TLS/SSL
https       443/udp
```

Private Netze

Was die Verwendung privater (nicht öffentlicher) IP-Adressen anbelangt, sollten Sie eigentlich bestens im Bilde sein. Da diese Adressen allerdings als prüfungsrelevant zu betrachten sind, folgt hier nochmal eine kurze Auflistung.

Für IPv4 wären das:

- 10.0.0.0/8 (10.0.0.0 bis 10.255.255.255.255)
- 172.16.0.0/12 (172.16.0.0 bis 172.31.255.255)
- 192.168.0.0/16 (192.168.0.0 bis 192.168.255.255)

Nennenswerte Sonderfälle sind das Loopbacknetzwerk und APIPA-Adressen. Das Loopbacknetzwerk mit der Adresse 127.0.0.0/8 dient der internen Kommunikation.

In der Regel ist lediglich eine Adresse, nämlich 127.0.0.1, für einen einzelnen Loopbackadapter konfiguriert.

APIPA (Automatic Private IP Addressing) wird zur automatischen Konfiguration von Netzwerkschnittstellen verwendet, wenn diese keinen DHCP-Server erreichen können. Es wird dann automatisch eine Adresse aus dem Netzwerk 169.254.0.0/16 an den entsprechenden Adapter gebunden.

Die Verwendung von privaten Adressen sieht bei IPv6 etwas anders aus, auch wenn immer wieder versucht wird, Vergleiche mit der IPv4-Welt anzustellen. Sie werden heutzutage bei fast jedem Computer eine *Link-lokale IPv6-Adresse* vorfinden. Sie erkennen diese Adressen sehr leicht, weil sie immer den Präfix fe80:: aufweisen. Diese Adressen können nur innerhalb eines Netzwerksegments verwendet werden, also nicht über Router hinweg. Die Konfiguration dieser Adressen geschieht automatisch.

Zusätzlich können private Adressen konfiguriert werden, so dass Computer auch mit Systemen in anderen Netzwerk-Segmenten, nicht aber direkt mit dem Internet kommunizieren können. Hierfür kommen *Unique Local Unicast*-Adressen in Frage. Sie weisen den Präfix fc00::/7 auf, reichen also von fc00 bis fdff.

Einen ähnlichen Verwendungszweck hatten *Site Local Unicast*-Adressen. Diese werden aber nicht mehr verwendet und sind auch für die Prüfung nicht von Belang.

iptables

`iptables` ist ein Programm, das zur Konfiguration der in den Linux-Kernel integrierten Firewall verwendet wird. Es gibt einige Frontends, welche die Konfiguration vereinfachen sollen. Diese setzen am Ende alle auf `iptables` auf, und wenn Sie mit `iptables` umgehen können, sind Sie in der Lage, Firewalls auf beliebigen Linux-Distributionen zu administrieren. Das Programm `iptables` kann nur für IPv4 verwendet werden. Die Firewall-Konfiguration für IPv6 wird mit `ip6tables` durchgeführt. `iptables` ist in Tabellen organisiert. Diese Tabellen (tables) bestehen aus sogenannten Ketten (chains) und Regeln (rules). Das folgende Beispiel gibt einen ersten Einblick in den Aufbau:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o ppp0 -j MASQUERADE
```

Kurz vorab gesagt, erlaubt dieses Kommando z. B. die gemeinsame Nutzung eines Internetzugangs, der über die Schnittstelle `eth0` erreichbar ist, für Computer, die sich im Netzwerksegment `192.168.0.0/24` befinden. Aber nun im Detail:

- `-t nat` wählt die Tabelle `nat` aus. Es gibt außerdem die Tabellen `filter` und `mangle` (dazu später mehr).

- `-A POSTROUTING` hängt die Regel (append) an die Kette (chain) `POSTROUTING` an. Standardmäßig sind weiterhin die Ketten `PREROUTING`, `INPUT`, `OUTPUT` und `FORWARD` vorhanden (auch hierzu später mehr).
- `-s` gibt die Quelle (source) an, von der die Pakete stammen müssen, damit die Regel zutrifft (match).
- `-o` gibt die Schnittstelle an, über die ein zutreffendes Paket das System wieder verlässt (out-interface).
- `-j MASQUERADE` springt (jump) zum Ziel (target) `MASQUERADE`, das eine auf dynamische IP-Adressen spezialisierte Form von NAT an der äußeren Netzwerkschnittstelle durchführt.

Lassen Sie sich durch die Vielzahl der Optionen und Elemente nicht abschrecken. Mit etwas Übung ist `iptables` einfacher zu administrieren, als es auf den ersten Blick aussieht.

Wenn man das Beispiel noch einmal heranzieht, kann man die einzelnen Elemente noch genauer beleuchten.

Tabellen (Option -t nat im Beispiel)

Das erste Element, das in einem `iptables`-Kommando angegeben wird, ist die zu bearbeitende Tabelle. Die wichtigsten Tabellen wurden bereits genannt, nämlich `filter`, `nat` und `mangle`. Wenn in einem Kommando keine Tabelle angegeben wird, wird standardmäßig die Tabelle `filter` bearbeitet.

- `filter` enthält die Ketten `INPUT`, `OUTPUT` und `FORWARD`. Diese Tabelle enthält diejenigen Regeln, die für das Blockieren oder Zulassen von Paketen verwendet werden.
- `nat` enthält die Ketten `PREROUTING`, `POSTROUTING` und `OUTPUT` und ist für Network Address Translation zuständig, wie der Name schon erahnen lässt. Das beinhaltet z. B. IP-Masquerading, IP-Forwarding und Port-Forwarding.
- `mangle` enthält alle standardmäßig vorhandenen Ketten, also `INPUT`, `OUTPUT`, `FORWARD`, `PREROUTING` und `POSTROUTING`. Diese Tabelle enthält Regeln zur Änderung von Paketen, während diese die jeweils verwendete Kette durchlaufen.

Sie können Tabellen wahlweise mit der Option `-t` oder `--table` angeben.

Kommandos (-A im Beispiel)

Das nächste Element ist das Kommando. Im Beispiel sorgt das Kommando `-A` dafür, dass die neue Regel hinten an die bestehende Kette angehängt wird. Wichtige Kommandos sind:

- `-A` bzw. `--append` hängt eine Regel an die angegebene Kette an.
- `-I` bzw. `--insert` fügt die Regel an einer numerisch angegebenen Position ein. Wenn keine Regelnummer angegeben wird, dann wird die neue Regel zur ersten Regel.
- `-D` bzw. `--delete` löscht die angegebene Regel.
- `-R` bzw. `--replace` ersetzt die angegebene Regel.
- `-L` bzw. `--list` listet die Regeln einer angegebenen Kette auf. Wenn keine Kette angegeben wird, werden alle Regeln aller Ketten angezeigt.
- `-F` bzw. `--flush` löscht alle Regeln einer angegebenen Kette. Wird keine Kette angegeben, werden alle Regeln aller Ketten gelöscht. Die Firewall ist dann praktisch vollständig geöffnet.

Es gibt noch einige weitere Kommandos, aber die hier aufgeführten sind für die meisten Fälle ausreichend.

Ketten (POSTROUTING im Beispiel)

Die standardmäßig vorhandenen Ketten (Chains) der Firewall sind sogenannten Hook Points im Kernel zugeordnet. Diese wiederum repräsentieren Positionen im Signalfluss eines Pakets. Die Beschreibungen der Ketten werden Aufschluss geben:

- `INPUT` ist die Kette, die unmittelbar vor einem lokalen Socket positioniert ist. Die hier platzierten Regeln steuern also den Fluss von Paketen zu lokalen Anwendungen hin und sind deshalb für Router normalerweise nicht von Belang.
- `OUTPUT` ist hinter lokalen Sockets positioniert und regelt den Fluss von Paketen, die eine lokale Anwendung gerade verlassen. Auch diese Kette ist deshalb für Router nicht von Belang.
- `FORWARD` ist eine Kette, die beim Routing verwendet wird. Sie regelt den Verkehr von Paketen, die nicht für das lokale System bestimmt sind. Diese Pakete würden die Ketten `INPUT` oder `OUTPUT` gar nicht erst erreichen.
- `PREROUTING` ist eine Kette, die sehr nah an der Netzwerkschnittstelle orientiert ist. Sie kontrolliert Pakete, die gerade beim System eintreffen.
- `POSTROUTING` ist ebenfalls nah an der Netzwerkschnittstelle orientiert, regelt aber den Fluss der Pakete, kurz bevor sie das System verlassen.

Wenn Sie der Firewall Regeln hinzufügen, kann Abbildung 212.1 Ihnen dabei helfen, die richtige Kette zu finden. Wenn Sie den Pfeilen folgen, können Sie z. B. sofort sehen, dass Pakete, die einen Router lediglich durchlaufen, nicht durch Regeln beeinflusst werden, die in den Ketten `INPUT` oder `OUTPUT` definiert wurden.

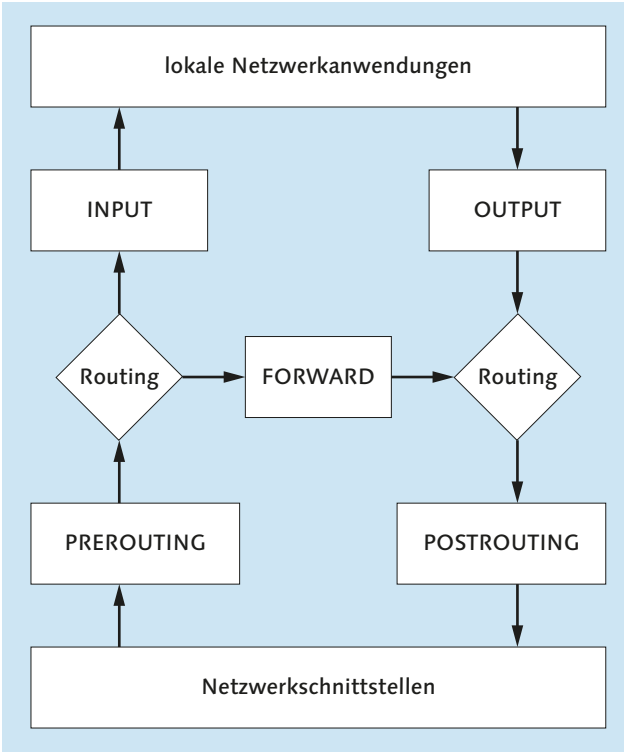

Abbildung 212.1 Die Chains der iptables-Firewall

Matches (-s 192.168.0.0/24 -o eth0 im Beispiel)

Matches sind die Beschreibungen der Pakete, die es mit einer Regel zu bearbeiten gilt. In diesem Fall sind das Pakete, die von Computern stammen, die sich im Netzwerksegment `192.168.0.0/24` befinden und über die Schnittstelle `ppp0` verlassen sollen. Gängige Matches sind:

- `-p` bzw. `--protocol`: z. B. `tcp`, `udp`, `icmp`, `gre`
- `-s` bzw. `--source`: Herkunft des Pakets
- `-d` bzw. `--destination`: Ziel des Pakets
- `-i` bzw. `--in-interface`: Schnittstelle, über die das Paket eintrifft
- `-o` bzw. `--out-interface`: Schnittstelle, über die ein Paket das System verlässt
- `--dport`: Zielport, den ein Paket verwendet
- `--sport`: Quellport, den ein Paket verwendet

Ein Match kann mehrere dieser Optionen enthalten. Wenn Sie viele Optionen verwenden, um Pakete möglichst genau zu beschreiben, benötigen Sie wahrscheinlich insgesamt ein größeres Regelwerk.

Ziele (-j MASQUERADE im Beispiel)

Ein Ziel (target) sagt dem Kernel, wie mit den durch Matches erkannten Paketen weiter verfahren werden soll. Dem Target ist grundsätzlich die Option `-j` (bzw. `--jump`) vorangestellt. Gängige Targets sind:

- `ACCEPT` lässt das Paket zur weiteren Verarbeitung zu.
- `DROP` verwirft ein Paket ohne Fehlermeldung.
- `REJECT` lehnt ein Paket ab. Der Absender erhält eine Fehlermeldung.
- `MASQUERADE` führt IP-Masquerading aus. Es handelt sich hierbei um eine Form von NAT, die auf dynamische IP-Adressierung an der externen Netzwerkschnittstelle hin optimiert wurde.
- `SNAT` (Source NAT) führt ebenfalls IP-Masquerading durch. `SNAT` ist aber ausschließlich für feste IP-Konfigurationen vorgesehen.
- `DNAT` (Destination NAT) maskiert nicht die Quelle, sondern das Ziel eines Paketes. `DNAT` wird für Port-Forwarding verwendet, wenn Sie z. B. einen Serverdienst, der sich hinter einer Firewall befindet, im Internet veröffentlichen müssen.
- `REDIRECT` leitet ein Paket um. Dieses Target wird z. B. für die Weiterleitung von Paketen an transparente Proxy-Server benötigt.

Computer absichern

Die folgenden Beispiele sollen Ihnen den Einstieg in `iptables` erleichtern. Sie sollten zum Üben natürlich kein Produktionssystem verwenden. Um zunächst alle bestehenden Firewall-Regeln zu löschen, verwenden Sie:

```
iptables -F
```

Um eine wasserdichte Grundkonfiguration zu erhalten, können Sie nun zuerst einmal alles blockieren:

```
iptables -t filter -A INPUT -j REJECT
iptables -t filter -A OUTPUT -j REJECT
```

Die Regeln sind fast selbsterklärend. Alle eingehenden und ausgehenden Pakete werden verworfen. Der Absender eines Pakets erhält eine Fehlermeldung. Da die Ketten `INPUT` und `OUTPUT` hinter der Kette `FORWARD` angeordnet sind (Sie sehen das in der schematischen Darstellung), ist ein Routing auf diesem Host weiterhin möglich. Wenn Sie auch die Weiterleitung von Paketen verhindern müssen, verwenden Sie:

```
iptables -t filter -A FORWARD -j REJECT
```

Der Computer ist jetzt absolut sicher. Es kommt kein Paket mehr herein noch heraus, und es findet auch keine Weiterleitung mehr statt.

Jetzt können Sie Schritt für Schritt die Ports wieder öffnen, die Sie benötigen. Da Namensauflösung für fast alles benötigt wird, sollten Sie vielleicht mit DNS beginnen:

```
iptables -I OUTPUT -p udp --dport 53 -j ACCEPT
iptables -I INPUT -p udp --sport 53 -j ACCEPT
iptables -I OUTPUT -p tcp --dport 53 -j ACCEPT
iptables -I INPUT -p tcp --sport 53 -j ACCEPT
```

DNS kommuniziert sowohl über TCP als auch über UDP. Das muss natürlich bei der Erstellung der Regeln berücksichtigt werden. Die erste Regel lässt ausgehende UDP-Pakete, die an den Zielport (destinationport) 53 gerichtet sind, zu. Damit die Antwort des DNS-Servers das System erreicht, lässt die zweite Regel eingehende UDP-Pakete vom Quellport (sourceport) 53 zu. Die dritte und vierte Regel erstellen entsprechende Regeln für das TCP-Protokoll. Beachten Sie, dass die Regeln in diesem Fall mit `-I` eingefügt werden müssen. Es würden sonst die anfänglich erstellten Blockierungsregeln zuerst abgearbeitet werden und die Namensauflösung verhindern.

Nach diesem Muster können Sie jetzt weitere Regeln erstellen, damit das System wieder nutzbar wird. Rückblickend auf die Datei */etc/services* können Sie natürlich auch die Namen der Dienste in Ihren Regeln verwenden. Damit Sie wieder im Internet surfen können, verwenden Sie diese Kommandos:

```
iptables -I OUTPUT -p tcp --dport http -j ACCEPT
iptables -I INPUT  -p tcp --sport http -j ACCEPT
```

Um das Routing für dieselben Dienste ebenfalls zu erlauben, müssen die entsprechenden Regeln auch noch in die Kette `FORWARD` aufgenommen werden. Das erledigen die folgenden Zeilen:

```
iptables -I FORWARD -p udp --dport 53   -j ACCEPT
iptables -I FORWARD -p udp --sport 53   -j ACCEPT
iptables -I FORWARD -p tcp --dport 53   -j ACCEPT
iptables -I FORWARD -p tcp --sport 53   -j ACCEPT
iptables -I FORWARD -p tcp --dport http -j ACCEPT
iptables -I FORWARD -p tcp --sport http -j ACCEPT
```

Es ist übrigens zumindest bei Computern, die nicht als Router fungieren, nicht zwingend erforderlich, alle Regeln symmetrisch anzulegen. Weitere Informationen darüber finden Sie in Abschnitt 212.4, »Sicherheitsmaßnahmen«.

Network Address Translation (NAT)

Sie können Linux auch als NAT-Router verwenden, um den Internetzugang für Benutzer bereitzustellen. Sie sollten dann einen Computer mit mindestens zwei Netzwerkkarten verwenden. Wenn die Schnittstelle `ppp0` (z. B. ein DSL-Modem) mit

dem Internet und die Schnittstelle `eth1` mit dem inneren Netzwerk verbunden ist, dann können Sie 1:1 das Kommando aus dem anfänglichen Beispiel verwenden:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o ppp0 -j MASQUERADE
```

Wenn Sie ausgehendes NAT verwenden, wird immer die Kette `POSTROUTING` verwendet. Damit wird sichergestellt, dass Pakete, die in einem lokalen Netzwerk zugestellt werden können, zunächst durch die Routing-Tabelle behandelt werden. Ausgehende Pakete werden entsprechend maskiert. Wenn die dem Internet zugewandte Schnittstelle über eine statische IP-Adresse verfügt, sollten Sie nicht das Target `MASQUERDE` verwenden, sondern `SNAT`. `SNAT` besagt, dass die Quelle maskiert werden soll (Source NAT). Die zum Maskieren verwendete (statische!) IP-Adresse wird dann als Option übergeben. Beispiel:

```
iptables -t nat -A POSTROUTING -s 192.168.0.0/24 -o eth0 -j SNAT
--to-source 207.144.6.7
```

Port-Forwarding und IP-Forwarding

Wenn Sie einen Webserver, der sich hinter einer Firewall befindet, veröffentlichen wollen, dann müssen Sie den entsprechenden Port (80) von der dem Internet zugewandten Schnittstelle des NAT-Routers an die IP-Adresse des Webservers weiterleiten. Dieser Vorgang wird als *Port-Forwarding* bezeichnet. Da solche Regeln sehr frühzeitig innerhalb des Regelwerks greifen müssen (noch vor dem Routing), kommt hier die Kette `PREROUTING` zum Einsatz. Ausgehend von einem Szenario, bei dem ein DSL-Modem zum Einsatz kommt und der Webserver die IP-Adresse 192.168.0.5 verwendet, sieht das entsprechende Kommando so aus:

```
iptables -t nat -A PREROUTING -i ppp0 -p tcp --dport http -j DNAT
--to-destination 192.168.0.5:http
```

Wenn Sie einen Server vollständig im Internet veröffentlichen wollen, weil alle auf diesem Computer vorhandenen Dienste vom Internet aus erreichbar sein müssen, dann können Sie stattdessen IP-Forwarding verwenden. Eine entsprechende Regel würde so konfiguriert:

```
iptables -t nat -A PREROUTING -i ppp0 -j DNAT --to-destination 192.168.0.5
```

Beachten Sie aber, dass dieser Server dann ohne zusätzlichen Schutz vom Internet aus erreichbar ist und selbst über eine Firewall verfügen sollte.

iptables-save und iptables-restore

Sie können die Konfiguration Ihrer Firewalls sichern, indem Sie das Programm `iptables-save` verwenden. Dieses Programm ist aber auch sehr nützlich, wenn Sie sich einfach einen Überblick über Ihre `iptables`-Konfiguration verschaffen wollen.

Zur Sicherung Ihrer `iptables` leiten Sie die Ausgabe des Programms einfach in eine Textdatei um:

```
root@arch-deb-book:/# iptables-save > /etc/meineRegeln
```

Um die Firewall-Regeln wiederherzustellen, verwenden Sie das Gegenstück `iptables-restore`. Da dieses Programm von *stdin* liest, benötigen Sie auch hier wieder einen Redirektor:

```
root@arch-deb-book:/# iptables-restore < /etc/meineRegeln
```

Da `iptables`-Regeln nach einem Neustart des Computers nicht erhalten bleiben, können Sie `iptables-restore` auch sehr gut in Startskripten verwenden, um die Firewall zu konfigurieren.

ip6tables

Zur Konfiguration von Firewallregeln für IPv6 gibt es Tools, die denen von IPv4 entsprechen. Auch die Syntax der Tools ist mit denen für IPv4 weitestgehend identisch. Es ergeben sich diese einfachen Entsprechungen:

- `ip6tables` entspricht `iptables`
- `ip6tables-save` entspricht `iptables-save`
- `ip6tables-restore` entspricht `iptables-restore`

Spätestens das Wissen um die beinahe identische Syntax mit `iptables` sollte die Befürchtungen, IPv6 sei aufgrund mangelnder Firewall-Technologien gegenüber IPv4 unsicherer, widerlegen. Diese Ängste sind zum Teil entstanden, weil aufgrund des großen Adressraums von IPv6 keine Notwendigkeit für NAT mehr besteht. Um eine sichere Grundkonfiguration zu erhalten, können Sie Ihre Firewallregeln einfach mit diesen drei Zeilen abschließen:

```
ip6tables -t filter -A INPUT -j DROP
ip6tables -t filter -A OUTPUT -j DROP
ip6tables -t filter -A FORWARD -j DROP
```

Das Resultat ist offensichtlich.

212.2 Verwalten von FTP-Servern

Wichtung: 2

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen FTP-Server für anonyme Downloads und Uploads zu konfigurieren. Dieses Lernziel beinhaltet die Vor-

sichtsmaßnahmen, die getroffen werden, falls anonyme Uploads erlaubt sein sollen, sowie die Konfiguration des Benutzerzugriffs.

Wichtigste Wissensgebiete:

- Konfigurationsdateien, Werkzeuge und Dienstprogramme für Pure-FTPd und *vsftpd*
- Kenntnis von ProFTPD
- Grundlegendes Verständnis von passiven gegenüber aktiven FTP-Verbindungen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- *vsftp.conf*
- wichtige Kommandozeilenoptionen von Pure-FTPd

Allgemeines

Das File Transfer Protocol (FTP) dient der Übertragung von Dateien über das Netzwerk, wie der Name schon vermuten lässt. Das Protokoll verwendet zwei unterschiedliche TCP-Ports. TCP-Port 21 wird zum Übertragen von Kommandos genutzt, während die eigentliche Datenübertragung serverseitig den TCP-Port 20 verwendet. Diese Trennung bewirkt, dass auch während der Übertragung großer Datenmengen weiterhin Kommandos an den FTP-Server gesendet werden können. Man unterscheidet zwei Modi, nämlich passives und aktives FTP.

Aktiver Modus

Im aktiven Modus initiiert der Client die Kommunikation mit dem FTP-Server über den Port 21. Er teilt dem Server dann einen zufällig gewählten Port mit, an dem der Client auf Daten wartet. Der Server sendet angeforderte Daten dann von seinem Port 20 aus *aktiv* an den vom Client genannten Port. Sollte sich der Client hinter einer NAT-Firewall befinden, kann der Server den vom Client zufällig gewählten Port nicht erreichen, und die Kommunikation schlägt fehl.

Passiver Modus

Wenn sich der Client-Computer hinter einer NAT-Firewall befindet, sollten Sie den passiven Modus verwenden. Der Kommunikationsaufbau geschieht dann ebenfalls vom Client aus an den TCP-Port 21 des Servers. Der Client baut aber bei Bedarf ebenfalls die Verbindung für die Datenübertragung zu TCP-Port 20 des FTP-Servers auf. Der Server bleibt also *passiv*. Da beide Kommunikationskanäle durch den Client initiiert wurden, gibt es normalerweise keine Probleme, die durch NAT verursacht werden könnten.

Allgemeine Sicherheitsüberlegungen

Mit Sicherheit kennen Sie einige FTP-Server, die im Internet erreichbar sind. Die meisten dieser Server erfordern keine Authentifizierung. Stattdessen wird gerne `anonymous` als Benutzername verwendet und ein beliebiges Passwort. Das ist auch in Ordnung, solange die Daten, die auf dem FTP-Server lagern, nicht vertraulich sind. Wenn Sie aber Daten für eine Unternehmensumgebung bereitstellen müssen, dann sollten Sie natürlich eine Authentifizierung konfigurieren.

Wenn auf einen Server anonym zugegriffen und auch Uploads erlaubt werden sollen, müssen Sie sich einerseits Gedanken um die Legalität der heraufgeladenen Dateien machen und andererseits darum, dass die Benutzer nicht Ihr Dateisystem überfluten. Es ist normalerweise sinnvoll, für den FTP-Zugang eine separate Partition auf einer Festplatte bereitzustellen und den Zugriff des Benutzers auf diese Partition zu begrenzen. Authentifizierte Benutzer können bei Bedarf auch auf ihr Heimatverzeichnis beschränkt werden.

Das FTP-Protokoll bietet von Haus aus keinerlei Verschlüsselung. Das gilt leider nicht nur für die Datenübertragung, sondern auch für die Übermittlung von Benutzernamen und Passwörtern. Abhilfe schafft hier FTP über SSL. Sie sollten unverschlüsselte Kommunikation mit FTP nur in Netzwerken verwenden, in denen es Fremden nicht möglich ist, die Datenkommunikation abzuhören.

Die wichtigsten typischen Sicherheitseinstellungen eines FTP-Servers sollen als Erstes am Beispiel von *vsftpd*, dem Very Secure FTP Daemon, erläutert werden.

vsftpd

Der *Very Secure FTP Daemon* (*vsftpd*) ist ein FTP-Server, der vor allem auf Sicherheit, hohe Geschwindigkeit und Stabilität hin optimiert wurde. Die Konfiguration ist verhältnismäßig einfach, weil der Server mit einer einzigen Konfigurationsdatei, nämlich */etc/vsftpd.conf* auskommt. Eine dokumentierte Beispielkonfiguration wird Ihnen dabei helfen, sich zurechtzufinden. Sie finden hier vor allem Optionen, die mit den Prüfungsthemen im Zusammenhang stehen.

Sie können `vsftpd` eigenständig als Daemon ausführen oder die Kontrolle einem Super-Daemon wie `inetd` überlassen. Wenn Sie einen eigenständigen Server wünschen, dann verwenden Sie diese Option:

```
listen=YES
```

Grundsätzlich unterstützt dieser Daemon auch das Protokoll IPv6. Es ist allerdings zu beachten, dass der Server derzeit nur eines der IP-Protokolle gleichzeitig unterstützt, wenn er als Stand-alone-Server ausgeführt wird. Die folgende Option schließt also die vorangehenden aus:

```
listen_ipv6=YES
```

Die folgenden Zeilen machen den Server beschreibbar, aktivieren den Zugriff für `anonymous` und legen gleichzeitig das Hauptverzeichnis für `anonymous`-Benutzer fest:

```
write_enable=YES
anonymous_enable=YES
anon_root=/ftp
```

Damit `anonymous`-User auch Uploads durchführen und Verzeichnisse auf dem FTP-Server erstellen können, sind zusätzlich die beiden folgenden Optionen notwendig:

```
anon_upload_enable=YES
anon_mkdir_write_enable=YES
```

Wenn ein Zugriff als `anonymous` stattfindet, dann werden die Zugriffe auf das Dateisystem des Servers stellvertretend durch den User `ftp` durchgeführt. Deshalb sollten Sie den User ftp (mittels `chown`) zum Besitzer des entsprechenden Verzeichnisses machen. Das Verzeichnis */ftp* darf für den Benutzer `ftp` allerdings nicht beschreibbar sein, weil der Server den Login von `anonymous` sonst aus Sicherheitsgründen ablehnt. Wenn `anonymous`-Benutzer auch Dateien auf den FTP-Server hochladen müssen, legen Sie unterhalb des FTP-Roots (in diesem Fall */ftp*) ein für diesen Zweck vorgesehenes Unterverzeichnis (z. B. */ftp/upload*) an. Hier können Sie dann großzügig sein und mit `chmod 777 /ftp/upload -R` gefahrlos Zugriff für alle gewähren. Ein FTP-User kann seine Uploads dann in diesem Verzeichnis durchführen.

Wenn Sie Benutzern den Zugriff auf den FTP-Server gestatten wollen, die über ein Benutzerkonto auf dem Server verfügen, dann verwenden Sie diese Zeile:

```
local_enable=YES
```

Damit sich die Benutzer nicht frei auf dem Server bewegen und in beliebigen Verzeichnissen operieren können, ist die folgende Option hilfreich:

```
chroot_local_user=YES
```

Der Zugriff durch die Benutzer wird dann auf deren jeweiliges Heimatverzeichnis begrenzt. Umgangssprachlich ist der User *gechrootet* oder im *Jail* (Gefängnis). Zum Datenaustausch können dann auch diese Benutzer die `anonymous`-Anmeldung verwenden.

Damit Sie feststellen können, welcher Benutzer welche Dateien auf den Server geladen hat, sollten Sie die Protokollierung aktivieren:

```
xferlog_enable=YES
```

Um die unerwünschte Benutzung des Servers einzuschränken, können Sie die Benutzer durch eine Nachricht darauf aufmerksam machen, dass die Nutzung des FTP-Servers protokolliert wird:

```
ftpd_banner=Willkommen auf unserem FTP-
Server! Alle Zugriffe werden protokolliert und Missbrauch entsprechend geahnde
t. Der BOFH
```

Wenn Sie die Kommunikation – insbesondere die Authentifizierung – des FTP-Servers absichern wollen, benötigen Sie ein RSA-Zertifikat. Den Pfad zu diesem Zertifikat geben Sie so an:

```
rsa_cert_file=/etc/ssl/private/vsftpd.pem
```

Es gibt noch einige andere Optionen, aber die hier aufgeführten Optionen sind sowohl für die Sicherheit als auch für die Prüfung am wesentlichsten.

Pure-FTPd

Der *Pure-FTPd*-Server weist eine gegenüber anderen Servern ungewöhnliche Eigenart auf. Er wird nämlich nicht, wie gewohnt, über Konfigurationsdateien konfiguriert. Stattdessen erwartet er alle benötigten Optionen direkt auf der startenden Kommandozeile. Es gibt allerdings trotzdem Konfigurationsdateien, die aber auf eine besondere Art verarbeitet werden. Ein Wrapper-Skript liest die Konfigurationsdateien vor dem Start des Servers ein und erzeugt aus den enthaltenen Informationen ein entsprechendes Startkommando mitsamt Optionen. Die Konfigurationsdateien finden Sie im Verzeichnis */etc/pure-ftpd*. Deren Inhalt variiert in Abhängigkeit von der verwendeten Linux-Distribution.

Die ausführbaren Dateien des Pure-FTPd-Servers finden Sie im Verzeichnis */usr/sbin*:

```
root@arch-deb:/usr/sbin# ls pure-ftp*
pure-ftpd   pure-ftpd-control   pure-ftpd-virtualchroot   pure-ftpd-
wrapper   pure-ftpwho
```

- `pure-ftpd` ist das eigentliche Serverprogramm.
- `pure-ftpd-wrapper` ist ein Perl-Skript und zuständig für das Generieren der Kommandozeilen aus den Konfigurationsdateien. Es wird normalerweise von `init`-Skripten aufgerufen.
- `pure-ftpd-control` ist ebenfalls ein Perl-Skript und sollte zum manuellen Starten, Stoppen oder Neustarten des Servers verwendet werden.
- `pure-ftpd-virtualchroot` ist ein Binärprogramm, das verwendet wird, wenn `pure-ftpd` in einer gechrooteten Umgebung ausgeführt werden soll.

- `pure-ftpwho` ist ebenfalls ein Binärprogramm. Sie können es verwenden, um aktuelle Sitzungen auf dem Server anzuzeigen.

Sollten Sie den Server aus einem `tar`-Ball heraus installiert haben, finden Sie diese Dateien stattdessen unter */usr/local/sbin*.

Pure-FTPd starten

Normalerweise werden Sie den Server natürlich durch ein `init`-Skript automatisch starten lassen. Sie können den Daemon aber auch von Hand ausführen, wenn Sie die von Ihnen benötigten Optionen übergeben. Ein Start mit den Default-Einstellungen ist einfach:

```
root@arch-deb:/# pure-ftpd &
```

Wie bei vielen anderen Programmen auch gibt es für diesen Server Optionen in Kurzform oder in ausgeschriebener Form. Die beiden folgenden Kommandos starten den Server z. B. so, dass er ausschließlich IPv4 als Kommunikationsprotokoll zulässt und als Daemon läuft:

```
root@arch-deb:/# pure-ftpd -4 -B
root@arch-deb:/# pure-ftpd --ipv4only --daemonize
```

Die Option `-B` erspart einem auch die Methode, den Server mithilfe des »&« in den Hintergrund zu schicken. Einige weitere wichtige Optionen sind:

- `-6 --ipv6only` erlaubt ausschließlich die Kommunikation über das IPv6-Protokoll.
- `-A --chrooteveryone` sperrt alle Benutzer in ihrem Heimatverzeichnis ein, außer den User *root*.
- `-B --daemonize` führt *pure-ftpd* als Daemon aus.
- `-c --maxclientsnumber` legt die maximale Anzahl gleichzeitiger Client-Verbindungen fest.
- `-C --maxclientsperip` legt die maximale Anzahl der Verbindungen von einer IP-Adresse aus fest. Bedenken Sie hierbei, dass Benutzer, die sich hinter derselben NAT-Firewall befinden, nur eine gemeinsame IP-Adresse verwenden.
- `-d --verboselog` sorgt für eine sehr umfangreiche Protokollierung. Jedes einzelne FTP-Kommando wird protokolliert, wenn Sie diese Option verwenden.
- `-D --displaydotfiles` zeigt auch versteckte Dateien an.
- `-e --anonymousonly` weist authentifizierte Benutzer zurück.
- `-E --noanonymous` lässt nur authentifizierte Benutzer zu.
- `-i --anonymouscantupload` sorgt dafür, dass nicht authentifizierte Benutzer nur lesend auf den FTP-Server zugreifen können.

- `-M --anonymouscancreatedirs` lässt `anonymous`-Benutzer Verzeichnisse auf diesem Server erstellen.
- `-N --natmode` sagt dem Server, dass er sich hinter einer NAT-Firewall befindet und deshalb nur im aktiven Modus arbeiten darf. Sie sollten diese Option möglichst vermeiden, weil Client-Computer, die ihrerseits hinter einer NAT-Firewall positioniert sind, nicht auf diesen FTP-Server zugreifen könnten.
- `-Y --tls` legt die SSL-/TLS-Einstellungen fest. Es muss hier ein numerischer Parameter folgen.
  - `0` deaktiviert SSL/TLS komplett.
  - `1` akzeptiert sowohl normale als auch gesicherte Sitzungen.
  - `2` lehnt Verbindungen ab, die ohne SSL/TLS initiiert werden. Das gilt auch für `anonymous`-Verbindungen. Diese Einstellung betrifft nicht die Datenübertragung, sondern nur die Steuerung.
  - `3` lehnt zusätzlich Verbindungen ab, bei denen die Datenübertragung im Klartext stattfinden soll.

Die zu verwendenden Optionen werden, wie bereits erwähnt, während der Ausführung des zuständigen `init`-Skripts aus den Konfigurationsdateien durch das Perl-Skript `pure-ftpd-wrapper` generiert.

Normalerweise sollten Sie den Server über das Skript `pure-ftpd-control` steuern. Das ist ziemlich einfach, weil das Skript nur drei verschiedene und obendrein selbsterklärende Optionen versteht, nämlich `start`, `stop` und `restart`.

ProFTPD

Die Konfigurationsdateien des FTP-Servers ProFTPD finden Sie im Verzeichnis */etc/proftpd*. Die Hauptkonfigurationsdatei heißt *proftpd.conf*. Den genauen Aufbau dieser Datei brauchen Sie für die LPI-Prüfung nicht zu beherrschen. Im Übrigen gibt es für die Konfiguration ein grafisches Frontend, mit dessen Hilfe Sie sogar auf sehr komfortable Weise ein Zertifikat für die sichere Kommunikation generieren können. Sie müssen dazu einfach nur die für ein Zertifikat üblichen Angaben in eine Maske eintragen. Die Erstellung des Zertifikats geht ansonsten vollautomatisch vonstatten. Das grafische Konfigurationsprogramm heißt übrigens `gadmin-proftpd` und Sie können es zumindest bei Debian recht einfach über die Paketverwaltung installieren.

Wenn Sie den Server nach Ihren Vorstellungen fertig konfiguriert haben, können Sie ihn einfach mithilfe der Schaltfläche ACTIVATE starten. Mit `gadmin-proftpd` können Sie auch Benutzerkonten erstellen, indem Sie die Registerkarte USERS auswählen.

Auf der Registerkarte TRANSFERS können Sie laufende Transaktionen auf dem Server überwachen und bei Bedarf auch abbrechen.

212.3 Secure Shell (SSH)

Wichtung: 4

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, einen SSH-Daemon zu konfigurieren. Dieses Lernziel umfasst die Verwaltung von Schlüsseln und die Konfiguration von SSH für die Benutzer. Auch sollten die Prüflinge dazu in der Lage sein, ein Anwendungsprotokoll über SSH weiterzuleiten und die SSH-Anmeldung zu regeln.

Wichtigste Wissensgebiete:

- OpenSSH-Konfigurationsdateien, -Begriffe und -Dienstprogramme
- Anmeldebeschränkungen für den Superuser und den normalen Benutzer
- Verwalten und Benutzen von Server- und Client-Schlüsseln zur Anmeldung mit und ohne Passwort
- Nutzung mehrerer Verbindungen über mehrere Hosts, um einem Verbindungsabbruch zum Remote-Host bei Konfigurationsänderungen vorzubeugen

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `ssh`
- `sshd`
- */etc/ssh/sshd_config*
- Dateien für private und öffentliche Schlüssel
- `PermitRootLogin`, `PubKeyAuthentication`, `AllowUsers`, `PasswordAuthentication`, `Protocol`

Allgemeines

Damit Sie einen Computer, der sich nicht in unmittelbarer Nähe befindet, *remote* administrieren können, benötigen Sie eine entsprechende Zugriffsmöglichkeit. Solche Aufgaben wurden früher mit `telnet` durchgeführt. Heute sollten Sie `telnet` nur noch der Allgemeinbildung wegen kennen und wissen, dass `telnet` eine Konsolensitzung über den Port 23 herstellt. Da `telnet` keinen Mechanismus zur Verschlüsselung von Authentifizierungs- oder gar Nutzdaten in sich birgt, wird es heute fast nur noch zur Diagnose von Verbindungen verwendet. Als sichere Alternative kommt SSH (Secure Shell) zum Einsatz. Mit SSH (inzwischen SSH2) ist es aber nicht nur möglich, eine Konsolensitzung zu initiieren, sondern Sie können über SSH auch Daten transferieren und sogar andere Protokolle tunneln.

SSH verwenden

In den meisten Linux-Distributionen ist SSH nicht nur enthalten, sondern standardmäßig auch installiert. Sie können den SSH-Daemon dann einfach starten wie jeden anderen Daemon auch:

```
archangel:~ # /etc/rc.d/sshd start
```

Zwar unterstützt `sshd` auch die Verwendung eines *TCP-Wrappers* und ist auch von `inetd` bzw. `xinetd` aus startbar, aber diese Vorgehensweise wird nicht empfohlen.

SSH-Client-Verbindung

Wenn eine Verbindung von einem Linux-System zu einem anderen aufgebaut werden soll, starten Sie einfach `ssh` und übergeben den Hostnamen als Parameter. Sie werden dann zur Eingabe eines Kennworts aufgefordert:

```
archangel:~ # ssh ubuntu-server
root@ubuntu-server's password:
```

Sollten Sie sich an der entfernten Maschine nicht mit dem gleichen Benutzerkonto anmelden wollen, das Sie derzeit lokal verwenden, müssen Sie die Option `-l` verwenden:

```
archangel:~ # ssh -l willi ubuntu-server
willi@ubuntu-server's password:
```

Sie können über diese Verbindung auch grafische Anwendungen, die unter X Window laufen, tunneln. Das setzt allerdings voraus, dass Sie die SSH-Verbindung von einem X-Terminal aus initiieren, damit ein Display für die Anwendung auf dem Rechner verfügbar ist, an dem Sie sitzen. Es ist sehr einfach, eine solche Anwendung zu tunneln. Initiieren Sie die Verbindung einfach zusätzlich mit der Option `-X` (großes `X`). Auf der erscheinenden Konsole des Remote-Systems führen Sie die Anwendung dann aus:

```
archangel:~ # ssh -l willi -X ubuntu-server
willi@ubuntu-server's password:
willi@ubuntu-server:~ # openoffice
```

Die Anwendung wird auf dem entfernten System ausgeführt, während die grafische Ausgabe und die Bedienung lokal an dem SSH-Client-Rechner erfolgen. Das hier beschriebene Verfahren wird als *X11-Tunnel* bezeichnet.

Sie können durch einen SSH-Tunnel auch beliebige andere Ports weiterleiten. Zu diesem Zweck ordnen Sie einen lokalen Port einer entfernten IP-Adresse in Kombination mit einem (entfernten) Zielport zu. Im folgenden Beispiel wird auf einen

Terminalserver in einem entfernten Netzwerk zugegriffen, der hinter einem SSH-Server residiert. Die private IP-Adresse des Terminalservers sei 192.168.50.10. Der Terminalserver lauscht an Port 3389. Der SSH-Server ist über die DynDNS-Adresse *meinserver.dyndns.org* erreichbar. Der Verbindungsaufbau aus der Ferne geht so vonstatten, dass zunächst die SSH-Verbindung initiiert wird:

```
Linux:~ # ssh meinserver.dyndns.org -L 4711:192.168.50.10:3389
```

Mit der Option `-L` wird der lokale Port 4711 an die entfernte IP-Adresse 192.168.50.10 mit der Portnummer 3389 weitergeleitet. Alle Verbindungen, die mit diesem lokalen Port hergestellt werden, werden nun an 192.168.50.10:3389 weitergeleitet. Sie können also in diesem Fall einen Terminaldienste-Client (`tsclient`) starten und als Ziel `localhost:4711` angeben, wie in Abbildung 212.2 gezeigt.

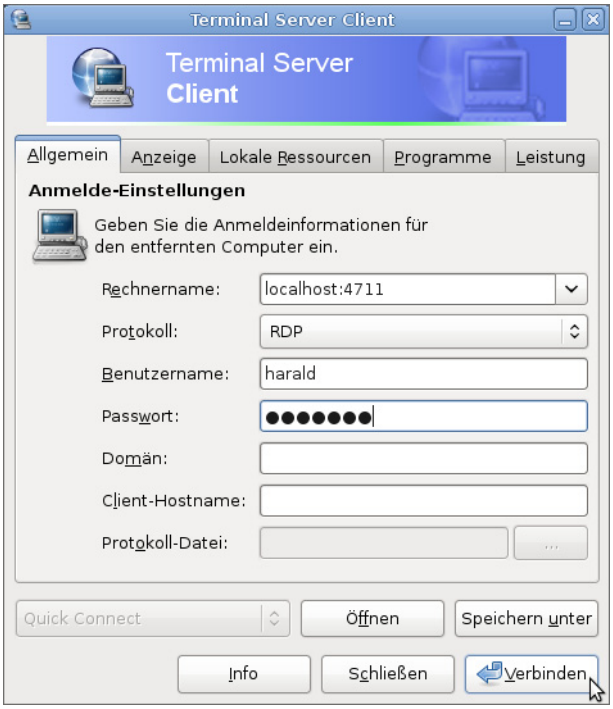

Abbildung 212.2 Es ist üblich, auf localhost zuzugreifen, wenn eine Netzwerkverbindung (hier RDP) über SSH getunnelt wird.

Hinweis

Privilegierte Ports können nur durch den Benutzer `root` weitergeleitet werden. Alle anderen Ports (auch der oben verwendete RDP-Port) können auch von normalen Benutzern umgeleitet werden.

Wenn Sie von einem Windows-Computer aus auf einen Linux-Host zugreifen wollen, dann kommt das frei verfügbare SSH-Client-Programm PuTTY zum Einsatz. Das Tunneln von Ports, inklusive von X11, ist auch mit PuTTY durchführbar. Sie erhalten PuTTY auf dieser Webseite:

http://www.chiark.greenend.org.uk.

Es ist auch möglich, die Authentifizierung über RSA- oder DSA-Schlüssel durchzuführen. Um diese Methode geht es etwas später in diesem Kapitel.

SSH-Konfigurationsdateien

/etc/ssh/sshd_config

Die Datei *sshd_config* wird für die Konfiguration von `sshd`, also dem SSH-Server verwendet. Hier können z. B. folgende Parameter festgelegt werden:

- `Port 22`
- `Protocol 2,1`
- `ListenAddress 192.168.0.58`
- `PermitRootLogin no`
- `AllowUsers harald`
- `PasswordAuthentication no`

In diesem Beispiel verwendet `sshd` den Standardport 22. Es können sowohl SSH2- als auch »normale« SSH-Verbindungen hergestellt werden. Sicherheitshalber kann sich `root` nicht direkt einloggen. Der Zugriff ist auf die interne Schnittstelle des Rechners 192.168.0.58 beschränkt. Ausschließlich der User `harald` darf eine SSH-Verbindung aufbauen. Allen anderen Benutzern wird der Zugriff implizit verweigert. Die letzte Option `PasswordAuthentication no` sorgt dafür, dass eine Authentifizierung ausschließlich mit Schlüsseln möglich ist. Passwortauthentifizierungsversuche werden abgelehnt.

Prüfungshinweis

Die oben aufgelisteten Optionen werden vom LPI ausdrücklich als Prüfungsthemen genannt.

/etc/ssh/ssh_config

Die Konfigurationsdatei *ssh_config* befasst sich ausschließlich mit der clientseitigen Konfiguration von SSH. Hier wird z. B. festgelegt, ob die clientseitige X11-Weiterleitung oder Passwortauthentifizierung erlaubt werden soll. Sie können hier auch die RSA-Authentifizierung aktivieren und den Standardport für ausgehende Verbindun-

gen festlegen. Die Optionsnamen innerhalb dieser Datei sind ansonsten selbsterklärend. Sie sollten in der Prüfung vor allem darauf achten, dass Sie die Dateien *ssh_config* und *sshd_config* nicht aufgrund ihrer Namensähnlichkeit verwechseln.

/etc/hosts.allow und /etc/hosts.deny

Der Zugriff auf SSH kann auch mit den beiden Dateien */etc/hosts.allow* und */etc/hosts.deny* gesteuert werden. Wie Sie diese Dateien verwenden können, ist aber bereits in Abschnitt 205.3, »Kernpunkte der Fehlerbehebung in Netzwerken«, ausführlich beschrieben worden. Aufgrund der Wichtigkeit für die Prüfung sei hier nur noch einmal darauf hingewiesen, dass die *hosts.allow*-Datei Vorrang vor der Datei *hosts.deny* hat. Wenn einem Host also aufgrund eines Eintrags in der Datei *hosts.allow* Zugriff auf SSH gewährt wurde, kann das nicht durch einen Eintrag in der Datei *hosts.deny* zurückgenommen werden.

/etc/nologin

Wenn Sie vorübergehend den Zugriff auf SSH verhindern müssen, können Sie einfach eine leere Datei */etc/nologin* erstellen. Es kann sich dann nur noch `root` per SSH verbinden. Sie können in der Datei auch eine informative Nachricht an die Benutzer hinterlassen, die sich anmelden wollen. Die Nachricht wird dann bei einem Anmeldeversuch im SSH-Client-Programm angezeigt.

/etc/ssh/ssh_known_hosts

Die Datei *ssh_known_hosts* kann sich als systemweite Datei im Verzeichnis */etc/ssh* befinden oder auch unterhalb des Heimatverzeichnisses eines Benutzers in *~/.ssh/know_hosts*. In jedem Fall beinhaltet sie die öffentlichen RSA-Schlüssel bereits bekannter Hosts.

/etc/sshrc

Das Skript `sshrc` wird (falls vorhanden) ausgeführt, sobald sich ein Benutzer über SSH authentifiziert hat. Das geschieht noch vor dem Laden einer Shell.

Authentifizierung der Server mit Schlüsseln

Bei der Anmeldung über SSH authentifiziert sich der Server gegenüber dem Client mit seinem *Hostkey*. Bei der ersten Anmeldung ist der Zielhost dem Client aber noch nicht bekannt, weshalb eine Warnmeldung ausgegeben wird. In der Ausgabe erscheint auch der *Fingerprint* des *Hostkeys*. Theoretisch wäre jetzt eine telefonische Rückversicherung möglich, ob der Zielserver wirklich der ist, der er vorgibt zu sein. Wenn der Benutzer die Warnmeldung mit `yes` bestätigt, wird der Hostkey des Servers

in die Datei *~/.ssh/known_hosts* des Benutzers eingetragen. Bei wiederholten Anmeldungen entfällt dann deshalb die Warnmeldung:

```
root@DRO2:~# ssh 192.168.0.150
The authenticity of host '[192.168.0.150]:22 ([192.168.0.150]:22)'
can't be established.
RSA key fingerprint is 6a:aa:a2:b2:a8:25:21:42:3e:25:49:81:b3:8d:3d:98.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[
192.168.0.150]:22' (RSA) to the list of known hosts.
root@192.168.0.150's password:
```

Es gibt auch eine systemweite Datei */etc/ssh/known_hosts*. Sie können hier die *Hostkeys* aller zu kontaktierenden SSH-Server direkt einbinden, um Warnmeldungen zu vermeiden.

Generieren von Schlüsseln

Mit `ssh-keygen` können Sie u. a. neue *Hostkeys* für Ihr System generieren. Typischerweise werden drei Schlüsselpaare erstellt:

```
root@DRO2:/# ssh-keygen -t dsa
Generating public/private dsa key pair.
Enter file in which to save the key (/root/.ssh/id_dsa):
/etc/ssh/ssh_host_dsa_key
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
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:
68:b0:73:be:0f:fe:b2:13:09:4a:c7:e7:c8:0b:61:d4 root@DRO2
```

Nachdem Sie das Kommando abgeschickt haben, müssen Sie interaktiv die Zieldatei angeben (hier: */etc/ssh/ssh_host_dsa_key*), weil das Programm in der Voreinstellung davon ausgeht, dass ein Schlüsselpaar für einen Benutzer erstellt werden soll, und nicht wie hier für einen Computer. Optional (aber empfohlenermaßen) können Sie eine Passphrase für das Schlüsselpaar angeben. Das Ergebnis werden zwei Dateien sein. Die Datei */etc/ssh/ssh_host_dsa_key* enthält sowohl den privaten als auch den öffentlichen Schlüssel des Hosts. In der Datei */etc/ssh/ssh_host_dsa_key.pub* (die Erweiterung steht für public) ist nur der öffentliche Schlüssel enthalten. Dieser öffentliche Schlüssel kann in die *known_hosts*-Dateien der Client-Computer verteilt werden:

```
root@client:/# cat ssh_host_dsa_key.pub >> /etc/ssh/known_hosts
```

Der *DSA-Schlüssel* aus dem vorangehenden Beispiel dient ausschließlich der Signatur. Für eine Verschlüsselung ist ein *RSA-Key* nötig. Auch er kann mit einem `ssh-keygen`-Kommando erzeugt werden:

```
root@DRO2:/# ssh-keygen -t rsa
```

Das weitere Vorgehen entspricht der Verfahrensweise bei DSA. Als Zieldatei wird diesmal allerdings */etc/ssh/ssh_host_rsa_key* angegeben. In dieser Datei liegt dann entsprechend sowohl der private als auch der öffentliche RSA-Schlüssel. In der Datei */etc/ssh/ssh_host_rsa_key.pub* wird entsprechend nur der öffentliche Schlüssel hinterlegt. Dieser kann wiederum zur Verteilung in *known_hosts*-Dateien verwendet werden:

```
root@client:/# cat ssh_host_rsa_key.pub >> /etc/ssh/known_hosts
```

Wenn noch SSH-Clients der Version 1 verwendet werden, benötigen Sie einen weiteren Hostkey. Erzeugen Sie diesen mit:

```
root@DRO2:/# ssh-keygen -t rsa1
```

Sie sollten diesen Schlüssel in */etc/ssh/ssh_host_key* abspeichern. Wie Sie wahrscheinlich schon vermuten, liegt der öffentliche Schlüssel exportfähig in der Datei */etc/ssh/ssh_host_key.pub*. In der Realität sollten Sie allerdings die Verwendung von `ssh 1` vermeiden.

Benutzerauthentifizierung mit Schlüsseln

Wenn Sie die Benutzerauthentifizierung für SSH-Sitzungen ohne Passwort durchführen möchten, können Sie auch mit Schlüsseln arbeiten. Es müssen dann Schlüsselpaare für den Benutzer generiert werden. Anschließend muss der öffentliche Schlüssel des Benutzers auf dem Zielsystem in die zum Benutzer gehörende Datei *~/.ssh/authorized_keys* integriert werden. Die Vorgehensweise ist ähnlich wie bei der Erstellung der Hostkeys, nur dass Sie diesmal keinen Pfad zur Zieldatei angeben müssen. Wenn bei der Authentifizierung kein Passwort verwendet werden soll, dann geben Sie auch keine Passphrase an:

```
martha@archangel:/$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/martha/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/martha/.ssh/id_rsa.
Your public key has been saved in /home/martha/.ssh/id_rsa.pub.
The key fingerprint is:
2e:99:f4:2e:70:8f:f3:c3:80:7f:df:f7:12:8f:3d:d9 martha@archangel
```

Durch die Ausführung dieses Kommandos (das muss der User übrigens selbst machen) werden die Dateien *~/.ssh/id_rsa* und *~/.ssh/id_rsa.pub* erstellt. Ähnlich wie beim Hostkey enthält die Datei *id_rsa* den privaten und den öffentlichen Schlüssel, während die Datei *id_rsa.pub* nur den öffentlichen Schlüssel enthält.

Für das DSA-Schlüsselpaar gilt entsprechend das Kommando:

```
martha@archangel:/$ ssh-keygen -t dsa
```

Es werden dann die Dateien *~/.ssh/id_dsa* und *~/.ssh/id_dsa.pub* entsprechend generiert.

Damit Sie die Schlüssel zur Authentifizierung verwenden können, müssen diese, wie bereits erwähnt, in die Datei *authorized_keys* des Zielsystems implementiert werden. Das geht am bequemsten mit `scp`. Vorher sollten Sie die Dateien aber umbenennen (bzw. umkopieren), damit eventuell auf dem Zielsystem existierende Dateien nicht überschrieben werden. Sie benötigen hierfür nur die beiden Dateien mit den öffentlichen Schlüsseln:

```
martha@archangel:~/.ssh$ cp id_dsa.pub temp1
martha@archangel:~/.ssh$ cp id_rsa.pub temp2
```

Im nächsten Schritt werden die beiden Dateien auf den Zielserver kopiert. Das kann der Benutzer selbst durchführen. Er wird allerdings jetzt noch nach seinem Passwort gefragt:

```
martha@archangel:~/.ssh$ scp temp? \
martha@217.18.182.170:~/.ssh/
martha@217.18.182.170's password:
temp1                         100 %  608     0.6KB/s   00:00
temp2                         100 %  400     0.4KB/s   00:00
```

Zum Schluss müssen die beiden temporären Dateien ganz einfach mit `cat` in die Datei *authorized_keys* auf dem Zielserver implementiert werden. Dazu ist eine gewöhnliche SSH-Sitzung nötig:

```
martha@archangel:~/.ssh$ ssh 217.18.182.170
martha@217.18.182.170's password:
Linux DRINET 2.6.24-16-generic #1 SMP Thu Apr 10 13:23:42 UTC 2008 i686
martha@DRINET:~$ cat .ssh/temp1 >> .ssh/authorized_keys
martha@DRINET:~$ cat .ssh/temp2 >> .ssh/authorized_keys
```

Ab sofort kann sich die Benutzerin `martha` auch ohne Eingabe eines Passwortes bei diesem Server anmelden.

Eigentlich wird für die Authentifizierung nur ein einziges Schlüsselpaar benötigt. Server und Client müssen sich nur einig darüber werden, ob RSA oder DSA für die Authentifizierung verwendet werden sollen. Um für alle Eventualitäten gerüstet zu sein, empfiehlt es sich, beide Algorithmen zu unterstützen.

Der Authentifizierungsagent

Ein anderer Weg, Authentifizierungen auszuführen, ohne Passwörter eingeben zu müssen, ist über `ssh-agent`. `ssh-agent` ist dazu in der Lage, mehrere Schlüssel für einen Benutzer zu verwalten. Zu diesem Zweck sollte `ssh-agent` frühzeitig während der Startphase von *X* ausgeführt werden.

Mit dem Kommandozeilentool `ssh-add` können Sie dem `ssh-agent` weitere Schlüssel hinzufügen. Ohne Optionen ausgeführt sucht das Programm automatisch nach *~/.ssh/id_rsa*, *~/.ssh/id_dsa* und *~/.ssh/identity*. Es können aber auch gezielt Schlüssel angegeben werden. Wichtige Optionen sind:

- `-l` listet die Fingerabdrücke der verfügbaren Schlüssel auf.
- `-d` entfernt einen einzelnen (angegebenen) Schlüssel vom Agenten.
- `-D` entfernt alle Schlüssel vom Agenten.
- `-s` liest Schlüssel von Smartcards.
- `-e` entfernt Schlüssel von Smartcards.
- `-x` sperrt den Agenten (mit Passwortschutz).
- `-X` entsperrt den Agenten.

Warnhinweis

Sie sollten normalerweise automatische Anmeldungen jeder Art vermeiden. Ein kurzzeitig unbeaufsichtigter Computer kann sonst zu einer schweren Sicherheitsbedrohung für Ihr Netzwerk werden. Bei einem Betriebssystem, das ohnehin überwiegend durch Tastatureingaben verwaltet wird, ist die Eingabe eines Passwortes wohl zumutbar.

212.4 Sicherheitsmaßnahmen

Wichtung: 3

Beschreibung: Die Prüflinge sollten dazu in der Lage sein, Sicherheitswarnungen aus verschiedenen Quellen zu empfangen. Die Installation, die Konfiguration und der Betrieb von Systemen zur Erkennung von Angriffsversuchen, das Einspielen von Sicherheits-Patches sowie Fehlerbehebungen sind auch Teil dieses Lernziels.

Wichtigste Wissensgebiete:

- Werkzeuge und Dienstprogramme zum Scannen und Prüfen von Ports eines Servers
- Quellen und Organisationen, die von Sicherheitslücken berichten, wie Bugtraq, CERT, CIAC usw.
- Werkzeuge und Dienstprogramme, um ein Erkennungssystem für Angriffsversuche (*Intrusion Detection System*, *IDS*) zu implementieren
- Kenntnis von OpenVAS und Snort

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- `telnet`
- `nmap`
- `snort`
- `fail2ban`
- `nc`
- `iptables`

Allgemeines

Die Sicherheitsmaßnahmen, über die Sie bisher in diesem Kapitel gelesen haben, waren eher auf bestimmte Servertypen spezialisiert. Abschließend sollen noch ein paar allgemein anwendbare Informationsquellen und Techniken durchleuchtet werden.

Sicherheitsinstitutionen

Es gibt einige Institutionen, die sich mit der Aufdeckung und Behebung von Sicherheitslücken in Software beschäftigen.

Bugtraq

Bugtraq ist eine (inzwischen) moderierte Mailingliste, die sich mit Sicherheitslücken in Softwareprodukten beschäftigt. Sie soll dazu dienen, frühzeitig Sicherheitslücken aufzudecken und zu beheben, bevor sich Angreifer diese Fehler zunutze machen. Sie können sich auf folgender Webseite in diese Mailingliste eintragen:

http://www.securityfocus.com

CERT

CERT ist das Akronym für *Computer Emergency Response Team*. Das erste dieser Teams wurde 1988 gegründet und zumindest indirekt durch das amerikanische Ver-

teidigungsministerium finanziert. Der Zweck eines CERT ist nicht das Aufdecken sämtlicher bekannt werdender Sicherheitslücken, wie es bei Bugtraq der Fall ist. Vielmehr kümmern sich CERTs um konkrete aktuelle Sicherheitsbedrohung (z. B. Viren oder massiv auftretende Fälle von Phishing).

In Deutschland ist der CERT-Bund des Bundesamts für Sicherheit in der Informationstechnik (kurz BSI) sehr aktiv. Sie können sich auch hier in eine Mailingliste eintragen und sich so über aktuelle Sicherheitsrisiken informieren lassen:

https://www.buerger-cert.de/subscription-new-request

Diese Mailingliste wird von Bürger-CERT betrieben, einer Institution, die vom Bundesamt für Sicherheit in der Informationstechnik (BSI) betrieben wird.

CIAC

CIAC stand für *Computer Incident Advisory Capability*. Die Organisation wurde vom amerikanischen Energieministerium gegründet und beschäftigte sich grundsätzlich mit den gleichen Aufgaben, wie es CERT tun. CIAC wurde bereits vor einigen Jahren in DOE-CIRC umbenannt.

Manuelle Untersuchung

Wenn auf einem Server Netzwerkanwendungen laufen, die nicht (mehr) benötigt werden, dann stellen diese Dienste eine unnötige Angriffsfläche für Hacker und Skriptkiddies dar und sollten dementsprechend deaktiviert oder deinstalliert werden. Wenn Sie offene Ports auf einem System suchen, an dem Sie gerade angemeldet sind, können Sie `netstat` verwenden. Um entfernte Systeme zu untersuchen, benötigen Sie allerdings andere Werkzeuge. Diese Tools kennen Sie bereits aus anderen Kapiteln, weshalb sie hier auch nur noch einmal kurz im Zusammenhang mit dem aktuellen Thema besprochen werden sollen.

Manchmal sind auf einem Computer, den man zur Diagnose einsetzen muss, nicht alle Werkzeuge installiert, die man sich wünscht. Dann ist es gut, wenn man sich mit jeweils anderen Mitteln zu helfen weiß.

telnet

`telnet` (damit ist hier die clientseitige Komponente gemeint) ist eigentlich auf jedem Linux-System vorhanden. Sie können das Kommando `telnet` verwenden, um auf einem Server einen bestimmten Port anzusteuern und dann zu sehen, wie er reagiert. Wenn Sie etwa auf einem Server prüfen müssen, ob IMAP erreichbar ist, können Sie das folgende Kommando verwenden:

```
harald@arch-deb-book:/$ telnet imap.lpic-2.de 143
Trying 46.252.17.134...
Connected to popserver131.ispgateway.de.
Escape character is '^]'.
* OK [CAPABILITY IMAP4rev1 SASL-IR SORT THREAD=REFERENCES MULTIAPPEND
UNSELECT LITERAL+ IDLE CHILDREN NAMESPACE LOGIN-REFERRALS UIDPLUS
LIST-EXTENDED I18NLEVEL=1 QUOTA AUTH=PLAIN AUTH=LOGIN] Dovecot ready.
```

Offensichtlich läuft der IMAP-Server, ein Dovecot-Server, um genau zu sein. Es reagieren nicht alle Dienste mit einer gut lesbaren Antwort, wenn Sie mit `telnet` angesprochen werden. Manche antworten sogar überhaupt nicht, aber wenn ein Port geschlossen ist (und das galt es hier eigentlich zu ermitteln), ist die Fehlermeldung immer gleich und eindeutig:

```
harald@arch-deb-book:/$ telnet imap.lpic-2.de 80
Trying 46.252.17.134...
telnet: Unable to connect to remote host: Connection refused
```

Da dieser Server nicht auf Port 80 antwortet, kann davon ausgegangen werden, dass auf dieser Maschine kein Webserver ausgeführt wird.

Letztendlich ist `telnet` kein Ersatz für einen Portscanner, aber in manchen Situationen ist es für einen kleinen Test ausreichend.

netcat (nc)

`netcat` (Sie können auch das kurze Kommando `nc` verwenden) kann zunächst für dieselben Tests eingesetzt werden wie `telnet`. Die Syntax bei der Angabe des Ziels ist hierbei sogar mit der Syntax von `telnet` identisch. Sie sehen hier ein Beispiel, in dem getestet werden soll, welcher Dienst an TCP-Port 4711 eines Servers lauscht:

```
harald@arch-deb-book:/$ netcat web1.lpic-2.de 4711
SSH-2.0-OpenSSH_5.8p1 Debian-7ubuntu1
Protocol mismatch.
```

Offensichtlich hat hier ein Administrator den SSH-Port auf 4711 geändert, um Angriffe zu minimieren. Der Server hat die Verbindung allerdings getrennt, weil das falsche Protokoll verwendet wurde.

Sie können `netcat` auch verwenden, um zu testen, ob ein Router bzw. eine Firewall die Weiterleitung bestimmter Ports zwischen zwei Computern verhindert. Zu diesem Zweck müssen Sie noch nicht einmal die entsprechende Serversoftware installieren. Auf einem der beiden Computer erstellen Sie mittels `netcat` einen Port Listener, indem Sie ein solches Kommando ausführen:

```
root@system1:~# netcat -l 25
```

System1 wartet nun an Port 25 auf eingehende Verbindungen. So kann geprüft werden, ob SMTP-Kommunikation zwischen den beiden Systemen möglich ist. Auf *System1* darf zu diesem Zeitpunkt kein MTA laufen, der ebenfalls Port 25 verwendet. Führen Sie anschließend auf dem anderen Computer dieses Kommando aus:

```
root@system2:/# netcat system1 25
```

Die Verbindung sollte hergestellt werden. Tastatureingaben, die Sie an *System2* vornehmen, erscheinen auch auf dem Bildschirm von *System1*.

nmap

`nmap` ist ein hervorragender Portscanner. Er wurde schon detailliert in Abschnitt 205.2, »Fortgeschrittene Netzwerkkonfiguration«, beschrieben. Lesen Sie in diesem Kapitel die entsprechenden Abschnitte noch einmal nach, wenn Sie im Umgang mit `nmap` nicht mehr sicher sind.

Automatische Sicherheitssysteme

Es gibt verschiedene Techniken, mit denen Sie Ihre Computer und Netzwerkkomponenten gegen Angriffe schützen können. Wie eine Firewall arbeitet, haben Sie in Abschnitt 212.1, »Router-Konfiguration«, schon erfahren. Eine Firewall ist aber nicht die einzige Komponente, die Sie einsetzen können. In manchen Fällen können zusätzliche Programme zur Einbruchsversuchserkennung (*IDS*, *Intrusion Detection Systems*) oder gar -Verhinderung (*IPS, Intrusion Prevention Systems*) erforderlich werden. Sie sollten die hierfür verwendbare Software kennen.

iptables (Firewall)

Wenn Sie einen einzelnen Computer absichern, z. B. eine Arbeitsstation oder einen Webserver, dann benötigen Sie in der Hauptsache die Ketten `INPUT` und `OUTPUT`. Die für Router wichtigen Ketten `FORWARD`, `PREROUTING` und `POSTROUTING` kommen hier eigentlich nicht vor. Für eine einfache Grundkonfiguration können Sie zunächst alle Regeln löschen:

```
iptables -F
```

Von diesem Moment an ist die Firewall ausgeschaltet. Es muss also jetzt eine Regel zumindest alle eingehenden Verbindungen blockieren. Das erreichen Sie mit:

```
iptables -t filter -A INPUT -j DROP
```

Die Angabe der Tabelle `filter` ist übrigens optional. Wenn Sie keine Tabelle angeben, wird `filter` automatisch ausgewählt. Momentan werden keine Pakete daran gehin-

dert, das System zu verlassen. Die Antworten angesprochener Systeme werden allerdings verworfen. Es gibt eine Regel, die hier Abhilfe schafft:

```
iptables -t filter -I INPUT -m conntrack --ctstate
ESTABLISHED,RELATED -j ACCEPT
```

Diese Regel erlaubt jeglichen Verkehr, der zu bereits existierenden Verbindungen gehört (`RELATED`), und Verbindungen, die von Ihrem System aus initiiert wurden (`ESTABLISHED`). `RELATED` beinhaltet auch, dass zusätzliche Ports auf der Firewall geöffnet werden, wenn die Netzwerkanwendung das erfordert. Bei FTP-Verbindungen wird also auch Port 20 zugelassen, selbst wenn nur Port 21 ausdrücklich erlaubt wurde. Sollte das Match `conntrack` auf Ihrem System nicht vorhanden sein, dann können Sie alternativ mit dem Match `state` arbeiten:

```
iptables -t filter -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Hinweis

Bis hierhin ist der Computer dazu in der Lage, alle ausgehenden Verbindungen einzurichten. Alle eingehenden Verbindungen werden geblockt. Es gibt einige wenige Programme, die in dieser Umgebung Probleme bereiten. NFS läuft mit diesen Grundeinstellungen z. B. nicht.

Wenn die Grundkonfiguration abgeschlossen ist, dann wollen Sie den Computer vielleicht zumindest für die Fernadministration zugänglich machen. Wenn Sie SSH zulassen und dabei den Standardport verwenden, ist es nur eine Frage von kurzer Zeit, bis der erste Brute-Force-Angriff auf Ihr System stattfindet. Erfahrungsgemäß muss man auf einen Angriff weniger als einen Tag warten, wenn man statische IP-Adressen verwendet. Einen ersten Schutz bieten die folgenden `iptables`-Regeln:

```
iptables -A INPUT -m tcp -p tcp --dport 22 -m state
--state ESTABLISHED,RELATED -j ACCEPT

iptables -A INPUT -m tcp -p tcp --dport 22 -m state
--state NEW -m limit --limit 3/hour --limit-burst 3 -j ACCEPT

iptables -A INPUT -m tcp -p tcp --dport 22 -j DROP
```

Die erste Regel erlaubt es bestehenden Verbindungen erhalten zu bleiben. Mit der zweiten Regel wird die Anzahl der Zugriffe via SSH auf drei Zugriffe pro Stunde begrenzt. Leider kann `iptables` nicht feststellen, ob diese Zugriffe erfolgreich waren oder nicht. Wenn Sie also einmal tatsächlich häufiger als dreimal pro Stunde per SSH auf das System zugreifen müssen, werden auch Sie geblockt. Das bewirkt dann die dritte und letzte Regel. Es gibt noch etliche andere `iptables`-Konstrukte zur Absiche-

rung von SSH, aber ich rate Ihnen eher zu anderen Maßnahmen (z. B. zu einem alternativen Port für SSH oder zu `fail2ban`).

fail2ban (IPS, Intrusion Prevention System)

Wenn Sie einen Angriff auf eines Ihrer Systeme live erleben (z. B. weil Sie sich gerade routinemäßig Auffälligkeiten in Logdateien ansehen), dann können Sie natürlich sofort Maßnahmen ergreifen. Sie könnten z. B. mit einer einfachen `iptables`-Regel den Verkehr von und zu der IP-Adresse des Angreifers verwerfen. Genau so arbeitet `fail2ban`. Hierbei handelt es sich um ein Intrusion Prevention System (IPS), das im Gegensatz zum Administrator immer zugegen ist.

`fail2ban` beobachtet verdächtige Aktivitäten, indem es die Logfiles der zu schützenden Programme durchsucht. Es legt bei Bedarf eine Regel in `iptables` an, um einen Angriff zu stoppen. Der absolute Klassiker ist hierbei die Blockierung wiederholter Versuche, sich über SSH an einem System anzumelden (normalerweise Brute-Force-Attacken). Der zusätzliche Einsatz eines IPS ist natürlich mit einigen Vorteilen gegenüber dem Schutz durch eine reine Firewall verbunden. So werden z. B. Ihre erfolgreichen Anmeldungen per SSH nicht als potenzielle Bedrohung gewertet und Sie können sich beliebig oft anmelden.

Installation und Konfiguration

Die Installation über `apt-get` verläuft wie immer:

```
root@arch-deb-book:/# apt-get install fail2ban
```

Auf Red Hat und seinen Derivaten verwenden Sie entsprechend `yum`:

```
root@arch-fedora:/# yum install fail2ban
```

Die Konfigurationsdateien finden Sie im Verzeichnis */etc/fail2ban*.

```
root@arch-deb-book:/etc/fail2ban# ls -l
insgesamt 10
drwxr-xr-x 2 root root 1024 24. Mär 14:22 action.d
-rw-r--r-- 1 root root  859 27. Feb 2008  fail2ban.conf
drwxr-xr-x 2 root root 1024 24. Mär 14:22 filter.d
-rw-r--r-- 1 root root 6683 29. Jun 2010  jail.conf
```

Das Verzeichnis *filter.d* enthält Dateien mit Informationen zu jedem Dienst, den `fail2ban` schützen kann, während *action.d* Dateien enthält, die Aktionen beschreiben, die `fail2ban` ausführen kann. Das beinhaltet die Bedienung verschiedener Firewalls sowie den Versand von E-Mails (vorzugsweise an den Administrator). In der Konfigurationsdatei *fail2ban.conf* können Sie das Loglevel und den Pfad für die Logdatei einstellen. In der Standardeinstellung finden Sie die Logdatei unter */var/log/fail2ban.log*, und das Loglevel ist mit `3` (`INFO`) festgelegt. Es gibt die folgenden Loglevel:

```
1 = ERROR
2 = WARN
3 = INFO
4 = DEBUG
```

Sie können das Loglevel an Ihre Bedürfnisse anpassen, indem Sie diese Einstellung ändern:

```
loglevel = 3
```

Die eigentlich interessante Konfigurationsdatei ist die Datei *jail.conf*. Im Fachjargon werden hier die Jails (Gefängnisse) für die einzelnen zu schützenden Dienste konfiguriert bzw. aktiviert. In Bezug auf SSH sieht das z. B. so aus:

```
[ssh]
enabled = true
port    = ssh
filter  = sshd
logpath  = /var/log/auth.log
maxretry = 3
```

Der Aufbau des Jails ist leicht zu durchschauen.

Die Regel ist aktiviert, und es wird der Port für *SSH* verwendet, wie er in der Datei */etc/services* eingetragen ist. Das Programm durchsucht die Datei */var/log/auth.log* und wendet dabei den Filter `sshd` an. Das klingt logisch, denn SSH-Anmeldungen werden in diesem Logfile protokolliert. Da diese Datei aber auch von anderen Programmen zur Protokollierung verwendet wird (z. B. `login`, `cron`, `su` usw.), werden nur die Zeilen herausgesucht, die das Wort `sshd` enthalten. Nach drei Fehlversuchen wird eingegriffen. Beachten Sie bitte, dass nicht jeder einzelne Versuch, ein falsches Passwort einzugeben, gezählt wird, sondern die Anzahl der Zugriffe über das Netzwerk. Sie sollten also bei einem Selbsttest hartnäckig bleiben. Die Dauer der Aussperrung können Sie übrigens ebenfalls in der Konfigurationsdatei *jail.conf* festlegen. Modifizieren Sie hierfür diese Zeile:

```
bantime  = 600
```

Die Einheit für den Wert dieses Parameters ist Sekunden.

fail2ban testen

Wenn Sie sich mehrfach absichtlich falsch angemeldet haben, werden Sie ausgesperrt. Sie können das am automatisch geänderten Inhalt Ihrer Firewall-Regeln überprüfen. In diesem Fall wurden zwei Angreifer ausgesperrt:

```
root@archangel:/# iptables-save |grep fail2ban
:fail2ban-ssh - [0:0]
-A INPUT -p tcp -m multiport --dports 22 -j fail2ban-ssh
-A fail2ban-ssh -s 87.193.135.42/32 -j DROP
-A fail2ban-ssh -s 79.193.45.193/32 -j DROP
-A fail2ban-ssh -j RETURN
```

Beachten Sie die beiden Regeln mit den `DROP`-Targets. Sie können auch das `fail2ban-client`-Programm verwenden, um den Status des Jails für SSH abzurufen. So bekommen Sie schnell einen Überblick:

```
root@archangel:/# fail2ban-client status ssh
Status for the jail: ssh
|- filter
|  |- File list: /var/log/auth.log
|  |- Currently failed:   0
|  `- Total failed: 114
`- action
   |- Currently banned:   2
   |  `- IP list: 79.193.45.193 87.193.135.42
   `- Total banned: 31
```

Die letzte Möglichkeit, vorangehende Angriffe nachzuvollziehen, ist ein Blick in die Logdatei */var/log/fail2ban.log*. Nur hier finden Sie Details zu Vorgängen, die schon etwas länger zurückliegen:

```
2012-03-24 22:49:29,426 fail2ban.actions: WARNING [ssh] Ban 79.193.45.193
2012-03-24 22:59:30,077 fail2ban.actions: WARNING [ssh] Unban 79.193.45.193
2012-03-24 23:37:51,538 fail2ban.actions: WARNING [ssh] Ban 79.193.45.193
2012-03-24 23:39:46,670 fail2ban.actions: WARNING [ssh] Ban 87.193.135.42
```

Hier wird sichtbar, dass es sich offensichtlich bei dem Computer mit der IP-Adresse 79.193.45.193 um einen »Wiederholungstäter« handelt.

Snort (IDS/IPS, Intrusion Detection and Prevention System)

Wenn Sie über Einbruchsversuche immer auf dem Laufenden gehalten werden wollen, dann sollten Sie Snort verwenden. Im Gegensatz zu `fail2ban` horcht Snort direkt an den Netzwerkschnittstellen und vergleicht den Netzwerkverkehr mit verdächtigen Mustern. Auf diese Art kann Snort ereignisorientiert eingreifen und Angreifer blockieren. Diese Muster werden als Signaturen bezeichnet und sind mit den Mustern von Virenscannern vergleichbar. Da sich Einbruchsmuster im Laufe der Zeit ändern, sollten Sie die Vergleichsdatenbank regelmäßig aktualisieren. Sie können diesen Vorgang automatisieren, indem Sie das Programm *PulledPork* verwenden, oder Sie laden die Regeln regelmäßig von dieser Webseite herunter:

http://www.snort.org/downloads/#rule-downloads

Es gibt auf dem Markt einige Geräte, die als dedizierte IDS-/IPS-Systeme verkauft werden, auf denen Snort zum Einsatz kommt.

Abgesehen von seiner Aufgabe als IDS und IPS können Sie Snort auch als Paket-Sniffer verwenden. Er verfügt dabei über dieselben Funktionen wie das Programm `tcpdump`. Zusätzlich kann Snort zu Debug-Zwecken als Packet Logger eingesetzt werden.

Es ist nicht zu erwarten, dass in der Prüfung Detailfragen zu Snort gestellt werden, aber Sie sollten wissen, wozu diese Software verwendet wird.

OpenVAS

OpenVAS ist eine komplette Lösung zur Untersuchung eines Netzwerks auf Schwachstellen. Es gibt fertige Pakete für etliche Linux-Distributionen und sogar für Windows. Die Kernkomponente ist ein Scanner, der ein Netzwerk auf Schwachstellen prüft. Zurzeit kennt und prüft dieser Scanner ca. 25.000 Sicherheitsrisiken. OpenVAS enthält ein Managementsystem mit einem grafischen Frontend und einer Weboberfläche. Sie können aber auch das Command-Line Interface verwenden. Im Backendbereich verwendet die Software eine Datenbank zur Verwaltung der gesammelten Informationen und der bekannten Sicherheitslücken. Nähere Informationen und die Software selbst erhalten Sie unter:

http://www.openvas.org

In der Prüfung werden keine Fragen zur Installation oder Benutzung von OpenVAS gestellt. Sie sollten lediglich den Zweck dieser Software kennen.

212.5 OpenVPN

Wichtung: 2

Beschreibung: Die Kandidaten sollten dazu in der Lage sein, ein VPN zu konfigurieren und sichere Punkt-zu-Punkt- bzw. Standort-zu-Standort-Verbindungen zu erstellen.

Wichtigste Wissensgebiete:

- OpenVPN

Liste wichtiger Dateien, Verzeichnisse und Anwendungen:

- */etc/openvpn/**
- openvpn

Allgemeines

Es ist leicht zu erraten, was sich hinter dem Namen OpenVPN verbirgt. Es handelt sich natürlich um eine VPN-Lösung auf Open-Source-Basis. OpenVPN ist für viele verschiedene Betriebssysteme verfügbar und sorgt auch für Interoperabilität zwischen diesen unterschiedlichen Systemen. Es gibt OpenVPN für Linux, OpenBSD, FreeBSD, NetBSD, Solaris, Windows 2000 bis Windows 10 und Mac OS X.

Es ist mit OpenVPN möglich, zwei oder mehr Standorte miteinander zu verbinden. Sie können aber auch einen Server bereitstellen, mit dem sich lediglich Road Warriors verbinden. Die kleinste und einfachste Implementierung dient lediglich einer verschlüsselten Peer-to-Peer-Verbindung zweier Einzelcomputer durch das Internet. Da letztere Methode leicht zu implementieren und auch gut zu verstehen ist, soll auf den nächsten Seiten zunächst eine entsprechende Beispielkonfiguration folgen.

Peer-to-Peer-VPN

Sie benötigen für dieses Beispiel zwei Computer. Diese Computer können notfalls im selben Netzwerksegment stehen, aber es macht natürlich mehr Spaß, wenn ein Transitnetzwerk (am besten das Internet) zwischen den Maschinen liegt.

Installieren Sie zunächst das Paket *openvpn* auf den beiden Computern. Sie können es, wie gewohnt, mit `aptitude` oder dem `yum`-Installer installieren. Es gibt bei OpenVPN keine Default-Konfiguration. Es wird zwar das Verzeichnis */etc/openvpn* angelegt, aber Sie werden dort keine Konfigurationsdateien finden.

Unter Umständen müssen Sie das Kernel-Modul für das Tunnel-Device noch laden. Das können Sie wie immer mittels `modprobe` erledigen:

```
root@arch-deb-book:/# modprobe tun
```

Da in diesem Beispiel eine Peer-to-Peer-Konfiguration erstellt werden soll, kann man einen statischen Schlüssel verwenden. Am besten wechseln Sie auf einer der Maschinen in das Verzeichnis */etc/openvpn* und erstellen den Schlüssel mit folgendem Kommando:

```
root@arch-deb-book:/etc/openvpn# openvpn --genkey --secret static.key
```

Bei der Verwendung eines statischen Schlüssels müssen beide Computer ein und denselben Schlüssel verwenden. Kopieren Sie deshalb die Datei *static.key* auch in das Verzeichnis */etc/openvpn* des zweiten Computers. Bei der Verschlüsselung der Datenkommunikation verwendet OpenVPN übrigens die Mechanismen von OpenSSL.

Als Nächstes wird für beide Computer eine Konfigurationsdatei erstellt. Diese Konfigurationsdateien müssen die Dateierweiterung *.conf* aufweisen und im Verzeichnis

/etc/openvpn erstellt werden. Die beiden Konfigurationsdateien unterscheiden sich nur geringfügig voneinander. Sie finden die Erläuterungen zu den einzelnen Zeilen der Dateien jeweils als Kommentare direkt im Quelltext. Sie können die Konfigurationsdateien z. B. *open_p2p_vpn.conf* nennen. Verwenden Sie für den Computer, der die Verbindung herstellt, diese Konfiguration:

```
# Modus der Verbindung festlegen:
mode p2p
# Remote-VPN-Computer und Port angeben!
# Es kann als Ziel entweder eine IP-Adresse oder
# ein FQDN verwendet werden.
remote computer4711.dnsalias.net 1194
# Protokoll der Transportschicht auswählen:
proto udp
# Tunnelgerät definieren:
dev tun
# Zuerst die lokale und dann die Remote-IP-Adresse angeben:
ifconfig 10.111.0.2 10.111.0.1
# Auf die Schlüsseldatei verweisen:
secret static.key
# Zyklisch die Verbindung prüfen und ggf. neu aufbauen:
ping 5
ping-restart 120
ping-timer-rem
# Protokollierungsgrad festlegen:
verb 3
mute 20
```

Für den Computer, der auf die eingehende Verbindung wartet, ist die Konfiguration sehr ähnlich. Es wird lediglich kein Zielcomputer, sondern nur ein Port angegeben. Außerdem sind die lokalen und die Remote-IP-Adressen miteinander vertauscht. Auf redundante Kommentare wird in dieser Datei verzichtet:

```
mode p2p
port 1194
proto udp
dev tun
# Beachten Sie die Umkehrung der Reihenfolge
# bei den IP-Adressen:
ifconfig 10.111.0.1 10.111.0.2
secret static.key
ping 5
ping-restart 120
```

```
ping-timer-rem
verb 3
mute 20
```

Wenn Sie fertig sind, sollten Sie auf beiden Computern OpenVPN neu starten, indem Sie folgendes Kommando verwenden:

```
root@arch-deb-book:/# /etc/init.d/openvpn restart
```

Die neuen Konfigurationsdateien werden dann eingelesen und das VPN aufgebaut. Führen Sie mittels `ping` einen Funktionstest aus, indem Sie die jeweils gegenüberliegende VPN-IP-Adresse angeben. Sollte etwas nicht klappen, dann konsultieren Sie die Dateien */var/log/messages* oder */var/log/syslog*, je nachdem, welches System Sie verwenden.

VPN-Server für mehrere gleichzeitige Zugriffe

Wenn Sie einen Server bereitstellen wollen, auf den mehrere VPN-Clients gleichzeitig zugreifen können, dann gibt es ein paar Unterschiede gegenüber der Peer-to-Peer-Konfiguration zu berücksichtigen. Bei der Peer-to-Peer-Konfiguration wurden die Adressen beider Parteien statisch festgelegt. Bei einer VPN-Server-Konfiguration benötigen Sie mehrere IP-Adressen für die Client-Computer, die nach Möglichkeit automatisch vergeben werden sollten. Außerdem müssen Sie Zertifikate generieren, weil bei einer Serverkonfiguration aus Sicherheitsgründen keine statischen Schlüssel unterstützt werden. Generieren Sie zunächst die benötigten Zertifikate. Zu diesem Zweck wechseln Sie in folgendes Verzeichnis:

/usr/share/doc/openvpn/easy-rsa/1.0

In Abhängigkeit von der verwendeten Distribution könnte das Verzeichnis geringfügig variieren. Es kommt z. B. auch dieser Pfad infrage:

/usr/share/doc/openvpn/examples/easy-rsa/1.0

Sie müssen zunächst die Datei *vars* anpassen. Insbesondere der letzte Teil dieser Datei ist interessant, weil Sie hier Ihre Zertifikate personalisieren können. Bei meinem Testsystem sieht das im Ergebnis so aus:

```
export KEY_COUNTRY=DE
export KEY_PROVINCE=Berlin
export KEY_CITY=Berlin
export KEY_ORG="Maaßen"
export KEY_EMAIL="harald@archangel.homelinux.net"
```

Damit die Variablen, die Sie gerade gesetzt haben, vom System übernommen werden, können Sie entweder die Datei *vars* mittels `chmod a+x` ausführbar machen und

anschließend mit `./vars` ausführen oder einfach das Kommando `source` verwenden. Die zweite Variante ist wohl die elegantere:

```
root@archangel:/usr/...../easy-rsa/1.0# source vars
NOTE: when you run ./clean-all, I will be doing a rm -rf on /usr/share/doc/
openvpn/examples/easy-rsa/1.0/keys
```

Als Nächstes werden eventuell vorhandene alte Schlüssel und Zertifikate gelöscht:

```
root@archangel:/usr/...../easy-rsa/1.0# ./clean-all
```

Nachdem alles sauber ist, erstellen Sie eine neue Zertifizierungsstelle (CA). Die Abkürzung CA steht übrigens für Certificate Authority:

```
root@archangel:/usr/...../easy-rsa/1.0# ./build-ca
Generating a 1024 bit RSA private key
.............++++++
.......++++++
writing new private key to 'ca.key'
```

Die meisten Fragen, die dieses Skript Ihnen stellen wird, können Sie einfach durch Betätigen der Eingabetaste bestätigen, weil die vorgeschlagenen Werte aus den Variablen der Datei *vars* resultieren. Wichtig ist aber die Vergabe eines Wertes für `"CommonName"`. Sie bekommen sonst keine funktionierende Konfiguration. Der Common Name kann frei gewählt werden und ist der Name der Zertifizierungsstelle.

Jetzt, da Sie eine Zertifizierungsstelle besitzen, können Sie als Erstes ein Zertifikat für den VPN-Server generieren.

```
root@archangel:/usr/..../1.0# ./build-key-server  vpn-server
Generating a 1024 bit RSA private key
.++++++
...................++++++
writing new private key to 'vpn-server.key'
```

Mit dem Ausfüllen der Dialoge verhält es sich hier ähnlich wie bei der CA. Sie sollten für den `CommonName` den Namen eintragen, den ein Client bei der Verbindung zu Ihrem Server angibt. Das ist auch hier die Hauptfehlerquelle. Die restlichen Dialoge sind dann wieder selbsterklärend.

```
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'DE'
stateOrProvinceName   :PRINTABLE:'Berlin'
localityName          :PRINTABLE:'Berlin'
```

```
organizationName      :T61STRING:'Maa\0xFFFFFFC3\0xFFFFFF9Fen'
commonName            :PRINTABLE:'archangel.homelinux.net'
emailAddress          :IA5STRING:'harald@archangel.homelinux.net'
Certificate is to be certified until May 19 17:52:28 2021 GMT (3650 days)
```

Nun können Sie ein oder mehrere Client-Zertifikate generieren. Sie sollten für jeden Client ein eigenes Zertifikat ausstellen:

```
root@archangel:/usr/.../easy-rsa/1.0# ./build-key client-01
Generating a 1024 bit RSA private key
.....++++++
.++++++
writing new private key to 'client-01.key'
```

Sie können jetzt auf dieselbe Art weitere Client-Zertifikate ausstellen. Vermutlich ist es nicht mehr nötig zu erwähnen, dass Sie jeweils einen Common Name angeben sollten. Wechseln Sie anschließend in das Unterverzeichnis *keys* und überprüfen Sie den Inhalt. Kopieren Sie die Dateien, die mit *vpn-server* beginnen, und die Datei *ca.crt* in das Verzeichnis */etc/openvpn* des Servers. Die Client-Computer benötigen in ihrem Verzeichnis */etc/openvpn* entsprechend ebenfalls das CA-Zertifikat *ca.crt* und die jeweils zum Client gehörenden Zertifikate und Schlüssel. Diese Dateien gelten als vertraulich und sollten normalerweise nicht über ein ungesichertes Netzwerk übertragen werden.

Auf dem Server benötigen Sie auch noch einen kryptografischen Schlüssel. Diesen sollten Sie gleich im Verzeichnis */etc/openvpn* erstellen. Da OpenVPN Mechanismen von OpenSSL verwendet, wird der Schlüssel auch mit dem entsprechenden Programm erzeugt:

```
root@archangel:/etc/openvpn# openssl dhparam -out dh1024.pem 1024
```

Nachdem alle Zertifikate an Ort und Stelle sind, können Sie sich den Konfigurationsdateien widmen. Diese funktionieren grundsätzlich ähnlich wie in der Peer-to-Peer-Konfiguration. Sie finden die Erläuterungen zu den einzelnen Einstellungen wieder als Kommentare innerhalb der Konfigurationsdateien. Betrachten Sie zunächst die Konfigurationsdatei für den Server:

```
### Serverkonfigurationsdatei ###
port 1194
proto udp
dev tun
# Zertifikate und Schlüssel angeben:
ca ca.crt
cert vpn-server.crt
key vpn-server.key
```

```
dh dh1024.pem
# VPN-Netzwerkadressen zur automatischen Verteilung
# Die erste Adresse verwendet der Server selbst.
server 172.20.0.0 255.255.255.0
# Für wiederkehrende Clients verwendete IP-Adressen
# in dieser Datei speichern:
ifconfig-pool-persist ipp.txt
# Routing-Eintrag dem Client hinzufügen
push "route 192.168.50.0 255.255.255.0"
# Dem Client einen DNS-Server zuweisen
push "dhcp-option DNS 192.168.50.1"
# Ermögliche Kontakt zwischen Clients
client-to-client
# Verbindung alle 10 Sek. testen (ICMP). Wenn nach
# 120 Sek. keine Antwort Tunnel  ggf. neu aufbauen
keepalive 10 120
# Kompression einschalten
comp-lzo
# Maximale Anzahl von Clients festlegen
max-clients 40
# Statusdatei festlegen
status openvpn-status.log
# Mittleren Protokollierungsgrad festlegen
verb 3
```

Nach der Konfiguration müssen Sie `openvpn` starten bzw. neu starten, je nachdem:

```
root@archangel:/etc/openvpn# /etc/init.d/openvpn restart
```

Eine entsprechende Client-Konfigurationsdatei könnte so aussehen:

```
### Client-Konfiguration ###
client
dev tun
proto udp
# VPN-Server:
remote archangel.homelinux.net 1194
# Zertifikate und Schlüssel:
ca ca.crt
cert client-01.crt
key client-01.key
# Kompression einschalten
comp-lzo
verb 3
```

Auch auf dem Client-System muss anschließend `openvpn` neu gestartet werden. Wenn Sie Ihre Konfiguration genauso gemacht haben wie auf den vorangehenden Seiten dieses Buchs, dann sollten Sie jetzt die IP-Adresse des Servers (172.20.0.1) erreichen können. Das können Sie einfach mittels `ping` nachprüfen. Mittels `ifconfig` können Sie auch den Tunneladapter überprüfen:

```
root@chrouter1:~# ifconfig tun0
tun0   Link encap:UNSPEC  Hardware Adresse 00-00-00-00-00-00-00-00-00-00-00-
00-00-00-00-00
       inet Adresse:172.20.0.6  P-z-P:172.20.0.5  Maske:255.255.255.255
       UP PUNKTZUPUNKT RUNNING NOARP MULTICAST  MTU:1500  Metrik:1
       RX packets:6 errors:0 dropped:0 overruns:0 frame:0
       TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
       Kollisionen:0 Sendewarteschlangenlänge:100
       RX bytes:504 (504.0 B)  TX bytes:504 (504.0 B)
```

Die Verbindung steht offensichtlich, denn es sind schon ein paar Bytes übertragen worden.

Übungsfragen zu LPI 117-202

Die folgenden Fragen sollen Ihnen helfen, sich an die Art der Fragestellung in der wirklichen Prüfung zu gewöhnen. Es hat keinen Zweck, die Fragen einfach auswendig zu lernen, denn es sind keine echten Prüfungsfragen. Sie sollten versuchen, die Antworten zu jeder einzelnen Frage zu verstehen. Deshalb werden sowohl die richtigen als auch die falschen Antworten im Lösungsteil des Buchs detailliert besprochen. Das Üben mit diesen Fragen soll Ihnen auch die Herangehensweise bei eventuell Ihnen unbekannten Themen näher bringen. Ein unbekanntes Kommando in einer Frage ist nämlich noch längst kein Grund, eine Frage einfach nicht zu beantworten. Oft führt ein wenig Logik oder das Ausschlussverfahren dennoch zum Ziel.

Fragen

Frage 1:

Sie verwalten einen Apache Webserver, auf dem vertrauliche Dokumente gespeichert sind. Um die Verwaltung zu vereinfachen, wollen Sie die Benutzer, die auf den Server zugreifen dürfen, in einer Gruppe zusammenfassen. Welche Datei werden Sie erstellen bzw. bearbeiten?

- ☐ A: */etc/passwd*
- ☐ B: */etc/groups*
- ☐ C: *.htaccess*
- ☐ D: *.htgroup*
- ☐ E: *httpd.conf*

Frage 2:

Sie verwenden zu Testzwecken einen Apache Webserver auf einer Workstation. Sie müssen den Zugriff auf den Server über das Netzwerk verhindern. Mit welcher Direktive erreichen Sie das?

- ☐ A: `Port 81`
- ☐ B: `Listen 127.0.0.1:80`
- ☐ C: `Listen 82`
- ☐ D: `ServerType standalone`
- ☐ E: `ServerType inetd`

Frage 3:

Sie stellen bei einem hoch frequentierten Apache Webserver fest, dass der Zugriff auf Webseiten nur verzögert erfolgt. CPU-Ressourcen und Arbeitsspeicher sind jedoch nicht ausgelastet. Wie erhöhen Sie die Serverperformance? (Wählen Sie zwei Antworten.)

- ☐ A: Fügen Sie dem Server weitere IP-Adressen hinzu.
- ☐ B: Erhöhen Sie den Wert für `MinSpareServers`.
- ☐ C: Legen Sie den `ServerType` mit `standalone` fest.
- ☐ D: Legen Sie den `ServerType` mit `inetd` fest.
- ☐ E: Konfigurieren Sie einen anderen Port.

Frage 4:

Sie müssen einen Apache Webserver neu starten. Sie wollen hierfür ein Skript verwenden, das ausdrücklich für solche administrativen Eingriffe gedacht ist. Welches Kommando werden Sie verwenden? (Geben Sie ggf. benötigte Optionen mit an.)

Frage 5:

Sie müssen einer existierenden Passwortdatei eines Apache Webservers einen neuen Benutzer namens `ritchie` hinzufügen. Wie lautet das richtige Kommando, wenn der aktuelle Pfad dem `ServerRoot` entspricht?

- ☐ A: `bin/htpasswd passwortdatei ritchie`
- ☐ B: `bin/htpasswd -c passwortdatei ritchie`
- ☐ C: `useradd ritchie -m`
- ☐ D: `adduser ritchie`
- ☐ E: `echo ritchie >> .htaccess`

Frage 6:

Welche Direktive eines Apache Webservers legt fest, welches Verzeichnis aus der Sicht eines Benutzers das Hauptverzeichnis des Webservers ist?

- ☐ A: `ServerRoot`
- ☐ B: `ApacheRoot`
- ☐ C: `DocumentRoot`
- ☐ D: `Path`
- ☐ E: `httpd-Path`

Frage 7:

Sie müssen Fehlermeldungen eines virtuellen Hosts, der auf einem Apache Webserver läuft, in einer separaten Datei aufzeichnen. Welche Direktive werden Sie konfigurieren?

- ☐ A: `LogFile`
- ☐ B: `syslog.conf`
- ☐ C: `DocumentRoot`
- ☐ D: `CustomLog`
- ☐ E: `ErrorLog`

Frage 8:

Sie wollen mehrere Webseiten auf einem einzigen Apache Webserver hosten. Welche Möglichkeiten der Unterscheidung gibt es beim Zugriff auf die verschiedenen Webseiten? (Wählen Sie drei Antworten.)

- ☐ A: mehrere Verzeichnisse
- ☐ B: mehrere IP-Adressen
- ☐ C: mehrere HTML-Dateien
- ☐ D: mehrere TCP-Ports
- ☐ E: mehrere `ServerName`-Direktiven

Frage 9:

Sie müssen Ihren Benutzern den Zugriff auf Webseiten ermöglichen, die durch OpenSSL-Zertifikate gesichert wurden. Welchen Port müssen Sie auf der Firewall zulassen? (Geben Sie nur den Port ohne Protokoll an.)

_____

Frage 10:

Sie benötigen ein Zertifikat zur Absicherung einer Webseite. Mit welchem Programm können Sie das Zertifikat generieren?

- ☐ A: `httpd`
- ☐ B: `openssh`
- ☐ C: `openssl`
- ☐ D: `openswan`
- ☐ E: `https`

Frage 11:

Sie wollen die Speichermenge, die ein neuer Squid-Proxy auf der Festplatte belegen darf, konfigurieren. Welche Einstellung ändern Sie?

- ☐ A: `cache_mem`
- ☐ B: `cache_dir`
- ☐ C: `reply_body_max_size`
- ☐ D: `disk_usage`
- ☐ E: `access_log`

Frage 12:

Sie haben die Option `cache_dir` in der Datei *squid.conf* eines neuen Servers konfiguriert. Welches Kommando werden Sie anschließend ausführen? Vervollständigen Sie das Kommando!

`/usr/local/squid/sbin/squid`_____

Frage 13:

Sie müssen verhindern, dass Benutzer beim Zugriff auf einen Samba-Server Dateien mit einer bestimmten Dateierweiterung zu sehen bekommen. Wenn ein Benutzer auf eine solche Datei direkt zugreift, soll er diese dennoch öffnen können. Welchen Eintrag werden Sie in der Datei *smb.conf* verwenden?

- ☐ A: `lock directory`
- ☐ B: `security = user`
- ☐ C: `security = share`
- ☐ D: `hide files`
- ☐ E: `veto files`

Frage 14:

Bei einem Samba-Server soll verhindert werden, dass Benutzer auf Dateien mit einer bestimmten Dateierweiterung zugreifen können. Das soll auch dann gelten, wenn User Dateinamen direkt auswählen. Welchen Eintrag benötigen Sie hierfür?

- ☐ A: `lock directory`
- ☐ B: `security = user`
- ☐ C: `security = share`
- ☐ D: `hide files`
- ☐ E: `veto files`

Frage 15:

Sie haben umfangreiche Änderungen an der Datei *smb.conf* eines Servers durchgeführt. Bevor Sie den Samba-Server neu starten, wollen Sie die Datei auf Syntaxfehler hin überprüfen. Welches Programm können Sie dafür verwenden?

- ☐ A: `testparm`
- ☐ B: `hdparm`
- ☐ C: `nmblookup`
- ☐ D: `smbstatus`
- ☐ E: `smbcontrol`

Frage 16:

Wie heißt die Komponente eines Samba-Servers, die für Windows-Clients eine NetBIOS-Namensauflösung bereitstellt?

- ☐ A: *WINS*
- ☐ B: *Broadcast*
- ☐ C: *named*
- ☐ D: *nmbd*
- ☐ E: *smbd*

Frage 17:

Sie müssen einen Samba-Server zur Wartung herunterfahren. Mit welchem dafür vorgesehenen Kommando können Sie überprüfen, ob noch Benutzer mit den Shares des Servers verbunden sind?

Frage 18:

Sie müssen die Benutzerkonten von Windows-Computern den Benutzerkonten eines Samba-Servers zuordnen. Welche Datei müssen Sie bearbeiten?

- ☐ A: */etc/samba/passwd*
- ☐ B: */etc/samba/smbusers*
- ☐ C: */etc/passwd*
- ☐ D: */etc/shadow*
- ☐ E: */etc/samba/smb.conf*

Frage 19:

Sie verwenden einen Linux-Client-Computer und müssen in das lokale Verzeichnis */data* die Share »*files*« des Samba-Server »*fs01*« einhängen. Welche der folgenden Kommandos können Sie hier verwenden? (Wählen Sie zwei Antworten.)

- ☐ A: `mount -t smbfs //fs01/files /data`
- ☐ B: `mount -t smbfs /data //fs01/files`
- ☐ C: `smbmount -t smbfs //fs01/files /data`
- ☐ D: `smbmount /data //fs01/files`
- ☐ E: `smbmount //fs01/files /data`

Frage 20:

Sie haben auf einem Samba-Server zur Absicherung `iptables` verwendet. Nun kann kein Benutzer mehr auf die Shares des Servers zugreifen. Welche TCP-Ports sollten Sie auf der Firewall zulassen? (Wählen Sie zwei Antworten.)

- ☐ A: 80
- ☐ B: 25
- ☐ C: 443
- ☐ D: 139
- ☐ E: 445

Frage 21:

Sie planen den Einsatz eines NFS-Servers. Welcher Daemon muss zusätzlich laufen, damit Sie NFS nutzen können?

- ☐ A: `smbd`
- ☐ B: `nmbd`
- ☐ C: `portmap`
- ☐ D: `dhcpd`
- ☐ E: `inetd`

Frage 22:

Sie müssen ein Verzeichnis über NFS bereitstellen. Welche Konfigurationsdatei werden Sie bearbeiten?

- ☐ A: */etc/samba/smb.conf*
- ☐ B: */etc/nfs.conf*
- ☐ C: */etc/exportfs*
- ☐ D: */etc/fstab*
- ☐ E: */etc/exports*

Frage 23:

Sie haben die Datei */etc/exports* erweitert. Welches Kommando werden Sie ausführen, damit sich die Änderung sofort auswirkt? (Geben Sie das Programm und ggf. benötigte Optionen an.)

Frage 24:

Sie wollen das Verzeichnis */files* temporär exportieren, sodass nur ein Computer mit dem Namen `desktop1` schreibend darauf zugreifen kann. Welches Kommando verwenden Sie?

- ☐ A: `exportfs -o rw desktop1:/files`
- ☐ B: `mount -t nfs desktop1:/files /mnt/files`
- ☐ C: `vi /etc/exports`
- ☐ D: `exportfs -o r desktop1:/files`
- ☐ E: `exportfs -o rw //desktop1/files`

Frage 25:

Sie müssen überprüfen, von welchen Client-Computern aus auf einen NFS-Server zugegriffen wird und welche Verzeichnisse hierbei eingehängt wurden. Welches Kommando werden Sie verwenden?

- ☐ A: `showmount -a`
- ☐ B: `showmount -r`
- ☐ C: `showmount -d`
- ☐ D: `rpcinfo`
- ☐ E: `nfsstat`

Frage 26:

Sie benötigen eine Aufstellung der NFS-Aktivitäten auf einem Server. Hierbei interessieren Sie sich besonders für die Anzahl der auf dem Server erstellten Dateien und Verzeichnisse. Welches Kommando gibt die von Ihnen benötigten Informationen aus?

- ☐ A: `showmount -a`
- ☐ B: `showmount -r`
- ☐ C: `showmount -d`
- ☐ D: `rpcinfo`
- ☐ E: `nfsstat`

Frage 27:

Sie müssen unabhängig von exportierten Dateisystemen global festlegen, von welchen IP-Subnetzen aus auf einen NFS-Server zugegriffen werden darf. Welche Methoden kommen hier infrage? (Wählen Sie zwei.)

- ☐ A: `iptables`
- ☐ B: `portmapper`
- ☐ C: Konfiguration in */etc/exports*
- ☐ D: Konfiguration in */etc/hosts*
- ☐ E: TCP-Wrapper

Frage 28:

Sie wollen sehen, von welchen Computern aus auf einen NFS-Server zugegriffen wird. Welches Kommando ist hierfür gedacht? (Geben Sie ggf. keine Optionen an, sondern nur das einfache Kommando.)

Frage 29:

Sie konfigurieren die Firewall eines DHCP-Servers. Welche Ports sollten Sie auf dem DHCP-Server zulassen?

- ☐ A: 68 eingehend, 67 ausgehend
- ☐ B: 67 eingehend, 68 ausgehend
- ☐ C: 53 eingehend, ausgehend dynamisch
- ☐ D: 67 eingehend, ausgehend dynamisch
- ☐ E: 139 eingehend, 445 eingehend

Frage 30:

Welche der folgenden Konfigurationseinstellungen der Datei *dhcpd.conf* werden nicht an einen DHCP-Client-Computer übermittelt? (Wählen Sie zwei Antworten.)

- ☐ A: `ddns-updates on`
- ☐ B: `option domain-name-servers`
- ☐ C: `option netbios-name-servers`
- ☐ D: `ignore client-updates`
- ☐ E: `option routers`

Frage 31:

Welche der folgenden Zeilen sind keine gültigen Komponenten einer Reservierung in der Datei *dhcpd.conf*? (Wählen Sie zwei Antworten.)

- ☐ A: `host client4`
- ☐ B: `fixed address 192.168.50.104;`
- ☐ C: `fixed-address 192.168.50.104;`
- ☐ D: `hardware ethernet 00:1c:bf:c5:e1:8e:f7:80;`
- ☐ E: `hardware ethernet 00:1c:bf:c5:e1:8e;`

Frage 32:

Sie müssen feststellen, wann die Lease eines DHCP-Clients abläuft. Welche Datei können Sie auf dem DHCP-Server einsehen, um diese Information zu bekommen? (Geben Sie nur die Datei ohne Pfad an.)

_____

Frage 33:

Sie müssen verhindern, dass die Benutzer der Computer, die Sie betreuen, einfache Passwörter verwenden. Welches PAM-Modul werden Sie in Ihre Konfiguration mit einbeziehen?

- ☐ A: *pam_env.so*
- ☐ B: *pam_unix.so*
- ☐ C: *pam_cracklib.so*
- ☐ D: *pam_permit.so*
- ☐ E: *pam_deny.so*

Frage 34:

Sie wollen die Authentifizierung in Ihrem Netzwerk umstellen, sodass LDAP zur Speicherung der Benutzerkonten verwendet wird. Die nötigen PAM-Module haben Sie bereits eingebunden. Welche Datei müssen Sie noch ändern, damit PAM zuerst auf die LDAP-Datenbank zugreift und erst danach auf lokale Dateien?

- ☐ A: */etc/pam.d/common-password*
- ☐ B: */etc/pam.d/common-auth*
- ☐ C: */etc/pam.conf*
- ☐ D: */etc/nsswitch.conf*
- ☐ E: */etc/ldap/ldap.conf*

Frage 35:

Sie planen den Einsatz der Bibliothek *pam_cracklib.so*. Welcher der folgenden PAM-Verwaltungsgruppen werden Sie diese Bibliothek zuordnen?

- ☐ A: `auth`
- ☐ B: `account`
- ☐ C: `password`
- ☐ D: `session`
- ☐ E: keiner der genannten

Frage 36:

Sie versuchen sich erstmalig an einem System mit einem Benutzerkonto anzumelden, das in einer LDAP-Datenbank gespeichert wurde. Die Anmeldung schlägt jedoch fehl. Was sind wahrscheinliche Ursachen für diesen Authentifizierungsfehler? (Wählen Sie zwei Antworten.)

- ☐ A: */etc/nsswitch.conf* wurde nicht richtig konfiguriert.
- ☐ B: */etc/resolv.conf* enthält einen Fehler.
- ☐ C: *pam_ldap.so* wurde nicht installiert bzw. konfiguriert.
- ☐ D: Der Benutzer existiert nicht.
- ☐ E: Der Benutzer existiert sowohl lokal als auch in LDAP.

Frage 37:

In welcher Konfigurationsdatei können Sie festlegen, dass PAM zur Authentifizierung zuerst LDAP und erst im zweiten Anlauf die Datei *passwd* konsultieren soll? (Geben Sie die Datei und den Pfad an.)

Frage 38:

Welches der folgenden Verzeichnisse enthält üblicherweise die Bibliotheken der PAM-Module?

- ☐ A: */etc/pam.d*
- ☐ B: */var/lib/pam*
- ☐ C: */lib/pam*
- ☐ D: */usr/pam/modules*
- ☐ E: */lib/security*

Frage 39:

Welche der folgenden Zeilen enthält einen gültigen DN (Distinguished Name) nach Standard X.500?

- ☐ A: `ronnie@example.org`
- ☐ B: `CN=Ronnie,DC=example,DC=org`
- ☐ C: `www.example.org`
- ☐ D: `DN=Ronnie,DC=example,DC=org`
- ☐ E: `example.org`

Frage 40:

Sie vermuten eine fehlerhafte Namensauflösung bei einem DNS-Server und wollen nun den Namen *www.lpic-2.de* in eine IP-Adresse auflösen. Es soll bei der Diagnose der DNS-Server mit der IP-Adresse 172.16.0.10 verwendet werden – unabhängig davon, welcher DNS-Server in der Datei *resolv.conf* des Testsystems eingetragen ist. Welche der folgenden Kommandos können Sie verwenden, um dieses Ziel zu erreichen? (Wählen Sie zwei Antworten.)

- ☐ A: `nslookup www.lpic-2.de`
- ☐ B: `dig @172.16.0.10 www.lpic-2.de`
- ☐ C: `host www.lpic-2.de`
- ☐ D: `ping www.lpic-2.de`
- ☐ E: `dig www.lpic-2.de @172.16.0.10`

Frage 41:

Welches Programm wird verwendet, wenn das Passwort eines Benutzers geändert werden soll, dessen Benutzerkonto in OpenLDAP gespeichert wurde? (Geben Sie nur das Programm, ohne Optionen und Parameter an.)

Frage 42:

Sie wollen einem OpenLDAP-Verzeichnis neue Objekte hinzufügen. Welche Kommandos können Sie für diese Aufgabe verwenden? (Wählen Sie zwei Antworten.)

- ☐ A: `ldappasswd`
- ☐ B: `ldapadd`
- ☐ C: `ldapsearch`
- ☐ D: `addldap`
- ☐ E: `ldapmodify`

Frage 43:

Bei welchen der folgenden Produkte handelt es sich um reine MTAs? (Wählen Sie zwei Antworten.)

- ☐ A: *Postfix*
- ☐ B: *Sendmail*
- ☐ C: *Courier*
- ☐ D: *Dovecot*
- ☐ E: *Cyrus*

Frage 44:

Sie müssen für einen Benutzer zusätzliche E-Mail-Adressen zur Verfügung stellen. Welche Aktionen sind erforderlich, wenn Sie den E-Mail-Client des Benutzers nicht umkonfigurieren wollen? (Wählen Sie zwei Antworten.)

- ☐ A: Erstellen Sie weitere Benutzerkonten.
- ☐ B: Modifizieren Sie die Datei *.forward* des Benutzers.
- ☐ C: Führen Sie `newaliases` aus.
- ☐ D: Erstellen Sie Einträge in der Datei */etc/aliases*.
- ☐ E: Diese Konfiguration ist nicht möglich.

Frage 45:

Sie haben Änderungen an der Datei */etc/named.conf* vorgenommen. Bevor Sie BIND anweisen die Konfiguration neu einzulesen, wollen sie sicherstellen, dass Ihnen bezüglich der Syntax kein Fehler unterlaufen ist. Welches Kommando werden Sie zur Überprüfung ausführen?

Frage 46:

Nach Änderungen an mehreren Zonendateien wollen Sie überprüfen, ob Ihnen keine Syntaxfehler bei der Eingabe unterlaufen sind. Welches der folgenden Programme können Sie zu Überprüfung verwenden?

- ☐ A: `named-compilezone`
- ☐ B: `named-checkzone`
- ☐ C: `named-checkconf`
- ☐ D: `rndc`
- ☐ E: `nslookup`

Frage 47:

Sie müssen *Postfix* so konfigurieren, dass E-Mails, die an die Domäne *example.org* adressiert sind, als lokal erachtet werden. Welche Einstellung werden Sie ändern?

- ☐ A: `myorigin` in *main.cf*
- ☐ B: `mynetworks` in *master.cf*
- ☐ C: `mynetworks` in *main.cf*
- ☐ D: `mydestination` in *main.cf*
- ☐ E: `mydestination` in *master.cf*

Frage 48:

Sie müssen die Funktionstüchtigkeit eines SMTP-Servers prüfen. Welche der folgenden Programme können Ihnen dabei helfen? (Wählen Sie zwei.)

- ☐ A: `traceroute`
- ☐ B: `ping`
- ☐ C: `telnet`
- ☐ D: `nslookup`
- ☐ E: `netcat`

Frage 49:

Welche der folgenden Komponenten eines E-Mail-Servers ist für die Filterung der Nachrichten zuständig (z. B. zur Spam-Bekämpfung)?

- ☐ A: *Postfix*
- ☐ B: *Sendmail*
- ☐ C: *Dovecot*
- ☐ D: *Sieve*
- ☐ E: *Courier*

Frage 50:

In einer DNS-Zone stoßen Sie auf einen Eintrag vom Typ TLSA. Für welchen Mechanismus wird dieser Eintrag verwendet?

- ☐ A: *MX*
- ☐ B: *CNAME*
- ☐ C: *TSIG*
- ☐ D: *DNSSEC*
- ☐ E: *DANE*

Frage 51:

Welches der folgenden Kommandos verwirft Pakete aus einem angegebenen öffentlichen IPv6-Netzwerk, ohne den Absender über die Ablehnung des Pakets zu informieren?

- ☐ A: `ip6tables -A INPUT -s 2001:db8:0:c4f8::/64 -j REJECT`
- ☐ B: `iptables -A INPUT -s 2001:db8:0:c4f8::/64 -j DROP`
- ☐ C: `ip6tables -A INPUT -s 2001:db8:0:c4f8::/64 -j DROP`
- ☐ D: `iptables -A INPUT -s 2001:db8:0:c4f8::/64 -j REJECT`
- ☐ E: `ip6tables -A OUTPUT -s 2001:db8:0:c4f8::/64 -j DROP`

Frage 52:

Um einen definierten Ausgangspunkt für eine neue Firewallkonfiguration zu erhalten, wollen Sie zunächst alle Regeln aller IPv6-bezogenen Filter-Ketten eines Systems löschen. Welchen Befehl (ggf. mit Optionen und Parametern) werden Sie eingeben?

Frage 53:

Um eine nicht autorisierte Verwendung eines *Postfix*-Servers einzuschränken, soll die Weiterleitung (Relay) von E-Mails auf lokale Netzwerke beschränkt werden. Welche Einstellungen in der Datei *main.cf* werden Sie konfigurieren? (Wählen Sie zwei Antworten.)

- ☐ A: `mydestination`
- ☐ B: `myorigin`
- ☐ C: `smtp_relay_restrictions`
- ☐ D: `myhostname`
- ☐ E: `mynetworks`

Frage 54:

Sie müssen einen Dovecot-Server anpassen, sodass E-Mails bei der Übertragung immer verschlüsselt werden. Welche Konfiguration in der Datei */etc/dovecot/dovecot.conf* stellt die Erfüllung dieser Anforderung sicher?

- ☐ A: `protocols: imap imaps pop3 pop3s`
- ☐ B: `protocols: imaps pop3s`
- ☐ C: `protocols: imap pop3`
- ☐ D: `disable_plaintext_auth: no`
- ☐ E: `disable_plaintext_auth: yes`

Frage 55:

Sie müssen das auf dem entfernten Host *compi.example.com* installierte Betriebssystem identifizieren. Welches Kommando können Sie verwenden?

- ☐ A: `dig compi.example.com`
- ☐ B: `ping compi.example.com`
- ☐ C: `nmap compi.example.com -p 139`
- ☐ D: `nslookup compi.example.com`
- ☐ E: `nmap compi.example.com -O`

Frage 56:

Sie müssen dafür Sorge tragen, dass E-Mails aus dem Internet Ihren *Postfix*-Mailserver erreichen. Ein DNS-Eintrag welchen Typs ist im Internet notwendig, um dieses Ziel zu erreichen?

- ☐ A: `CNAME`
- ☐ B: `PTR`
- ☐ C: `SOA`
- ☐ D: `NS`
- ☐ E: `MX`

Frage 57:

Welche der folgenden Zeilen enthält eine gültige Direktive, damit *nginx* als Reverse-Proxy fungiert, wenn er selbst an Port 80 lauschen soll?

- ☐ A: `location / { proxy_pass http://127.0.0.1:8000; }`
- ☐ B: `location / { proxy_pass http://127.0.0.1:80; }`
- ☐ C: `listen 80`
- ☐ D: `listen 8000`
- ☐ E: `index index.html;`

Frage 58:

Sie konfigurieren die Firewall-Einstellungen eines Routers und wollen als Basis alle bestehenden Firewall-Filterregeln verwerfen. Welches Kommando werden Sie ausführen? (Geben Sie das Kommando und ggf. Optionen an.)

Frage 59:

Bei der Konfiguration eines Routers wollen Sie als sicheren Ausgangspunkt zunächst alle Pakete verwerfen, die der Router sonst weiterleiten würde. Pakete sollen mit einer Fehlermeldung an den Client verworfen werden. Welches Kommando werden Sie ausführen?

- ☐ A: `iptables -t filter -A INPUT -j DROP`
- ☐ B: `iptables -t filter -A FORWARD -j DROP`
- ☐ C: `iptables -t filter -A FORWARD -j REJECT`
- ☐ D: `iptables -t filter -F FORWARD`
- ☐ E: `iptables -t filter -L FORWARD`

Frage 60:

Ein Router soll IP-Masquerading für ein Netzwerksegment mit der IP-Adresse 192.168.40.0/24 unterstützen. Die dem Internet zugewandte Schnittstelle des Routers ist ppp0. Welches Kommando sollten Sie verwenden?

☐ A: `iptables -t nat -I PREROUTING \`
`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

☐ B: `iptables -t nat -I POSTROUTING \`
`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

☐ C: `iptables -t filter -I PREROUTING \`
`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

☐ D: `iptables -t nat -I INPUT \`
`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

☐ E: `iptables -t filter -I POSTROUTING \`
`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

Frage 61:

Sie stellen eine DoS-Attacke auf einen Ihrer Router fest. Die IP-Adresse des Angreifers stammt aus dem inneren Netzwerk und ist 192.168.40.87. Wie können Sie den Angriff abwenden? Wählen Sie zwei funktionierende Möglichkeiten aus!

☐ A: `iptables -t filter -I OUTPUT -s 192.168.40.87 -j DROP`

☐ B: `route add 192.168.40.87 gw 127.0.0.1`

☐ C: `iptables -t nat -I INPUT -s 192.168.40.87 -j DROP`

☐ D: `iptables -t filter -I INPUT -s 192.168.40.87 -j DROP`

☐ E: `iptables -t filter -I FORWARD -s 192.168.40.87 -j DROP`

Frage 62:

Sie müssen einen SMTP-Server zum Internet hin veröffentlichen, indem Sie den entsprechenden Port auf der äußeren Schnittstelle Ihrer Firewall weiterleiten. Welches Kommando werden Sie verwenden, wenn die äußere Netzwerkschnittstelle des Internet-Routers `ppp0` und die IP-Adresse des SMTP-Servers 192.168.40.27 ist?

☐ A: `iptables -t nat -I PREROUTING -i ppp0 -p tcp \`
`--dport 25 -j DNAT --to-destination 192.168.40.27:25`

☐ B: `iptables -t nat -I PREROUTING -i ppp0 -p tcp \`
`--dport 110 -j DNAT --to-destination 192.168.40.27:110`

☐ C: `iptables -t nat -I POSTROUTING -i ppp0 -p tcp \`
`--dport 25 -j DNAT --to-destination 192.168.40.27:25`

☐ D: `iptables -t filter -I PREROUTING -i ppp0 -p tcp \`
`--dport 25 -j DNAT --to-destination 192.168.40.27:25`

☐ E: `iptables -t nat -I PREROUTING -i ppp0 -p tcp \`
`--dport 25 -j MASQUERADE --to-destination 192.168.40.27:25`

Frage 63:

Sie müssen die Kommunikation mit einem FTP-Server absichern, der sich hinter einer Firewall befindet. Welche eingehenden TCP-Ports sollten in den Firewall-Einstellungen zugelassen werden, wenn der Server im passiven Modus läuft? (Wählen Sie zwei Antworten.)

☐ A: 20

☐ B: 21

☐ C: 22

☐ D: 23

☐ E: Ein weiterer Port ist nicht erforderlich.

Frage 64:

Welcher der folgenden FTP-Server verwendet einen Perl-basierten Wrapper, um aus einer Datei die Optionen für den Start des Daemons zu generieren?

☐ A: Alle FTP-Server verwenden einen solchen Wrapper.

☐ B: Kein FTP-Server verwendet einen solchen Wrapper.

☐ C: vsftpd

☐ D: ProFTPD

☐ E: Pure-FTPd

Frage 65:

Sie müssen Dateien auf einen Server kopieren, auf dem ein *vsftpd* läuft. Es müssen Dateien nach */usr* kopiert werden, aber der Server begrenzt Sie auf Ihr Home-Directory. Welche Einstellung müssen Sie in der Datei *vsftpd.conf* ändern?

- ☐ A: `anon_root=/anonymous`
- ☐ B: `anonymous_enable=YES`
- ☐ C: `local_enable=YES`
- ☐ D: `chroot_local_user=YES`
- ☐ E: `anon_upload_enable=YES`

Frage 66:

Sie betreuen einen Server, auf dem Pure-FTPd läuft. Sie wollen sehen, welche FTP-Verbindungen gerade zu diesem Server bestehen. Welches Werkzeug aus dem Lieferumfang von Pure-FTPd können Sie hierfür verwenden? (Geben Sie keine Optionen mit an.)

Frage 67:

Sie wollen SSH verwenden, um ein Programm auf einem entfernten Rechner zu starten. Das Programm basiert auf der Bibliothek GTK+. Wie beginnen Sie das Kommando zum Verbindungsaufbau?

- ☐ A: `ssh -6`
- ☐ B: `ssh -l`
- ☐ C: `ssh -p`
- ☐ D: `ssh -L`
- ☐ E: `ssh -X`

Frage 68:

Sie müssen sich mit einem älteren SSH-Server verbinden. Die laufende Version des `sshd` ist Ihnen unbekannt. Aus Sicherheitsgründen müssen Sie sicherstellen, dass die Verbindung nicht zustande kommt, wenn der Server lediglich die SSH-Version 1 unterstützt. Wie beginnen Sie das Kommando zum Verbindungsaufbau?

- ☐ A: `ssh -2`
- ☐ B: `ssh -4`
- ☐ C: `ssh -6`
- ☐ D: `ssh -l`
- ☐ E: `ssh -p`

Frage 69:

Sie haben einen Server nach einem Festplattenfehler neu installiert. Bei einem Versuch, den Server von Ihrem Arbeitsplatz aus per SSH zu kontaktieren, erhalten Sie lediglich eine Fehlermeldung:

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now ...
```

Welche Datei müssen Sie bearbeiten, damit die Anmeldung wieder funktioniert?

- ☐ A: *authorized_keys* auf dem Client-Computer
- ☐ B: *known_hosts* auf dem Client-Computer
- ☐ C: *authorized_keys* auf dem Server-Computer
- ☐ D: *known_hosts* auf dem Server-Computer
- ☐ E: *hosts.allow* auf dem Server-Computer

Frage 70:

Sie wollen neue RSA-Hostkeys für ein System generieren. Welches Programm werden Sie verwenden? (Geben Sie nur das Programm ohne Optionen an.)

Frage 71:

Bei welchen der folgenden Dateien handelt es sich um die Hauptkonfigurationsdatei des SSH-Server-Daemons?

- ☐ A: *ssh_host_dsa_key.pub*
- ☐ B: *ssh_config*
- ☐ C: *sshd_config*
- ☐ D: *ssh_host_dsa_key*
- ☐ E: *ssh_host_rsa_key*

Frage 72:

Aus Gründen der Sicherheit wollen Sie den direkten `root`-Zugriff auf einen Server über SSH verhindern. Welchen Eintrag werden Sie in welcher Konfigurationsdatei erstellen?

- ☐ A: `AllowRootLogin no` in der Datei *ssh_config*
- ☐ B: `AllowRootLogin no` in der Datei *sshd_config*
- ☐ C: `PermitRootLogin no` in der Datei *ssh_config*
- ☐ D: `PermitRootLogin no` in der Datei *sshd_config*
- ☐ E: `PermitRootLogin yes` in der Datei *ssh_config*

Frage 73:

Um übermäßige Anmeldeversuche via SSH auf einem Ihrer Server zu reduzieren, wollen Sie diesen so konfigurieren, dass der *sshd* an einem anderen Port lauscht. Welche Konfigurationsdatei werden Sie modifizieren? Geben Sie den kompletten Pfad und die Datei an.

Frage 74:

Sie müssen SSH auf einem Server so umkonfigurieren, dass eine Authentifizierung nur noch mit Schlüsseln möglich ist. Welche Option werden Sie modifizieren?

- ☐ A: `PermitRootLogin no`
- ☐ B: `Protocol 2,1`
- ☐ C: `Port 22`
- ☐ D: `AllowUsers harald`
- ☐ E: `PasswordAuthentication yes`

Frage 75:

Welche der folgenden Konfigurationsdateien steuern das Verhalten von TCP-Wrappern? (Wählen Sie zwei Antworten.)

- ☐ A: */etc/services*
- ☐ B: */etc/hosts.allow*
- ☐ C: */etc/hosts*
- ☐ D: */etc/ssh/ssh_config*
- ☐ E: */etc/hosts.deny*

Frage 76:

Welche der folgenden Produkte verwenden die Bibliothek *libwrap*? (Wählen Sie zwei Antworten.)

- ☐ A: `iptables`
- ☐ B: `inetd`
- ☐ C: `xinetd`
- ☐ D: `tcpd`
- ☐ E: `dhcpd`

Frage 77:

Welche der folgenden Organisationen befassen sich mit Sicherheitslücken in Softwareprodukten? (Wählen Sie drei Antworten.)

- ☐ A: *CERT*
- ☐ B: *Bugtraq*
- ☐ C: *IANA*
- ☐ D: *ISC*
- ☐ E: *CIAC*

Frage 78:

Welche der folgenden Programme können Sie verwenden, wenn Sie eine Liste der auf einem Server geöffneten Ports benötigen? (Wählen Sie zwei Antworten.)

- ☐ A: `telnet`
- ☐ B: `netcat`
- ☐ C: `netstat`
- ☐ D: `ifconfig`
- ☐ E: `nmap`

Frage 79:

Welche der folgenden Programme sind in der Lage, Einbruchsversuche zu erkennen oder zu verhindern? (Wählen Sie zwei Antworten.)

- ☐ A: `nmap`
- ☐ B: `ipchains`
- ☐ C: `iptables`
- ☐ D: `fail2ban`
- ☐ E: `snort`

Frage 80:

Nennen Sie das Target, das Sie in `iptables` verwenden müssen, wenn Pakete ohne eine Fehlermeldung an den Client verworfen werden sollen.

Frage 81:

Ein Benutzer ist nach mehrfacher Fehleingabe seines Passworts via SSH von einem IDS gesperrt worden. Welche Kommandos können Sie verwenden, um zu überprüfen, ob die Sperrung weiterhin besteht? (Wählen Sie drei Möglichkeiten.)

- ☐ A: `netstat -an`
- ☐ B: `iptables -L`
- ☐ C: `iptables-save`
- ☐ D: `fail2ban-client status ssh`
- ☐ E: `nmap localhost`

Frage 82:

Sie planen den Einsatz einer Software, auf die von Remote-Standorten aus über den TCP-Port 2704 zugegriffen werden soll. Sie müssen vor der Anschaffung der Software testen, ob der Zugriff auf diesen Port durch alle Firewalls hindurch möglich ist. Welches Programm kann hierbei helfen?

- ☐ A: `netcat`
- ☐ B: `netstat`
- ☐ C: `fail2ban`
- ☐ D: `iptables`
- ☐ E: `nmap`

Frage 83:

Welche der folgenden Programme können Sie verwenden, um die Funktionalität eines DNS-Servers zu überprüfen? (Wählen Sie drei der angebotenen Antworten.)

- ☐ A: `route`
- ☐ B: `nslookup`
- ☐ C: `ping`
- ☐ D: `dig`
- ☐ E: `host`

Frage 84:

Die Datenbank eines OpenLDAP-Servers soll neu indexiert werden. Welches Kommando (ohne Optionen oder Parameter) kann hierfür verwendet werden?

Frage 85:

Welche Technologie kommt zum Einsatz, wenn mehr als eine HTTPS-Seite auf einem Apache-Webserver gehostet wird, der über nur eine einzige IP-Adresse verfügt?

- ☐ A: *Round Robin*
- ☐ B: *SNI*
- ☐ C: *DANE*
- ☐ D: *SSL*
- ☐ E: *TSIG*

Frage 86:

Sie wollen die Konfigurationseinstellungen eines Dovecot-Servers einsehen. Sie sind nur an den Einstellungen interessiert, die nicht standardmäßig vorhanden sind. Welches Kommando sollten Sie verwenden?

Frage 87:

Welche der folgenden sind neue, verbesserte Eigenschaften von NFS Version 4? (Wählen Sie drei Antworten.)

- ☐ A: Verwendung von UDP als Transportprotokoll
- ☐ B: integrierte Verschlüsselung
- ☐ C: bessere Performance
- ☐ D: Authentifizierung auf Basis des Benutzers
- ☐ E: Authentifizierung auf Basis des Computers

Frage 88:

Sie müssen für eine Samba-Share sicherstellen, dass Dateien, die Benutzer in dieser Share ablegen, in jedem Fall von beliebigen Benutzern gelesen werden können. Andere Berechtigungseinstellungen sollen durch den Eingriff nicht beeinflusst werden. Welche Einstellung werden Sie in der Datei *smb.conf* für diese Share vornehmen?

- ☐ A: `create mode = 0664`
- ☐ B: `force create mode = 0444`
- ☐ C: `create mask = 0664`
- ☐ D: `directory mode = 0555`
- ☐ E: `force create mode = 0660`

Frage 89:

Sie haben *Postfix* so konfiguriert, dass Verbindungen mit TLS gesichert werden. An welchem Port sollte der Server nun lauschen? (Geben Sie nur den numerischen Wert ohne die Angabe des Transportprotokolls an.)

Frage 90:

Sie müssen *Postfix* so konfigurieren, dass alle E-Mails, deren Ziele außerhalb Ihrer Domäne liegen, an einen *Smarthost* weitergeleitet werden. Welche Option werden Sie in welcher Datei konfigurieren?

- ☐ A: `relayhost` in *master.cf*
- ☐ B: `relay` in *main.cf*
- ☐ C: `smarthost` in *master.cf*
- ☐ D: `smarthost` in *main.cf*
- ☐ E: `relayhost` in *main.cf*

Frage 91:

Sie planen die automatische Vergabe von internettauglichen IPv6-Adressen an Clientcomputer. Broadcastverkehr, der von den Clientcomputern ausgeht, soll nach Möglichkeit vermieden werden. Welche Funktionalität werden Sie einsetzen?

- ☐ A: *isc-dhcp3-server*
- ☐ B: *DHCP-Relay*
- ☐ C: *radvd*
- ☐ D: *bind*
- ☐ E: *link-lokale* Adressen

Frage 92:

Bei welchen der folgenden Produkte handelt es sich um Mailfilter? (Wählen Sie zwei Antworten.)

- ☐ A: *postfix*
- ☐ B: *procmail*
- ☐ C: *exim*
- ☐ D: *sieve*
- ☐ E: *dovecot*

Frage 93:

Für welche der folgenden Dienste kann *nginx* als Proxy fungieren? (Wählen Sie alle zutreffenden.)

- ☐ A: *SMB*
- ☐ B: *IMAP*
- ☐ C: *RDP*
- ☐ D: *POP*
- ☐ E: *HTTP*

Frage 94:

Ein Portscan an einem Server ergibt, dass die in den Antworten vorgegebenen Ports des Servers abgehört werden. Welche der Ports geben einen Hinweis darauf, dass hier wahrscheinlich ein Samba-Server läuft? (Wählen Sie zwei Antworten.)

- ☐ A: `22/tcp`
- ☐ B: `443/tcp`
- ☐ C: `80/tcp`
- ☐ D: `139/tcp`
- ☐ E: `445/tcp`

Frage 95:

Welches Kommando wird verwendet, um die Domänen-Funktionen eines Samba-Servers zu steuern?

- ☐ A: `samba-tool`
- ☐ B: `testparm`
- ☐ C: `smbcontrol`
- ☐ D: `smbstatus`
- ☐ E: `rndc`

Frage 96:

Welche Aufgabe hat der Daemon *winbindd*?

- ☐ A: DNS-Namensauflösung für Windows-Clients
- ☐ B: Authentifizierung von Benutzern aus NT-Domänen
- ☐ C: Bereitstellung von Dateien für Windows-Clients
- ☐ D: NetBIOS-Namensauflösung
- ☐ E: Binden von Windowsdiensten

Frage 97:

Sie beabsichtigen, den Inhalt einer OpenLDAP-Datenbank in eine LDIF-Datei zu exportieren. Welches der folgenden Programme sollten Sie hierfür auswählen?

- ☐ A: `ldapadd`
- ☐ B: `slapadd`
- ☐ C: `slapcat`
- ☐ D: `slapindex`
- ☐ E: `slapdump`

Frage 98:

Sie müssen eine OpenLDAP-Datenbank neu indexieren. Welches Kommando werden Sie hierfür verwenden?

Frage 99:

Sie planen die Abschaltung eines NFS-Servers, von dem Sie vermuten, dass dieser nicht mehr genutzt wird. Sie müssen zunächst prüfen, ob noch Clients mit dem Server verbunden sind. Welche Kommandos können Ihnen Hinweise auf NFS-Aktivitäten auf diesem Server geben? (Wählen Sie zwei Antworten.)

- ☐ A: `watch rpcinfo`
- ☐ B: `watch nfsstat -s -o nfs`
- ☐ C: `watch showmount`
- ☐ D: `watch smbstatus`
- ☐ E: `netstat -an |grep 2049`

Frage 100:

Welche der hier gelisteten Zeilen enthält eine gültige Option zur Zuweisung eines DNS-Servers an einen DHCP-Client?

- ☐ A: `option domain-name "example.com";`
- ☐ B: `option domain-name-server 192.168.50.1;`
- ☐ C: `option DNS-servers 192.168.50.1;`
- ☐ D: `option domain-name-servers 192.168.50.1;`
- ☐ E: `option name-servers 192.168.50.1;`

Frage 101:

Welche der folgenden Mechanismen unterstützen die Authentifizierung von Benutzern aus Active-Directory-Domänen mittels PAM? (Wählen Sie zwei Antworten.)

- ☐ A: *sssd*
- ☐ B: *smbusers*
- ☐ C: *nsswitch.conf*
- ☐ D: *passwd*
- ☐ E: *winbind*

Frage 102:

In welcher Konfigurationsdatei wird festgelegt, dass eine Authentifizierung mittels OpenLDAP gegenüber einer lokalen Anmeldung über die Datei *passwd* bevorzugt durchgeführt werden soll? (Nennen Sie den kompletten Pfad zur Datei.)

Frage 103:

Welche Module von Apache müssen wahlweise geladen sein, damit eine Authentifizierung des Zugriffs basierend auf den IP-Adressen der Clients durchgeführt werden kann? (Wählen Sie zwei.)

- ☐ A: *mod_auth_basic*
- ☐ B: *mod_authz_host*
- ☐ C: *mod_access_compat*
- ☐ D: *mod_authz_user*
- ☐ E: *mod_ssl*

Frage 104:

Sie müssen einem LDAP-Verzeichnis neue Objekte hinzufügen. Welches der folgenden Programme sollten Sie vorzugsweise verwenden?

- ☐ A: `slapadd`
- ☐ B: `slapcat`
- ☐ C: `slapindex`
- ☐ D: `slapd-config`
- ☐ E: `ldapadd`

Frage 105:

Sie planen die automatische Vergabe von IPv4-Adressen für 50 Netzwerksegmente. Aus Kostengründen soll nicht jedes Netzwerksegment einen eigenen DHCP-Server bekommen. Welche Funktionalität können Sie den Routern der jeweiligen Netzwerke hinzufügen, um DHCP-Anfragen von Clients an einen DHCP-Server weiterzuleiten?

- ☐ A: *isc-dhcp3-server*
- ☐ B: *DHCP-Relay*
- ☐ C: *radvd*
- ☐ D: *BIND*
- ☐ E: *link-lokale* Adressen

Frage 106:

Ein Eintrag in der Zugriffssteuerungsliste (ACL) eines OpenLDAP-Servers beginnt mit folgender Zeile:

```
access to *
```

Welche der folgenden Einträge sollten Sie löschen, weil es sich jeweils um Sicherheitsbedrohungen für die gesamte Datenbank handelt? (Wählen Sie zwei.)

- ☐ A: `by self write`
- ☐ B: `by * write`
- ☐ C: `by anonymous auth`
- ☐ D: `by * read`
- ☐ E: `by anonymous manage`

Frage 107:

Sie müssen die dynamischen Aktualisierungen von Client-Computern in DNS in Ihrem Netzwerk absichern. Sie wollen keine PKI-basierte Methode verwenden, weil die Implementierung zu zeitaufwändig wäre. Welche Technologie setzen Sie stattdessen ein?

- ☐ A: `DNSSEC`
- ☐ B: `TSIG`
- ☐ C: `RRSIG`
- ☐ D: `DNSKEY`
- ☐ E: `KEY`

Frage 108:

Sie planen den Einsatz von DNSSEC. Welchen der folgenden Eintragstypen werden Sie verwenden, um den öffentlichen Schlüssel des Servers zu veröffentlichen?

- ☐ A: `DNSKEY`
- ☐ B: `RRSIG`
- ☐ C: `SIG`
- ☐ D: `KEY`
- ☐ E: `NS`

Frage 109:

Sie konfigurieren einen DNS-Server, der über mehrere Netzwerkschnittstellen verfügt. Es soll festgelegt werden, dass der Server lediglich die Schnittstelle mit der IP-Adresse 192.168.5.1 und das Loopback-Device abhorcht. Wie müssen Sie die entsprechende Option konfigurieren, wenn der standardmäßige Port für DNS verwendet werden soll?

- ☐ A: `listen-on port 53 { 127.0.0.1; 192.168.5.1; };`
- ☐ B: `listen-on port 53 {127.0.0.1; 192.168.5.1;};`
- ☐ C: `listen-on port 25 { 127.0.0.1; 192.168.5.1; };`
- ☐ D: `allow-query { 127.0.0.1; 192.168.5.0/24; };`
- ☐ E: `allow-query {127.0.0.1; 192.168.5.0/24;};`

Frage 110:

Sie konfigurieren einen DNS-Server, der in einem segmentierten Netzwerk eingesetzt wird. Der Zugriff auf den Server muss nun so begrenzt werden, dass nur noch vom Netzwerk mit der ID 192.168.5.0/24 aus und über das Loopback-Device zugegriffen werden kann. Welche der folgenden Optionen werden Sie in der Datei *named.conf* verwenden?

- ☐ A: `listen-on port 53 { 127.0.0.1; 192.168.5.1; };`
- ☐ B: `listen-on port 53 {127.0.0.1; 192.168.5.1;};`
- ☐ C: `listen-on port 25 { 127.0.0.1; 192.168.5.1; };`
- ☐ D: `allow-query {127.0.0.1; 192.168.5.0/24;};`
- ☐ E: `allow-query { 127.0.0.1; 192.168.5.0/24; };`

Frage 111:

Sie haben einer DNS-Zone auf einem BIND-Server neue Host (A)-Einträge hinzugefügt. Sie wollen, dass der Server die Zonendateien neu einliest, damit die neuen Einträge im Netzwerk sofort verfügbar werden. Welches Kommando sollten Sie auf dem DNS-Server verwenden?

Frage 112:

Sie beabsichtigen, einer DNS-Zone weitere Einträge hinzuzufügen. Sie müssen verhindern, dass der Daemon `named` gleichzeitig versucht, diese Zonendatei zu beschreiben. Welches Kommando sollten Sie verwenden, bevor Sie die Zonendatei bearbeiten?

- ☐ A: `rndc stop`
- ☐ B: `rndc thaw`
- ☐ C: `rndc freeze`
- ☐ D: `rndc reload`
- ☐ E: `rndc halt`

Frage 113:

Sie konfigurieren die Ausfallsicherheit für einen SMTP-Server, der als Mail Exchanger fungiert. Die Priorität des bestehenden Mailservers ist in DNS mit einem Wert von 50 konfiguriert. Wie könnte der entsprechende Eintrag in DNS für einen Reserve-SMTP-Server aussehen? (Wählen Sie zwei mögliche Einträge!)

- ☐ A: `IN MX 100 smtp02.lpic-2.de.`
- ☐ B: `IN MX 80 smtp02.lpic-2.de.`
- ☐ C: `IN MX 10 smtp02.lpic-2.de.`
- ☐ D: `IN MX 80 47.11.8.15`
- ☐ E: `IN MX 100 47.11.8.15`

Frage 114:

Wie heißt der Eintragstyp in einer DNS-Zonendatei, der steuert, wie oft Sekundärzonen zu anderen Servern transferiert werden dürfen? (Geben Sie nur das Akronym an. Beachten Sie bei Ihrer Antwort die Groß- und Kleinschreibung).

Frage 115:

Welcher der folgenden Eintragstypen ist nicht in einer Forward-Lookup-Zone anzutreffen?

- ☐ A: `SOA`
- ☐ B: `PTR`
- ☐ C: `NS`
- ☐ D: `AAAA`
- ☐ E: `CNAME`

Frage 116:

Sie haben Änderungen an der Konfigurationsdatei *named.conf* vorgenommen. Da es sich hierbei um sicherheitskritische Einstellungen handelt, müssen Sie dafür sorgen, dass diese sofort angewendet werden. Welche der folgenden Kommandos können Sie verwenden, damit BIND die Konfiguration sofort verwendet? (Wählen Sie drei Antworten.)

- ☐ A: `rndc reload`
- ☐ B: `rndc thaw`
- ☐ C: `rndc stop`
- ☐ D: `/etc/init.d/named restart`
- ☐ E: `kill -HUP `pidof named``

Frage 117:

Sie konfigurieren die Sicherheitseinstellungen eines DNS-Servers. Sie müssen festlegen, von welchen Netzwerken aus eine dynamische Aktualisierung der Zonen durch Clients erlaubt werden soll. Welche Option werden Sie konfigurieren?

- ☐ A: `allow update`
- ☐ B: `allow-recursion`
- ☐ C: `allow-query`
- ☐ D: `allow-transfer`
- ☐ E: `allow-update`

Frage 118:

Sie konfigurieren die Sicherheitseinstellungen eines DNS-Servers. Sie müssen festlegen, welche Server sekundäre Zonen von diesem Server beziehen dürfen. Welche Option werden Sie konfigurieren?

- ☐ A: `allow update`
- ☐ B: `allow-recursion`
- ☐ C: `allow-query`
- ☐ D: `allow-transfer`
- ☐ E: `allow-update`

Frage 119:

Welche der folgenden Kombinationen aus Transportprotokoll und Portnummer werden von *openvpn* verwendet?

- ☐ A: TCP/443
- ☐ B: TCP/500
- ☐ C: TCP/1701
- ☐ D: TCP/4500
- ☐ E: UDP/1194

Frage 120:

Welches der folgenden Kommandos verwirft Pakete aus einem angegebenen öffentlichen Netzwerk und informiert den Absender über die Ablehnung des Pakets?

- ☐ A: `ip6tables -A INPUT -s fe80::/64 -j REJECT`
- ☐ B: `ip6tables -A INPUT -s 2001:db8:0:c4f8::/64 -j DROP`
- ☐ C: `ip6tables -A INPUT -s ff00::/8 -j DROP`
- ☐ D: `ip6tables -A INPUT -s 2001:db8:0:c4f8::/64 -j REJECT`
- ☐ E: `ip6tables -A OUTPUT -s 2001:db8:0:c4f8::/64 -j DROP`

Antworten und Erklärungen zu den Prüfungsfragen

Hier finden Sie die Erläuterungen zu allen Fragen des zweiten Teils. Sie sollten unbedingt auch die Kommentare zu den falschen Antworten lesen. Einige Fakten werden hier nicht zufällig mehrfach erwähnt, sondern weil wesentliche Prüfungsinhalte auf diese Weise besser in Ihrem Gedächtnis haften bleiben.

Frage 1:

D: *.htgroup* ist die richtige Datei für diese Aufgabe. Hier können Sie Benutzer, die zuvor in einer Passwortdatei angelegt wurden, gruppieren. Sie können die Datei auch anders nennen, aber *.htgroup* ist der übliche Name.

zu A: */etc/passwd* enthält normale Benutzerkonten.

zu B: */etc/groups* enthält Benutzergruppen, die aber üblicherweise nicht zur Zugriffssteuerung auf Webserver verwendet werden.

zu C: *.htaccess* wird ebenfalls zur Zugriffssteuerung auf Apache-Servern verwendet. Sie müssen im vorliegenden Fall sogar die *.htgroup*-Datei in der Datei *.htaccess* als `AuthGroupFile` angeben.

zu E: *httpd.conf* ist die Hauptkonfigurationsdatei des Apache-Servers.

Frage 2:

B: `Listen 127.0.0.1:80` bewirkt, dass `httpd` nur noch an der Loopback-Adresse lauscht. Ein Zugriff vom Netzwerk aus ist dann nicht mehr möglich.

zu A: `Port 81` wird bei älteren Apache-Versionen verwendet, um den Server mit einem alternativen Port zu konfigurieren. Aber der Server wäre weiterhin vom Netzwerk aus erreichbar.

zu C: `Listen 82` sorgt dafür, dass Apache an Port 82 lauscht. Hierbei werden aber weiterhin alle Netzwerkschnittstellen verwendet.

zu D: `ServerType standalone` klingt natürlich wie eine sehr gute Antwort, bewirkt aber lediglich, dass Apache alleinstehend, also ohne vorgeschalteten Super-Daemon (`inetd`) läuft. Das ist übrigens die Standardeinstellung.

zu E: `ServerType inetd` teilt dem Webserver mit, dass er durch `inetd` aufgerufen wird. Eine solche Konfiguration sollte man übrigens aus Gründen der schlechteren Performance vermeiden.

Frage 3:

B: »Erhöhen Sie den Wert für `MinSpareServers`.« Diese Maßnahme erhöht die Anzahl der Prozesse, die auf eingehende Verbindungen lauschen. Benutzer können so schneller bedient werden.

C: »Legen Sie den `ServerType` mit `standalone` fest.« Das ist zwar die Standardeinstellung, muss aber gegenüber den anderen Antworten in Erwägung gezogen werden – insbesondere, weil die anderen Antworten entweder keine oder sogar eine negative Auswirkung auf die Serverleistung haben.

zu A: »Fügen Sie dem Server weitere IP-Adressen hinzu.« Das bringt nichts in Bezug auf die Leistung des Servers.

zu D: »Legen Sie den `ServerType` mit `inetd` fest.« Das verzögert sogar die Antwort bei jedem Benutzerzugriff.

zu E: »Konfigurieren Sie einen anderen Port.« Das bringt nichts in Bezug auf die Leistung des Servers.

Frage 4:

`apachectl restart` oder `apache2ctl restart` sind jeweils richtig. Es gibt natürlich auch andere Methoden, Apache neu zu starten, aber hier war ausdrücklich nach einem dafür vorgesehenen Skript gefragt.

Frage 5:

A: `bin/htpasswd passwortdatei ritchie` ist richtig, wenn Sie einen Benutzer einer existierenden Passwortdatei hinzufügen wollen. Da der aktuelle Pfad `ServerRoot` entspricht, muss dem `htpasswd`-Kommando der relative Pfad `bin/` vorangestellt werden.

zu B: `bin/htpasswd -c passwortdatei ritchie` ist hier falsch, weil die Option `-c` nur bei der Erstellung einer neuen Passwortdatei verwendet wird.

zu C und D: `useradd ritchie -m` und `adduser ritchie` sind falsch, weil diese Kommandos zur Erstellung normaler Benutzerkonten verwendet werden und nicht für Apache-Benutzer in Passwortdateien.

zu E: `echo ritchie >> .htaccess` ist völlig falsch, weil in dieser Datei keine Benutzerkonten angelegt werden. Es wird hier lediglich die Passwortdatei angegeben.

Frage 6:

C: `DocumentRoot` ist die Direktive, die das Hauptverzeichnis für den Content des Webservers angibt.

zu A: `ServerRoot` ist das Serverhauptverzeichnis, in dem sich auch Module und Konfigurationsdateien des Servers befinden. Hierauf darf ein Benutzer natürlich nicht zugreifen.

zu B, D und E: `ApacheRoot`, `Path` und `httpd-Path` sind einfach frei erfunden, also falsch.

Frage 7:

E: `ErrorLog` ist die Direktive, die das Logfile für die Fehlerprotokollierung angibt.

zu A: `LogFile` gibt es beim Apache-Webserver nicht.

zu B: `syslog.conf` konfiguriert den `syslogd`, ist aber nicht zur Konfiguration der Apache-Logfiles geeignet.

zu C: `DocumentRoot` ist das Hauptverzeichnis für den Webcontent.

zu D: `CustomLog` enthält nicht nur Fehler, sondern auch Aufzeichnungen über erfolgreiche Zugriffe auf den Webserver.

Frage 8:

B: mehrere IP-Adressen, D: mehrere TCP-Ports oder E: mehrere `ServerName`-Direktiven können hier jeweils verwendet werden. IP-Adressen oder Ports können über die Direktive `LISTEN` für mehrere Webseiten unterschiedlich konfiguriert und zugewiesen werden. Die Direktive `ServerName` kann für mehrere virtuelle Hosts jeweils innerhalb der Direktiven `VirtualHost` unterschiedlich konfiguriert werden.

zu A und C: Mehrere Verzeichnisse oder mehrere HTML-Dateien können zur Unterscheidung von Webseiten aus der Sicht des Netzwerks nicht herangezogen werden.

Frage 9:

443 ist die korrekte Antwort. Es handelt sich hier letztendlich um das ganz normale HTTPS-Protokoll.

Frage 10:

C: `openssl` ist ein Programm, mit dem man unter anderem X.509-Zertifikate ausstellen kann.

zu A: `httpd` ist der eigentliche Webserver-Daemon.

zu B: `openssh` ist ein Paket, das die Secure Shell enthält.

zu D: `openswan` ist ein Paket zur Implementierung von IPSec.

zu E: `https` ist lediglich das Protokoll, das für HTTP über SSL verwendet wird.

Frage 11:

B: `cache_dir` ist die Direktive, in der die Verzeichnisstruktur des Proxy-Caches konfiguriert wird. Hier wird auch die Größe des festplattenseitigen Caches festgelegt.

zu A: `cache_mem` legt fest, wie viel Arbeitsspeicher Squid für das Caching verwenden darf.

zu C: `reply_body_max_size` wird verwendet, um Benutzer am Herunterladen übergroßer Dateien zu hindern.

zu D: `disk_usage` gibt es nicht bei Squid.

zu E: `access_log` legt den Pfad zur Protokolldatei fest.

Frage 12:

`-z` wäre hier die benötigte Ergänzung. Nach der Konfiguration der Direktive `cache_dir` muss die Verzeichnisstruktur des Caches aufgebaut werden. Das wird durch dieses Kommando erreicht:

```
/usr/local/squid/sbin/squid -z
```

Frage 13:

D: `hide files` ist hier die richtige Option. Sie können so auch einzelne, bestimmte Dateien verstecken.

zu A: `lock directory` ist das Verzeichnis, in dem Samba-Lockfiles anlegt, damit überprüft werden kann, ob es bereits laufende Instanzen von Samba gibt.

zu B: `security = user` legt fest, dass Zugriffe nur durch Benutzer mit gültigen Benutzerkonten erfolgen dürfen.

zu C: `security = share` lässt Zugriffe ohne Authentifizierung von Benutzern zu. Diese Konfiguration wird von neueren Samba-Versionen nicht mehr unterstützt.

zu E: `veto files` ähnelt vom Verhalten her der richtigen Antwort. Es wird hier aber auch der Zugriff auf Dateien verweigert, die ein Benutzer konkret angibt.

Frage 14:

E: `veto files` verhindern den Zugriff auf angegebene Dateien oder auch Erweiterungen. Das gilt auch dann, wenn ein Benutzer eine konkrete Datei auswählt.

zu D: `hide files` versteckt die Dateien nur. Ein direkter Zugriff bleibt weiterhin möglich.

zu A, B und C: Für `lock directory`, `security = user` und `security = share` gelten die Erläuterungen zu den Antworten der vorherigen Frage.

Frage 15:

A: `testparm` überprüft den Inhalt der Datei *smb.conf* auf Korrektheit.

zu B: `hdparm` dient der Anzeige und Überprüfung von Festplattenparametern.

zu C: `nmblookup` ist ein Tool, das zur NetBIOS-Namensauflösung verwendet wird.

zu D: `smbstatus` zeigt den Status bestehender Samba-Verbindungen von Clients an.

zu E: `smbcontrol` wird verwendet, um Signale an die Daemons `smbd` oder `nmbd` zu senden.

Frage 16:

D: *nmbd* ist der hierfür zuständige Dienst. Er unterstützt auch die Anzeige von Samba-Ressourcen in der Netzwerkumgebung von Windows-Computern.

zu A: *WINS* ist der entsprechende Dienst auf einem Windows-Server.

zu B: *Broadcast* ist ein Mechanismus, den Windows-Clients u. a. zur Namensauflösung verwenden, wenn kein unterstützender Server (*nmbd* oder *WINS*) im Netzwerk vorhanden ist.

zu C: *named* unterstützt keine NetBIOS-Namensauflösung, sondern ausschließlich DNS.

zu E: *smbd* ist die Komponente des Samba-Servers, welche die eigentliche Bereitstellung der Shares durchführt.

Frage 17:

`smbstatus` ist die richtige Antwort. Das Programm zeigt den Status bestehender Client-Verbindungen zu einem Samba-Server an.

Frage 18:

B: */etc/samba/smbusers* ist korrekt. Diese Datei ist für diese Zuordnung zuständig. Die Benutzerkonten werden unter Verwendung eines Gleichheitszeichens miteinander verknüpft (`linuxkonto = windowskonto`).

zu A: */etc/samba/passwd* ist mit der Datei */etc/passwd* vergleichbar. Sie gilt aber für Samba-Benutzer.

zu C: */etc/passwd* enthält die normalen Linux-Benutzerkonten.

zu D: */etc/shadow* enthält die verschlüsselten Passwörter der Benutzer.

zu E: */etc/samba/smb.conf* ist die Hauptkonfigurationsdatei des Samba-Servers. Hier wird diese Zuordnung nicht durchgeführt.

Frage 19:

A: `mount -t smbfs //fs01/files /data`

E: `smbmount //fs01/files /data`

Beide Antworten führen zu demselben Ergebnis.

zu B: `mount -t smbfs /data //fs01/files` – Hier sind Mountpoint und Ziel vertauscht worden.

zu C: `smbmount -t smbfs //fs01/files /data` – Das Kommando `smbmount` benötigt (und akzeptiert) keine Angabe über das verwendete Dateisystem.

zu D: `smbmount /data //fs01/files` – Hier sind ebenfalls Mountpoint und Ziel vertauscht worden.

Frage 20:

D: 139 wird für NetBIOS-Sessions benötigt.

E: 445 wird vom CIFS-Protokol verwendet.

zu A: 80 ist der Standardport für HTTP.

zu B: 25 ist der Standardport für SMTP.

zu C: 443 ist der Standardport für HTTPS.

> **Hinweis**
>
> Sie sollten die TCP- und UDP-Ports für gängige Protokolle kennen. Das bezieht sich insbesondere auf alle Dienste, die in diesem Buch Thema sind.

Frage 21:

C: `portmap` ist richtig. Der RPC-Portmapper wird zum Einrichten der NFS-Sitzungen benötigt.

zu A und B: `smbd` und `nmbd` gehören zum Samba-Server. NFS benötigt diese beiden Daemons nicht.

zu D: `dhcpd` ist nicht nötig. NFS bedient auch Clients, die mit statischen IP-Adressen konfiguriert sind.

zu E: `inetd` NFS ist autark und benötigt keinen Super-Daemon.

Frage 22:

E: */etc/exports* ist hier die richtige Konfigurationsdatei.

zu A: */etc/samba/smb.conf* gehört zum Samba-Server.

zu B: */etc/nfs.conf* gibt es nicht.

zu C: */etc/exportfs* gibt es ebenfalls nicht. Verwechseln Sie den Dateinamen nicht mit dem Kommando `exportfs`.

zu D: */etc/fstab* enthält statische Informationen über Dateisysteme, kann in Zusammenhang mit NFS aber nur clientseitig genutzt werden.

Frage 23:

`exportfs -a` exportiert automatisch alle Dateisysteme, die in */etc/exports* aufgeführt sind.

Frage 24:

A: `exportfs -o rw desktop1:/files` exportiert das Verzeichnis mit den gewünschten Optionen.

zu B: `mount -t nfs desktop1:/files /mnt/files` würde versuchen, das Verzeichnis */files* eines NFS-Servers mit dem Namen `desktop1` in das lokale Verzeichnis */mnt/files* einzuhängen.

zu C: `vi /etc/exports` leitet die Bearbeitung der Datei */etc/exports* ein. In der Frage ist allerdings ausdrücklich von einem temporären Export die Rede. Hier würde die Bereitstellung dauerhaft eingerichtet werden.

zu D: `exportfs -o r desktop1:/files` erteilt nur die Berechtigung zum Lesen. In der Frage war aber auch Schreibzugriff gefordert.

zu E: `exportfs -o rw //desktop1/files` ist von der Syntax her falsch.

Frage 25:

A: `showmount -a` zeigt Ihnen genau die gewünschten Informationen an.

zu B: `showmount -r` ergibt eine Fehlermeldung, weil `showmount` die Option `-r` nicht kennt.

zu C: `showmount -d` zeigt nur an, welche Exportverzeichnisse durch Clients in Gebrauch sind. Es fehlen jedoch die gewünschten Informationen über die Client-Computer.

zu D: `rpcinfo` zeigt lediglich RPC-Informationen an, wie der Name bereits vermuten lässt.

zu E: `nfsstat` zeigt umfangreiche NFS-Statistiken an. Diese beinhalten jedoch nicht die gewünschten Informationen.

Frage 26:

E: `nfsstat` liefert unter vielen anderen genau die hier gesuchten Informationen. Die Erläuterungen zu den falschen Antworten entsprechen denen der vorangehenden Frage.

Frage 27:

A: `iptables` ist richtig, weil Sie einfach den Zugriff auf den Port 2049 auf bestimmte Netzwerke beschränken können.

E: TCP-Wrapper funktioniert ebenfalls, indem Sie den Zugriff auf den Portmapper in der Datei *hosts.allow* einschränken.

zu B: `portmapper` wird von NFS benötigt, ist aber zur Zugriffssteuerung ungeeignet.

zu C: Die Konfiguration in */etc/exports* ist falsch. Es sollte eine globale Einstellung vorgenommen werden. In dieser Datei können Sie den Zugriff aber nur pro Exportverzeichnis einzeln konfigurieren.

zu D: Die Konfiguration in */etc/hosts* ist eine Datei des DNS-Client und hier praktisch im falschen Thema.

Frage 28:

`showmount` zeigt genau die gewünschten Informationen an, wenn Sie keine Optionen verwenden.

Frage 29:

B: 67 eingehend, 68 ausgehend ist die optimale Kombination.

zu A: 68 eingehend, 67 ausgehend ist von der Richtung her falsch herum.

zu C: 53 eingehend, ausgehend dynamisch wäre für einen DNS-Server passend.

zu D: 67 eingehend, ausgehend dynamisch funktioniert ebenfalls, ist aber keine saubere Konfiguration.

zu E: 139 eingehend, 445 eingehend wären die eingehenden Ports für einen Samba-Server.

Frage 30:

A: `ddns-updates on` und D: `ignore client-updates` werden nicht an den Client übermittelt.

zu B: `option domain-name-servers` ist eine Client-Option.

zu C: `option netbios-name-servers` ist eine Client-Option.

zu E: `option routers` ist ebenfalls eine Client-Option.

Prüfungstipp

DHCP-Einstellungen, die an den Client übergeben werden, beginnen mit dem Schlüsselwort `option`. Optionen, die aus mehreren einzelnen Wörtern bestehen, sind immer durch Bindestriche miteinander verbunden (außer dem Wort `option` selbst). Wenn Sie sich das merken, können Sie in der Prüfung meist die falschen Antworten ausschließen.

Frage 31:

B: `fixed address 192.168.50.104;` ist aufgrund des fehlenden Bindestrichs (im Vergleich zu Antwort C) falsch.

D: `hardware ethernet 00:1c:bf:c5:e1:8e:f7:80;` kann aufgrund der zu langen MAC-Adresse nicht stimmen. MAC-Adressen weisen eine Länge von nur 48 Bit auf.

zu A: `host client4` ist richtig.

zu C: `fixed-address 192.168.50.104;` ist ebenfalls richtig.

zu E: `hardware ethernet 00:1c:bf:c5:e1:8e;` ist auch richtig. Merken Sie sich, dass hier ausnahmsweise kein Bindestrich zwischen `hardware` und `ethernet` verwendet werden darf.

Frage 32:

dhcpd.leases heißt die Datei, die diese Information enthält.

Frage 33:

C: *pam_cracklib.so* vergleicht Passwörter mit einem Wörterbuch und führt andere Prüfungen auf Schwachstellen in Benutzerpasswörten durch.

zu A: *pam_env.so* ist für Umgebungsvariablen zuständig.

zu B: *pam_unix.so* wird für die normale Passwortanmeldung benötigt.

zu D: *pam_permit.so* erteilt Zugriff.

zu E: *pam_deny.so* verweigert den Zugriff.

Frage 34:

D: */etc/nsswitch.conf* wird unter anderem verwendet, um festzulegen, in welcher Reihenfolge PAM bei der Authentifizierung auf lokale Dateien oder LDAP zugreifen soll.

zu A und B: */etc/pam.d/common-password* und */etc/pam.d/common-auth* sind standardmäßig vorhandene PAM-Konfigurationsdateien. Sie steuern aber nicht die Reihenfolge der Suche nach Benutzerkonten.

zu C: */etc/pam.conf* ist die Hauptkonfigurationsdatei. Auch diese Datei steuert nicht die Reihenfolge der Suche nach Benutzerkonten.

zu E: */etc/ldap/ldap.conf* ist die LDAP-Client-Konfigurationsdatei. Sie ist bei der Authentifizierung von Benutzern nicht beteiligt.

Frage 35:

C: `password` ist die einzig sinnvolle PAM-Verwaltungsgruppe für *cracklib.so*. Schließlich prüft das Modul die Sicherheit der Passwörter.

Frage 36:

A: »*/etc/nsswitch.conf* wurde nicht richtig konfiguriert.« Das wäre ein typischer Fehler. Standardmäßig ist diese Datei so konfiguriert, dass PAM bei der Suche nach Benutzerkonten lediglich die lokale Datei */etc/passwd* konsultiert und nicht LDAP.

C: »*pam_ldap.so* wurde nicht installiert bzw. konfiguriert.« Auch das wäre ein gängiger Fehler. Bei vielen Linux-Distributionen ist das Modul *pam_ldap.so* (Paket *libpam-ldap*) standardmäßig nicht installiert.

zu B: »*/etc/resolv.conf* enthält einen Fehler.« Ein Fehler in dieser Datei verursacht lediglich Probleme mit der Namensauflöung.

zu D: »Der Benutzer existiert nicht.« Doch, er existiert. Die Frage impliziert die Existenz des Benutzers.

zu E: »Der Benutzer existiert sowohl lokal als auch in LDAP.« Das ist unwahrscheinlich, weil dann zumindest die lokale Anmeldung gegriffen hätte.

Frage 37:

/etc/nsswitch.conf ist die richtige Datei. Sie wird unter anderem verwendet, um festzulegen, in welcher Reihenfolge PAM bei der Authentifizierung auf lokale Dateien oder LDAP zugreifen soll.

Frage 38:

E: */lib/security* enthält die PAM-Module.

zu A: */etc/pam.d* enthält PAM-Konfigurationsdateien.

zu B, C und D: */var/lib/pam*, */lib/pam* und */usr/pam/modules* sind frei erfundene Dateinamen.

Frage 39:

B: `CN=Ronnie,DC=example,DC=org` ist ein DN (Distinguished Name) nach dem X.500-Standard.

zu A: `ronnie@example.org` ist lediglich eine E-Mail-Adresse.

zu C: `www.example.org` ist lediglich eine Webadresse.

zu D: `DN=Ronnie,DC=example,DC=org` enthält einen Fehler, weil »DN« keine Komponente eines DN ist.

zu E: `example.org` ist ein Domainname, aber kein Distinguished Name.

Frage 40:

Richtig sind:

B: `dig @172.16.0.10 www.lpic-2.de` und

E: `dig www.lpic-2.de @172.16.0.10`

Beide Kommandos weisen `dig` an, den DNS-Namen *www.lpic-2.de* unter Verwendung des DNS-Servers mit der IP-Adresse 172.16.0.10 aufzulösen. Die Reihenfolge der Parameter kann also tatsächlich variieren.

zu A: `nslookup www.lpic-2.de` enthält keine Angabe darüber, welcher DNS-Server zu verwenden ist. Es war aber die Verwendung des DNS-Servers mit der IP-Adresse 172.16.0.10 gefordert.

zu C: `host www.lpic-2.de` enthält ebenfalls keine Angabe darüber, welcher DNS-Server zu verwenden ist.

zu D: `ping www.lpic-2.de` verwendet ein zur Diagnose von DNS-Problemen ungeeignetes Programm. Der Name würde zwar auch hier grundsätzlich in eine IP-Adresse übersetzt werden, aber die Angabe eines konkreten DNS-Servers wäre gar nicht möglich.

Frage 41:

`ldappasswd` wird verwendet, wenn das Passwort eines Benutzers geändert werden muss, dessen Benutzerkonto in LDAP gespeichert wurde.

Frage 42:

B: `ldapadd` ist das für diese Aufgabe gedachte Kommando. Es handelt sich hier allerdings lediglich um einen Link auf `ldapmodify`.

E: `ldapmodify` kann ebenfalls verwendet werden, um Objekte zum Verzeichnis hinzuzufügen. Verwenden Sie hierzu die Option `-a`.

zu A: `ldappasswd` ändert Passwörter von Benutzern, deren Konten in OpenLDAP angelegt wurden.

zu C: `ldapsearch` durchsucht LDAP-Datenbanken nach angegebenen Objekten.

zu D: `addldap` gibt es nicht. Alle Kommandos, die mit OpenLDAP in Zusammenhang stehen, beginnen grundsätzlich mit `ldap`.

Frage 43:

A: *Postfix* und B: *Sendmail* sind Mail Transfer Agents (MTAs).

zu C, D und E: *Courier*, *Dovecot* und *Cyrus* sind E-Mail-Server, die POP und IMAP unterstützen.

Frage 44:

D: Erstellen Sie Einträge in der Datei */etc/aliases*.

C: Führen Sie `newaliases` aus.

Diese beiden Aktionen müssen in dieser Reihenfolge ausgeführt werden. Eingehende E-Mails, die an die zusätzlichen Adressen gerichtet sind, werden dann im Mail-Eingang des Benutzerkontos zugestellt.

zu A: »Erstellen Sie weitere Benutzerkonten.« Das ist zu aufwendig für alle Beteiligten und setzt eine Anpassung des Mail-Client-Programms voraus. Das war ausdrücklich unerwünscht.

zu B: »Modifizieren Sie die Datei *.forward* des Benutzers.« Hier kann ein Benutzer lediglich seine E-Mails an andere Adressen weiterleiten.

zu E: »Diese Konfiguration ist nicht möglich.« Diese Aussage ist falsch.

Frage 45:

`named-checkconf` führt, ohne Optionen gestartet, eine Überprüfung der Syntax der Datei *named.conf* durch.

Frage 46:

B: `named-checkzone` führt genau diese Aufgabe durch.

zu A: `named-compilezone` wird verwendet, um Zonendateien in ein anderes Format umzuwandeln. Man kann damit zwar auch die Syntax von Zonendateien prüfen, aber das ist nicht der Hauptverwendungszweck dieses Programms.

zu C: `named-checkconf` können Sie verwenden, um die Syntax der Datei *named.conf* zu überprüfen.

zu D: `rndc` wird zur Steuerung des Nameservers verwendet.

zu E: `nslookup` richtet zu Diagnosezwecken Anfragen an Nameserver.

Frage 47:

D: `mydestination` in *main.cf* enthält den gewünschten Parameter.

zu A: `myorigin` in *main.cf* wird verwendet, um das Suffix für ausgehende E-Mails zu definieren.

zu B: `mynetworks` in *master.cf* verwendet die falsche Option und die falsche Datei.

zu C: `mynetworks` in *main.cf* legt fest, welche IP-Subnetze als lokal zu erachten sind (z. B. um Relaying zu begrenzen).

zu E: `mydestination` in *master.cf* verwendet die falsche Konfigurationsdatei.

Frage 48:

C: `telnet` und E: `netcat` sind geeignet, um die SMTP-Kommunikation mit einem Server durchzuführen. Das ist ein guter Funktionstest.

zu A und B: `traceroute` und `ping` würden lediglich feststellen, dass der Server-Computer als solcher erreichbar ist. Die SMTP-Funktionalität kann damit nicht geprüft werden.

zu D: `nslookup` kann lediglich feststellen, ob der Name des Computers in eine IP-Adresse auflösbar ist.

Frage 49:

D: *Sieve* ist die für Filterung zuständige Komponente.

zu A und B: *Postfix* und *Sendmail* sind jeweils MTAs und für die Zustellung via SMTP zuständig.

zu C und E: *Dovecot* und *Courier* stellen die E-Mails für die Benutzer zur Abholung bereit.

Frage 50:

E: *DANE* verwendet Einträge vom Typ TLSA, um damit Hashwerte zu veröffentlichen, die zu X.509-basierten Zertifikaten gehören.

zu A: *MX* ist ein Mailexchanger-Eintrag.

zu B: *CNAME* ist ein Aliaseintrag.

zu C: *TSIG* wird zur Unterstützung von Transaktionssignaturen verwendet.

zu D: *DNSSEC* wird als Basis von DANE benötigt.

Frage 51:

C: `ip6tables -A INPUT -s 2001:db8:0:c4f8::/64 -j DROP`

ist die richtige Antwort. `ip6tables` ist das für IPv6 zuständige Kommando. `-A INPUT -s 2001:db8:0:c4f8::/64` adressiert Pakete, die ihren Ursprung in einem öffentlichen (globalen) Netzwerk haben. `-j DROP` sorgt dafür, dass die Pakete ohne eine Fehlermeldung an den Absender verworfen werden.

zu A: `ip6tables -A INPUT -s 2001:db8:0:c4f8::/64 -j REJECT`

Das Kommando funktioniert ebenfalls, aber der Absender eines Pakets wird über die Abweisung informiert (`-j REJECT`).

zu B: `iptables -A INPUT -s 2001:db8:0:c4f8::/64 -j DROP`

und zu D: `iptables -A INPUT -s 2001:db8:0:c4f8::/64 -j REJECT`

Bei diesen beiden Antworten wird das entsprechende Kommando für IPv4 verwendet. Es wird hier zu einer Fehlermeldung kommen.

zu E: `ip6tables -A OUTPUT -s 2001:db8:0:c4f8::/64 -j DROP`

Dieses Kommando verwirft ausgehende Pakete. Die Frage bezieht sich jedoch auf eingehenden Verkehr.

Frage 52:

`ipt6ables -F` oder `ip6tables -t filter -F` erfüllen diese Aufgabe. Wenn keine Kette explizit angegeben wird, löschen diese Kommandos alle Regeln aller Ketten.

Frage 53:

C: `smtp_relay_restrictions`

Hier kann festgelegt werden, für wen der Host als Relay fungieren darf. Hierfür kommt z. B. der Parameter `permit_mynetworks` in Frage. Dazu passend werden bei E: `mynetworks` die Netzwerke festgelegt, für die E-Mails im Sinne eines Relay weitergeleitet werden sollen.

zu A: `mydestination` sagt dem Server, für welche Adressen er selbst das Ziel ist.

zu B: `myorigin` sagt dem Server, welches Suffix er an versendete Mails anhängen soll, falls noch keines vorhanden ist.

zu D: `myhostname` legt den Namen für den Mailserver fest. Das ist in der Regel zugleich der FQDN des Hosts.

Frage 54:

B: `protocols: imaps pop3s` ist richtig. Diese Konfiguration erlaubt den Datentransfer ausschließlich über eine gesicherte Verbindung (SSL).

zu A: `protocols: imap imaps pop3 pop3s` lässt zusätzlich auch unverschlüsselte Verbindungen zu.

zu C: `protocols: imap pop3` erlaubt ausschließlich ungesicherte Verbindungen.

zu D: `disable_plaintext_auth: no` erlaubt Authentifizierung im Klartext, hat aber mit der eigentlichen Übertragung der Nachrichten nichts zu tun.

zu E: `disable_plaintext_auth: yes` verhindert eine unverschlüsselte Authentifizierung. Ob die eigentliche E-Mail-Übermittelung geschützt ist, wird hier nicht festgelegt.

Frage 55:

E: `nmap compi.example.com -O` ist richtig. Die Option -O steht hierbei für OS detection.

zu A: `dig compi.example.com` führt lediglich eine Namensauflösung durch.

zu B: `ping compi.example.com` sendet einen ICMP Echo-Request an den Host.

zu C: `nmap compi.example.com -p 139` prüft, ob auf dem Zielsystem Port 139 geöffnet ist.

zu D: `nslookup compi.example.com` führt, genau wie Antwort A, lediglich eine Namensauflösung durch.

Frage 56:

E: `MX` ist der benötigte Eintragstyp. Ein MX-Record (Mail Exchanger) ist ein Eintrag, der auf den (oder die) Eingangs-SMTP-Server einer Domäne zeigt.

zu A: `CNAME` ist ein Alias, der auf einen Host-A Eintrag zeigt.

zu B: `PTR` ist eine Pointer (Zeiger), der nur in Reverse-Lookup-Zonen verwendet wird.

zu C: `SOA` zeigt auf den für eine Zone autoritativen Server (Start of a Zone of Authority).

zu D: `NS` ist ein Eintrag, der auf den (oder die) DNS-Server (Nameserver) einer Domäne zeigt.

Frage 57:

A: `location / { proxy_pass http://127.0.0.1:8000; }` ist die richtige Direktive unter der Annahme, dass der lokal installierte Webserver an Port 8000 lauscht.

zu B: `location / { proxy_pass http://127.0.0.1:80; }` kann nicht sein, weil *nginx* selbst am lokalen Anschluss 80 lauscht. Dann kann die Umleitung nicht ebenfalls an Port 80 stattfinden.

zu C: `listen 80` legt lediglich fest, dass *nginx* den Port 80 abhören soll.

zu D: `listen 8000` bewirkt, dass *nginx* Port 8000 verwendet.

zu E: `index index.html;` ist keine Direktive für einen Reverse-Proxy.

Frage 58:

`iptables -F` oder `iptables -t filter-F` sind richtig. Diese Kommandos löschen alle bestehenden Regeln.

Frage 59:

C: `iptables -t filter -A FORWARD -j REJECT` erfüllt genau die Anforderungen der Frage.

zu A: `iptables -t filter -A INPUT -j DROP` verwirft lediglich die Pakete, die direkt an den Router gerichtet sind. Weiterleitungen würden immer noch stattfinden.

zu B: `iptables -t filter -A FORWARD -j DROP` sorgt zwar dafür, dass der Router keine Pakete mehr weiterleitet, aber die sendenden Client-Computer erhalten keine Fehlermeldung, wie in der Frage gefordert.

zu D: `iptables -t filter -F FORWARD` löscht alle Regeln aus der Kette `FORWARD`.

zu E: `iptables -t filter -L FORWARD` listet die Regeln der Kette `FORWARD` auf.

Frage 60:

B: `iptables -t nat -I POSTROUTING \`

`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

Das ist die richtige Antwort. Wichtig sind hier die Tabelle `nat`, die Kette `POSTROUTING` (weil das Masquerading erst nach dem Routing-Modul ausgeführt werden soll) und das Ziel `MASQUERADE`.

zu A: `iptables -t nat -I PREROUTING \`

`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

Das Kommando verwendet die falsche Kette. `POSTROUTING` wäre hier richtig gewesen.

zu C: `iptables -t filter -I PREROUTING \`

`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

Das Kommando verwendet die falsche Tabelle und die falsche Kette. Statt `filter` wäre hier `nat` nötig gewesen.

zu D: `iptables -t nat -I INPUT \`

`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

Hier wird ebenfalls eine falsche Kette (`INPUT`) verwendet.

zu E: `iptables -t filter -I POSTROUTING \`

`-s 192.168.40.0/24 -o ppp0 -j MASQUERADE`

In dieser Antwort wird die falsche Tabelle (`filter`) verwendet.

Frage 61:

B: `route add 192.168.40.87 gw 127.0.0.1` ist eine einfache Möglichkeit, einen Angriff abzuwehren. Der Angreifer erhält so keine Antworten von diesem Computer mehr und wird den Angriff abbrechen.

D: `iptables -t filter -I INPUT -s 192.168.40.87 -j DROP` verwirft alle Pakete, die vom Angreifer gesendet werden. Das ist eine gute Möglichkeit, einen Angriff abzuwehren.

zu A: `iptables -t filter -I OUTPUT -s 192.168.40.87 -j DROP`

Das ist eine ziemlich sinnlose Regel. Sie hindert Pakete daran, das System zu verlassen, die von der Quelle mit der IP-Adresse 192.168.40.87 stammen. Ein solches Paket kann es gar nicht geben.

zu C: `iptables -t nat -I INPUT -s 192.168.40.87 -j DROP` verwendet eine unbrauchbare Kombination der Tabelle `nat` mit der Kette `INPUT`.

zu E: `iptables -t filter -I FORWARD -s 192.168.40.87 -j DROP`

Dieses Kommando blockiert die Pakete des Angreifers, die den Router passieren sollen. Der Angriff betrifft aber den Router selbst, weshalb die Kette `INPUT` verwendet werden muss.

Frage 62:

A: `iptables -t nat -I PREROUTING -i ppp0 -p tcp \`

`--dport 25 -j DNAT --to-destination 192.168.40.27:25`

Das ist genau das richtige Kommando. Wichtig sind hier die Verwendung der Tabelle `nat`, die Kette `PREROUTING` (weil ein solcher Vorgang noch vor dem Routing ausgeführt wird) und das Target `DNAT` zwecks Port-Forwarding.

zu B: `iptables -t nat -I PREROUTING -i ppp0 -p tcp \`

`--dport 110 -j DNAT --to-destination 192.168.40.27:110`

verwendet den falschen TCP-Port. Port 110 wird von POP3 verwendet, nicht von SMTP.

zu C: `iptables -t nat -I POSTROUTING -i ppp0 -p tcp \`

`--dport 25 -j DNAT --to-destination 192.168.40.27:25`

verwendet die falsche Kette (`POSTROUTING`).

zu D: `iptables -t filter -I PREROUTING -i ppp0 -p tcp \`

`--dport 25 -j DNAT --to-destination 192.168.40.27:25`

verwendet die falsche Tabelle (`filter`).

zu E: `iptables -t nat -I PREROUTING -i ppp0 -p tcp \`

`--dport 25 -j MASQUERADE --to-destination 192.168.40.27:25`

verwendet das falsche Target (`MASQUERADE`).

Frage 63:

A: 20 wird für die Datenübertragung verwendet.

B: 21 ist der Steuerkanal für FTP.

zu C: 22 ist der von SSH verwendete Port.

zu D: 23 ist der Port, den Telnet verwendet.

zu E: »Ein weiterer Port ist nicht erforderlich.« Diese Antwort wäre (in Kombination mit B) richtig gewesen, wenn der Server im aktiven Modus laufen würde.

Frage 64:

E: Pure-FTPd weist als Einziger diese etwas ungewöhnliche Konfigurationsmethode auf.

Die anderen Antworten sind entsprechend falsch, was diesmal keiner weiteren Erläuterung bedarf.

Frage 65:

D: `chroot_local_user=YES` bewirkt das in der Frage beschriebene Verhalten. Sie sollten diese Option auf `NO` einstellen, wenn Sie auf alle Verzeichnisse des Servers FTP-Zugriff benötigen. Bedenken Sie in der Praxis, dass dann alle Benutzer den gesamten Server durchforsten können.

Die Antworten A, B und E sind nur für anonyme Zugriffe von Belang. Implizit steht in der Frage jedoch, dass eine Authentifizierung stattgefunden haben muss. Schließlich wird man auf das Home-Directory begrenzt.

zu C: `local_enable=YES` sollten Sie in diesem Fall nicht ändern, weil der authentifizierte Zugriff lokaler Benutzer sonst nicht mehr möglich wäre.

Frage 66:

`pure-ftpwho` ist richtig. Das Programm zeigt Ihnen in einer Tabelle an, wer gerade mit dem Server verbunden ist.

Frage 67:

E: `ssh -X` aktiviert X11-Forwarding. Ein Programm, das auf GTK+ basiert, kann nur auf einem X-Server ausgeführt werden. Deshalb ist das die richtige Option.

zu A: `ssh -6` erzwingt die Verwendung von IPv6.

zu B: `ssh -l` spezifiziert den Login-Namen des Users, wenn dieser vom lokal verwendeten Namen abweicht.

zu C: `ssh -p` wird benötigt, wenn der SSH-Server einen anderen Port als 22 abhört.

zu D: `ssh -L` wird verwendet, um einen angegebenen lokalen Port an einen Port einer entfernten Maschine weiterzuleiten.

Frage 68:

A: `ssh -2` erzwingt die Verwendung der SSH-Protokollversion 2.

zu B: `ssh -4` erzwingt die Verwendung von IPv4.

zu C: `ssh -6` erzwingt die Verwendung von IPv6.

zu D: `ssh -l` spezifiziert den Login-Namen des Users, wenn dieser vom lokal verwendeten Namen abweicht.

zu E: `ssh -p` wird benötigt, wenn der SSH-Server einen anderen Port als 22 abhört.

Frage 69:

B: `known_hosts` auf dem Client-Computer ist die Datei, die bearbeitet werden muss. Wenn der Client-Computer mit dem ursprünglichen Server bereits einmal Kontakt

hatte, dann ist der alte Hostkey des Servers in dieser Datei gespeichert. Bei dem Versuch, den neuen Server zu kontaktieren, bemerkt der Client den Unterschied zwischen den Hostkeys als mögliche Sicherheitsbedrohung durch einen ausgetauschten Server. Da die einzelnen Keys innerhalb der Datei nicht leicht voneinander unterscheidbar sind, werden Sie die Datei möglicherweise einfach löschen müssen.

zu A: `authorized_keys` auf dem Client-Computer wird zur Authentifizierung von Benutzern verwendet (anstatt Passwort).

zu C: `authorized_keys` auf dem Server-Computer wird zur Authentifizierung von Benutzern verwendet (anstatt Passwort).

zu D: `known_hosts` auf dem Server-Computer ist sinnlos, weil es sich hier um eine clientseitige Datei handelt.

zu E: `hosts.allow` auf dem Server-Computer ist hier nicht der Grund für den Fehler, wenn auch diese Datei den Zugriff natürlich auf bestimmte Clients beschränken kann.

Frage 70:

`ssh-keygen` ist richtig. Sie können mithilfe dieses Programms nicht nur Hostkeys, sondern auch Benutzerschlüssel für die Authentifizierung erstellen.

Frage 71:

C: `sshd_config` ist die Hauptkonfigurationsdatei des SSH-Serverdaemons.

zu A: `ssh_host_dsa_key.pub` ist der öffentliche DSA-Schlüssel des Servers.

zu B: `ssh_config` ist die Konfigurationsdatei des SSH-Clients.

zu D: `ssh_host_dsa_key` ist der private DSA-Key des Servers.

zu E: `ssh_host_rsa_key` ist der private RSA-Key des Servers.

Frage 72:

D: `PermitRootLogin no` in der Datei *sshd_config* ist die richtige Antwort.

zu A: `AllowRootLogin no` in der Datei *ssh_config* – Diesen Eintragstyp gibt es nicht in dieser Datei. Außerdem handelt es sich hier um die clientseitige Konfigurationsdatei von SSH.

zu B: `AllowRootLogin no` in der Datei *sshd_config* – Diesen Eintragstyp gibt es nicht in dieser Datei.

zu C: `PermitRootLogin no` in der Datei *ssh_config* – Es handelt sich hier um die clientseitige Konfigurationsdatei von SSH. Es muss aber die Serverkomponente konfiguriert werden.

zu E: `PermitRootLogin yes` in der Datei *ssh_config* – Es handelt sich hier wieder um die clientseitige Konfigurationsdatei. Außerdem würde dieser Eintrag den root-Zugriff erlauben, wenn es sich um die richtige Datei handeln würde.

Frage 73:

/etc/ssh/sshd_config ist die Konfigurationsdatei für den *sshd*. Legen Sie den Port mit der Option `Port` fest.

Frage 74:

E: `PasswordAuthentication yes` erlaubt ausdrücklich eine Authentifizierung mit Passwörtern. Sie sollten diese Option also hier auf »`no`« ändern.

zu A: `PermitRootLogin no` verhindert einen direkten Zugriff durch `root` via SSH.

zu B: `Protocol 2,1` legt fest, dass sowohl Protokollversion 1 als auch Protokollversion 2 verwendet werden dürfen.

zu C: `Port 22` legt den zu verwendenden Port für SSH fest.

zu D: `AllowUsers harald` erlaubt ausschließlich dem angegebenen User den SSH-Zugriff.

Frage 75:

B: */etc/hosts.allow* und E: */etc/hosts.deny* werden von TCP-Wrappern gelesen, um den Zugriff auf Dienste einzuschränken.

zu A: */etc/services* enthält Zuordnungen von Diensten zu den entsprechenden TCP- und UDP-Ports.

zu C: */etc/hosts* ist eine Datei des DNS-Clients und dient der lokalen Namensauflösung.

zu D: */etc/ssh/ssh_config* ist die Konfigurationsdatei für den SSH-Client.

Frage 76:

C: `xinetd` und D: `tcpd` verwenden diese Bibliothek. Der Super-Daemon `xinetd` verwendet nämlich einen eigenen Wrapper und benötigt dafür diese Bibliothek.

zu A: `iptables` hat keinen Grund diese Bibliothek zu verwenden.

zu B: `inetd` ist selbst kein Wrapper, sondern verwendet den Wrapper `tcpd`. Deshalb benötigt `inetd` diese Bibliothek im Gegensatz zu `xinetd` nicht.

zu E: `dhcpd` hat ebenfalls keinen Grund diese Bibliothek zu verwenden.

Prüfungstipp

Das Thema der Frage 76 kommt auch in der Prüfung häufig vor.

Frage 77:

A: *CERT*, B: *Bugtraq* und E: *CIAC* sind richtig.

zu C: *IANA* ist die Internet Assigned Numbers Authority und beschäftigt sich mit der weltweiten Verwaltung von IP-Adressen und Namensauflösung.

zu D: *ISC* ist das Internet Systems Consortium, das sich u. a. mit der Weiterentwicklung von BIND und dem Betrieb der Root-Server beschäftigt.

Frage 78:

C: `netstat` und E: `nmap` sind für diese Aufgabe geeignet. Das Programm `netstat` muss auf dem Server lokal ausgeführt werden, während `nmap` diese Überprüfung auch remote ausführen kann.

zu A und B: `telnet` und `netcat` können zwar verwendet werden, um nach einzelnen geöffneten Ports zu suchen, aber keines der beiden Programme erstellt eine Liste.

zu D: `ifconfig` zeigt lediglich Schnittstelleninformationen an und ist deshalb für diese Aufgabe völlig ungeeignet.

Frage 79:

D: `fail2ban` und E: `snort` sind Intrusion-Detection- bzw. Intrusion-Prevention-Systeme und für diese Aufgabe konzipiert.

zu A: `nmap` ist lediglich ein Portscanner.

zu B: `ipchains` ist die Firewall bei alten Kernel-Versionen (2.2).

zu C: `iptables` ist die aktuelle Firewall. Sie kann keine Einbruchsversuche erkennen.

Frage 80:

`DROP` ist hier die richtige Antwort. Das Target `REJECT` wäre hier übrigens falsch gewesen, weil keine Fehlermeldung an den Client zurückgegeben werden sollte.

Frage 81:

B: `iptables -L` und C: `iptables-save` können Sie jeweils verwenden, um zu überprüfen, ob durch `fail2ban` erstellte Firewall-Regeln immer noch aktiv sind.

D: `fail2ban-client status ssh` kann ebenfalls verwendet werden, um aktuell aufgrund von SSH-Fehlauthentifizierungen gebannte IP-Adressen einzusehen.

zu A: `netstat -an` zeigt bestehende Verbindungen und lokal geöffnete Ports an.

zu E: `nmap localhost` zeigt an, welche lokalen Ports geöffnet sind.

Frage 82:

A: `netcat` ist für diese Aufgabe ideal. Sie können auf dem geplanten Server mithilfe von `netcat` einen Port-Listener erstellen, der Port 2704 abhört, und anschließend von entfernten Rechnern aus versuchsweise auf diesen Port zugreifen.

zu B und E: `netstat` und `nmap` können lediglich bereits geöffnete Ports anzeigen. Die Anwendung ist aber noch nicht installiert.

zu C und D: `fail2ban` und `iptables` dienen beide dem Schutz eines Computers, sind bei einer Analyse des Netzwerks aber wenig hilfreich.

Frage 83:

B: `nslookup`, D: `dig` und E: `host` sind Dienstprogramme, die zur Abfrage von DNS-Servern dienen.

zu A: `route` dient der Konfiguration und Anzeige von Routing-Tabellen.

zu C: `ping` sendet ICMP-Echo-Anforderungen an Netzwerkcomputer.

Frage 84:

`slapindex` lautet die richtige Antwort.

Frage 85:

B: *SNI* (*Server Name Indication*) kommt hier zum Einsatz.

zu A: *Round Robin* kann zur Lastverteilung verwendet werden, wenn mehrere Server dieselbe Webseite gemeinsam hosten.

zu C: *DANE* ist eine DNS-Seitige Absicherung zur Echtheitsprüfung von X.509-basierten Zertifikaten.

zu D: *SSL* liegt hier natürlich grundsätzlich vor, hilft aber nicht bei der Umsetzung des geschilderten Szenarios.

zu E: *TSIG* sichert DNS-Transaktionen mit Signaturen.

Frage 86:

`dovecot -n` oder `doveconf -n`. Beide Kommandos zeigen die Nicht-Standardeinstellungen von *Dovecot* an.

Frage 87:

B: integrierte Verschlüsselung

C: bessere Performance

D: Authentifizierung auf Basis des Benutzers

Das sind drei der Verbesserungen von NFS 4 gegenüber NFS 3.

zu A: »Verwendung von UDP als Transportprotokoll« – UDP wurde früher von NFS verwendet, was aus Sicht der Performance eher ein Nachteil ist.

zu E: »Authentifizierung auf Basis des Computers« – Diese Eigenschaft hatte NFS schon immer.

Frage 88:

B: `force create mode = 0444` bewirkt, dass zusätzlich zu den ohnehin gesetzten Berechtigungsbits immer die Bits für das Recht »lesen« gesetzt werden. Das gilt für den Eigentümer, die Gruppe und Andere.

zu A: `create mode = 0664` bewirkt eine Filterung der verwendbaren Berechtigungen, sodass maximal `644` möglich ist. Einer Datei, die ohne Leserechte gespeichert wird, werden diese nicht hinzugefügt.

zu C: `create mask = 0664` ist von der Wirkung her mit Antwort A identisch.

zu D: `directory mode = 0555` wirkt sich nur auf neu zu erstellende Verzeichnisse aus, jedoch nicht auf Dateien.

zu E: `force create mode = 0660` bewirkt, dass der Eigentümer und die Gruppe einer Datei mindestens Schreib- und Leserechte an einer gespeicherten Datei erhalten. Andere Benutzer profitieren hiervon jedoch nicht.

Frage 89:

`465` ist der Port, den SMTPS für Client zu Server Transfers verwendet.

Frage 90:

E: `relayhost` in *main.cf* ist die hierfür zuständige Option. Die in den anderen Antworten genannten Zuordnungen von Optionen zu Dateien gibt es jeweils nicht.

Frage 91:

C: *radvd* ist ein Daemon, der typischerweise auf einem Router eingesetzt wird. Er kündigt den Präfix eines konfigurierten IPv6 Netzwerks an. IPv6-Clients generieren dazu den Host-Anteil der IPv6-Adresse selbst.

zu A: *isc-dhcp3-server* kann diese Aufgabe zwar durchführen, erfüllt aber nicht die Anforderung, dass clientseitig Broadcastverkehr vermieden werden soll.

zu B: *DHCP-Relay* dient lediglich als Agent zur Vergabe von IP-Adressen über Routergrenzen hinweg.

zu D: *bind* dient zur Namensauflösung und vergibt keine IP-Adressen.

zu E: *link-lokale* Adressen werden tatsächlich ohne Broadcastverkehr automatisch generiert. Allerdings können diese Adressen nicht über Routergrenzen hinweg verwendet werden und sind deshalb nicht für Internetzugriffe geeignet.

Frage 92:

B: *procmail* und D: *Sieve* sind Mailfilter, wobei *Sieve* inzwischen die gängigere Lösung für solche Aufgaben ist.

zu A: *postfix und* C: *exim* sind jeweils MTAs.

zu E: *dovecot* stellt IMAP und POP-Postfächer bereit.

Frage 93:

B: *IMAP*, D: *POP* und E: *HTTP* sind hier zutreffend.

Die anderen genannten Dienste werden von *nginx* nicht unterstützt.

Frage 94:

D: `139/tcp` und E: `445/tcp` werden von Samba-Servern genutzt.

zu A: `22/tcp` wird für SSH verwendet.

zu B: `443/tcp` ist typisch für HTTPS-Server.

zu C: `80/tcp` ist der Standardport für Webserver.

Frage 95:

A: `samba-tool` wird genau für diese Aufgabe verwendet.

zu B: `testparm` überprüft die Samba-Konfigurationsdatei auf Fehler.

zu C: `smbcontrol` dient zum Starten und Stoppen der Daemons von Samba.

zu D: `smbstatus` überprüft den aktuellen Status des Samba-Servers.

zu E: `rndc` wird zur Steuerung von BIND genutzt.

Frage 96:

B: »Authentifizierung von Benutzern aus NT-Domänen« ist die richtige Antwort. Der Daemon befähigt einen Samba-Server, Benutzer zu authentifizieren, deren Benutzerkonten sich in Windows-NT bzw. Active-Directory-Domänen befinden. Die anderen Antworten sind entsprechend einfach falsch und bedürfen diesmal keiner weiteren Erklärung.

Frage 97:

C: `slapcat` ist ein Programm, das dem Export von OpenLDAP-Datenbankinhalten in das LDIF-Format dient.

zu A: `ldapadd` fügt dem LDAP-Verzeichnis Objekte hinzu.

zu B: `slapadd` fügt ebenfalls dem LDAP-Verzeichnis Objekte hinzu.

zu D: `slapindex` indexiert die OpenLDAP-Datenbank neu.

zu E: `slapdump` klingt passend, gibt es aber nicht.

Frage 98:

`slapindex` ist das Kommando zum Neuindexieren einer OpenLDAP-Datenbank.

Frage 99:

B: `watch nfsstat -s -o nfs` ist geeignet. Das Kommando zeigt wiederholt Statistiken zu NFS und somit auch zu entsprechenden Clientzugriffen an.

E: `netstat -an |grep 2049` zeigt an, ob Verbindungen zu Port 2049 hergestellt wurden. Dieser Port wird von NFS verwendet.

zu A: `watch rpcinfo` zeigt RPC-Informationen an. Das beinhaltet keine Zugriffe durch NFS-Clients.

zu C: `watch showmount` zeigt Verbindungen zu NFS-Servern an. Das Programm wird clientseitig verwendet, was hier nicht weiterhilft.

zu D: `watch smbstatus` zeigt den Status eines Sambaservers an.

Frage 100:

D: `option domain-name-servers 192.168.50.1;` ist die einzig richtige Antwort.

zu A: `option domain-name "example.com";` weist dem Client eine Domäne zu, aber keinen zu verwendenden DNS-Server. Bei den anderen Antworten ist jeweils die Syntax falsch.

Frage 101:

A: *sssd* und E: *winbind* authentifizieren Benutzer aus Windows-Domänen und arbeiten direkt mit PAM zusammen.

zu B: *smbusers* verknüpft lediglich Windows-Konten mit Linux-Konten, um den Zugriff auf Samba-Server zu steuern.

zu C: *nsswitch.conf* ist zwar bei der Steuerung der Authentifizierung beteiligt, ist aber selbst kein Mechanismus zur Authentifizierung von Benutzern.

zu D: *passwd* kann nur mit lokalen Benutzerkonten umgehen.

Frage 102:

/etc/nsswitch.conf ist die richtige Antwort. In dieser Datei wird unter anderem die Reihenfolge der zu verwendenden Authentifizierungsmechanismen festgelegt.

Frage 103:

B: *mod_authz_host* oder C: *mod_access_compat* sind für diese Aufgabe geeignet, wobei *mod_authz_host* die modernere Variante ist und mehr Möglichkeiten bietet.

zu A: *mod_auth_basic* wird zur Klartextauthentifizierung verwendet.

zu D: *mod_authz_user* authentifiziert nur Benutzer, wie man schon aus dem Namen des Moduls schließen kann.

zu E: *mod_ssl* sorgt für SSL-Unterstützung, hat aber mit der Authentifizierung selbst nicht zu tun.

Frage 104:

E: `ldapadd` ist ein clientseitiges Tool, das für diese Aufgabe richtig ist.

zu A: `slapadd` kann auch verwendet werden. Da Sie dafür zunächst den Server stoppen müssten, wäre das aber nicht die beste Lösung

zu B: `slapcat` zeigt nur den Inhalt der LDAP-Datenbank an.

zu C: `slapindex` indexiert die Datenbank neu.

zu D: `slapd-config` dient der Konfiguration des Servers, fügt aber keine Datensätze hinzu.

Frage 105:

B: *DHCP-Relay* führt genau diese Aufgabe durch. Er dient als Agent zur Vergabe von IP-Adressen über Routergrenzen hinweg.

zu C: *radvd* dient der Konfiguration von IPv6-Adressen innerhalb eines Netzwerksegments.

zu A: *isc-dhcp3-server* ist ein DHCP-Server als solcher. Er operiert ohne *DHCP-Relay* nicht über Routergrenzen hinweg.

zu D: *BIND* dient lediglich der Namensauflösung.

zu E: *link-lokale Adressen* werden von Computern zur IPv6-Kommunikation innerhalb eines Netzwerksegments verwendet.

Frage 106:

B: `by * write` würde Schreibzugriff für jeden erteilen.

E: `by anonymous manage` gibt nicht authentifizierten Benutzern Verwaltungsrechte.

Die in den Antworten A, C und D enthaltenen Einträge sind typisch für ACLs und nicht als gefährlich einzustufen.

Frage 107:

B: `TSIG` ist die richtige Lösung. Mit TSIG ist es möglich, die Kommunikation zwischen DNS-Servern, aber auch zwischen Clients und Servern abzusichern, ohne eine aufwendige Infrastruktur öffentlicher Schlüssel (PKI) bereitzustellen.

zu A: `DNSSEC` ist PKI-basiert, was ausdrücklich nicht erwünscht war.

zu C, D und E: `RRSIG`, `DNSKEY` und `KEY` sind jeweils DNS-Eintragstypen, die in Zusammenhang mit DNSSEC verwendet werden.

Frage 108:

A: `DNSKEY` ist der richtige Eintragstyp zur Veröffentlichung öffentlicher DNSSEC-Schlüssel.

zu B: `RRSIG` dient der Signatur einzelner Einträge (Resource Records) innerhalb einer Zonendatei.

zu C: `SIG` ist ein veralteter Vorgänger von `RRSIG` und sollte nicht mehr verwendet werden.

zu D: `KEY` ist ein veralteter Vorgänger von `DNSKEY` und sollte nicht mehr verwendet werden.

zu E: `NS` ist ein gewöhnlicher Nameserver-Eintrag und hat mit DNSSEC nichts zu tun.

Frage 109:

A: `listen-on port 53 { 127.0.0.1; 192.168.5.1; };` ist hier die einzig richtige Antwort. Die Option `listen-on` legt fest, welche Adressen und welchen Port ein BIND-Server abhören soll.

zu B: Hier fehlen innerhalb der geschweiften Klammer zwei Leerzeichen. Diese sind aber unbedingt erforderlich.

zu C: In dieser Antwort wird ein falscher Port verwendet. Es ist zwar theoretisch machbar, dass innerhalb eines Netzwerks ein alternativer Port für DNS eingesetzt wird, wenn sowohl Server als auch Clients entsprechend konfiguriert werden, aber das ist unüblich. Die Frage verlangt außerdem ausdrücklich nach dem Standardport. Port 25 wird normalerweise für SMTP-Server verwendet.

zu D und E: Diese Antworten befassen sich mit dem falschen Thema. Hier wird festgelegt, von wo aus auf den DNS-Server zugegriffen werden darf. Es soll aber konfiguriert werden, welche Verbindungen der DNS-Server abhört. Bei Antwort E fehlen zusätzlich wieder zwei Leerzeichen innerhalb der geschweiften Klammern.

Frage 110:

E: `allow-query { 127.0.0.1; 192.168.5.0/24; };` ist die richtige Lösung. Die Option `allow-query` legt fest, von welchen Netzwerken aus auf den DNS-Server zugegriffen werden darf.

zu A, B und C: Diese Antworten befassen sich mit dem falschen Thema. Es wird hier jeweils festgelegt, welche Adressen und welchen Port der DNS-Server abhorcht. Es soll hier aber festgelegt werden, welchen Clients (unterschieden nach Netzwerksegment) der Server antwortet. Bei Antwort B ist aufgrund fehlender Leerzeichen zusätzlich die Syntax falsch und Antwort C verwendet zusätzlich einen für DNS unüblichen Port.

zu D: Das ist wieder wegen fehlender Leerzeichen falsch.

Frage 111:

`rndc reload` ist das optimale Kommando für diese Situation. Es gibt natürlich auch die Möglichkeit, den DNS-Server oder gar den ganzen Computer neu zu starten, aber das würde zu Unterbrechungen führen, die sicherlich nicht erwünscht sind.

Frage 112:

C: `rndc freeze` friert DNS-Schreibvorgänge ein. Der Daemon `named` wird dann keine Schreibvorgänge mehr ausführen.

zu A: `rndc stop` beendet die Ausführung des BIND-Servers. Das verhindert zwar ebenfalls die Schreibzugriffe, leider aber auch die Namensauflösung im Netzwerk.

zu B: `rndc thaw` können Sie verwenden, wenn Sie die Bearbeitung der Zone abgeschlossen haben. Schreibvorgänge werden dann wieder aufgetaut, also wieder fortgesetzt.

zu D: `rndc reload` sollten Sie abschließend verwenden, damit BIND die geänderte Zone neu einliest und der geänderte Inhalt verwendet wird.

zu E: `rndc halt` wird nur in absoluten Notfällen verwendet, um BIND sofort und gewaltsam zu beenden. Eventuell ausstehende Schreibvorgänge und Zonenübertragungen werden sofort abgebrochen.

Frage 113:

A: `IN MX 100 smtp02.lpic-2.de.`

B: `IN MX 80 smtp02.lpic-2.de.`

Beide Antworten sind richtig. Wenn der neue SMTP-Server lediglich als Reserve verwendet werden soll, muss seine Priorität höher eingestellt werden als die des eigentlichen Mail Exchangers. So wird sichergestellt, dass der ursprüngliche SMTP-Server bevorzugt verwendet wird (auch wenn das paradox klingt).

zu C: `IN MX 10 smtp02.lpic-2.de.` bringt nicht das gewünschte Ergebnis, weil der Wert für die Priorität zu niedrig gewählt wurde.

zu D und E: `IN MX 80 47.11.8.15` und `IN MX 100 47.11.8.15` verweisen jeweils auf eine IP-Adresse. MX-Records müssen jedoch zwingend auf den FQDN eines Mailservers zeigen. Nur Host (A) Einträge enden auf IP-Adressen.

Frage 114:

`SOA` (Start of a Zone of Authority) ist die richtige Antwort.

Frage 115:

B: `PTR` ist richtig. Hierbei handelt es sich um einen Pointer-Eintrag, der ausschließlich in Reverse-Lookup-Zonen von DNS-Servern anzutreffen ist.

zu A: `SOA` (Start of a Zone of Authority) ist sowohl in Forward-Lookup-, als auch in Reverse-Lookup-Zonen anzutreffen.

zu C: `NS` (Nameserver) findet man ebenfalls in beiden Zonentypen.

zu D: `AAAA` (Quad-A) ist die Host (A)-Entsprechung für IPv6 und ausschließlich in Forward-Lookup-Zonen zu finden.

zu E: `CNAME` (Canonical Name) ist ein Alias-Eintrag, der fast ausschließlich in Forward-Lookup-Zonen vorkommt.

Frage 116:

A: `rndc reload` und E: `kill -HUP` `` `pidof named` `` führen jeweils dazu, dass `named` die Konfigurationsdateien im laufenden Betrieb neu einliest. In der Praxis wird man natürlich aufgrund der schnelleren Verwendbarkeit `rndc` verwenden.

D: `/etc/init.d/named restart` ist ebenfalls eine funktionierende Antwort. Sie müssen diese Antwort zusätzlich wählen, weil drei Antworten gefordert waren, auch wenn der Neustart von BIND natürlich nicht als optimale Lösung angesehen werden kann.

zu B: `rndc thaw` setzt Schreibvorgänge fort, die zuvor mit `freeze` eingefroren wurden.

zu C: `rndc stop` beendet die Ausführung von BIND.

Frage 117:

E: `allow-update` ist die richtige Lösung. Diese Option legt fest, von welchen Clients ein Server dynamische Updates entgegen nimmt.

zu A: `allow update` verwendet eine falsche Syntax. Es fehlt der Bindestrich.

zu B: `allow-recursion` legt fest, für welche Clients eine rekursive Namensauflösung durchgeführt werden soll. Für alle übrigen Clients wird lediglich eine iterative (verweisende) Auflösung angeboten.

zu C: `allow-query` legt fest, welche Clients den Server überhaupt abfragen dürfen.

zu D: `allow-transfer` steuert, welche Computer sekundäre Zonen von diesem Server beziehen dürfen.

Frage 118:

D: `allow-transfer` ist die richtige Lösung. Diese Option legt fest, welche Computer sekundäre Zonen von diesem Server beziehen dürfen.

zu A: `allow update` ist nicht nur die falsche Option, sondern auch die Syntax ist falsch. Es fehlt der Bindestrich.

zu B: `allow-recursion` legt fest, für welche Clients eine rekursive Namensauflösung durchgeführt werden soll.

zu C: `allow-query` steuert, welche Clients den Server zur Namensauflösung verwenden dürfen.

zu E: `allow-update` legt fest, von welchen Clients ein Server dynamische Updates entgegennimmt.

Frage 119:

E: UDP/1194 wird in der Standardeinstellung von *openvpn* verwendet.

zu A: TCP/443 kommt bei HTTPS zum Einsatz.

zu B: TCP/500 ist für ISAKMP reserviert.

zu C: TCP/1701 transportiert Daten in L2TP-Verbindungen.

zu D: TCP/4500 sorgt bei IPSEC-Verbindungen für NAT-Traversal.

Frage 120:

D: `ip6tables -A INPUT -s 2001:db8:0:c4f8::/64 -j REJECT` lehnt Pakete, die dem globalen Netzwerk `2001:db8:0:c4f8::/64` entspringen, ab und sendet eine Fehlermeldung an den Client.

zu A: `ip6tables -A INPUT -s fe80::/64 -j REJECT` lehnt Pakete aus linklokalen Netzwerken ab.

zu B: `ip6tables -A INPUT -s 2001:db8:0:c4f8::/64 -j DROP` lehnt Pakete, die dem globalen Netzwerk `2001:db8:0:c4f8::/64` entspringen, zwar ab, sendet jedoch keine Fehlermeldung an den Client.

zu C: `ip6tables -A INPUT -s ff00::/8 -j DROP` verwirft Pakete, die von IPv6-Multicastadressen stammen.

zu E: `ip6tables -A OUTPUT -s 2001:db8:0:c4f8::/64 -j DROP` verwirft ausgehende Pakete zur angegebenen Adresse ohne Fehlermeldung.

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X

Z

Die Serviceseiten

Im Folgenden finden Sie Hinweise, wie Sie Kontakt zu uns aufnehmen können.

Lob und Tadel

Wir hoffen sehr, dass Ihnen dieses Buch gefallen hat. Wenn Sie zufrieden waren, empfehlen Sie das Buch bitte weiter. Wenn Sie meinen, es gebe doch etwas zu verbessern, schreiben Sie direkt an die Lektorin dieses Buches: *anne.scheibe@rheinwerk-verlag.de*. Wir freuen uns über jeden Verbesserungsvorschlag, aber über ein Lob freuen wir uns natürlich auch!

Auch auf unserer Webkatalogseite zu diesem Buch haben Sie die Möglichkeit, Ihr Feedback an uns zu senden oder Ihre Leseerfahrung per Facebook, Twitter oder E-Mail mit anderen zu teilen. Folgen Sie einfach diesem Link: *http://www.rheinwerk-verlag.de/4353*.

Zusatzmaterialien

Zusatzmaterialien (Beispielcode, Übungsmaterial, Listen usw.) finden Sie in Ihrer Online-Bibliothek sowie auf der Webkatalogseite zu diesem Buch: *http://www.rheinwerk-verlag.de/4353*. Wenn uns sinnentstellende Tippfehler oder inhaltliche Mängel bekannt werden, stellen wir Ihnen dort auch eine Liste mit Korrekturen zur Verfügung.

Technische Probleme

Im Falle von technischen Schwierigkeiten mit dem E-Book oder Ihrem E-Book-Konto beim Rheinwerk Verlag steht Ihnen gerne unser Leserservice zur Verfügung: *ebooks@rheinwerk-verlag.de*.

Über uns und unser Programm

Informationen zu unserem Verlag und weitere Kontaktmöglichkeiten bieten wir Ihnen auf unserer Verlagswebsite *http://www.rheinwerk-verlag.de*. Dort können Sie sich auch umfassend und aus erster Hand über unser aktuelles Verlagsprogramm informieren und alle unsere Bücher und E-Books schnell und komfortabel bestellen. Alle Buchbestellungen sind für Sie versandkostenfrei.

Rechtliche Hinweise

In diesem Abschnitt finden Sie die ausführlichen und rechtlich verbindlichen Nutzungsbedingungen für dieses E-Book.

Copyright-Vermerk

Ihre Rechte als Nutzer

Sie sind berechtigt, dieses E-Book ausschließlich für persönliche Zwecke zu nutzen. Insbesondere sind Sie berechtigt, das E-Book für Ihren eigenen Gebrauch auszudrucken oder eine Kopie herzustellen, sofern Sie diese Kopie auf einem von Ihnen alleine und persönlich genutzten Endgerät speichern. Zu anderen oder weitergehenden Nutzungen und Verwertungen sind Sie nicht berechtigt.

So ist es insbesondere unzulässig, eine elektronische oder gedruckte Kopie an Dritte weiterzugeben. Unzulässig und nicht erlaubt ist des Weiteren, das E-Book im Internet, in Intranets oder auf andere Weise zu verbreiten oder Dritten zur Verfügung zu stellen. Eine öffentliche Wiedergabe oder sonstige Weiterveröffentlichung und jegliche den persönlichen Gebrauch übersteigende Vervielfältigung des E-Books ist ausdrücklich untersagt. Das vorstehend Gesagte gilt nicht nur für das E-Book insgesamt, sondern auch für seine Teile (z. B. Grafiken, Fotos, Tabellen, Textabschnitte).

Urheberrechtsvermerke, Markenzeichen und andere Rechtsvorbehalte dürfen aus dem E-Book nicht entfernt werden, auch nicht das digitale Wasserzeichen.

Digitales Wasserzeichen

Dieses E-Book-Exemplar ist mit einem **digitalen Wasserzeichen** versehen, einem Vermerk, der kenntlich macht, welche Person dieses Exemplar nutzen darf. Wenn Sie, lieber Leser, diese Person nicht sind, liegt ein Verstoß gegen das Urheberrecht vor, und wir bitten Sie freundlich, das E-Book nicht weiter zu nutzen und uns diesen Verstoß zu melden. Eine kurze E-Mail an *service@rheinwerk-verlag.de* reicht schon. Vielen Dank!

Markenschutz

Die in diesem Werk wiedergegebenen Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. können auch ohne besondere Kennzeichnung Marken sein und als solche den gesetzlichen Bestimmungen unterliegen.

Haftungsausschluss

Ungeachtet der Sorgfalt, die auf die Erstellung von Text, Abbildungen und Programmen verwendet wurde, können weder Verlag noch Autor, Herausgeber oder Übersetzer für mögliche Fehler und deren Folgen eine juristische Verantwortung oder irgendeine Haftung übernehmen.

Über den Autor

Harald Maaßen hat langjährige Erfahrung als EDV-Dozent und Berater im Linux-Umfeld. Er leitet Vorbereitungskurse für die Zertifizierungsprüfungen des Linux Professional Institute.