NEU!

www.tecChannel.de Juni/Juli/August 2002

tecCHANNEL

tecCHANNEL COMPACT

KOMPENDIUM FÜR IT-PROFIS

€ 9,90
Österreich € 10,90
Benelux € 11,40
Schweiz SFR 19,80

Linux professionell einsetzen

Know-How und Praxis-Lösungen für den erfolgreichen Einsatz von Linux

Tux goes Business
Linux auf Desktops, Servern, Clustern und Großrechnern

Nie mehr Windows
Linux als Router, File/Print-, Messaging- und Webserver

Tuning
Die besten Bootkonfigurationen und Kernel-Einstellungen

Security
Workstations abschotten, Hackerangriffe abwehren

Netze abdichten
Firewalls aufsetzen und optimieren, Masquerading im Detail

Intra- und Internet
TCP/IP und DSL unter Linux optimal einrichten

01
4 195914 909900

Editorial

tecCHANNEL-Compact – praktisch und kompetent

Mit tecCHANNEL-Compact halten Sie die erste Ausgabe des tecCHANNEL-Kompendiums in Händen, mit dem wir Ihnen ein kompetentes Nachschlagewerk zu jeweils einem bestimmten Themenkomplex bieten. Das handliche Format des Magazins verbindet die Vorteile einer Zeitschrift mit denen eines Buches. Das beliebte Pocket-Format mit auf diese Größe angepassten Schriften und Bildern und eine von Büchern übernommene Leseführung sind die ideale Voraussetzung für eine intensive Lektüre und das effektive Umsetzen der Inhalte am Arbeitsplatz. Daneben finden Sie in der Compact-Ausgabe die bewährten Elemente aus dem tecCHANNEL-Magazin wie beispielsweise Links zum Thema, weiter führende Links, Glossare und die Webcodes für zusätzliche Informationen im Online-Angebot von www.tecChannel.de. Downloads von Scripts, Listings und Tools zu den Beiträgen können Sie online über die zu jedem Beitrag am unteren Seitenrand angegebenen Webcodes vornehmen.

In der ersten Ausgabe behandeln wir das Thema Linux für den professionellen Einsatz. Neben allgemeinen Informationen zur Entwicklung und Verbreitung des populären Opensource-Betriebssystems bietet Ihnen tecCHANNEL-Compact zahlreiche Know-how-Beiträge und Workshops zu fast allen Einsatzgebieten von Linux. So lesen Sie im Kapitel „Linux-Optimierungen“ unter anderem, wie Sie die Bootoptionen und Kernel optimal einrichten oder Linux via DSL ans Internet anbinden.

Im Abschnitt „Linux im Servereinsatz“ zeigen unsere Workshops die möglichen Einsatzgebiete von Linux im Firmenumfeld. Speziell Umsteiger von Unix- oder Windows-Systemen finden hier die wichtigsten Grundlagen für eine Portierung ihrer Anwendungen auf Linux. In den Kapiteln „Linux und Sicherheit“ und „Linux als Firewall“ gehen wir auf die Besonderheiten von Linux in sicherheitsrelevanten Umgebungen ein und zeigen, wie Sie Desktops und Firmennetze effektiv gegen Viren und Angreifer von außen schützen.

Viel Spaß mit der ersten Ausgabe von tecCHANNEL-Compact

wünscht Ihnen

Frank Klinkenberg

Chefredakteur/Associate Publisher tecCHANNEL

Wir freuen uns über Kritik und Anregungen zu dieser Compact-Ausgabe. Unter www.tecchannel.de/compact können Sie uns in einem Online-Fragebogen Feedback geben.

Impressum

Chefredakteur / Ass. Publisher: Frank Klinkenberg, (verantwortlich, Anschrift der Redaktion)

Chef vom Dienst: Kerstin Lohr

Grafik: H2Design, München; stroemung, Michael Rupp, Oliver Eismann, Köln; Yvonne Reittinger

Redaktion tecCHANNEL:

Leopoldstraße 252b, 80807 München, Tel.: 0 89/3 60 86-897, Fax: -878

Homepage: www.tecChannel.de, E-Mail: redtecchannel@idginteractive.de

Autoren dieser Ausgabe: Dr. Peter Bieringer, Oliver Drees, Mike Hartmann, Jörg Luther, Peter Klau, Oliver Müller, Konstantin Pfliegl, Holger Reibold, Jörg Reitter, Michael Rupp

Textredaktion: Kerstin Lohr, Claudia Feige

Anzeigen:

Anzeigenleitung: Dominique Remus, Tel.: 0 89/3 60 86-871

Leitung Anzeigendisposition: Rudolf Schuster, Tel.: 0 89/3 60 86-135, Fax -328

Anzeigentechnik: Martin Mantel, Andreas Mallin

Digitale Anzeigenannahme: Thomas Wilms, leitend, Tel.: 0 89/3 60 86-604, Fax -328

Vertrieb:

Vertriebsleitung: Josef Kreitmair

Vertriebsmarketing: Peter Priewasser (leitend), Stefanie Kusseler

Vertrieb Handelsauflage: MZV Moderner Zeitschriften Vertrieb, Breslauer Straße 5, 85386 Eching, Tel.: 0 89/3 19 06-0, Fax: -113, E-Mail: mzv@mzv.de, Website: www.mzv.de

Produktionsleitung: Heinz Zimmermann

Druck: Schoder Druck, Gutenbergstraße 12, 86368 Gersthofen

Verlag:

IDG Interactive GmbH, Leopoldstraße 252b, 80807 München, Tel.: 0 89/3 60 86-02, Fax: -501

Leserservice:

CSJ, Postfach 140220, 80452 München, Tel.: 0 89/20 95 91 32, Fax: 0 89/20 02 81 11

Geschäftsführer: York von Heimburg

Verlagsleitung: Stephan Scherzer (Mitglied der Geschäftsleitung)

Veröffentlichung gemäß § 8, Absatz 3 des Gesetzes über die Presse vom 8.10.1949: Alleiniger Gesellschafter der IDG Interactive GmbH ist die IDG Communications Verlag AG, München, eine 100-prozentige Tochter der IDG Inc., Boston, Mass., USA.

Vorstand: Keith Arnot, Kelly P. Conlin, York von Heimburg, Ralph Peter Rauchfuss

Aufsichtsratsvorsitzender: Patrick McGovern

Inhalt

1 Linux im Überblick

Als Linus Torvalds 1991 mit der Entwicklung von Linux begann, war nicht abzusehen, dass sich aus dem Hobby des jungen Finnen ein ernst zu nehmendes Betriebssystem entwickelt. In diesem ersten Kapitel geben wir Ihnen einen Überblick über die Geschichte von Linux und darüber, wie sich das Opensource-Betriebssystem immer mehr auch im professionellen Umfeld etabliert.

1.1 Linux, wie alles begann

Vor über zehn Jahren hat Linus Torvalds im Usenet zum ersten Mal sein Projekt vorgestellt: „ein (kostenloses) Betriebssystem“. Von „es wird nichts Großes oder Professionelles wie GNU“ kann heute jedoch keine Rede mehr sein. Im Gegensatz zur weit verbreiteten Meinung ging es Linus Torvalds primär nicht darum, einen kostenlosen Unix-Ersatz zu schaffen. Es fing damit an, dass das Rechenzentrum seiner Universität 1990 zwar über eine microVAX mit Ultrix verfügte, aber nicht genug Rechenleistung für die Studenten bereitstellen konnte. Dennoch kam für Torvalds ein 386er nicht in Frage: „Dann hätte ich ja mit diesem lausigen Betriebssystem MS-DOS arbeiten müssen und nichts gelernt“, sagt er 1997 in einem Interview mit „Wired“.

Erst als er in einem Universitätskurs mit Andrew S. Tanenbaums Minix in Kontakt kommt, entscheidet Torvalds sich, einen PC zu kaufen. Zunächst geht es ihm lediglich darum, die Task-Switching-Fähigkeiten des 80386 zu verstehen. Sein erstes Erfolgserlebnis ist ein Minix-Programm aus zwei Prozessen, die abwechselnd die Zeichenfolgen AAAA und BBBB auf den Bildschirm bringen. Im nächsten Schritt erweitert Linus das Programm zu einem Newsreader: Der eine Task bringt die News vom Modem auf den Bildschirm und der andere von der Tastatur zum Modem - allerdings immer noch unter Minix.

Aber Linus Torvalds hat bereits Blut geleckt. „Zu diesem Zeitpunkt hatte ich bereits gemerkt, dass Minix nicht genug ist. Fehlende Jobkontrolle, hässliches Speichermanagement, keine Unterstützung für FPUs und so weiter“, erklärt er der „Linux News“ (http://alge.anart.no/linux/history/LinuxNews.03A) Mitte Oktober 1992 in einem Interview.

1.1.1 Der erste Schritt

Weitere Kritikpunkte von Torvalds waren, dass Minix ein rein „akademisches“ Betriebssystem ist und aus Gründen der Portierbarkeit nur den kleinsten gemeinsamen Nenner der damals verfügbaren Prozessorarchitekturen (8088, 68000, Sparc) verwendet. Dementsprechend nutzt es auch nicht die besonderen Fähigkeiten des 80386.

Torvalds beginnt mit seiner Mammutaufgabe: einem komplett neuen Betriebssystem, dessen erstes sichtbares Anzeichen sich am 3. Juli 1991 in Form eines Postings in comp.os.minix offenbart. Darin fragt er nach einer Definition der Posix-Standards, damit sich Anwendungsprogramme leichter auf das zu diesem Zeitpunkt noch namenlose Betriebssystem portieren lassen.

Doch die Posix-Spezifikationen sind nur gegen Bezahlung vom Standardkomitee erhältlich. Linus muss sich einen anderen Weg suchen, um eine Programmierschnittstelle für sein Betriebssystem zu schaffen. Zu diesem Zeitpunkt meldet sich Ari Lemmke bei Linus und verweist ihn auf die GNU libc.a (www.gnu.org), eine Bibliothek mit Funktionen des ANSI-C-Standards und Posix-Features.

Ari richtet auch gleich das erste öffentliche Linux-Verzeichnis (/pub/OS/Linux) auf nic.funet.fi ein, obwohl dort noch für einige Zeit lediglich ein README zu finden ist: „Dieses Verzeichnis ist für den frei verteilbaren Minix-Clone". Ari ist es übrigens zu verdanken, dass Linux heute „Linux" heißt: Linus will das Betriebssystem eigentlich „Freax" (Kunstwort aus free, freak und dem x von Unix) taufen. Den Arbeitstitel Linux will er nicht verwenden, weil er Angst hat, als „Egomane" beschimpft und nicht ernst genommen zu werden, erklärt er „Wired". Doch Ari findet Linux besser und legt somit den Grundstein.

1.1.2 Linux 0.01

Doch immer noch besteht Linux lediglich aus einem rudimentären Protected-Mode-System mit einem AT-Treiber und dem Minix-Dateisystem. Um nicht völlig richtungslos zu entwickeln, wendet sich Linus am 25. August 1991 erneut an die Benutzer von comp.os.minix:

```
Path: icdoc!ukc!mcsun!news.funet.fi!hydra!klaava!torvalds
From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Newsgroups: comp.os.minix
Subject: What would you like to see most in minix?
Summary: small poll for my new operating system
Keywords: 386, preferences
Message-ID: <1991Aug25.205708.9541@klaava.Helsinki.FI>
Date: 25 Aug 91 20:57:08 GMT
Organization: University of Helsinki
Lines: 20

Hello everybody out there using minix -

I´m doing a (free) operating system (just a hobby, won´t be
big and professional like gnu) for 386(486) AT clones.  This
has been brewing since april, and is starting to get ready.
I´d like any feedback on things people like/dislike in
minix, as my OS resembles it somewhat (same physical layout
of the file-system (due to practical reasons) among other
things).
I´ve currently ported bash(1.08) and gcc(1.40), and things
seem to work. This implies that I´ll get something practical
```

```
within a few months, and I'd like to know what features most
people would want.  Any suggestions are welcome, but I won´t
promise I´ll implement them :-)

                Linus (torvalds@kruuna.helsinki.fi)

PS. Yes - it´s free of any minix code, and it has a multi-
threaded fs. It is NOT protable (uses 386 task switching
etc), and it probably never will support anything other than
AT-harddisks, as that´s all I have :-(.
```

Damit ist Linux „offiziell"! Die erste Version linux-0.01.tar.gz geht dann im September an nur wenige Mitglieder von comp.os.minix. Genau weiß selbst Linus das nicht mehr, aber die jüngste Datei im tarball ist vom 17. September 1991 18:29. Den ersten „offiziellen" Linux-Kernel (Version 0.02) kündigt Linus am 5. Oktober in comp.os.minix an.

Bis Dezember „lernt" Linux (Version 0.11) virtuellen Speicher, Version 0.12 ist der erste Kernel, der auch „nicht essenzielle" Features enthält, etwa Unterstützung für 100 mal 40 Zeichen im SVGA-Modus oder die Punkte „." beim Laden des Kernels. Auch die virtuellen Konsolen, die sich mittels ALT- und Funktionstaste umschalten, geben hier ihr Debüt.

1.1.3 Torvalds gegen Tanenbaum

Die zunehmende Diskussion um Linux in comp.os.minix ruft Andrew S. Tanenbaum, den Entwickler von Minix, auf den Plan. Er schaltet sich mit einem Posting unter dem Titel „LINUX is obsolete" in die Diskussion ein und entfacht einen regelrechten Flamewar. Tanenbaums Hauptpunkte sind:

- Linux hat einen monolithischen Kernel und ist damit ein Rückschritt in die Siebziger. Microkernels wie bei Minix sind die Zukunft.
- Linux ist nicht portabel, weil es sich zu sehr auf die Features des 80386 stützt. Und der 80386 werde ohnehin von RISC-Chips abgelöst. Welch kolossaler Irrtum :-)

Tanenbaum nutzt die Gelegenheit und verweist auf sein neues Projekt Amoeba. Dabei handelt es sich um ein verteiltes, parallelisiertes Betriebssystem mit Microkernel-Architektur, das auf Sparcs, 386/486, 68030 sowie Sun 3/50 und Sun 3/60 läuft. Gleichzeitig weist er darauf hin, dass Minix nur ein Hobby von ihm sei und er ja hauptberuflich als Professor für Betriebssysteme arbeite - eventuell um die „Stichhaltigkeit" seiner Argumente qua Position zu untermauern.

Linus revanchiert sich mit dem Kommentar: „Ist das Ihre Entschuldigung für die Schwächen von Minix? Schade, aber ich habe mehr Ausreden und trotzdem ist Linux in fast allen Belangen überlegen".

Der Seitenhieb „Abgesehen davon scheint der meiste gute Code in Minix von Bruce Evans zu stammen“ ist da eigentlich schon gar nicht mehr nötig. Aber auch seine anderen Konter sind wirklich lesenswert:

- Minix ist Ihr Hobby? Dann schauen Sie doch mal, wie viel Geld mit Minix verdient wird und wer alles Linux kostenlos weitergibt. Ich betreibe Linux als echtes Hobby. Ich verdiene weder Geld damit, noch ist es in irgendeiner Form Teil meiner Studien.
- Ihr Beruf ist Professor? Das ist eine wirklich verdammt gute Erklärung für einige der „Gehirnschäden“ von Minix.
- Wenn die Programmierung als Microkernel das einzige Kriterium für einen „guten“ Kernel ist, dann wäre Minix tatsächlich besser. Was Sie aber nicht erwähnen ist, dass Minix das „Microkernel-Ding“ nicht besonders gut macht, und auch mit dem Multitasking seine Probleme hat. Wenn ich ein OS geschrieben hätte, das sogar mit einem multithreaded Dateisystem Schwierigkeiten hat, wäre ich nicht so schnell dabei, andere zu verurteilen. Ich würde alles tun, um dieses Fiasko geheim zu halten.
- Portabilität ist eine gute Sache, aber nur, wo es Sinn macht. Die Essenz eines Betriebssystems ist doch, die Hardware-Features der Plattform auszunutzen und sie hinter einer Schicht von APIs zu verstecken.

Sein Abschluss der Mail:

„P.S. Entschuldigung, wenn ich teilweise etwas hart klinge. Minix ist ein gutes Betriebssystem, wenn man gerade nichts anderes hat. Amoeba könnte nett sein, wenn man zufällig fünf bis zehn 386er in der Ecke stehen hat, ich habe das nicht. Normalerweise lasse ich mich nicht zu Flamewars hinreißen, aber ich bin ein wenig empfindlich, wenn es um Linux geht.“

Inzwischen sind zehn Jahre vergangen, und Andrew S. Tanenbaum schreibt in seiner persönlichen FAQ (www.cs.vu.nl/~ast/home/faq.html) auf die Frage, was er von Linux hält: „Ich möchte die Gelegenheit nutzen, Linus dafür zu danken, es gemacht zu haben. [...] Wir beide sind jetzt froh über die Ergebnisse. Die einzige Person, die vielleicht nicht so glücklich ist, ist Bill Gates.“

1.1.4 Linux, the gathering

Im März 1992 glaubt Linus, dass Linux für eine offizielle Version 1.0 bereit ist und erhöht die Versionsnummer von 0.12 auf 0.95. Inzwischen beteiligt sich schon eine Reihe von Hobbyprogrammierern an der Entwicklung von Linux. Version 0.95 ist auch die erste, die ein „Unix-ähnliches“ Login aufzuweisen hat. Doch bis zum endgültigen Release des Kernel 1.0 wird es noch über zwei Jahre dauern (16. April 1994). In diesem Zeitraum wächst die Größe des Source-Tarballs von 138 KByte auf 1,2 MByte und die Anzahl der in den Credits erwähnten Entwickler auf 79, darunter auch eine ganze Reihe von Deutschen.

Im Jahr 2002 sind es gut 200 „offiziell“ erwähnte Entwickler. Die Anzahl ungenannter Helfer ist nur schwer zu schätzen, ebenso wie die Anzahl der Rechner unter Linux. Gerade im (Web-)Serverbereich erfreut sich Linux inzwischen größter Beliebtheit. Für Linux-Desktops finden sich ebenfalls immer mehr Benutzer, da auch Linux inzwischen einen Komfort erreicht hat, der in vielen Belangen nahe an den von Windows heranreicht. Vom „Betriebssystem von einem Hacker für Hacker“, wie es Linus in seinem offiziellen Statement zur 0.02 genannt hat, ist Linux inzwischen weit entfernt. Nicht umsonst hat sich rund um Linux eine Firmenlandschaft entwickelt, die eine Vielzahl von Diensten und Programmen für das freie Betriebssystem anbietet.

1.1.5 Die Story geht weiter

Den letzten großen Schritt vollzieht Linux seit etwa zwei Jahren. Mehr und mehr große Firmen steigen auf den bereits rollenden Zug auf. Weitaus am energischsten in Richtung Linux im Enterprise zeigt sich dabei IBM.

Tatsächlich beweisen die Mannen aus Armonk schon geraume Zeit eine gesteigerte Affinität zu Linux. Die bewegte sich allerdings bislang in eher ungewöhnlichen Bahnen und brachte Linux auf Mainframes und ins Supercomputing. So verträgt sich das freie Unix inzwischen bestens mit den großen Eisen der S/390-Familie. Hier greift besonders die Kundschaft aus dem akademischen Bereich gern zu, aber auch kommerzielle Webserver operieren bereits auf S/390-Systemen unter Linux.

Big Iron: Mit den Mainframes der IBM z-Series, wie dieser S/390, kommt Linux inzwischen ebenfalls problemlos klar. (Bild: IBM)

Dabei will IBM es nicht bewenden lassen - über das ganze Firmenportfolio hinweg soll Linux das e-Business der Zukunft befördern. Für jede Hardware, vom eServer x-Series bis zum Netvista-Thin-Client, und für jede Software, von Lotus Domino bis zu Tivolis Netzwerkmanagement-Lösungen, will Big Blue den Einsatz des freien Unix unterstützen.

„Wir offerieren nicht nur grundlegenden Linux-Support für unsere Hard- und Software. Wir investieren auch in Linux-Services, um unseren Kunden den gewohnten Support für ihre Unternehmensumgebungen auch unter Linux zu bieten“, präzisiert Irving Wladawsky-Berger, Vizepräsident für Technologie und Strategie bei IBMs Server Group.

Aber auch HP und Intel zeigen mit ihren jüngsten Pressemitteilungen, dass sie das Thema Linux durchaus ernst nehmen. Daneben finden sich auch immer mehr Hersteller von PDAs, die Linux statt PalmOS oder PocketPC 2002 für die mobilen Kleinstrechner einsetzen. Erste Geräte und Projekte stammen beispielsweise von Agenda, Compaq, Gmate, Invair oder Sharp.

1.1.6 Linux gegen Windows

Die zunehmende Beliebtheit von Linux findet jedoch nicht jedermanns Zustimmung. Speziell Microsoft fühlt sich offenbar zunehmend unter Druck gesetzt. Das hat sich der Softwaregigant größtenteils selbst zuzuschreiben. Zu wenig hat sich das Unternehmen aus Redmond in den letzten Jahren um die offensichtlichen Bedürfnisse der professionellen Anwender gekümmert.

Nach einer Phase der völligen Ignoranz folgten vor drei Jahren die berüchtigten Halloween Papers (http://winnetou.lcd.lu/halloween.html): In der internen Studie, deren Autorschaft Microsoft niemals offiziell zugab, deren Herkunft aber als unzweifelhaft gilt, tat das Unternehmen die Opensource-Entwicklung in Bausch und Bogen als unzulänglich und lächerlich ab.

Auch in weiteren Reaktionen, wie einer bereits berüchtigten Anzeige (siehe nächste Seite) in einem bekannten deutschen Computermagazin, beschränkte sich Microsoft auf den Versuch, Linux ins Lächerliche zu ziehen.

Doch inzwischen bekommt es Microsoft wohl mit der Angst zu tun. Anders lässt sich der Versuch von Jim Allchin, Leiter der Windows-Entwicklung bei Microsoft, die US-Legislative gegen Linux aufzuwiegeln, kaum erklären. Mitte Februar 2001 beschuldigte Allchin in einem Interview des US-Nachrichtendienstes Bloomberg die Opensource-Bewegung, „intellektuelles Eigentum zu zerstören“, „Forschung und Entwicklung zu behindern“ und dadurch die „Innovation zu bremsen“. Er forderte die Gesetzgebung auf, sich „auf diese Bedrohung einzustellen“.

Ein loses Mundwerk hat nicht nur Vorteile: Mit diesem Anzeigenmotiv blamierte sich Microsoft gründlich.

Zumindest in Deutschland wird diese vorgebliche Bedrohung nicht gesehen. Die im April 2002 im deutschen Bundestag gefallene Entscheidung, Linux auf den 150 Servern des Bundestages zu installieren zeigt, dass der Staat die Probleme bei Microsoft sieht als bei Linux. Dies war ein herber Rückschlag für Microsoft, was sich auch in der Reaktion des deutschen Microsoft Chefs Kurt Siebold zeigte. Er wendete sich zuvor in einem offenen Brief an die „Erstunterzeichner" der Linux-Kampagne von Werk21 um das Vorhaben zu verhindern. Unter Bundestux.de hat die Initiative mit Unterschriften von Abgeordneten für einen Einsatz von Linux im Bundestag geworben. Einen Teilerfolg konnte Microsoft jedoch verbuchen: Bei der Aufrüstung der 5000 Arbeitsplatzrechner der Abgeordneten kommt jetzt Windows XP zum Einsatz.

1.1.7 Ausblick

In mehr als zehn Jahren hat Linux mehr erreicht, als eine ganze Reihe anderer Produkte oder Firmen:

- Es hat eine eingeschworene Gemeinde von Entwicklern geschaffen, wie es sie für kein zweites Betriebssystem gibt; vielleicht nicht unbedingt zahlenmäßig, aber hinsichtlich Einsatz und Enthusiasmus.

- Es hat eine eingeschworene Schar von Benutzern um sich gesammelt, die Linux benutzen, weil sie Spaß daran haben und nicht, weil sie keine andere Wahl haben.
- Es hat - trotz aller Unkenrufe - einen Level an Leistungsfähigkeit und Funktionalität erreicht, der „unerfahrene“ Endbenutzer und große Firmen gleichermaßen überzeugt.

Und last but not least: Es hat über all die Jahre - trotz so manchen heftigen System-Crashes - Spaß gemacht, und Spaß war schließlich die ursprüngliche Intention von Linus Torvalds.

tecCHANNEL Links zum Thema	Webcode
Interview mit Linus Torvalds	a551
Happy Birthday, Linux!	a758
Tux goes Biz	a654

1.2 Tux goes Biz

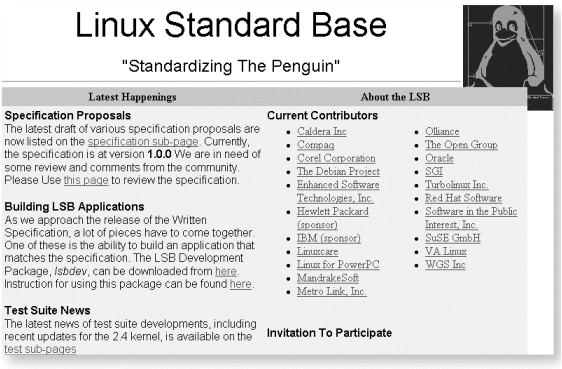
Die Kernel-Version 2.4 katapultiert Linux endgültig in die Riege ernst zu nehmender Profi-Betriebssysteme. Damit öffnen sich für Linux die letzten verschlossenen Türen: die unternehmenseigenen Rechenzentren.

Für Jon „Maddog“ Hall, seines Zeichens Director of Linux Evangelism bei VA Linux, ist die Sache klar: Linux hat den Durchbruch als Desktop-Betriebssystem schon geschafft. Nach seiner Zählung läuft das freie Unix auf 6 Prozent aller PCs und hat damit das Mac OS bereits überholt. Wie er gern mit einem Augenzwinkern anfügt, umfasst diese Zahl noch nicht einmal jene 1,3 Milliarden Chinesen, für die Linux auf Grund der Entscheidung ihrer Regierung das Betriebssystem „der Wahl“ sein wird.

Doch nicht alle Linux-Enthusiasten teilen den von Maddog Hall an den Tag gelegten Optimismus. Noch hält Microsoft mit seinen Windows-Versionen und einem Marktanteil von 90 Prozent auf Desktops die Vorherrschaft. Zwar bieten aktuelle Linux-Distributionen einen hohen Komfort bei Installation und Betrieb, erreichen aber nicht durchgängig die gleiche Benutzerfreundlichkeit wie die Produkte aus Redmond. Die größte Stärke des freien Unix - die durch die Opensource-Entwicklung bedingte Vielfalt - wird hier zum Hemmschuh.

1.2.1 Standardisierung für Linux

Ein Erfolg versprechender Ansatz, der verwirrenden Linux-Vielfalt eine gemeinsame Basis zu verleihen, ohne deswegen in rigide Reglementierung zu verfallen, ist die Linux Standard Base (LSB, www.linuxbase.org). Sie versucht, für alle Linux-Varianten einen Kern gemeinsamer Merkmale zu definieren, angefangen von der Verzeichnisstruktur bis hin zum Paketformat.



Who´s who: Auf der Mitgliederliste der Linux Standard Base stehen alle wichtigen Linux-Anbieter.

Die so erzielte Kompatibilität soll Entwicklern wie Anwendern gleichermaßen zugute kommen. Dem Programmierer bieten die LSB-Spezifikationen eine solide Grundlage, auf der er seine Applikationen mit einheitlicher Codebasis aufsetzen kann. Die Anwender bekommen die Sicherheit, jede Applikation ohne spezifische Anpassungen auf jeder LSB-konformen Distributionsplattform installieren und ausführen zu können.

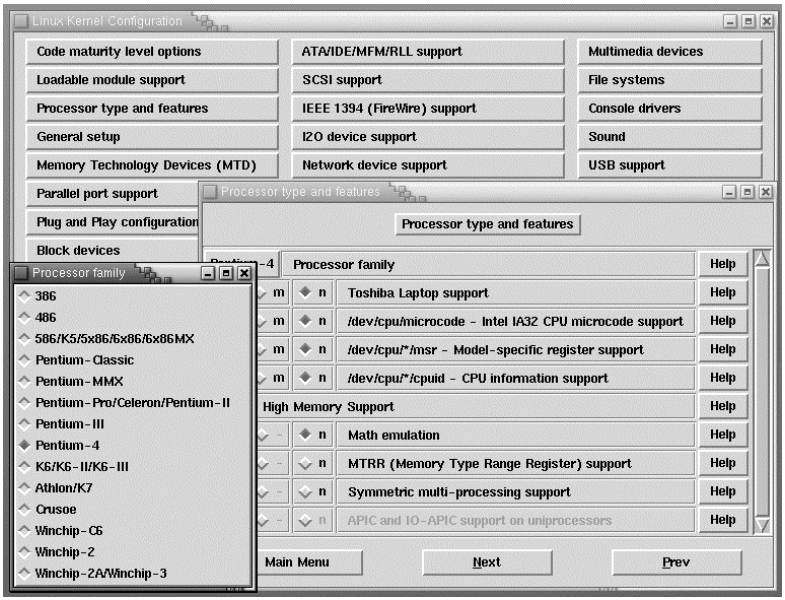
Das heute noch oft nötige Anpassen und Rekompilieren von Software, die verwirrende Vielfalt von Paketformaten und distributionsspezifischen Binaries, all das könnte schon bald der Vergangenheit angehören. Damit wäre ein wesentlicher Stolperstein für die weitere Verbreitung von Linux auf dem Desktop beseitigt. Das liegt nicht zuletzt auch im Interesse der großen Distributoren, sei es Red Hat, SuSE, TurboLinux, Mandrake, Caldera oder Debian, die deswegen das LSB-Projekt unterstützen.

1.2.2 Kernel 2.4

Das Release der neuen, in der Community schon im Vorfeld heiß diskutierten Kernel-Generation ließ Anfang Januar 2002 so manchen Tux-Jünger zum FTP-Client greifen. Auf den Download des aktuellen Tarball (www.kernel.org/pub/linux/kernel/) folgte allerdings oft die Ernüchterung: 119 MByte Sourcen - damit fällt der neue Kernel rund 30 Prozent größer aus als sein Vorgänger. Das Gros der „Gewichtszunahme“ entfällt auf die stark verbesserte Modulausstattung, Linux 2.4 bringt also deutlich mehr Treiber mit. Daneben verfügt der neue Kernel über zahlreiche neue Highlights.

Zu den interessantesten Features zählt der komplett integrierte USB-Support. Er erlaubt den Einsatz aller Devices von Maus und Tastatur bis hin zu Massenspeichern mit Hot-Plug-Funktionalität. Daneben bietet Linux 2.4 jetzt direkte Unterstützung für CPUs der Typen Athlon, Duron und Pentium 4 (siehe auch nächste Seite). Ab Version 2.4.1 integriert der Kernel ReiserFS, ein sehr effektives Btree-basiertes Dateisystem. Es verwaltet die Allokierungsinformationen in einem für schnellen Zugriff optimierten Binärbaum und macht so die feste Zuordnung von Inodes überflüssig. Dies spart ebenso wie das Abspeichern kleiner Dateien direkt im Verzeichnisbaum Plattenplatz.

Eine Direct Rendering Infrastructure (DRI) bietet verbesserten OpenGL-Support für Xfree86 Version 4. Die resultierende Beschleunigung von 3D-Bildschirmausgaben erfreut nicht nur CAD-Profis, sondern auch die Fans von Spielen wie Unreal Tournament oder Quake III.



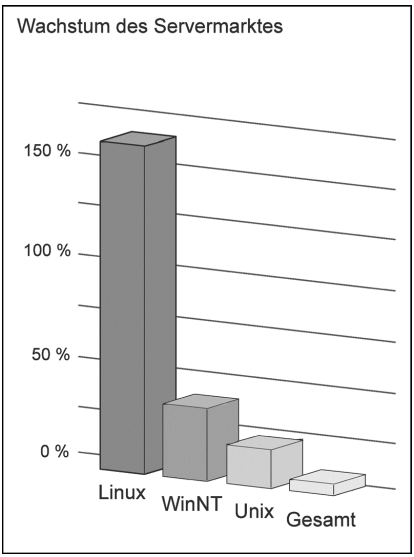
Aufgebohrt: Linux 2.4 unterstützt zahlreiche CPUs und bis zu 64 GByte RAM.

Auch beim Einsatz auf den kleinsten Rechnern bringt Linux 2.4 Vorteile. Seine hohe Modularität ermöglicht maßgeschneiderte, kompakte Systeme für Handhelds wie Compaq iPAQ, Agenda VR3d oder Embedded Systeme. Außerdem unterstützt der Kernel Winchip- und Crusoe-CPUs, also Prozessoren für den integrierten und mobilen Einsatz. ROM und Flash-Speicher spricht Linux 2.4 als Festplatten an (Disk on Chip). Als passendes Dateisystem fungiert das mit Kompression operierende cramfs, dessen Name sich vom englischen „to cram“ = vollstopfen ableitet.

1.2.3 Linux auf dem Server

Während Linux auf dem Desktop noch um seinen Platz kämpft, hat es im Bereich der professionellen Datenverarbeitung bereits fest Fuß gefasst. Schon letztes Jahr überholte das freie Unix den ehemaligen Klassenprimus Novell Netware als Serverbetriebssystem und schickte sich an, Windows den Rang als NOS Nr. 1 streitig zu machen. 2001 war laut IDC schon jeder vierte (27 Prozent) verkaufte Server mit Linux bestückt, zwei von fünf (41 Prozent) mit Microsoft-Produkten. Novell Netware und Unix mussten sich mit jeweils rund 14 Prozent Marktanteil bescheiden.

Wie die Zahlen der Marktforscher von IDC belegen, setzt sich dieser Trend weiterhin fort. Mitte 2001 war Linux das mit 170 Prozent am schnellsten wachsende Serverbetriebssystem, gefolgt von Windows NT/2000 mit einem Plus von nur 37 Prozent. Die Vergleichszahl für den Servermarkt insgesamt lag dagegen bei 8 Prozent. Der Erfolg von Linux demonstriert nach Meinung der IDC-Profis den wachsenden Bedarf nach einer „verlässlichen und kosteneffektiven Plattform für das Internet und andere Kernarbeitsbereiche“.



Überholspur: Auf dem Servermarkt zieht Linux den anderen Netzwerk-Betriebssystemen davon. (Umsatzwachstum, Quelle: IDC)

Da zudem immer mehr namhafte Hersteller Linux unterstützen und in ihr Lösungsprogramm einbauen, prophezeit IDC dem freien Betriebssystem ein nahezu unbegrenztes Wachstum bis wenigstens 2005.

Schon 2003 sollen die entsprechenden Umsätze ein Volumen von 4,4 Milliarden US-Dollar erreichen. In einer ausführlichen Studie bescheinigen die Bostoner Marktforscher Linux denn auch den Wandel vom Early-Adopter- zum Mainstream-System: „Linux wird 2005 einen Platz als fester Bestandteil der Enterprise-IT-Umgebung einnehmen.“

1.2.4 Linux 2.4 im Enterprise-Markt

Bislang kam das Opensource-Unix allerdings überwiegend in Kombination mit Apache (www.apache.org) als Webserver zum Einsatz. Die Türen der großen Rechenzentren blieben Linux dagegen in vielen Fällen verschlossen. Der Grund dafür lässt sich in zwei Wörtern zusammenfassen: mangelnde Skalierbarkeit.

Die in Enterprise-Szenarios übliche Versorgung tausender User erfordert den Einsatz leistungsfähiger Multiprozessormaschinen. Server mit acht CPUs gelten in diesem Segment als „Einstiegsmodelle". Linux kam bislang nur mit zwei Prozessoren und maximal 4090 Prozessen klar. Ähnlich hinderliche Einschränkungen betrafen den Datenbank-Support: So betrug die maximale Größe einzelner Dateien 2 GByte - ein Wert, über den etwa Oracle-Administratoren nur herzhaft lachen können.

Diese Limits räumt der Kernel 2.4 aus dem Weg. So geht Linux nun auch mit mehr als zwei Prozessoren im Rechner effektiv um. „Wir schätzen, dass der Kernel bis 32 Prozessoren skaliert, und vermutlich noch darüber hinaus", prognostiziert Michael Tiemann, CTO von Red Hat (www.redhat.com). Außerdem limitiert nur die Größe des Hauptspeichers noch die Anzahl der gleichzeitig bearbeiteten Prozesse. Da der Kernel mit bis zu 64 GByte RAM klarkommt, sind hier kaum Engpässe zu befürchten. Applikations- und Webserver können damit nun wesentlich mehr Anwender bedienen.

Auch den Datenbankbetreibern bietet Linux 2.4 neue Möglichkeiten: Der integrierte Large File Support hebt die bisherige Beschränkung des Dateiumfangs auf. Lediglich die Kapazität des Speichermediums setzt der Größe eines Files gewisse Grenzen. Die sind jedoch nicht allzu eng gezogen, da sich mit Hilfe des Logical Volume Manager (LVM) Dateien auch über mehrere Festplatten verteilen lassen.

1.2.5 Kernel 2.5 und IA64

Schon in der nächsten Developer-Release plant Linus Torvalds, den Kernel weiter in die Profi-Richtung auszubauen. Erste Ergebnisse dürften nicht mehr in allzu weiter Ferne liegen: In einem Posting in der Kernel-Mailing-Liste vom 11. Oktober 2001 kündigt der Linux-Guru an, demnächst („really soon now") mit der Arbeit an 2.5.x zu beginnen.

Zu den angepeilten Verbesserungen zählen neben aufpolierten Clustering-Fähigkeiten die Skalierbarkeit auf NUMA-Architekturen und „anderen großen Maschinen". Auch den I/O-Support sowie speziell den SCSI-Layer will Torvalds gründlich überarbeiten, wie er kürzlich in einem Interview präzisierte.

Der Terminus „andere große Maschinen" dürfte sich wohl auf Intels IA-64- und die neue AMD-64-Bit-Architektur beziehen. Die Distributoren Caldera (www.caldera.com), Red Hat, SuSE (www.suse.de) und Turbolinux (www.turbolinux.com) bieten bereits Pakete für IA-64-Maschinen an. SuSE ist zudem sehr

aktiv bei der Anpassung von Linux an die 64-Bit-AMD-Hammer-CPUs. Die Server-CPUs sollen Anfang 2003 unter dem Namen Opteron auf den Markt kommen.

Noch skalieren auf Maschinen mit mehr als acht Prozessoren klassische Business-Unixe wie HP-UX (www.hp.com) deutlich besser als Linux. Dieses Manko will Linus Torvalds mit der nächsten Stable Release (die möglicherweise nicht mehr 2.6, sondern 3.0 heißt) ausräumen.

Damit verlässt Linux endgültig seine Kinderstube und bringt alle wichtigen Voraussetzungen für den professionellen Einsatz in Unternehmen jeder Größenordnung mit.

1.2.6 Einstieg der Big Player

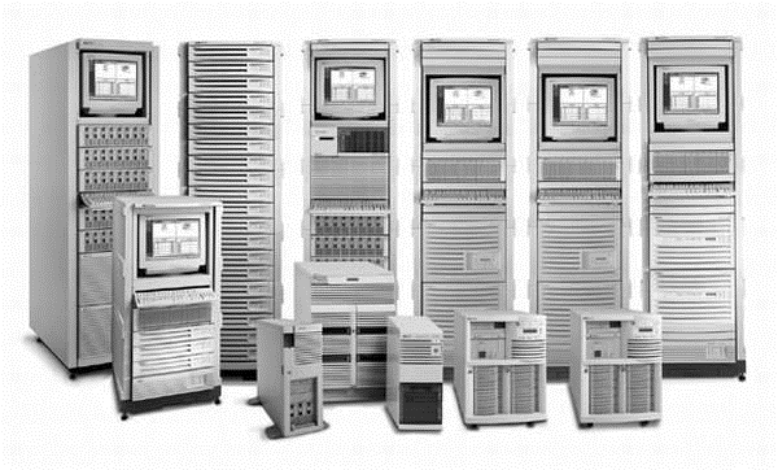
Solche Aussichten überzeugen zunehmend auch die Größen der IT-Branche davon, mit Linux auf das richtige Pferd zu setzen. Weitaus am energischsten in Richtung Linux im Enterprise bewegt sich dabei IBM (siehe bereits S. 15 ff).

Dass es sich dabei nicht nur um Lippenbekenntnisse handelt, zeigen die angedachten Investitionssummen. So bekräftigten CEO Lou Gerstner und Chief Operations Officer Sam Palmisano mehrfach, dass Big Blue eine Milliarde US-Dollar in das Opensource-Unix stecken will. Rund ein Drittel davon investiert IBM bis 2003 in den Aufbau professioneller Dienstleistungsangebote.

1.2.7 Linux-Elefantenhochzeit

Bei IBM handelt es sich jedoch nicht um den einzigen Big Player, der auf Linux setzt. Ein weiterer Linux-lastiger Gigant wurde gerade in Form der „neuen Hewlett-Packard“ aus der Taufe gehoben. Die Fusion von HP und Compaq hat zwei Unternehmen zusammengeführt, deren Portfolios schon für sich betrachtet jeweils einen hohen Linux-Anteil boten.

Schon vor der Elefantenhochzeit positionierte sich Compaq mit 24 Prozent aller ausgelieferten Linux-Server (in 2000) als größter Anbieter in diesem Markt. Sowohl auf den Zugpferden der Proliant-Serie als auch den Appliances der Tasksmart-Reihe kommt das freie Unix zum Einsatz. Das Sahnehäubchen des Angebots stellen jedoch zweifellos die Systeme auf Basis der Alpha-CPU dar. Einzeln oder im Cluster kommen diese Number Cruncher für anspruchsvollste Aufgaben wie Wettersimulation oder Filmproduktion zum Einsatz. So wurden beispielsweise die Special Effects für „Titanic“ auf Alpha-Servern unter Red Hat Linux gerendert.



Linux für alle: Hewlett-Packard offeriert alle NetServer-Systeme auch mit Linux.

Hewlett-Packard zählt wie IBM zu jenen Firmen, die Linux auf allen Ebenen von der Workstation über PC-Server und HP9000-Systeme bis hin zu Support und Schulung fest in ihr Angebot integriert haben. Auch in Sachen Software kann sich HPs Engagement sehen lassen: Das Unternehmen zählt beispielsweise zu den Gründungsmitgliedern von GNOME Foundation (www.gnome.org) und Open Source Development Lab (www.osdl.org). Insgesamt steht für HP schon lange fest, dass Linux in Zukunft Teile der IT-Welt dominieren wird. Als strategisches Betriebssystem soll es künftig parallel zum hauseigenen HP-UX auf allen IA64- und PA-RISC-Systemen (www.parisc-linux.org) laufen.

1.2.8 Engagement der Software-Industrie

Auch auf der Softwareseite des Geschäfts hat man inzwischen die Vorteile des freien Unix entdeckt. So portierte etwa Datenbankriese Oracle als erster der großen Anbieter seine Enterprise Edition auf Linux. Charles Rozwat, Executive Vice President für Servertechnologien, liefert dafür eine einleuchtende Begründung: „Die Portierung auf Linux bringt uns dahin, wo die Entwickler sind. Jedes Mal, wenn wir neue Produkte für Linux ankündigen, schießen die Registrierungen neuer Entwickler dramatisch in die Höhe“, so Rozwat.

Alles für den Pinguin: Auch der Datenbankspezialist Oracle setzt inzwischen voll auf Linux - zur Freude der Entwickler.

Die Entwicklerpakete für Linux würden bei Oracle weit häufiger geordert als jene für NT, Solaris oder jedes andere Betriebssystem. Entsprechend gestaltet sich auch der Absatz der Oracle-Pakete. „Es besteht eine ungeheure Nachfrage nach Geschäftsanwendungen und Clustering-Produkten für Linux“, freut sich Doug Kennedy, Oracles Vizepräsident für den Bereich Standard High Volume Systems.

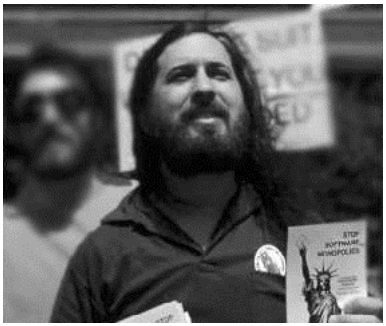
Ein weiteres Beispiel: Das deutsche Vorzeigeunternehmen SAP liefert bereits seit Ende 1999 R/3 respektive mySAP.com für Linux aus. Auch die Branchenlösungen der SAP.readytowork-Reihe setzen auf Linux-Servern auf. Allerdings erlauben die Walldorfer dabei lediglich den Einsatz zertifizierter Hard- und Softwarekombinationen. Als Distributionen kommen hier Red Hat Linux und der heimische SuSE 7.2 Enterprise Server (Kernel 2.4.7) zum Zug. Auch mit Opensource-Engagement in Form eines RDBMS (www.sapdb.org) kann SAP aufwarten.

Wie bereits im Abschnitt „Linux gegen Windows“ (Seite 15) beschrieben, teilen nicht alle Hersteller die positive Einstellung bezüglich Linux. Speziell Microsoft sieht Linux für eigene Produkte und Strategien mittlerweile als ernsthaften Gegner an und greift Linux sogar öffentlich an. Doch die Opensource-Gemeinde sieht dem nicht tatenlos zu.

1.2.9 Zukunft der Software

Auf Microsofts Holzhammer-Lobbyismus und ständige Agitation hat die Opensource-Gemeinschaft bereits passend geantwortet. In der an die amerikanische Unabhängigkeitserklärung angelehnten „Declaration of Software Freedom" kündigte das Freedevelopers.net im Jahr 2001 schon kurz nach den Angriffen von Jim Allchin, Leiter der Windows-Entwicklung bei Microsoft, an, sich zu einer „demokratischen Firma" der internationalen Opensource-Entwickler zusammenzuschließen.

Die Organisation unter Führung des FSF-Gründers (Free Software Foundation, www.fsf.org) Richard Stallman (www.stallman.org) plant offenbar, Anwender und Entwickler auf direktem Weg zusammenzuführen und so ein Finanzierungsmodell für freie Software zu Wege zu bringen. Mit den Einnahmen will man nicht nur die weitere Opensource-Entwicklung finanzieren, sondern auch „Softwarefirmen aufkaufen und deren Entwickler befreien".



Richard Stallman: „This is not one battle, it is a long war."

Das Modell könnte durchaus funktionieren, wie das Beispiel von MandrakeExpert.com (www.mandrakeexpert.com) zeigt. Dort bietet MandrakeSoft (www.mandrakesoft.com) den Benutzern und Entwicklern seiner Distribution einen offenen Marktplatz für Service, Support und Programmierung. Beide Seiten handeln eigenständig eine angemessene Entlohnung aus, einen kleinen Prozentsatz davon kassiert Mandrake und gibt ihn an die Free Software Foundation weiter.

Doch nicht nur durch den Zusammenschluss freier Entwickler, auch durch ganz konkrete Projekte planen die FSF, GNU und Freedevelopers.net Microsoft weiter die Zähne zu zeigen. Eines der interessantesten Vorhaben nennt sich DotGNU (www.dotgnu.org) und ist nichts anderes als ein Opensource-Ersatz für .NET. Das Hauptparadigma der DotGNU-Entwickler rund um Ximian-Mitbegründer Miguel de Icaza lautet: Sicherheit durch Dezentralisierung.

Keine einzelne Firma, Einrichtung oder Instanz darf die Kontrolle über Authentifizierung und Autorisierung im Netz erhalten. Damit steht DotGNU in direktem Widerspruch zu Microsofts Passport/Hailstorm-Offensive.

1.2.10 Ausblick

Dass Linux sich jetzt endgültig den Weg in den Mainstream-Markt gebahnt hat, zeigt nicht zuletzt auch das breite Produktangebot auf einer traditionell Business- und lösungsorientierten Messe wie der SYSTEMS.

Sein Erfolg im PC- und High-End-Servermarkt sowie die viel versprechenden Ansätze auf dem Desktop bieten Linux eine solide Basis für weiteres Wachstum. Die Akzeptanz durch Industrieriesen wie IBM und Oracle und vor allem die Initiative des Free-Software-Protagonisten Stallman könnten Linux und der Opensource-Software zum endgültigen Durchbruch verhelfen. Funktioniert FreeDevelopers.net wie geplant, ergänzt es die freie Software um jene stabile Support- und Dienstleistungsbasis, die ihr bislang fehlt.

Eines steht jedenfalls bereits jetzt fest: Die Release des Kernel 2.4 markierte für Linux das Ende der Kindheit. Von nun an ist das Opensource-Unix als ernst zu nehmendes Betriebssystem nicht mehr zu ignorieren.

tecCHANNEL Links zum Thema	Webcode
Happy Birthday, Linux!	a758
Interview mit Linus Torvalds	a551
Linux für den Server	a487
Linux 2.4 für den Desktop	a706
TCP/IP-Netze mit Linux	a562
Linux-Bootkonfigurationen	a546

1.3 Linux auf dem Mainframe

Die Portierung von Linux auf IBMs Enterprise-Mainframes beschert dem Big Iron eine neue Zukunft im Business-Intelligence- und E-Commerce-Einsatz.

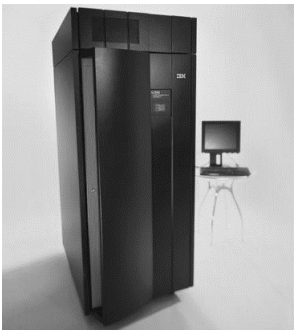
Ende der 90er Jahre galt der klassische Großrechner als Todeskandidat. Die Umsatzkurven der Millionen teuren Mainframes tendierten nach unten, schon eine Stagnation bei den Verkäufen werteten die Hersteller als Erfolg. Analysten titulierten die Maschinen gern als „Dinosaurier" - nicht nur, weil Serien wie IBMs S/390 seit den 60er Jahren auf dem Markt waren: Wie den Großechsen schien auch dem Großrechner die Ausrottung bevorzustehen.

Womit allerdings niemand rechnete, war die erfolgreiche Symbiose des ältesten Mitglieds der Business-IT-Familie mit dem jüngsten, anfangs noch misstrauisch beäugten Spross: Linux. Mittlerweile haben sich Dinosaurier und Pinguin jedoch zu einem Dream Team entwickelt, das den IBM-Mainframes der S/390- und zSeries-900-Reihe einen zweiten Frühling beschert. So konnte Big Blue im September 2001 den Verkauf des tausendsten zSeries-900-Systems vermelden. Bei einem Systempreis von mehreren Millionen Euro und zweistelligen Zuwachsraten verspricht das Mainframe-Geschäft nach einer langen Durststrecke wieder Milliardenumsätze.

„Noch vor einem Jahr hätten wir nicht gedacht, dass das so einschlägt", freut sich Joann Duguid, die bei IBM für Linux auf zSeries verantwortlich zeichnet. Über hundert der tausend zSeries-900-Kunden orderten ihren Großrechner bereits komplett mit Linux, viele andere haben es nach Einschätzung von IBM zusätzlich implementiert. Was Linux auf dem Mainframe kann und welche Aspekte es für den Unternehmenseinsatz so interessant machen, wollen wir im Folgenden unter die Lupe nehmen.

1.3.1 S/390 im Überblick

S/390 wird heute meist als Sammelbegriff für eine Familie von Großrechnern verwendet, deren Geschichte bereits Mitte der 60er Jahre begann. Als ersten Vertreter kündigte IBM im April 1964 die S/360 an, deren Name sich auf die Gradeinteilung eines Kompasses bezog und so symbolisch den weit gefächerten Anwendungsbereich symbolisieren sollte. Bereits nach eineinhalb Jahren folgte die S/370-Serie, die eine neue CPU-Architektur implementierte und bis 1972 mit Unterstützung für virtuellen Speicher und Multiprozessoreinsatz ausgebaut wurde. Sie gipfelte in der Enterprise System Architecture/370 (ESA/370) von 1988, die über zusätzliche Register den Zugriff auf den virtuellen Speicher ausbaute und beschleunigte.



Mutter aller Mainframes: Mit der S/390 dominiert IBM den Großrechnermarkt. Amdahl, Hitachi, Fujitsu und Siemens liefern sogar S/390-Clones. (Bild: IBM)

Im September 1990 folgte die S/390-Rechnerfamilie, mit der IBM das ESCON-Konzept zur Kommunikation von Mainframe und Peripherie einführte. ESCON steht für Enterprise System Connect und verbindet über Glasfaser den Mainframe mit anderen Großrechnern, Speichersubsystemen und lokalen Terminals. Als zweite Neuerung stellte IBM Sysplex vor, ein loses Clustering von S/390-Rechnern. 1994 folgte die Erweiterung zum Parallel Sysplex, bei dem eine Coupling Facility und ein Workload Manager die Verarbeitungskapazität und Lastverteilung verbesserte. Gleichzeitig führte IBM auch die CMOS-Technologie für die S/390-Serie ein.

Jüngster Spross der S/390-Familie ist seit Dezember 2000 die zSeries 900. Mit ihr steigt IBM von der 32-Bit-Architektur auf 64-Bit-Processing um. Anders als in der Intel-Welt bleibt IBM dabei jedoch voll abwärtskompatibel. Auf einer zSeries 900 können parallel sowohl 32- als auch 64-Bit-Programme laufen, einer direkten Verarbeitung selbst des ältesten S/360-Binärcodes steht nichts im Weg.

1.3.2 S/390-Betriebssysteme

Parallel zur Entwicklung der Hardware brachten auch IBMs Mainframe-Betriebssysteme diverse Evolutionsstufen hinter sich. Als Sammelbegriff für die diversen Varianten wird gern das Kürzel MVS verwendet, das für Multiple Virtual Storage steht.

Mit der S/360 von 1964 wurde als Betriebssystem OS/360 ausgeliefert. Ihm folgte mit der S/370-Generation das passende OS/370. Im Zuge des Ausbaus der Hardwarefähigkeiten, speziell der virtuellen Speicherarchitektur und der Enterprise Systems Architecture (ESA), entwickelte IBM es zu MVS/ESA weiter. Daraus entstand durch weitere Ergänzungen, speziell Unix-konforme Benutzer- und Systemschnittstellen, das heute verbreitete OS/390. Von ihm stammt die für die zSeries 900 überarbeitete 64-Bit-Variante z/OS ab.

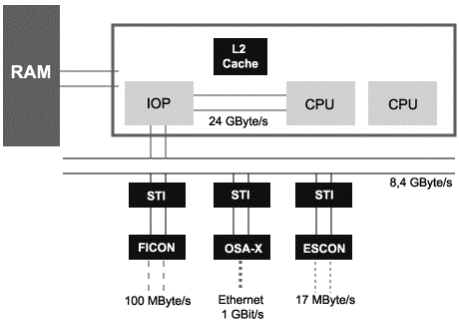
Mit VM/ESA schlug IBM eine für Mainframes neue Entwicklungsrichtung ein: Der Großrechner sollte von der Batch-orientierten Datenverarbeitungsanlage zur interaktiven Multiuser-Maschine werden. Dazu stellt VM (Virtual Machine) jedem Benutzer eine eigene „S/390“ samt CPU, Speicher und I/O-Ressourcen zur Verfügung. Für den interaktiven Teil zeichnet das Conversional Monitor System (CMS) verantwortlich, ein eigens dafür geschriebenes Single-User-OS.

Doch auf VM-Basis können durchaus auch andere Gastbetriebssysteme laufen. Die aktuelle VM-Weiterentwicklung z/VM (www.vm.ibm.com) unterstützt neben IBM-Varianten wie CMS, OS/390 oder z/OS auch den Einsatz von Linux als Guests. Dabei lassen sich die einzelnen Betriebssysteme je nach Bedarf mischen.

1.3.3 S/390-Architektur

Zu den Pluspunkten der S/390-Architektur zählen Skalierbarkeit, Robustheit, Zuverlässigkeit und insbesondere Performance. Dazu tragen eine ganze Reihe besonderer Merkmale des Mainframes bei.

In einer zSeries-Maschine arbeiten bis zu 20 Prozessoren. Lediglich 16 davon stehen für die unmittelbare Nutzung durch das Betriebssystem bereit. Zwei dienen als Reserve, zwei weitere kümmern sich ausschließlich um I/O-Operationen. Den Datentransport zwischen CPUs, I/O-Prozessoren, dem gemeinsamen Level-2-ECC-Cache und bis zu 64 GByte RAM übernimmt ein Bus mit einer maximalen Bandbreite von 24 GByte/s.

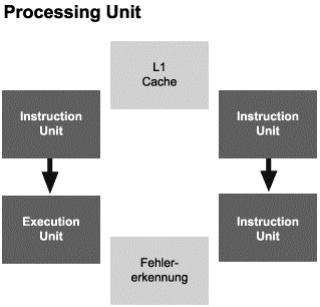


© tecChannel.de

Dicke Pipelines: Die Performance eines S/390-Systems beruht nicht zuletzt auf den breiten Datenkanälen und auf der entkoppelten I/O-Verarbeitung.

Alle Arbeiten im Zusammenhang mit Ein- und Ausgabe nehmen die dedizierten I/O-Prozessoren den CPUs ab. Über einen 8,4 GByte/s breiten Bus steuern sie bis zu 256 I/O-Kanäle an und erledigen dabei parallel bis zu 928 I/O-Operationen. Für den E-Business-Einsatz lassen sich die zSeries-Maschinen zudem um kryptographische Coprozessoren ergänzen, die maximal 2000 SSL-

Handshakes je Sekunde abwickeln. Als Schnittstellen nach außen stehen ESCON (bis zu 17 MByte/s), die Fibre-Channel-Variante FICON (100 MByte/s) sowie zur Netzwerkanbindung OSA-Extended (max. 1 GBit/s) zur Verfügung.



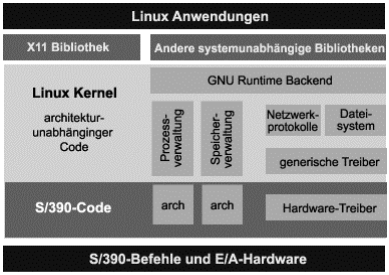
© tecChannel.de

Doppelt gemoppelt: Ein S/390-Prozessor prüft über doppelt ausgelegte Instruction- und Execution-Baugruppen seine Arbeit ständig nach.

Eine ganze Reihe von Merkmalen sorgt für Betriebssicherheit und Hochverfügbarkeit. So hat IBM etwa bei jedem Prozessor die Instruction- und Execution-Baugruppen doppelt ausgelegt. Mit Hilfe der doppelten Abarbeitung jeder Anweisung kann man Verarbeitungsfehler sofort erkennen, daneben finden noch Parity-Überprüfungen des Datenstroms und der Adressangaben statt. Neben Reserveprozessoren lässt sich auch Reservespeicher konfigurieren, die I/O-Kanäle können zu redundanten Gruppen gebündelt werden. Klassische High-Availability-Maßnahmen wie redundante Auslegung von Stromversorgung und Kühlung runden das Sicherheitsportfolio ab.

1.3.4 Linux auf dem Mainframe

Die systemimmanenten Vorteile des Mainframe hinsichtlich Stabilität, Zuverlässigkeit und Skalierbarkeit kommen Linux direkt zugute. Da das freie Betriebssystem von vornherein auf Portabilität ausgelegt wurde, erfordert die Implementation von Linux auf dem Mainframe nur einen relativ geringen Aufwand. Von den rund 2,2 Millionen Lines of Code des Kernel nehmen beispielsweise S/390-spezifische Instruktionen nur etwa 35.000 ein - das entspricht 1,5 Prozent. Für wichtige Komponenten wie gcc, glibc oder die binutils liegt das Verhältnis mit 0,5 bis 0,75 Prozent sogar noch niedriger. Von daher unterscheidet sich Linux auf dem Mainframe vom Look-and-Feel praktisch nicht von der Implementation für andere Plattformen.



© tecChannel.de

Fast identisch: Linux für den Mainframe unterscheidet sich lediglich auf der untersten Architekturebene von seinem PC-Pendant.

Auch auf der Anwendungsseite steht auf Grund der einfachen Portierung praktisch alles zur Verfügung, was des Anwenders Herz begehrt. „Als wir unseren Unix-basierten Code für Linux auf zSeries portieren wollten, stellten wir fest, dass es eigentlich gar nichts zu portieren gab“, kommentiert Fred Johannessen, der BMCs (www.bmc.com) Linux-Initiative leitet. „Wir haben das Ding einfach rekompiliert, und es lief. So einfach kann das nicht sein, dachten wir uns - war es aber doch.“

Die mittlerweile verfügbaren Applikationen umfassen nicht nur wichtige Opensource-Software wie Samba und Apache, Sendmail, MySQL oder KDE und GNOME. IBM selbst stellt neben DB/2 und Websphere beispielsweise Middleware wie MQSeries und CICS, Groupware wie Lotus und Management-Software wie Tivoli für Linux auf dem Mainframe zur Verfügung. Auch Drittanbieter aus allen Softwarebereichen sind mittlerweile auf den Zug aufgesprungen. Die Liste liest sich wie ein Who-is-who der Softwarebranche: Neben BMC (Patrol, Mainview) finden sich da auch Computer Associates (Unicenter, Inoculate, Arcserve), Oracle, SAP (R/3 Application Server), Software AG (Tamino) und andere.

1.3.5 Distributionen

Als Linux-Plattformen für den Mainframe stehen, ähnlich wie für andere Architekturen, verschiedene Distributionen zur Auswahl. Bislang haben sie alle eins gemeinsam: Sie bieten noch keine Unterstützung für die 64-Bit-Architektur der IBM zSeries 900, sondern operieren alle im S/390-Modus.

Red Hat bietet seine aktuelle OS-Version auch als Linux 7.2 for S/390 an. Sie basiert auf Kernel 2.4.9 und glibc 2.2.4, als journalbasiertes Dateisystem kommt Ext3 zum Zug. Als Kostenpunkt nennt Red Hat 20.000 US-Dollar pro CPU oder 50.000 US-Dollar für die unbeschränkte Premium-Version. Zum Paket gehört neben dem nackten Betriebssystem noch ein Bündel von Dienstleistungen, nicht jedoch die Installation: Sie schlägt noch einmal mit 10.000 US-Dollar zu Buche.

Recht viel bescheidener dürfte sich auch die SuSE Linux AG nicht geben. Das könnte man zumindest aus der Tatsache folgern, dass die Nürnberger Preise für

ihr Mainframe-Linux nur auf konkrete Kundenanfrage herausgeben mögen. SuSE Linux Enterprise Server 7 for S/390 and zSeries - so heißt das gute Stück - basiert auf Kernel 2.4.7 und der GNU C Library 2.2.2. In Sachen Dateisystem setzt SuSE auf ReiserFS. Zur Distribution zählen sieben CDs mit rund 1300 Applikationen, an Service bietet SuSE 30 Tage erweiterten Support sowie kostenlose Systempflege für zwölf Monate.

Der asiatische Platzhirsch Turbolinux (www.turbolinux.com) offeriert mit z/Linux Server 6 for zSeries and S/390 ebenfalls eine Distributionsvariante für den Mainframe. Im Gegensatz zur Konkurrenz gibt man sich dabei aber betont konservativ. So kommt hier noch Linux 2.2.19 zum Einsatz, Linux 2.4 / glibc 2.1.3 befindet sich bei Turbolinux noch in der Beta-Phase. Dafür steht als Dateisystem neben ReiserFS auch IBMs JFS zur Auswahl.

1.3.6 Betriebsarten

Prinzipiell bieten sich drei Modi an, um Linux auf einer S/390 zu betreiben. Die einfachste davon stellt der Native Mode dar. Hier läuft Linux direkt auf der Hardware und übernimmt unmittelbar die Kontrolle aller Komponenten. Die potenziellen Vorteile dieser Vorgehensweise halten sich jedoch in Grenzen, weswegen sie selten zum Einsatz kommt.

Viel interessanter erscheint die Methode, die vorhandene Hardware zu partitionieren. Im LPAR-Modus (logical partitions) unterstützt die S/390 den Betrieb von bis zu 15 Betriebssystem-Images - quasi jeweils einzelnen Servern. Dabei kann es sich wahlweise um 15 Linux-Instanzen handeln oder auch um einen Mix aus Linux und anderen Betriebssystemen. Die vorhandenen Hardwareressourcen (wie CPUs, Speicher und I/O-Kanäle) werden je nach Bedarf den einzelnen LPARs statisch oder dynamisch zugeteilt.

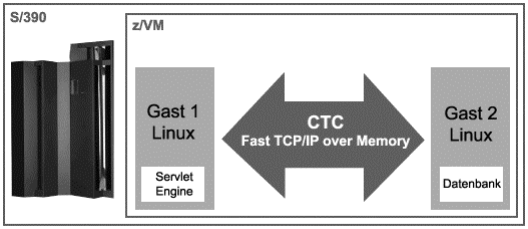
Die häufigste Betriebsart ist jedoch z/VM. Durch die Virtualisierung sämtlicher Ressourcen verschwindet die Beschränkung des LPAR-Modus auf 15 Images. Durch die dynamische Zuteilung von CPU-Zeit, Speicher und I/O-Kapazität an die Betriebssystem-Images lassen sich nahezu beliebig viele „Server“ auf der S/390 nachbilden. So zitiert IBM gern den Fall eines Kunden, der zu Testzwecken mehr als 41.000 Linux-Instanzen parallel auf einem System erzeugte. Auch im Praxiseinsatz lassen sich ganze Serverfarmen durch eine S/390-z/VM-Installation ersetzen.

1.3.7 Linux im z/VM-Modus

Neben der Virtualisierung und damit völlig dynamischen Zuteilung der Hardwareressourcen hat der z/VM-Modus weitere Vorteile. So lassen sich verschiedenste Gastbetriebssysteme je nach Bedarf parallel betreiben. Die S/390-Architektur bietet dabei genügend Performance, um im Produktivbetrieb eine hohe Anzahl von Guests zu unterstützen.

So berichtet IBM etwa von einer Installation, die 400 Sun-Server durch 400 Linux-Images auf einer S/390 ersetzte.

Die „Installation“ eines neuen „Servers“ gestaltet sich recht einfach: Der Benutzer meldet sich auf der Konsole eines virtuellen S/390-Systems an und konfiguriert über ein so genanntes Control Program die gewünschte Hardware für das virtuelle System. Anschließend bootet er ein auf Band oder Platte gelagertes Image des Betriebssystems. Im 390-Jargon heißt dieser Vorgang IPL (Initial Program Load). Ein Gastsystem lässt sich bei Bedarf auch im Betrieb klonen. Innerhalb von 45 Sekunden steht dann bei erhöhten Leistungsanforderungen ein zusätzlicher „Linux-Server“ bereit. Andererseits isoliert z/VM die Guests so voneinander, dass Abstürze einzelner Images die Arbeit der anderen virtuellen Maschinen nicht in Mitleidenschaft ziehen.



© tecChannel.de

LAN im RAM: Über Hipersockets stellt z/VM den Guests ein virtuelles TCP/IP-Netz im Speicher zur Verfügung.

Nun bliebe der Nutzwert der zahlreichen Images recht beschränkt, könnten die virtuellen Server nicht untereinander kommunizieren. Dafür bietet z/VM jedoch ebenfalls eine passende Lösung: Über bis zu 1024 Hipersockets stellt es virtuelle TCP/IP-Verbindungen zwischen den einzelnen Guests her. Diese Channel-to-Channel-Verbindungen (CTC) erfolgen direkt über den Speicher und somit extrem schnell.

1.3.8 Vorteile und Einsatzgebiete

Welche Vorteile der Betrieb multipler Linux-Images unter z/VM im praktischen Betrieb bietet, hat IBM in seinem Werbespot „The Heist“ publikumswirksam inszeniert:

Ein entsetzter IT-Manager präsentiert zwei Kriminalbeamten sein vermeintlich von Dieben leer geräumtes Rechenzentrum. Ein mit Donut und Kaffeetasse hinzukommender Techie klärt das Trio über die wahren Hintergründe auf. Er weist lässig auf eine einsam in der Ecke des Rechenzentrums stehende IBM zSeries, die jetzt unter z/VM Dutzende Maschinen ersetzt.

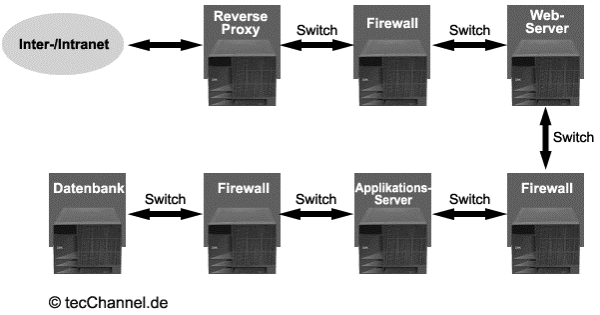
Im IT-Deutsch heißt dieser Effekt Serverkonsolidierung. Das Ersetzen vieler einzelner Server durch eine Maschine hat Vorteile. So spart man dadurch nicht nur - wie in „The Heist“ zu sehen - Standplatz. Gleichzeitig kann man die Be-

triebskosten reduzieren. So verbraucht ein einzelner Mainframe weniger Energie als Dutzende kleinerer Systeme. Dies erleichtert und verbilligt die Klimatisierung wesentlich.

Gleichzeitig vereinfachen sich die Anforderungen an die Netzwerkinfrastruktur. Durch den Wegfall zahlreicher Einzelmaschinen lassen sich die zu deren Verbindung notwendigen Switches und Router sowie Kilometer an Netzwerkkabel einsparen. Die geringe Komplexität der konsolidierten Umgebung erleichtert die Administration und Fehlersuche. Und last not least ermöglicht die dynamische Ressourcenzuweisung die optimale Skalierung und Auslastung der Hardware.

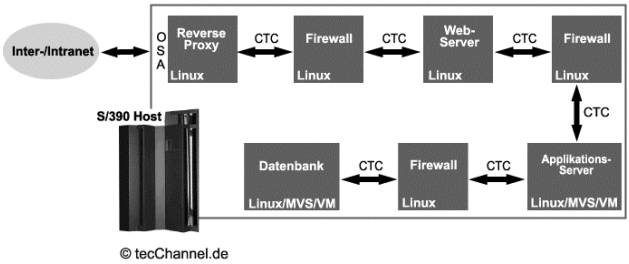
1.3.9 E-Business

Auf Grund der immanenten Vorteile kommt Linux auf dem Mainframe vor allen Dingen dort zum Einsatz, wo es mehrstufige Serverinstallationen ersetzen kann. Ein typisches Beispiel dafür liefern etwa klassische E-Business-Architekturen.



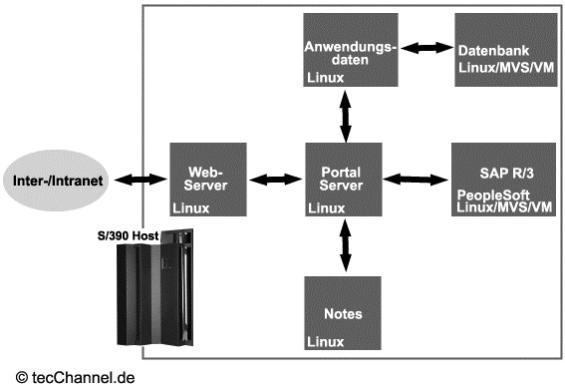
Vorher: Eine typische E-Business-Infrastruktur mit vielen Servern sowie Netzkomponenten.

Eine Installation im geschäftlichen Internet-Umfeld umfasst in der Regel mehr als nur einen Webserver. Ein zusätzlicher Applikationsserver bedient den Kunden interaktiv, die notwendigen Fakten dazu liefert ein Datenbankserver. Ein dem Gesamtsystem vorgeschalteter Proxy sorgt für die nötige Performanz. Aus Sicherheitsgründen isolieren Firewalls die einzelnen Zonen, aktive Netzkomponenten stellen die Kommunikation zwischen den Systemen sicher.



Nachher: Die Serverkonsolidierung auf eine S/390 reduziert sowohl Komplexität als auch Betriebskosten drastisch.

Bei der Portierung einer solchen Architektur auf den Mainframe ergibt sich neben einer deutlichen Vereinfachung auch ein Leistungszuwachs. So fallen nicht nur sieben Einzelmaschinen weg, die in der Praxis meist überdimensioniert werden, um Spitzenlasten standhalten zu können. Gleichzeitig realisiert die dynamische Ressourcenzuweisung auf dem Mainframe eine optimale Performance für die jeweils leistungshungrigste Komponente. Außerdem ersetzen schnelle Channel-to-Channel-Verbindungen die vorher notwendigen aktiven Netzwerkkomponenten.



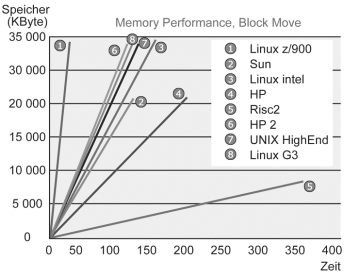
Portal kompakt: Gerade bei leistungshungrigen Applikationen und vielen Benutzern spielt Linux auf dem Mainframe seine Vorteile aus.

Auch Portallösungen lassen sich mit Linux auf dem Mainframe kompakt abbilden. Je leistungshungriger dabei die einzelnen Komponenten ausfallen, desto besser kommen die Performance-Vorteile des Großrechners zum Tragen. So können auch sehr hohe Benutzerzahlen problemlos mit leistungsfähigen Inter/Intranet-Anwendungen bedient werden.

1.3.10 Performance

Wie sich die viel zitierte Mainframe-Performance für Linux in der Praxis auswirkt, demonstrieren einige Daten, die Dr. Kolja Elsäßer von debis Systemhaus GmbH Leinfelden-Echterdingen auf der „Linuxworld Expo 2001“ vorstellte. Für das zu T-Systems gehörende Unternehmen führte er im dortigen Competence Center für Middleware und Application Integration eine Reihe von Leistungsmessungen durch. Als Linux-Variante kam SuSE 2.2.16 auf einer zSeries 900 mit zwei Prozessoren unter z/VM zum Einsatz. Die Rechenleistung lag bei rund 500 MIPS.

Performance Memory



© tecChannel.de

Deutlich schneller: Bei Speicherzugriffen ist der Mainframe nicht zu schlagen, hier kommt die performante Busarchitektur voll zum Tragen.

Wie die oben stehende Abbildung demonstriert, erreichte in dieser Konfiguration Linux auf der z/900 die mit Abstand höchsten Leistungsdaten beim Speicherzugriff. Weder High-End-Unix-Architekturen noch Linux auf anderen Plattformen können dem Mainframe hier das Wasser reichen.

Performance-I/0

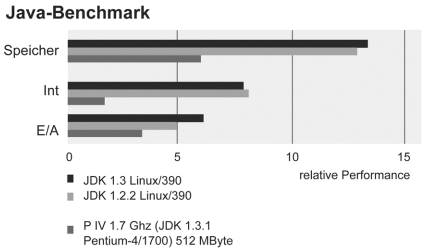
Rechentyp	Schreiben				Lesen			
	Char		Block		Char		Block	
	KByte/s	%CPU	KByte/s	%CPU	KByte/s	%CPU	KByte/s	%CPU
Intel PIII 600Mhz	9493	99,8	16382	15,7	9444	89,3	15884	12,3
HP PA 8600	30247	84,3	28889	21,1	27262	78,6	34497	22,7
UNIX High End	14732	26	17297	7,7	14574	30,1	17816	12
IBM z/900 native	5476	98,2	11230	4,3	4809	88,9	11164	3,5
IBM z/900 2 stripes	4061	99,5	23759	10,8	3440	95,8	19726	7,3
IBM z/900 4 stripes	4560	99,8	45423	19,3	4839	95,2	27971	10,5
IBM z/900 7 stripes	5591	100	67051	23,2	5212	96,9	32465	13

© tecChannel.de

Je größer, desto besser: Eingabeoperationen verarbeitet der Mainframe am liebsten in Blöcken. Bei kleinen I/O-Aufgaben hängen ihn PCs und Midrange-Maschinen ab.

Ein weitaus differenzierteres Bild ergab sich bei der Untersuchung der I/O-Performance. Wie die obige Tabelle zeigt, kann beim zeichenbasierten Zugriff

selbst ein Pentium-4-basierter Server durchaus mit dem Mainframe konkurrieren. High-End-Unix-Systeme hängen die zSeries in diesem Modus sogar ab. Ein ganz anderes Resultat ergibt sich jedoch beim blockbasierten Umgang mit großen Datenmengen. Hier kann der Mainframe seine architekturbedingten Vorteile voll ausspielen und deklassiert die anderen Architekturen.



Applikationsbeschleuniger: Bei der Abarbeitung von Anwendungen schlägt der Linux-Guest die PC-basierte Konkurrenz deutlich.

Auch auf der Anwendungsebene ergaben sich bei den Tests deutliche Leistungsvorteile für den Linux-Guest gegenüber einer typischen PC-Umgebung. So ließ Dr. Elsäßer einen Java2-JDK-Benchmark auf dem 2-CPU-Mainframe sowie einem Pentium-4-Server unter Windows 2000 antreten. Wie das obige Balkendiagramm erkennen lässt, lieferte Linux auf dem Mainframe bei Speicher- und I/O-Operationen in etwa doppelte Performanz. Bei der Verarbeitung von Integerwerten ließ Linux auf der zSeries die PC-Variante deutlich hinter sich.

1.3.11 Fazit

Linux auf dem Mainframe bietet eine interessante Alternative für jedes Unternehmen, das zur Bewältigung seiner Geschäftsprozesse komplexe IT-Architekturen betreiben muss. Zwar fallen hohe Anfangsinvestitionen im Millionen-Euro-Bereich an, die sich jedoch durch ein immenses Einsparungspotenzial bei Infrastruktur und Betriebskosten relativieren. Als weiteres Plus kommen eine Ausfallsicherheit und Skalierbarkeit hinzu, die sich bei PC- und selbst High-End-Unix-Lösungen nur schwer oder gar nicht realisieren lassen.

Unternehmen, die bereits Großrechner im Einsatz haben, erschließt Linux auf dem Mainframe eine Vielzahl neuer Anwendungen. Die leichtere Interaktion mit ebenfalls hostbasierten Legacy-Applikationen verspricht einen besseren Zugang zu den vorhandenen Unternehmensinformationen. Für die Entwicklung von eigenen Anwendungen kann zudem auf eine deutlich breitere Entwicklerbasis zurückgegriffen werden, was Dauer und Kosten für das Ausrollen neuer Applikationen senkt.

Auch für ISPs und ASPs könnte sich der Linux-Einsatz auf dem Mainframe rechnen. Gegenüber den heute üblichen Rack-Reihen voller „Pizzaboxen“ ver-

spricht eine Mainframe-Lösung stark reduzierten Platzbedarf, niedrigere Betriebskosten und einfacheres Management. Außerdem lassen sich via z/VM in sehr kurzer Zeit neue Kunden-„Server“ aufsetzen.

Mit diesen Vorteilen hat sich Linux nicht nur einen festen Platz auf dem Mainframe erobert, sondern dem Dinosaurier wieder neues Leben eingehaucht. Der Brontosaurus ist zum Velociraptor-Rudel mutiert.

tecCHANNEL-Links zum Thema	Webcode
Happy Birthday, Linux!	a758
Linux für den Server	a487
Interview mit Linus Torvalds	a551
Tux goes Biz	a654
Supercomputing	a696

2 Linux-Optimierungen

Egal ob Desktop oder Server, ein Linux-System bietet viel Spielraum für Optimierungen und systemspezifische Anpassungen. Im zweiten Kapitel zeigen wir, wie Sie Bootkonfigurationen einrichten, den Kernel tunen und Linux optimal in TCP/IP- und DSL-Netze einbinden. Auch die wichtigsten Punkte zum Absichern einer Linux-Workstation kommen in diesem Abschnitt zur Sprache.

2.1 Linux-Bootkonfigurationen

Die Installation eines einsteigergerechten Linux-Systems mit grafischer Oberfläche lässt sich mit Hilfe aktueller Distributionen problemlos in 30 Minuten erledigen. Ein endloser Weg durch die Online-Dokumentation steht jedoch dem bevor, der die Startup-Konfiguration des Systems an seine eigenen Bedürfnisse anpassen möchte - ob es sich nun um das automatische Einbinden von Netzlaufwerken oder das schlichte Customizing des grafischen Logins handelt.

Tatsächlich nutzt Linux jedoch klar strukturierte und gut nachvollziehbare Startup-Mechanismen, die sich mit einigen Grundkenntnissen problemlos auch für eigene Zwecke nutzen lassen. Wir zeigen im Folgenden, wie der Start eines Linux-Systems vom Einschalten des Rechners bis zum Login abläuft, und an welchen Stellen der gezielte Einsatz eigener Modifikationen Sinn macht. Bei der Arbeit mit den Startup-Dateien empfiehlt es sich, Vorsicht walten zu lassen: Je nach Distribution kann sowohl der Inhalt der Files als auch der Pfad zu ihnen leicht variieren. Die Angaben in unserem Workshop beziehen sich auf die in Deutschland verbreitetste Distribution, SuSE Linux. Falls Sie mit einer anderen Distribution arbeiten, empfiehlt sich ein kurzer Blick ins Handbuch.

2.1.1 Notbremse: Bootdisketten erstellen

Wer mit dem Betriebssystem experimentiert, benötigt auf jeden Fall eine Boot-Diskette als Rettungsanker. Manche Linux-Distributionen - etwa Red Hat oder SuSE - bringen schon im Lieferumfang eine entsprechende Floppy mit. Doch nicht nur als Rescue-Disk macht der Einsatz einer Bootdiskette Sinn. Auch beim Austesten maßgeschneiderter Kernel erweist es sich als hilfreich, den selbst erstellten Kernel erst einmal auf Diskette auszulagern: Treten beim Hochfahren Fehler auf, starten Sie Linux wieder mit dem alten Kernel von der Festplatte.

Um die Bootdiskette anzulegen, wechseln Sie nach der Kompilierung des Kernels mit *make bzImage* in das Verzeichnis */usr/src/linux*. Dort befinden sich die Quellen des Linux-Kernel. Legen Sie jetzt eine leere Diskette - sie muss nicht formatiert sein - ins Laufwerk und geben Sie dann den Befehl *make zdisk* ein. Linux kopiert dann das Kernel-Image auf die Diskette.

Planen Sie intensive Tests, dann sollten Sie sich neben einer Bootdiskette auch noch eine Utility-Disk anlegen. Auf diese kopieren Sie alle Werkzeuge, die Sie zur Reparatur des Systems benötigen. Dazu müssen Sie eine Initial Ramdisk anlegen, in der der Kernel ein Mini-Root-Dateisystem anlegen kann. Mehr zu diesem Thema finden Sie im Linux-Bootdisk-HOWTO (www.linuxdoc.org/HOWTO/Bootdisk-HOWTO/) beziehungsweise in Ihrem Linux-Handbuch.

2.1.2 Systemstart unter Linux

Auch unter Linux beginnt der Bootvorgang mit dem POST des Rechner-BIOS. Anschließend liest das System den Bootsektor einer Floppy respektive den MBR der Harddisk aus. Der dort enthaltene Code identifiziert den Lagerungsort des Kernel und befördert diesen in den Arbeitsspeicher.

Während es beim Laden des Kernel von Diskette genügt, eine Reihe sequenzieller Sektoren einzulesen, gestaltet sich die Sache beim Booten von Harddisk etwas komplizierter. Hier dient der Linux Loader LILO dazu, die richtige Partition sowie - über das installierte Filesystem - jene Sektoren anzusteuern, die den Kernel-Code enthalten.

Da der Linux-Kernel komprimiert auf der Platte lagert, wird er zunächst entpackt. Danach prüft der Betriebssystemkern, ob bestimmte Hardwarekomponenten vorhanden sind, und passt die Treiberkonfiguration entsprechend an. Dabei gibt er viele Verlaufsmeldungen aus. Das Einbinden („mounten“) des Root-Filesystems schließt diese Bootphase ab.

Nun übernimmt das Programm *init* die Steuerung des weiteren Ablaufs. Es sorgt für den Start diverser Systemdienste – diese werden als Daemons bezeichnet. Zu guter Letzt startet *init* das Programm *getty*, das eine Anmeldung der Benutzer über virtuelle Konsolen oder serielle Terminals erlaubt. Damit endet der Bootvorgang, das Linux-System steht zur Verwendung durch die User bereit.

An zwei Stellen dieses Ablaufs bietet sich dem Anwender Gelegenheit, nach seinen Vorstellungen in den Bootvorgang einzugreifen: Beim Laden des Kernels über LILO sowie bei der Diensteverwaltung durch *init*.

2.1.3 LILO-Basics, Teil 1

Für gefahrlose Experimente mit Linux erweist sich - neben einer funktionierenden Rettungsdiskette - vor allem das Grundverständis des Linux-Bootmanagers LILO als wichtige Voraussetzung. Da die meisten Funktionen eines Linux-Systems von der Konfiguration des Kernels abhängen, ergibt sich beim Tuning regelmäßig die Notwendigkeit, diverse Kernel-Varianten wahlweise zu laden. Am bequemsten erfolgt dies via LILO.

Legen Sie für einen neuen Betriebssystemkern zuerst ein eigenes Verzeichnis unter */boot* an - zum Beispiel */boot/kernel-version/*. Um den Kernel auch dort hin zu bugsieren (einfaches Kopieren genügt nicht), müssen Sie den Pfad in

Makefile im Verzeichnis */usr/src/linux/* angeben. Suchen Sie dort den Eintrag INSTALL_PATH und modifizieren Sie ihn entsprechend:

```
INSTALL_PATH=/boot/<kernel-version>/
```

Anschließend konfigurieren und übersetzen Sie den neuen Kernel und tragen ihn in die Datei */etc/lilo.conf* ein. Diese informiert LILO, wie der Kernel heißt (meist *vmlinuz*), wo er sich auf der Platte befindet und welchen Namen er im Bootmenü erhalten soll. Haben Sie Linux von einer Distribution installiert, befindet sich bereits mindestens ein Kernel-Eintrag in *lilo.conf*:

```
image = /boot/vmlinuz
root = /dev/hda1
label = linux
```

Kopieren Sie einfach diesen Abschnitt und passen Sie ihn anschließend für den neuen Kernel an:

```
image = /boot/2.3.16-000922/vmlinuz
label = linux-2.3.16-000922
```

Den Root-Eintrag übernehmen Sie ohne Änderungen. Er informiert LILO, welche Partition das Root-Dateisystem enthält.

2.1.4 LILO-Basics, Teil 2

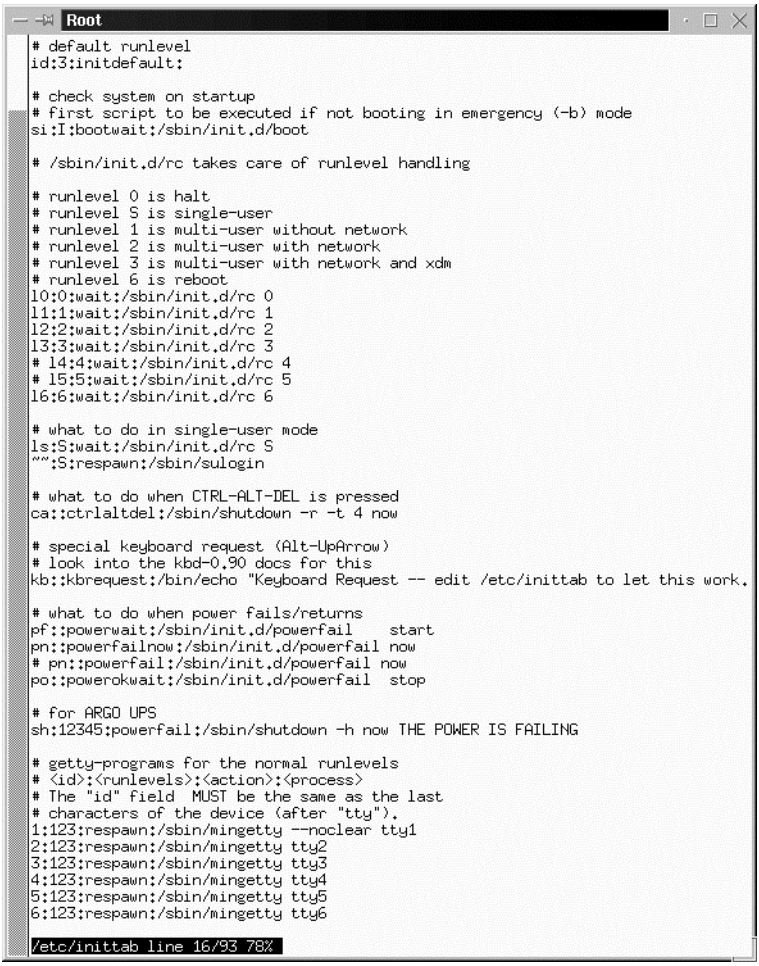
Nachdem Sie die modifizierte *lilo.conf* gespeichert haben, geben Sie nur noch den Befehl *make bzlilo* ein - alles Weitere geschieht automatisch: Der neue Kernel wird von seinem Speicherort */usr/src/linux/arch/i386/boot/* nach */boot/kernel-version/* transferiert. Das gleichzeitig aktualisierte LILO-Bootmenü reagiert beim nächsten Systemstart auf Ihren neuen Eintrag (*label=...*).

Falls Sie sich unsicher sind, welchen Namen Sie Ihrer Kreation gegeben haben, drücken Sie die Tabulatortaste, sobald der LILO-Prompt erscheint. Der Bootmanager zeigt dann alle Konfigurationen, die sein Menü enthält.

LILO setzt der Experimentierfreude kaum Grenzen: *lilo.conf* kann unbegrenzt Einträge aufnehmen. Achten Sie jedoch stets darauf, die Struktur der Datei eindeutig zu halten. Nicht benutzte Abschnitte sollten Sie löschen oder zumindest mit dem #-Zeichen auskommentieren. Daneben verfügt LILO über einige weitere Fähigkeiten. So lässt sich über die Option *password* die Eingabe eines Bootkennworts erzwingen. Das Schlüsselwort *message* erlaubt die Angabe einer Datei, die den User beim Booten mit einem informativen Text begrüßt. Weitere Informationen zu den LILO-Optionen finden Sie in der deutschen Man-Page von *lilo.conf* sowie im LILO User Guide unter */usr/doc/lilo/*.

2.1.5 Schaltzentrale init

Nachdem der Kernel das Root-Dateisystem gemountet hat, startet er das Programm *init*. Diese Schaltzentrale startet und stoppt verschiedene Dienste. So fährt *init* etwa den Druckerspooler hoch, startet mit *fsck* eine Überprüfung der Dateisysteme und bindet via *mount* Dateisysteme anderer Partitionen lokal ein.



Kommandozentrale: Die Datei /etc/inittab legt unter anderem den Default-Runlevel fest, in den Linux nach dem Einschalten des Rechners booten soll.

Welche Tätigkeiten es genau vornehmen soll, entnimmt *init* der Datei */etc/inittab*. Dabei folgt es dem System-V-Bootschema, das seinen Namen der ersten kommerziellen Unix-Variante (AT&T Unix System V) verdankt. Das Schema sieht acht so genannte Runlevels vor, von denen jeder einer speziellen Systemkonfiguration entspricht. Die Runlevels S, 0, 1 und 6 sind fest definiert und reserviert, die Runlevels 2 bis 5 lassen sich frei konfigurieren.

2.1.6 Konfiguration über Runlevels

Die System-V-Runlevels lassen sich in gewissem Sinn mit den Optionen vergleichen, die das Windows-9x-Bootmenü offeriert. Jedes steht für eine bestimmte Systemkonfiguration.

System-V-Runlevels	
Runlevel	**Funktion**
0	Systemhalt
S	Single-User
1	Multi-User ohne Netzwerk
2	Multi-User mit Netzwerk
3	Multi-User mit Netzwerk und grafischem Login
4	Frei
5	Frei
6	Reboot

Runlevel: Das System-V-Bootschema sieht insgesamt acht verschiedene Runlevels vor. Hier die Belegung unter SuSE Linux.

Die einzelnen Runlevels lassen sich manuell von der Kommandozeile aus starten. Um das System etwa neu zu booten, wechseln Sie mit *init 6* in den entsprechenden Level.

Diese Fähigkeit zum manuellen Wechsel des Runlevels erlaubt weit gehende Modifikationen der Systemkonfiguration, ohne den Rechner dazu neu booten zu müssen. Wollen Sie etwa die Netzwerkeinstellungen modifizieren, bringen Sie Linux mit *init 1* zuerst in den Modus *Multi-User ohne Netzwerk* (Runlevel 1). Nun nehmen Sie die notwendigen Änderungen vor und wechseln anschließend mit *init 2* wieder in den Netzwerkmodus. Alle Netzwerkprogramme starten jetzt neu und lesen dabei die modifizierten Konfigurationsdateien ein. Das Beste daran: Alle nicht auf das Netzwerk angewiesenen Applikationen - etwa Editoren oder Dateimanager - laufen währenddessen unbeeinflusst weiter.

2.1.7 Login und Shut-down

Wie die Runlevel-Tabelle auf der vorigen Seite zeigt, erfolgt das grafische Login über *kdm* (KDE Display Manager) erst in Runlevel 3. Bootet Ihr Rechner bei installiertem *kdm* nur in den Textmodus (Runlevel 2), kommt Linux offensichtlich nicht weiter. Das Problem lässt sich über die Konfigurationsdatei *inittab* im Verzeichnis /etc beheben. Suchen Sie dazu die folgende Zeile:

```
id:2:initdefault:
```

Ändern Sie hier die Zahl 2 (für Runlevel 2) in eine 3 (für Runlevel 3) um. Beim nächsten Start bootet das System bis in den Grafikmodus und fährt direkt den installierten Display Manager hoch. Dasselbe Verhalten lässt sich manuell - etwa gleich nach der Installation - durch Eingabe des Befehls *init 3* erzielen. Auch in der umgekehrten Richtung kann eine entsprechende Anpassung durchaus Sinn machen: So geben einige Konfigurationen per Default ein grafisches Logon vor, was jedoch bei Serverinstallationen wenig Sinn macht.

Über *inittab* lässt sich auch das Verhalten des Systems bei Betätigung der Tastenkombination [Strg]+[Alt]+[Entf] vorgeben. Die meisten Distributionen sehen dafür den Reboot des Rechners vor - also den Wechsel in Runlevel 6. Wollen Sie stattdessen den „Affengriff“ zum schnellen Herunterfahren des Systems nutzen, erfordert dies einen Wechsel in Runlevel 0. Suchen Sie dazu in *inittab* die Zeile:

```
ca::ctrlaltdel:/sbin/shutdown -r -t 4 now
```

und tauschen Sie die Option *r* für Reboot gegen ein *h* für Halt aus. Wenn Sie nun die Tastenkombination im Textmodus benutzen, fährt der Rechner herunter. Benutzen Sie X-Windows, müssen Sie vorher die grafische Oberfläche beenden.

2.1.8 inittab und boot

Besonders behutsam sollten Sie bei Veränderungen an *inittab* mit den Einstellungen in den ersten Zeilen des Scripts umgehen. So gibt die Zeile:

```
si:I:bootwait:/sbin/init.d/boot
```

etwa den Pfad zum Script boot an. Darin finden sich die Programme, die *init* unmittelbar nach dem Laden des Kernels ausführen muss. Dazu zählen unter anderem *swapon* (aktiviert die Swap-Partition), *fsck* (Prüfung des Dateisystems), *mount* (Einbinden von Laufwerken in Dateisystem) oder *hostname* (setzt den Rechnernamen).

Startverhalten: Init führt bei Systemstart als Erstes das Script boot aus. Dort startet es systemnahe Programme wie fsck und mount.

Bei Änderungen an den hier aufgeführten Programmen und Optionen sollten Sie äußerste Vorsicht walten lassen. Allzu leicht verweigert das System andernfalls den Dienst.

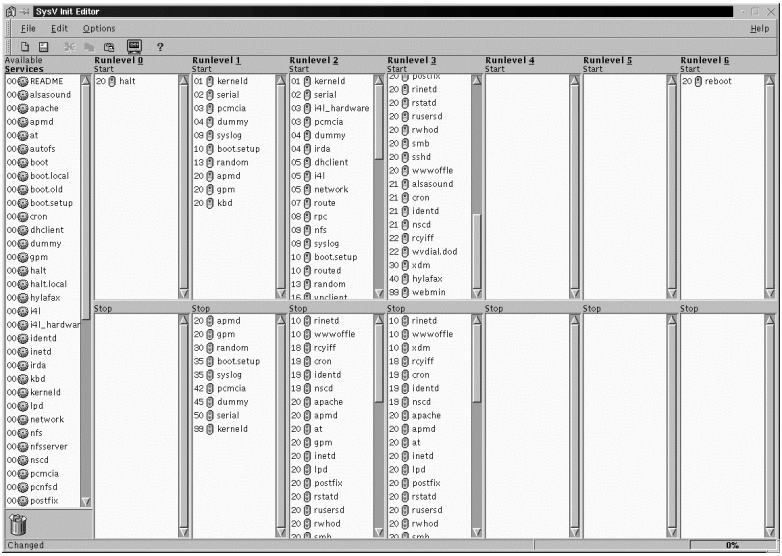
Falls Sie sich trotzdem einmal aus dem System aussperren, starten Sie den Kernel mit folgender Zeile:

```
kernelname init=/bin/sh
```

So booten Sie direkt in eine Shell und können Ihre Änderungen rückgängig machen. Aus Sicherheitsgründen sollten Sie die Nutzung dieser Funktion allerdings über LILO einschränken. Vergeben Sie dazu in *lilo.conf* ein Passwort (*password=geheim*) und ergänzen Sie zusätzlich eine Zeile mit dem Parameter restricted. Dies erzwingt bei der Übergabe von Bootparametern wie *init=/bin/sh* die Eingabe eines Passworts.

2.1.9 Runlevel-Änderung über ksysv

Wenn *init* den Runlevel wechselt, startet es das Script */sbin/init.d/rc*, das weitere Scripts anstößt. Jedes davon startet oder stoppt einen Systemdienst. Zu dieser Regel existieren nur zwei Ausnahmen: In Runlevel 6 kommt nur das Script *reboot* zum Einsatz, in Runlevel 0 lediglich *halt*. Auch wenn Ihr System nicht als Server operiert, benötigen Sie einige der Daemons, etwa den Tastaturdienst *kbd*, den Line Printer Daemon *lpd* und den Protokolldienst *syslogd*.



Komfortables Werkzeug: Mit dem SysV Init Editor aus dem KDE konfigurieren Sie die Runlevels per Drag-and-Drop.

Welche Scripts momentan auf Ihrem System aktiviert sind, zeigt Ihnen beispielsweise das Programm *ksysv* aus dem KDE (*K-Menü/System/SysV Init Editor*). Mit dem SysV Init Editor stellen Sie ein, welche Dienste *init* in welchem Runlevel starten soll. Ziehen Sie dazu das Script aus dem linken Fenster (*Available Services*) mit der Maus in den gewünschten Runlevel. Ein kleiner Tipp zur Orientierung: In Runlevel 1 finden sich alle Scripts, die zum Minimalbetrieb des Systems erforderlich sind.

Die Zahl vor jedem Script zeigt dessen Start/Stopp-Position an. Je höher die Zahl, desto später startet respektive stoppt *init* das Script auf dem Weg zum gewünschten Modus. Manche Scripts setzen den vorherigen Start eines anderen voraus: So benötigen Server-Daemons meist das Script *network*, um überhaupt via LAN kommunizieren zu können.

2.1.10 Mit Runlevels experimentieren

Für eigene Versuche mit selbst gestrickten Konfigurationen benutzen Sie am besten einen freien Runlevel, zum Beispiel die Nummer 4. Als Basisgerüst kopieren Sie alle Scripts von Runlevel 1. Außerdem sollte stets *kbd* auftauchen, das den Keyboard-Dienst startet.

Für Stand-alone-Workstations benötigen Sie auf jeden Fall zwei weitere Scripts. *sendmail* kümmert sich um den gleichnamigen Mailserver. Er sorgt dafür, dass die System-Daemons Sie bei Fehlern über das interne Netzwerk (loopback) per Mail alarmieren können. Der Jobmanager *cron* erledigt die regelmäßige Ausführung von Routine-Aufgaben, etwa das Löschen der Daten in temporären Verzeichnissen.

Auf Netzwerk-Clients oder Servern darf zudem das Script *network* nicht fehlen, das die LAN-Anbindung übernimmt. Seine Betriebsparameter bezieht es aus den Angaben in der Datei */etc/rc.config*. Als File- und Print-Server für Windows-Maschinen schließlich dient Samba, das zugehörige Script heißt *smb*.

Hinsichtlich der Start- und Stopp-Nummern für die Daemons orientieren Sie sich am besten an den von SuSE konfigurierten Runlevels. Anschließend speichern Sie die Einstellungen und entfernen in *inittab* noch das Kommentarzeichen aus der Zeile, die für Runlevel 4 verantwortlich ist:

```
l4:4:wait:/sbin/init.d/rc 4
```

2.1.11 Weitere Dienste einbinden

Nicht nur die Netzwerk-Daemons, auch beliebige andere Dienste - als prominentester wohl *httpd* alias Apache - lassen sich über die Konfiguration eines entsprechenden Runlevels automatisch in das Linux-Startup einbinden. Nicht im-

mer jedoch kann man dabei Daemon- und Script-Namen auf den ersten Blick eindeutig zuordnen.

Wie finden Sie aber heraus, welches Script zu welcher Anwendung gehört? Dabei hilft die Online-Dokumentation *man*. Deren Manual Pages führen auf, welche Aufgaben ein Dienst übernimmt und wie man ihn konfiguriert. Um eine Man-Page anzuzeigen, geben Sie den Befehl *man*, gefolgt von einem Dienstnamen ein (zum Beispiel: *man sshd*).

Oft stimmen Script- und Dienstnamen allerdings nicht exakt überein. Meist fehlt aber nur der Buchstabe d (für „Daemon“) am Ende. So ruft etwa das Script *smb* den Daemon *smbd* auf, der den Samba-Server startet und stoppt.

Werden Sie weder mit noch ohne d am Ende fündig, übergeben Sie den Dienstnamen an *apropos*: Der Befehl *apropos smb* etwa gibt daraufhin eine Liste aller Man-Pages aus, in denen sich der Begriff „smb“ findet.

2.1.12 Scripts unter der Lupe

Selbst ein Script für einen Dienst herzustellen, setzt beträchtliche Grundkenntnisse über Shell-Programmierung und reichlich Geduld voraus. Allerdings ist es auch nur in den seltensten Fällen notwendig: Linux-Distributionen bringen für jede auf den CDs vorhandene Software ein Init-Script mit.

Die Scripts finden sich im Verzeichnis */sbin/init.d/*, das auch die RunlevelKonfigurationen vorhält. Für jeden Runlevel existiert ein eigenes Verzeichnis (*rc0.d* bis *rc6.d*). Die dort enthaltenen symbolischen Links verweisen auf das tatsächliche Script in */sbin/init.d/*. Löschen Sie mit dem SysV Init Editor zum Beispiel einen Dienst im Runlevel 2, entfernt das Programm einfach den entsprechenden symbolischen Link im Verzeichnis *rc2.d*. Fügen Sie umgekehrt einen Dienst hinzu, setzt *ksysv* einen neuen Link.

Ein typisches Beispiel liefert das Script *syslog*: Hier finden sich sechs Optionen, um den Protokolldienst Syslog manuell zu bedienen: *start*, *stop*, *restart*, *reload*, *status* und *probe* (siehe auch Bild auf der nächsten Seite).

Diese Optionen offeriert jedes Systemscript. Sie dienen der Konfiguration des jeweiligen Programms, ohne dass Sie jedes Mal den Rechner neu booten müssen.

Umfangreiche Optionen: Alle Scripts im Verzeichnis /sbin/init.d/ enthalten die Optionen "start", "stop", "restart", "reload", "status" und "probe". Damit hantieren Sie manuell mit dem jeweiligen Dienst.

2.1.13 Fazit

Mit den Beschreibungen aus unserem Workshop sowie etwas Mut zum Experiment sollte es Ihnen keine Schwierigkeiten bereiten, für Ihr Linux-System eine maßgeschneiderte Bootkonfiguration zu erstellen.

Falls Ihnen unser kleiner Einstieg Lust auf mehr Informationen rund um Linux-Bootkonfigurationen gemacht hat, sollten Sie einmal beim Linux Documentation Project (www.linuxdoc.org) vorbeischauen. Hier empfiehlt sich speziell die Lektüre folgender HOWTOs:

- Bootdisk-HOWTO (www.linuxdoc.org/HOWTO/Bootdisk-HOWTO/)
- Bootprompt-HOWTO (www.linuxdoc.org/HOWTO/BootPrompt-HOWTO.html)
- Config-HOWTO (www.linuxdoc.org/HOWTO/Config-HOWTO.html)
- Kernel-HOWTO (www.linuxdoc.org/HOWTO/Kernel-HOWTO.html)
- LILO-crash-rescue-HOWTO (www.linuxdoc.org/HOWTO/LILO-crash-rescue-HOWTO.html)
- LILO-mini-HOWTO (http://www.linuxdoc.org/HOWTO/mini/LILO.html)

Vergessen Sie auf keinen Fall bei der Arbeit mit den Startup-Dateien, die nötige Vorsicht walten zu lassen. Halten Sie für den Fall der Fälle immer eine funktionsfähige Bootdiskette in der Hinterhand. Beachten Sie zudem, dass Linux nicht immer gleich Linux ist: Je nach Distribution können sowohl Name als auch Inhalt der beschriebenen Files sowie der Pfad zu ihnen leicht variieren.

tecCHANNEL-Links zum Thema	Webcode
Linux für den Desktop	a494
Linux für den Server	a487
Linux als Webserver	a442
Linux als Dial-up-Router	a322
Linux als Windows-Printserver	a392
Linux als Windows-Server	a248

2.2 Linux-Kernel-Tuning

Für die meisten Windows-Konvertiten lässt sich der Augenblick ihrer Bekehrung zu Linux exakt definieren: Es ist in der Regel der Moment, in dem das noch ungewohnte Betriebssystem zum ersten Mal mit dem selbst optimierten Kernel bootet. Die Erkenntnis schlägt ein wie eine Bombe: So einfach ist es also, ein modernes, leistungsfähiges und stabiles Betriebssystem zielgenau auf Maschine und Einsatzzweck hin anzupassen. Und unweigerlich stellt sich die Frage, warum Microsoft dem Systemverwalter nicht dieselben Möglichkeiten einräumt, wenn sich deren Produkte derart schmerzhaft im Budget bemerkbar machen.

Bei etwas Anpassung hier und ein wenig Kompilierung da stellt sich jenseits aller Euphorie jedoch irgendwann die Frage, was die Handarbeit am Kernel denn eigentlich bringt. Lohnt es sich wirklich, die zig Megabyte an Sourcen Zeit raubend zu konfigurieren und zu kompilieren? Bringt der Feinschliff an den Quellen tatsächlich noch merklichen Performance-Schub, obwohl heute alle gängigen Distributionen ohnehin schon für Pentium-Maschinen voroptimiert sind? Dieser Frage wollen wir im vorliegenden Artikel genauer nachgehen. Als Versuchsobjekt soll uns dabei ein SuSE Linux 7.3 Professional in der Standardinstallationsvariante dienen.

2.2.1 Werkzeugkasten

„You can't manage what you can't measure" lautet ein alter Grundsatz des Performance-Managements. Bevor wir uns also an die Optimierung des Systems machen, müssen wir uns einige Messwerkzeuge bereitlegen. Die meisten davon hat jede gängige Linux-Distribution ohnehin an Bord. So lässt sich beispielsweise mit dem Befehl *time* die Zeit messen, die Programme oder Scripts zur Abarbeitung benötigen. Dabei stellt man *time* einfach dem jeweiligen Kommando voran. Für die Kompilation des Kernels mit

```
time make dep clean bzImage modules
```

liefert unser Testrechner beispielsweise die Ausgabe:

```
real    36m42.392s
user    33m25.870s
sys     1m48.700s
```

Dabei gibt der *real*-Anteil die gesamte verbrauchte Laufzeit wieder. Die Werte für user und sys signalisieren die vom getesteten Programm/Script respektive die vom System verbrauchte Zeit. Differenzen zwischen der Summe der *user*- sowie *sys*-Werte und der Gesamtlaufzeit ergeben sich aus dem Ressourcenverbrauch anderer Prozesse.

Die momentane Belegung des Arbeitsspeichers lässt sich jederzeit durch die Ausgabe der Datei */proc/meminfo* feststellen. Der Befehl

```
cat /proc/meminfo
```

liefert eine umfangreiche Ausgabe, an der uns speziell vier Werte interessieren: *used* zeigt den gesamten belegten Speicher, *MemFree* die verbleibende Reserve. *Buffers* und *Cached* zeigen an, wie viel des belegten Speichers Linux für den jeweiligen Zweck verbraucht.

2.2.2 x11perf, bonnie und unixbench

Die XFree86-Suite beinhaltet mit *x11perf* auch ein Werkzeug zum Benchmarken des grafischen Subsystems. Falls Sie sehr viel Zeit mitbringen, können Sie über den Aufruf des Programms ohne Parameter sämtliche Tests vornehmen. Je nach Rechner und Grafikkarte beansprucht ein Durchlauf dann mehrere Stunden. Glücklicherweise lässt sich der Test aber auch per Parametrisierung auf einzelne Funktionsprüfungen einschränken. Erstellen Sie sich dazu am besten ein kleines Script wie das unten abgebildete.

```
#
# XFree86-Performance-Test
#
time x11perf    -time 1 \
                -bigsrect500 \
                -vseg500 -wvseg500 \
                -fspcircle100 -tr24itext \
                -copywinwin500 -copypixpix500 \
                -copyplane500 -deepcopyplane500 \
                -putimage500 -shmputxy500 \
                -getimagexy500 \
| tee xtest.out
#
```

Grafiktest: Ein einfaches Script steuert gezielt die Einzeltests des x11perf an.

Mit *bonnie* (www.textuality.com/bonnie/) stellen die meisten Distributionen ein Tool zur Verfügung, das die Performance des Filesystems misst. Dazu nutzt *bonnie* sequenziellen In- und Output sowie wahlfreien Zugriff auf ein Testfile, dessen Größe sich beim Aufruf angeben lässt. Als Vorgabe arbeitet der Benchmark mit einer 100 MByte großen Datei. Rechner mit 128 MByte oder mehr Arbeitsspeicher können jedoch einen Großteil der dabei entstehenden Zugriffe puffern. Daher sollten Sie über den Parameter *-s* die Größe der Testdatei auf solchen Rechnern erhöhen. Der Aufruf

```
bonnie -s 1000
```

veranlasst das Tool beispielsweise, mit einem 1000 MByte großen File zu arbeiten. So erhalten Sie durch Saturieren des Cache realistische Performance-Werte für die I/O auf Dateisystemebene.

Zu den nützlichsten generellen Messwerkzeugen für Linux zählt unixbench (www.tux.org/pub/tux/benchmarks/System/unixbench/). Dabei handelt es sich um eine Modifikation der Byte Unix Benchmarks, die für den Einsatz unter Linux angepasst wurde. Die aktuelle Variante trägt die Versionsnummer 4.0.1 und stellt neben Dateisystem-Benchmarks auch Tests für arithmetische Operationen und Kernel-Funktionen bereit.

2.2.3 Kernel-Kompilierung als Benchmark

Die Kompilierung eines Kernel eignet sich hervorragend als Benchmark. Zu den Faktoren, die hier die Geschwindigkeit beeinflussen, zählen neben der schieren CPU-Leistung auch die Geschwindigkeit von Festplatte und Dateisystem sowie die Performance des RAM und der Speicherverwaltung.

Jedoch lassen sich die Ergebnisse nur dann vergleichen, wenn in allen getesteten Konfigurationen derselbe Kernel mit identischen Optionen erstellt wird. Dazu kann man im Regelfall die Standardeinstellungen des Distributors einsetzen. Im Fall von SuSE finden sich diese zum Beispiel in der Datei */boot/vmlinuz.config*. Bei einer entsprechenden Einstellung des Kernel lagern darüber hinaus in der Datei */proc/config.gz* die Settings für die momentan laufende Variante des Betriebssystemkerns. Vor Gebrauch gilt es, diese Datei in ein beliebiges Verzeichnis zu kopieren und zu entpacken.

Um die gewünschten Einstellungen zu aktivieren, starten Sie aus dem Verzeichnis */usr/src/linux* mit make xconfig die Kernel-Konfiguration. Dort laden Sie mit Hilfe des Buttons *Load Configuration from File* die fragliche Konfigurationsdatei. Bei SuSE Linux können Sie also */boot/vmlinuz* oder ersatzweise die kopierte und entpackte */proc/config.gz* angeben. Über den Button *Save and Exit* speichern Sie die Einstellungen anschließend als Vorgabe für den nächsten Kompilationslauf. Nun genügt beispielsweise ein

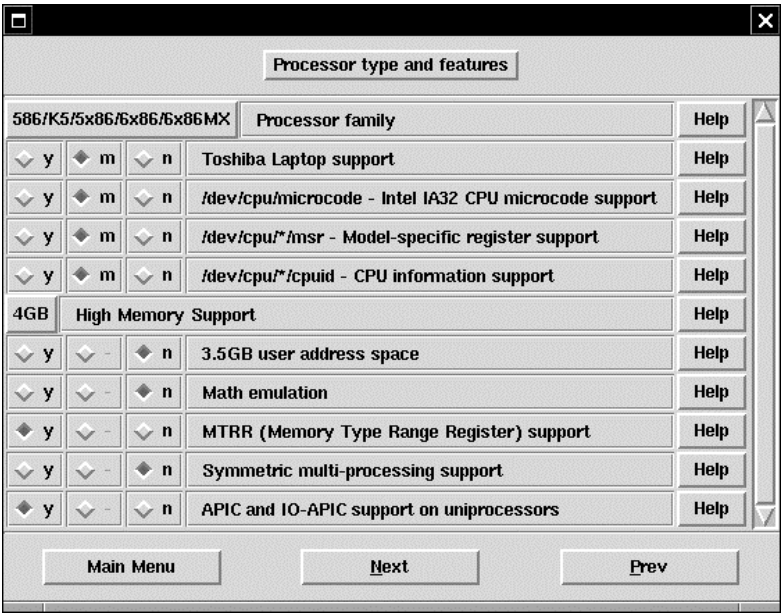
```
time make dep clean bzImage modules
```

um ein Messergebnis zu generieren. Bei dem von uns verwendeten SuSE Linux 7.3 ist der Rechner damit eine gute halbe Stunde beschäftigt.

2.2.4 CPU-gerechte Kompilierung

Als ersten Schritt zum optimierten Kernel erstellen wir eine Variante in der vom Hersteller vorgegebenen Default-Konfiguration, die wir aber für den tatsächlichen Prozessor optimieren. In unserem Fall handelt es sich dabei um einen

Pentium-III, weswegen wir bei *Processor Family* die Option *Pentium-III/Celeron/Coppermine* anwählen. Darüber hinaus verzichten wir auf die Unterstützung für vier GByte Memory und begnügen uns mit maximal zwei GByte Speicherausbau.



Minimal-Tuning: Hier wählen Sie den eingesetzten Prozessor an und konfigurieren den Hauptspeicherausbau.

Die Kompilierung des so angepassten Kernels und der Module nimmt auf unserem Testrechner knapp 35 Minuten in Anspruch. Nach der Installation des Kernels messen wir mit einer Stoppuhr sowie den bereits erwähnten Benchmarks einige Leistungsdaten. Dabei erweisen sich die erzielten Verbesserungen als recht moderat.

So sinkt die Zeit für den Bootvorgang um drei Sekunden, die Durchlaufzeit für eine Kernel-Kompilierung verringert sich um 1,2 Prozent. Beim Test mit *bonnie* ergibt sich ein Performance-Plus von rund sechs Prozent für sequenzielles Lesen und Schreiben sowie wahlfreien Zugriff. Zudem bleiben rund drei Prozent mehr Arbeitsspeicher frei. *x11perf* und *unixbench* können dagegen keinen Geschwindigkeitsvorteil vermelden, mit einer Ausnahme: Die Pipe-basierten Tests des Unix-Benchmarks verzeichnen ein Plus von rund sieben Prozent.

Ergebnisse – Prozessoranpassung			
	SuSE 7.3 (i586, 4GByte)	**Angepasst (i686, 2GByte)**	**Differenz (Prozent)**
Bootdauer (min)	1:13	1:10	4,1
RAM frei (Byte)	75788288	78159872	3,1
Kernel erstellen (min)	36:46,1	36:19,9	1,2
bonnie -s 1000			
Block write (KByte/s)	19495	19497	0,0
Block read (KByte/s)	17864	19466	9,0
Random seek (op./s)	96,8	102,6	6,0
Unixbench (loops/s)			
Dhrystone2	1362894,5	1371010,6	0,6
Arithmetic (double)	154133,6	155140,2	0,7
Pipe throughput	329216,9	349317,6	6,1
Pipe-based context switching	166980,7	179567,3	7,5

Distribution: SuSE 7.3 Professional, Standardinstallation

2.2.5 Kompilierung ohne Module

Zusätzlich zu gezielten Optimierungen für den Prozessor lässt sich ein Kernel „out-of-the-box“ auch deutlich verschlanken. Er muss ja Module für verschiedenste Zwecke bereitstellen, die jedoch für die meisten Installationen nicht gebraucht werden. Im Regelfall werden ganze Modulgruppen nicht benötigt. So können auf Desktops beispielsweise die IrDA-Komponenten entfallen, auf Netzwerk-Clients in der Regel das ISDN-Subsystem. Die Amateur-Radio-Unterstützung oder die Bluetooth-Komponenten zählen ebenfalls zu den typischen Entsorgungskandidaten. Bei der gezielten Anpassung der Kernel-Konfiguration an die vorhandene Hardware hilft unter anderem der Befehl

```
cat /proc/modules
```

der eine Liste der tatsächlich geladenen Module liefert. Daneben empfiehlt sich auch ein Blick in die Datei */var/log/boot.msg*. Dort finden Sie ein Protokoll aller vom Kernel beim Systemstart ausgegebenen Meldungen, das ebenfalls Informationen zur Kernel-Konfiguration liefern kann.

In der schlanken Konfiguration ohne überfüssigen Ballast sinkt der Zeitauf wand für eine komplette Kernel-Kompilierung von 35 auf knapp 7 Minuten. Der

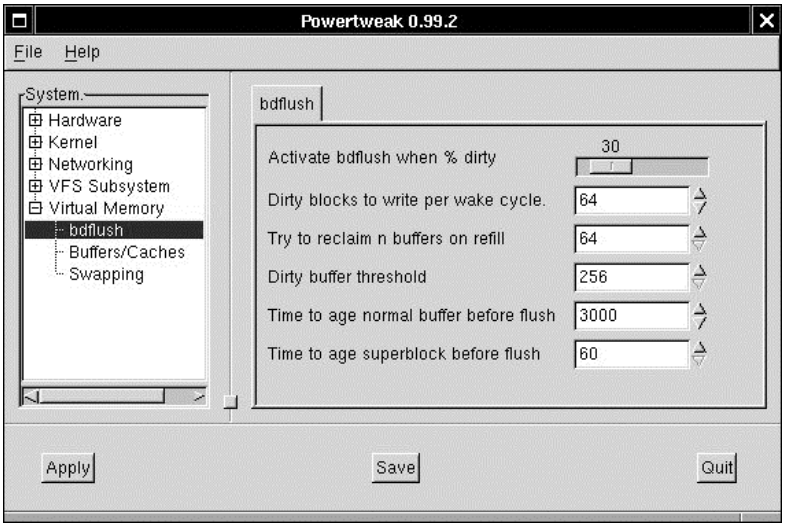
Speicherbedarf des Systems reduziert sich erneut, auch bei sequenziellen Lesevorgängen erreichen wir nochmals einen Performance-Vorteil. Ansonsten weisen unsere Benchmarks jedoch keine wesentliche Beschleunigung gegenüber einem modularen Kernel aus.

Ergebnisse – Statischer Kernel			
	SuSE 7.3 (i586, 4 GByte)	**Angepasst (i686, 2 GByte stat.)**	**Differenz (Prozent)**
Bootdauer (min)	1:13	1:10	4,1
RAM frei (Byte)	75788288	82812928	9,3
Kernel erstellen (min)	36:46,1	36:15,2	1,4
Bonnie -s 1000			
Block write (KB/s)	19495	19544	0,3
Block read (KB/s)	17864	20134	12,7
Random seek (op./s)	96,8	102,6	6,0
Unixbench (loops/s)			
Dhrystone2	1362894,5	1371702,1	0,6
Arithmetic (double)	154133,6	155144,2	0,7
Pipe throughput	329216,9	349319,8	6,1
Pipe-based context switching	166980,7	179571,1	7,5

Distribution: SuSE 7.3 Professional, Standardinstallation

2.2.6 Powertweak für Linux

Einige Performance-relevante Kernel-Variablen lassen sich auch während des laufenden Betriebs modifizieren. Dies betrifft vornehmlich Parameter rund um Caching und Swapping, also Virtual-Memory-Komponenten. Linux hält diese Variablen im *proc*-Pseudo-Filesystem unter */proc/sys/vm* in mehreren Dateien vor. Der klassische Weg, dort Einstellungen vorzunehmen, führt über einen *sysctl()*-Aufruf, etwa per Script.



Komfortabel: Mit Powertweak für Linux lassen sich zahlreiche Kernel-Parameter dynamisch konfigurieren.

Es geht allerdings auch deutlich bequemer: *powertweak-linux* (http://powertweak.sourceforge.net) von Dave Jones bietet sowohl ein bequemes User-Interface für die Einstellung als auch einen Daemon (*powertweakd*), der die veränderten Parameter bei jedem Systemstart automatisch wieder setzt. Neben Variablen, die für die Arbeitsweise des Virtual Memory (VM) zuständig zeichnen, kann *powertweak-linux* auch zahlreiche andere Einstellungen beeinflussen. Dazu zählen unter anderem Settings rund um die Hardware, das Netzwerksubsystem oder das Kernel-Logging. Wir gehen im Folgenden jedoch nur auf die VM-spezifischen Parameter ein.

Diese können für die Leistung des Rechners in bestimmten Einsatzumgebungen deutliche Folgen haben. Generelle Vergleiche lassen sich jedoch nicht anstellen, da die Auswirkungen je nach konkretem Verwendungszweck und abzuarbeitenden Anwendungen völlig unterschiedlich ausfallen können.

Auch die Hardware des Rechners ist hier ausschlaggebend. Ältere PCs mit langsamen Festplatten gewinnen etwa bei einer Optimierung des Festplatten-Cache deutlich mehr als moderne Systeme. Auf Grund der starken Abhängigkeit von Anwendung und Hardware verzichten wir im Folgenden auf die Angabe von konkreten Einstellungen und Benchmarks.

2.2.7 Powertweak: bdflush-Parameter

Der Kernel-Daemon *bdflush* kontrolliert, wie und wann der Inhalt der Buffer nach Modifikationen auf die Platte zurückgeschrieben wird. Zur Kontrolle dieser Vorgänge stellt powertweak-linux im Abschnitt *Virtual Memory/bdflush* sechs Parameter zur Verfügung. Über drei davon lässt sich das System gezielt auf bestimmte Anwendungen hin optimieren. Das *Setting Activate bdflush when % dirty* gibt an, wie viele modifizierte Puffer im Cache gehalten werden sollen. Je höher diese Anzahl, desto seltener muss der Kernel auf die Platte zugreifen. Andererseits dauert ein Schreibvorgang dann im Einzelfall länger.

Die Einstellung *Dirty blocks to write per wake cycle* begrenzt die Anzahl der modifizierten Puffer, die der Kernel in einem Rutsch auf die Platte schreibt. Hohe Parameter verursachen seltenere, dafür aber längere Zugriffe. Niedrige Werte sorgen für ein verteiltes und unauffälligeres Schreiben.

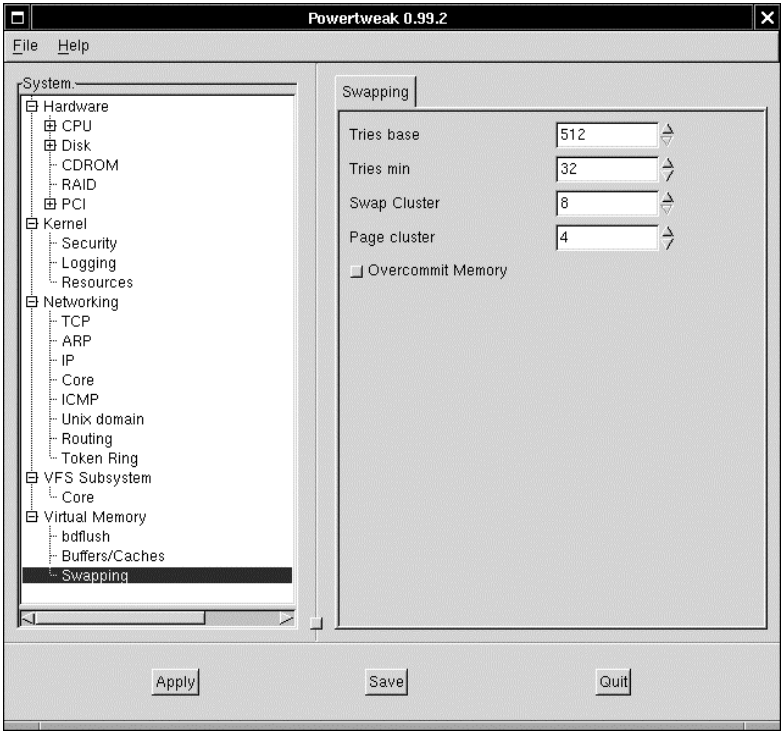
Mit *Try to reclaim n buffers on refill* geben Sie die Anzahl der Puffer an, die der Kernel jeweils neben der aktuellen Pufferbelegung in Reserve halten soll. Je höher diese Zahl ausfällt, desto mehr Arbeitsspeicher beansprucht *bdflush*. Dafür lassen sich aktuelle Pufferanforderungen sehr viel schneller befriedigen.

2.2.8 Powertweak: Swapping-Parameter

Über den Abschnitt *Virtual Memory/Swapping* lässt sich das Verhalten des Kernel-Swapdaemon *kswapd* beeinflussen. Als interessantester Parameter findet sich hier *Swap cluster*. Er gibt an, wie viele Pages *kswapd* in einem Zug auf die Platte schreiben soll. Je höher dieser Wert, desto effektiver kann der Daemon die Auslagerung abwickeln. Zu hohe Angaben bremsen das System jedoch wieder aus, da dann die Request-Warteschlange zuläuft. Über einen höheren Wert für *Page cluster* lässt sich zudem der Read-ahead auf Kosten der Auslagerungslatenz steigern.

Eine ebenso nützliche wie gefährliche Einstellungsmöglichkeit bietet der Punkt *Overcommit Memory*. Manche Programme versuchen sich per *malloc()* „vorsorglich“ große Speichermengen zu sichern, die sie dann doch nur teilweise in Anspruch nehmen. Dadurch scheitert die Zuweisung, obwohl das Programm eigentlich ausgeführt werden könnte. Schalten Sie *Overcommit Memory* ein, signalisiert der Kernel in jedem Fall ausreichend vorhandenen Speicher. Andererseits besteht damit das Risiko eines Speicherüberlaufs.

Mit dem Wert für *Tries base* definieren Sie die maximale Anzahl von Pages, die *kswapd* in einem Zug freimachen soll. *Tries min* gibt an, wie oft *kswapd* eine Page freizumachen versucht. Änderungen an diesen Werten führen im Regelfall selten zu mehr Performance, sondern nur zu intensiverem Swapping.



Plattenschoner: Über diese Registerkarte konfigurieren Sie die Parameter rund um das Swap-Verhalten.

2.2.9 Fazit

Soll Linux auf dem Desktop zum Einsatz kommen, drängt wohl mehr der Optimierungswille des Anwenders als die tatsächlich erzielbare Leistungssteigerung zum Tuning des Kernel. Mehr als zehn Prozent Performance-Plus lassen sich bei gängigen Distributionen nicht erreichen - was übrigens für die Qualität und das durchdachte Design des Betriebssystemkerns spricht. Eine exakt auf die Hardware zugeschnittene Konfiguration erleichtert jedoch ein laufendes Update des Kernel: Der Verzicht auf den modularen Aufbau drückt die Kompilierungszeiten um über 80 Prozent. Ganz anders stellt sich die Situation beim Servereinsatz dar. Hier kommt jede Leistungssteigerung direkt den Clients zugute. Allerdings muss die optimale Konfiguration jeweils gezielt für den Einzelfall erarbeitet werden. Dabei leisten Werkzeuge zur dynamischen Konfiguration - wie powertweaklinux - wertvolle Dienste.

2.2.10 Testkonfiguration

Als Testsystem diente uns ein Pentium-III/650 mit 128 MByte RAM und 20,5-GByte-EIDE-Festplatte und SuSE Linux 7.3 Professional.

Testkonfiguration	
Komponente	**Daten**
Mainboard	Asus P3B-F Rev. 1004, Slot 1, BX-Chipsatz
Prozessor	Pentium III/650, 100 MHz FSB
RAM	128 MByte PC100 SDRAM
Festplatte	IBM DPTA-372050, 20,5 GByte UltraDMA/66
CD-ROM	Toshiba DVD-ROM SD-M1402, 40x, UltraDMA/33
Grafikkarte	Elsa Erazor III Pro, AGP, Riva TNT2, 32 Mbyte
Netzwerkkarte	3Com 3C905C-TX, PCI, 10/100 MBit/s
Soundkarte	Ensoniq ES-1371 Rev.8, PCI

Für jede Konfiguration ermitteln wir:

- Die Zeit zum Booten des Systems: Dazu starten wir über *kdm* den PC neu und messen die Dauer bis zum Wiedererscheinen der Anmeldeaufforderung.
- Die Speicherauslastung: Diese werten wir über *cat /proc/meminfo* aus.
- Die Leistung des X11-Systems: Sie wird mittels *x11perf* gemessen. Dazu führt ein Script je einen Test aus jeder Funktionsgruppe von *x11perf* aus.
- Die Performance des Dateisystems: Hierzu kommt *bonnie* mit den Parametern *-s 1000 -y* zum Einsatz.
- Die allgemeine Systemleistung: Als High-Level-Benchmark wird *unixbench-4.0.1* eingesetzt.

tecChannel-Links zum Thema	Webcode
Systemtools für Linux	a831
Security-Tools für Linux	a814
Linux 2.4 für den Desktop	a706
Linux für den Server	a487
Sichere Linux-Workstation	a720
Desktop-Firewall mit Linux 2.4	a751
TCP/IP-Netze mit Linux	a562

2.3 TCP/IP-Netze mit Linux

Das Internet-Protokoll TCP/IP gehört zur Standardausstattung von Linux. Damit bauen Sie mit minimalem Aufwand große und kleine Netzwerke auf.

Ob zu Hause, im kleinen Büro oder im Großkonzern - Einzelplatz-PCs sind out, es lebe das Netzwerk! Als Grundlage für die Kommunikation vernetzter Rechner dient heute TCP/IP, das universelle Netzwerkprotokoll des Internets. Innerhalb weniger Jahre hat es alle Konkurrenten - selbst so mächtige Gegner wie Novells IPX oder NetBIOS von IBM/Microsoft - fast völlig verdrängt.

Den Betriebssystemen der Unix-Familie wurde TCP/IP schon als Muttersprache mit in die Wiege gelegt. Daher eignen gerade sie sich ideal zum Aufbau TCP/IP-basierter LANs. Dies gilt nicht nur für die teuren kommerziellen Unix-Derivate, sondern genauso für das für jedermann erschwingliche Linux.

Mit dem Opensource Unix steht ein ideales Netzwerkbetriebssystem praktisch kostenlos zur Verfügung. Und auch die Netzwerkhardware kostet nicht mehr die Welt: Network-Starter-Kits mit Netzwerkkarten, Hub und Verkabelung sind für rund 100 Euro zu haben.

2.3.1 Linux und die Netzwerkkarte

Hubs - und auch ihre leistungsfähigeren und teureren Vettern, die Switches - fungieren im Netz als automatische Verteilerkomponenten. Hier gibt es praktisch nichts zu konfigurieren. Ganz anders bei den Netzwerkkarten: Sie dienen als zentrale Schnittstelle zwischen dem Datenbus innerhalb und dem Netzwerk außerhalb des Rechners. Jedes IP-Datenpaket von und zum Computer muss dieses Tor passieren.

Bei einer Neuinstallation erkennt Linux alle gängigen und selbst einige eher ungewöhnliche Netzwerkkarten automatisch und lädt das zugehörige Treibermodul. Mit welchen NICs das Betriebssystem dabei klar kommt, können Sie im Linux Networking HOWTO nachlesen (www.linuxdoc.org/HOWTO/Net-HOWTO/). Hardware, die sich in dieser Aufstellung nicht findet, läuft mit ziemlicher Sicherheit nicht unter Linux.

Falls Sie in der Liste ihre NIC weder nach Hersteller- noch Modellnamen finden, ist das jedoch noch kein Grund, sich graue Haare wachsen zu lassen. Linux identifiziert Netzwerkkarten nur im Ausnahmefall nach diesen Kriterien, stattdessen fast immer anhand des verwendeten Ethernet-Controllers. Es lohnt sich also im Zweifelsfall, die Netzwerkkarte zur Hand zu nehmen und den Aufdruck des größten Chips darauf genauer zu studieren.

Das gilt speziell, falls Sie eine Netzwerkkarte nachinstallieren oder gegen eine NIC anderen Typs austauschen wollen. Dazu müssen Sie dem installierten Linux zunächst einmal beibringen, beim Booten des Systems den neuen Treiber mit einzubinden.

Das können Sie auf zwei Wegen erledigen: Entweder teilen Sie bei modularem Kernel mit, welches Modul zu laden ist, oder Sie kompilieren einen maßgeschneiderten Kernel neu.

2.3.2 NIC-Treiber als Modul nachladen

Linux lagert die Kernel-Module im Verzeichnis */lib/modules/kernel-version*, bei aktuellen Distributionen also üblicherweise in */lib/modules/2.2.16*. Hier finden sich zahlreiche, nach dem Einsatzbereich der Module benannte Subdirectories. Uns interessiert speziell das Unterverzeichnis *net*, in dem sich die Module für die Netzwerkkarte befinden.

Hier picken Sie sich den Treiber für Ihre NIC heraus und merken sich den Namen. Im Falle einer 3Com FastEtherlink 10/100 etwa wäre das *3c59x.o*, für eine der sehr verbreiteten Karten mit DEC21x4x-Controller *tulip.o*, für eine der preiswerten NICs mit Realtek-Chipsatz *rtl8139.o*.

Nun müssen Sie den gewünschten Treiber noch dem Betriebssystem bekannt machen. Öffnen Sie als root mit Hilfe eines Texteditors das File */etc/conf modules* und fügen Sie hier eine Zeile für das zu ladende Modul ein:

```
alias eth0 3c59x
```

Auf diese Weise teilen Sie dem Kernel mit, dass er für das Device *eth0* den Treiber *3c59x.o* verwenden soll. Nun starten Sie per *init 6* den Rechner neu, um nachzuprüfen, ob das Modul bei Systemstart auch wirklich automatisch mit eingebunden wird.

```
nulldevice:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:DA:E3:8E:27
          BROADCAST 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:100
          Interrupt:10 Base address:0xb800
```

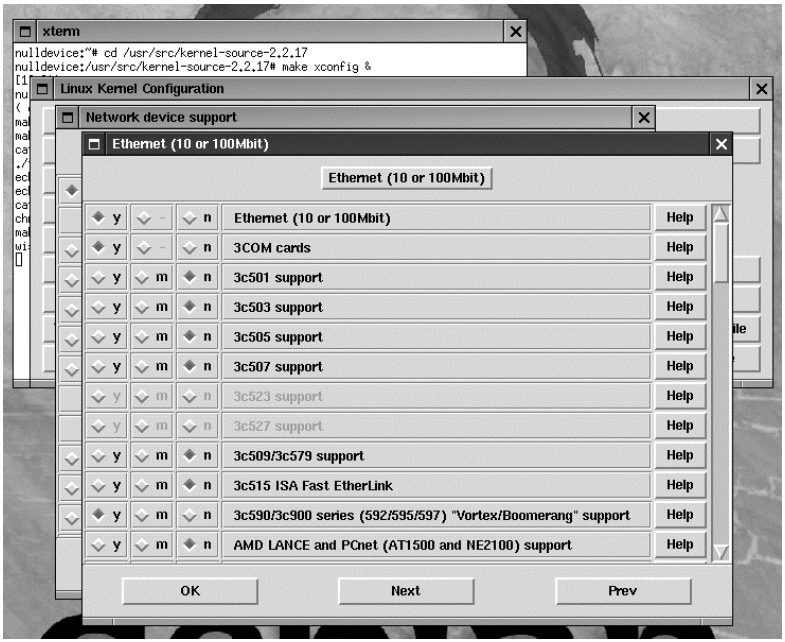
Erster Funktions-Check: Mit Hilfe des Kommandos ifconfig prüfen Sie, ob das Treibermodul eingebunden wurde.

Vom Erfolg des Vorgangs können Sie sich durch den Aufruf des Kommandos *ifconfig eth0* überzeugen. Dessen Ausgabe sieht - je nach der von Ihnen eingebundenen NIC - in etwa wie in der obigen Abbildung aus.

2.3.3 Kernel für die NIC neu kompilieren

Alternativ lässt sich der Support für die gewünschte Netzwerkkarte auch in den Kernel kompilieren. Das funktioniert garantiert bei jeder Distribution und bietet zudem die Gelegenheit, den Kernel auf den neuesten Stand zu bringen.

Holen Sie sich dazu die aktuellen Kernel-Quellen bei Kernel.org und installieren Sie diese in das Verzeichnis */usr/src/kernel-version*. Wechseln Sie nun in dieses Verzeichnis und starten Sie mit *make menuconfig* (textbasiert) beziehungsweise *make xconfig* (grafisch) das Konfigurations-Utility.



Kernel-Konfiguration: Über den Punkt Ethernet in der Option Network Device Support binden Sie NIC-Treiber (hier für eine 3C905B-TX) fest in den Kernel ein.

Falls Sie eine relativ neue oder ungewöhnliche Netzwerkkarte Ihr Eigen nennen, aktivieren Sie unter *Code maturity level options* den Punkt *Prompt for development and/or incomplete code/drivers*. Wählen Sie anschließend aus dem Hauptmenü die Option *Network Device Support* und dort den Unterpunkt *Ethernet (10 or 100MBit)*. Hier finden Sie alle verfügbaren NIC-Treiber versammelt.

Suchen Sie den passenden Treiber und markieren Sie dort die Checkbox *y*, um ihn fest in den Kernel einzubinden. Vergessen Sie nicht, nach der Rückkehr

ins Hauptmenü die Konfiguration zu speichern. Jetzt können Sie den neuen Kernel kompilieren und via LILO einbinden. Eine ausführliche Beschreibung (www.linuxdoc.org/HOWTO/Kernel-HOWTO.html) dafür findet sich beim Linux Documentation Project.

2.3.4 IP-Adressen auf die Schnelle

Bevor wir daran gehen, das Netzwerk-Interface weiter zu konfigurieren, ist noch ein kurzer Ausflug in die Theorie angesagt. Motto: Woher kommen die IP-Adressen? Wir beschränken unsere Ausführungen dabei auf so genannte Class-C-Netze, die maximal 254 Endgeräte aufnehmen können - für unsere Zwecke mehr als genug.

Für private LANs, die nicht direkt mit dem Internet in Verbindung stehen, hat IANA unter anderem die Netze mit den Nummern 192.168.0.0 bis 192.168.255.0 reserviert. Suchen Sie sich nach Belieben eines dieser Netze für Ihren Gebrauch heraus, sagen wir die 192.168.80.0.

Innerhalb dieses Netzes können Sie die Adressen von 192.168.80.1 bis 192.168.80.254 für Endgeräte frei verwenden. Die 192.168.80.0 bleibt als Netzwerknummer reserviert. Über 192.168.80.255 lassen sich alle Rechner des Netzes gleichzeitig ansprechen. Zur Adressierung in unserem Beispiel wird noch die Netzmaske 255.255.255.0 benötigt, in der alle Bits des Netzwerk-Deskriptors gesetzt, alle Bits der möglichen Maschinenadressen aber gelöscht sind.

Jeder Rechner Ihres Netzes muss eine eindeutige Adresse aus dem zur Verfügung stehenden Bereich erhalten. Traditionell nutzt ein Gateway, über den die Verbindung zu anderen Netzen erfolgt, die Adresse 1 oder 254. Somit stehen für die anderen Rechner in unserem Beispiel die IP-Adressen 192.168.80.2 bis 192.168.80.253 zur Disposition.

2.3.5 Netzanbindung mit ifconfig

Um die Netzwerkkarte mit einer eigenen IP-Adresse ins Netz einzubinden, verwendet man den Befehl *ifconfig*. Um also Ihren Rechner mit der Host-Adresse 77 in das Class-C-Netz 192.168.80.0 einzubinden, lautet das vollständige Kommando:

```
ifconfig eth0 192.168.80.77 netmask 255.255.255.0 up
```

Das Anhängsel *up* teilt dem Interface mit, dass jetzt die Aktivierung ansteht. Umgekehrt lässt sich die Karte mit einem einfachen *ifconfig eth0 down* auch wieder deaktivieren.

Die erfolgreiche Einbindung der Karte prüfen Sie erneut mit Hilfe des Befehls *ifconfig eth0*. Diesmal sollte die Ausgabe in etwa wie in der unten stehenden Abbildung aussehen.

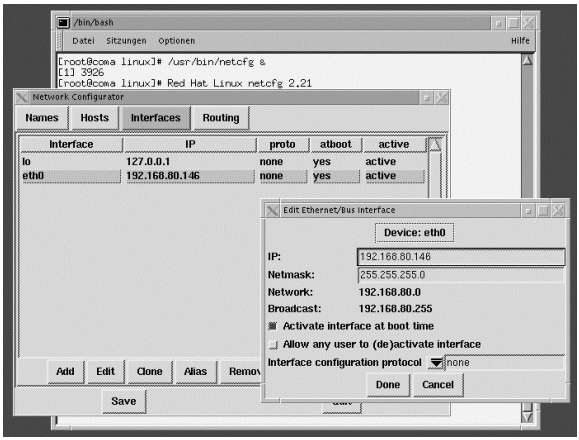
```
nulldevice:~# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:DA:E3:8E:27
          inet addr:192.168.80.77  Bcast:192.168.80.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:606 errors:1 dropped:0 overruns:0 frame:2
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:100
          Interrupt:10 Base address:0xb800
```

Jetzt läuft's: Nach dem Hochziehen des Netzwerk-Interface mit ifconfig laufen die ersten Broadcast-Pakete ein.

Im Gegensatz zu der Meldung unmittelbar nach Einbindung des Treibers zeigt *ifconfig* nun auch die IP- und Broadcast-Adresse sowie die Netzwerkmaske an. Ist der Rechner an ein schon laufendes Netz angebunden, findet sich unter *RX packets* zudem eine bei jedem Aufruf des Kommandos steigende Zahl: Sie signalisiert die als Broadcasts eingegangenen Pakete.

2.3.6 Speichern der NIC-Einstellungen

Zwar eignet sich *ifconfig* wunderbar, um die Einstellung der NIC zu testen. Auf Dauer wäre es allerdings lästig, die Settings jedes Mal manuell vorzunehmen. Also gilt es, sie zu speichern - aber wo?



Wichtiges Helferlein: Beinahe alle Distributionen bringen Tools zur Konfiguration der Netzwerk-Interfaces mit. Hier der netcfg von Red Hat.

Generell ließe sich das *ifconfig*-Kommando durchaus auch direkt in die Bootkonfiguration einarbeiten. Praktisch jede Distribution regelt jedoch den Vorgang etwas anders. Eine generelle Leitlinie lässt sich hier also nicht geben, im Zweifelsfall hilft das Studium des Distributionshandbuchs sowie unseres Artikels über Linux-Bootkonfigurationen (webcode: a546) weiter.

Die meisten Distributionen - von vereinzelten Ausnahmen wie Debian einmal abgesehen - bringen jedoch dankenswerterweise Tools zur Netzwerkkonfiguration mit. Es empfiehlt sich dringend, diese auch zu benutzen: Sie ersparen sich auf diese Weise eine Menge Schweiß und Zeit.

2.3.7 Debian-Special

Der Debian-Distribution allerdings liegt selbst in der aktuellen Version 2.2 („Potato“) noch kein Tool zur Konfiguration der Netzwerkkarte bei. Hier hilft nur das intensive Studium des Handbuchs weiter.

Ein kleiner Tipp für Debian-Jünger: Sie finden die Settings für die Netzwerkkarten in */etc/network/interfaces*. Eine typische statische Zuordnung für eine NIC sieht folgendermaßen aus:

```
# 3C905B-TX auf eth0
iface eth0 inet static
  address 192.168.80.77
  network 192.168.80.0
  netmask 255.255.255.0
  broadcast 192.168.80.255
  gateway 192.168.80.254
```

Den Rest erledigt das System im Zuge des Bootvorgangs dann selbst. Falls der Rechner als DHCP-Client laufen soll, können Sie sich die Parameterangaben schenken: Ersetzen Sie in diesem Fall in der Kopfzeile das Schlüsselwort *static* durch *dhcp*.

2.3.8 Pfad nach außen

Damit der Rechner auch mit anderen Geräten innerhalb des Netzwerks kommunizieren kann, benötigt er passende Wegangaben, die so genannten Routen. Den Pfad für die Kommunikation mit den anderen Rechnern im lokalen Netz legen Sie mit dem Aufruf des Kommandos *route* fest, in unserem Beispiel:

```
route add -net 192.168.80.0 netmask 255.255.255.0 eth0
```

Eine Klartextübersetzung würde in etwa lauten: „Schicke allen Datenverkehr an das Class-C-Netz 192.168.80.0 über das Interface *eth0* nach draußen.“

Wollen Sie allerdings nicht nur im eigenen LAN, sondern auch mit Rechnern in anderen Netzen kommunizieren, erweist sich diese Art der Wegangabe als äußerst umständlich: Sie müssten für jedes anzusprechende Netz eine eigene Route eintragen. Dem beugt die Möglichkeit zur Angabe einer Default-Route vor:

```
route add default gw 192.186.80.254 eth0
```

Deutsche Übersetzung: „Schicke alles, was nicht ins lokale Netz soll, über *eth0* an das Gateway mit der IP-Adresse 192.168.80.254“.

```
nulldevice:~# route
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
192.168.80.0    *               255.255.255.0   U     0      0        0 eth0
default         192.168.80.254  0.0.0.0         UG    0      0        0 eth0
nulldevice:~#
```

Kontrolle ist besser: Beim Aufruf ohne Parameter gibt route die bereits aktiven Leitwege aus.

Diese beiden Standardrouten legen Sie im Normalfall bereits während der Konfiguration der NIC mit fest. Ob sie tatsächlich eingetragen wurden, überprüfen Sie mit einem Aufruf von *route* ohne Parameterangabe. Sie sollten dann - in unserem Beispiel - eine Ausgabe ähnlich jener in der oben stehenden Abbildung erhalten.

2.3.9 Testen der Basiskonfiguration

Nachdem Sie die beschriebenen Konfigurationsarbeiten an mindestens zwei der Rechner abgeschlossen haben, steht ein erster Funktionstest Ihres Netzwerks an. Dazu benutzen Sie das Kommando *ping*, das ein spezielles Datenpaket an die angegebene Gegenstelle verschickt. Sofern diese die Daten erhält, versieht sie sie mit einigen Zusatzinformationen und sendet sie umgehend zurück.

In unserem Beispiel haben Sie den PC *client* (192.168.80.77) und den Gateway-Rechner 192.168.80.254 konfiguriert und senden nun drei Testpakete vom Desktop an den Gateway. Eine saubere Konfiguration vorausgesetzt, erhalten Sie umgehend Antwort:

```
client:~# ping -c3 192.168.80.254
PING 192.168.80.254 (192.168.80.254): 56 data bytes
64 bytes from 192.168.80.254: icmp_seq=0 ttl=64 time=3.2 ms
64 bytes from 192.168.80.254: icmp_seq=1 ttl=64 time=3.0 ms
64 bytes from 192.168.80.254: icmp_seq=2 ttl=64 time=3.0 ms
```

```
--- 192.168.80.254 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 3.0/3.0/3.2 ms
```

Erhalten Sie stattdessen von der Gegenstelle keinerlei Lebenszeichen, gilt es, auf beiden Rechnern noch einmal Interface- und Routeneinstellungen sowie die Verkabelung zu prüfen.

2.3.10 Namensauflösung über /etc/hosts

Um die anderen Rechner im lokalen Netz nicht nur über deren IP-Adressen, sondern auch über leichter zu behaltende Namen ansprechen zu können, erstellen Sie eine Datei namens */etc/hosts*. In drei durch beliebigen Whitespace getrennten Spalten enthält sie IP-Adressen, den offiziellen Rechnernamen sowie einen Kurznamen für das bequemere Handling:

```
# /etc/hosts
127.0.0.1             localhost              loopback
192.168.80.77  client.mydom.de client
192.168.80.254 gw.mydom.de     gw
(... und so weiter ...)
```

Diese Art der Namensauflösung bringt allerdings einige Nachteile mit sich: So muss die */etc/hosts* auf jedem Rechner erstellt und bei Veränderungen im Netz auf jedem Computer angepasst werden.

Andererseits erfolgt die Auflösung wesentlich schneller als bei der Nutzung über einen Nameserver, so dass selbst in großen LANs jede Maschine über eine */etc/hosts* mit den Namen und Adressen der am häufigsten genutzten Verbindungspartner verfügen sollte.

2.3.11 Namensauflösung über DNS

Um die Namensauflösung über einen Nameserver zu regeln, speichern Sie einige Einstellungen in einer Datei namens */etc/resolv.conf*. Sie sieht etwa folgendermaßen aus:

```
# /etc/resolv.conf
domain mydom.de
search mydom.de mydom.local
nameserver 195.30.0.2
nameserver 195.30.0.1
```

Hinter dem Schlüsselwort domain findet sich der Name des lokalen Netzwerks. Die Eintragungen bei *search* legen fest, dass bei Angabe eines Rechners ohne

voll qualifizierten Namen die Maschine in den Domains *mydom.de* und *my dom.local* gesucht werden soll.

Die Angabe für die Nameserver - im Regelfall wohl diejenigen Ihres Providers - lässt sich beliebig oft wiederholen. Bei der Namensauflösung über DNS fragt der Rechner dann die aufgeführten Nameserver der Reihe nach ab, bis er die gewünschte Antwort erhält.

2.3.12 Regeln für die Namensauflösung

Nun müssen Sie dem Rechner noch mitteilen, in welcher Reihenfolge er die beiden Möglichkeiten der Namensauflösung nutzen soll. Dazu erstellen Sie eine Datei namens */etc/host.conf*. Sinnigerweise sieht diese folgendermaßen aus:

```
# /etc/host.conf
order hosts,bind
multi on
```

Diese Standardkonfiguration weist den Rechner an, zur Namensauflösung zunächst die schnelle Methode über die Auswertung der */etc/hosts* zu nutzen. Erst wenn dies nicht zum gewünschten Erfolg führt, wendet er sich an einen Nameserver.

Das Schlüsselwort *multi* mit der Parameterangabe *on* spezifiziert, dass beim Vorliegen mehrerer gültiger Adressen für einen Host in der */etc/hosts* alle davon zurückgegeben werden sollen. Andernfalls wird der Host stets nur mit der ersten gefundenen Adresse assoziiert.

2.3.13 Fazit

Nach Abschluss der beschriebenen Konfigurationsarbeiten auf den Rechnern in Ihrem Netzwerk können Sie sich als stolzer Benutzer eines funktionsfähigen TCP/IP-LANs betrachten. Zwar war die Einrichtung nicht besonders schwierig, einen höheren Nutzwert hat Ihr frisch aufgesetztes Netzwerk allerdings auch noch nicht.

Den erhält es erst, indem Sie Ihre Rechner mit nützlichen Netzwerkdiensten aufpäppeln. So eignet sich Linux ideal als Dienstleister für Windows-Clients, wie unsere Workshops Linux als Windows-Server (webcode: a248) und Linux als Samba-Printserver (webcode: a392) zeigen.

Auch gegenüber der Außenwelt gibt sich Linux gern kommunikativ - lesen Sie dazu unsere Workshops Linux als Webserver (webcode: a442) und Linux als Dial-up-Router (webcode: a322).

Bei Bedarf macht sich Linux auch als DNS- und DHCP-Server nützlich, befördert als Faxmaschine Papierenes über Datenleitungen oder unterhält sich via

NFS und NIS mit anderen „Unixen“. Wie Sie Ihr Linux zu solch nützlichen Tätigkeiten bewegen, lesen Sie in den folgenden Kapiteln.

tecCHANNEL-Links zum Thema	Webcode
TCP/IP-Grundlagen	a209
Linux-Bootkonfigurationen	a546
Linux als Webserver	a442
Linux als Dial-up-Router	a322
Linux als Samba-Printserver	a392
Linux als Windows-Server	a248

2.4 DSL unter Linux

Die Einrichtung des High-Speed-Zugangs ist unter Linux mit einigen Hürden verbunden. Schon immer galt Linux als ausgesprochenes Netzwerkbetriebssystem, da es von Anfang an als solches konzipiert und entwickelt wurde. Linux kann nicht nur mit großen zu übertragenden Datenmengen umgehen. In Netzwerken übernimmt es oft als Router oder Gateway zentrale Aufgaben.

Damit eignet sich Linux sehr gut für die Anbindung ans Internet. Soll der Zugang per ADSL erfolgen, sind aber einige Hürden zu nehmen. Die erste beginnt bereits bei der Wahl des geeigneten Treibers und dessen Installation. Beim Einbinden der Netzwerkkarte und deren Aktivierung gilt es, etliche wichtige Punkte zu beachten.

Auch wenn vieles von den Installationsroutinen der Distributionen automatisch erledigt wird, lohnt sich ein Blick in die Logfiles, die bei den einzelnen Arbeitsschritten angelegt werden. Speziell beim Verbindungsaufbau und den ersten Tests können Sie Fehlern so auf die Schliche kommen. Angefangen von diesen Punkten bis hin zum Verbindungsaufbau via manueller oder automatischer Einwahl finden Sie auf den folgenden Seiten die wichtigsten Informationen zur Nutzung der ADSL-Technologie unter Linux.

2.4.1 Treibervarianten

Das beim Einsatz von ADSL in Deutschland am häufigsten eingesetzte Übertragungsprotokoll PPPoE ist im RFC2516 („A Method for Transmitting PPP Over Ethernet“) definiert. Linux unterstützt dieses Übertragungsprotokoll bereits seit längerem. Es sind mittlerweile drei populäre Treibervarianten gebräuchlich:

- Kernel-Treiber für den Kernel 2.2 (*pppoed*)
- Kernel-Treiber für den Kernel 2.4
- Userspace-Treiber Roaring Penguin (*rp-pppoe*)

Keine so große Popularität wie der Roaring-Penguin-Treiber hat der Kernel-2.2-Treiber erlangt, weil ihm einige wichtige Funktionen fehlen. Daher wird er auch schon seit einiger Zeit nicht mehr weiterentwickelt.

Der Kernel-2.4-Treiber ist der Neueste unter den PPPoE-Treibern. Er wurde, wie an seiner Bezeichnung abzulesen ist, für den 2.4er Kernel entwickelt. Schon heute ist absehbar, dass dieser Treiber in naher Zukunft den Standard unter den PPPoE-Implementationen bilden wird. Durch das Kernel-Konzept weist er beachtliche Vorteile gegenüber dem Userspace-Treiber *rp-pppoe* auf. So müssen die Daten nicht Ressourcen verbrauchend zwischen Userspace und Kernel umkopiert werden. Damit werden Router beziehungsweise Zugangsrechner erheblich entlastet.

2.4.2 Einzig empfehlenswert: Roaring Penguin

Der Haken am Kernel-2.4-Treiber ist, dass er sich derzeit nicht für den kritischen Einsatz empfiehlt. Die Kernel-Entwickler stufen den Treiber immer noch als experimentell ein, so dass das Patchen der Point-to-Point-Software *pppd* (ppp-Daemon) erforderlich ist. Dieser Patch wird erst in der Version 2.4.2 des *pppd* Einzug halten.

Somit ist der Roaring-Penguin-Treiber (www.roaringpenguin.com/pppoe) der einzige zurzeit - auch für den kritischen Einsatz - empfehlenswerte Treiber. Er lässt kaum Wünsche offen. So bietet er beispielsweise umfangreiche Diagnosemöglichkeiten und eine Unterstützung von Dial-on-Demand. Bei der Konfiguration des Systems bietet er einen großen Vorteil: Das Patchen beziehungsweise Neukompilieren des Linux-Kernels und der *pppd*-Software entfällt. Dadurch werden speziell bei unerfahrenen Administratoren potenzielle Fehlerquellen ausgeschlossen.

Der Userspace-Treiber *rp-pppoe* läuft außerhalb des Kernels. Damit ist er unabhängig von der verwendeten Kernel-Version. Deshalb bietet dieser Treiber auch die einzige Möglichkeit, einem betagten Linux-System, zum Beispiel mit einem 2.0er Kernel, zur PPPoE-Unterstützung zu verhelfen.

2.4.3 Installation

Unter Linux gibt es meist zwei Wege zur Software-Installation: Zum einen kann man die gewünschte Software von den Distributions-CDs einspielen oder aus den Quelltexten übersetzen.

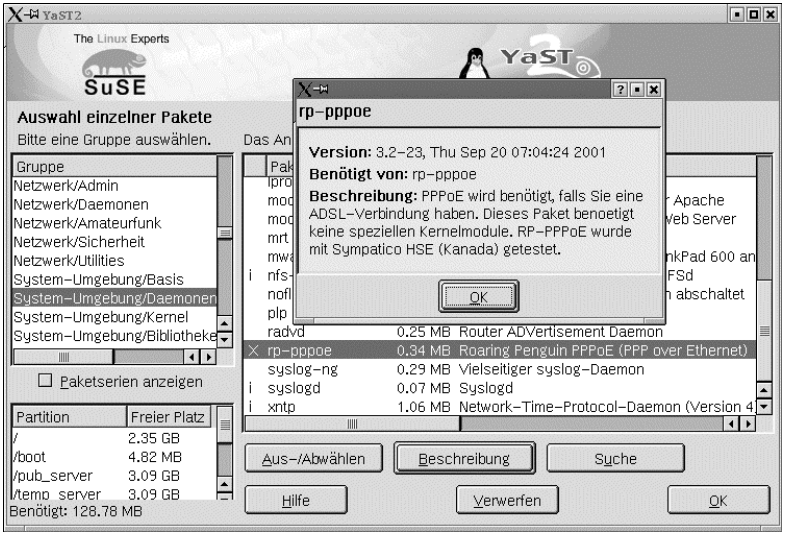
Empfehlenswert ist, die *rp-pppoe*-Komponenten der jeweiligen Distribution einzurichten. In diesem Fall sollte man darauf achten, dass man mindestens die Version 2.8 des Treibers verwendet. Ältere Versionen weisen zum Teil erhebliche Sicherheitslücken auf.

Für die manuelle Einrichtung lädt man sich die aktuelle Version der Treiber herunter, kopiert diese in ein beliebiges Verzeichnis, beispielsweise nach */usr/src/packages/SOURCES*, und entpackt sie dann:

```
tar xvfz rp-pppoe-3.3.tar.gz
```

Anschließend wechselt man in das neu erzeugte Verzeichnis und führt die Befehle für die Übersetzung der Quelltexte und die anschließende Installation aus:

```
./configure
make
make install
```



Bequem: Am einfachsten ist es, wenn man zum Konfigurationswerkzeug der jeweiligen Distribution greift. Hier die Installation des Roaring-Penguin-Treibers mit SuSEs YaST2.

Die Installationsroutine fragt nun noch, ob *adsl-setup* ausgeführt werden soll. Das sollte man an dieser Stelle zunächst verneinen. Denn neben dem PPPoE-Treiber für den Zugang ist auch der PPP-Daemon (*pppd*), der Point-to-Point-Verbindungen in einem Linux-System regelt, erforderlich. Ob dieser auch installiert ist, können Sie mit dem Kommando

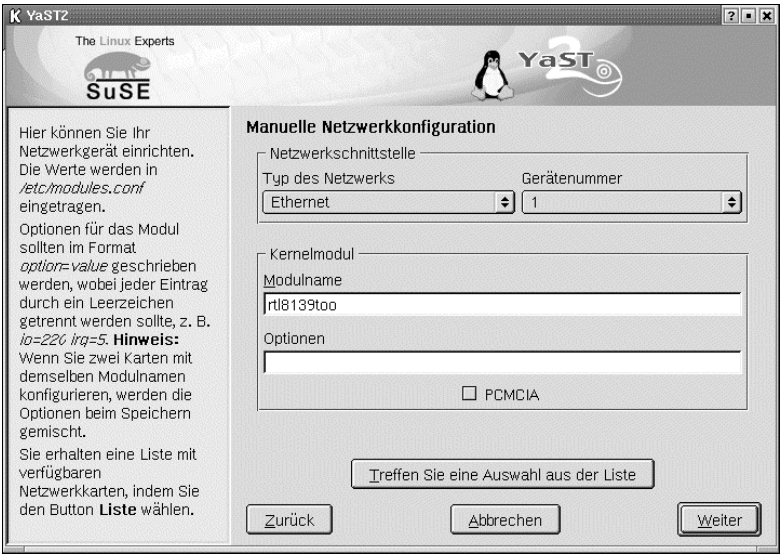
```
pppd --version
```

überprüfen. Folgt dem Befehl eine Fehlermeldung, ist der *pppd* nicht im System installiert. Bei diesem Modul handelt es sich um Standardsoftware, die bei allen Distributionen enthalten ist und sich auch nachträglich installieren lässt.

2.4.4 Treiberaktivierung der Netzwerkkarte

Für den ADSL-Zugang ist eine Netzwerkkarte erforderlich, die über ein 1:1-verschaltetes CAT5-PATCH-Kabel an das ADSL-Modem angeschlossen wird. Über die ADSL-Leitung ist das Modem direkt mit dem so genannten DSLAM in der Vermittlungsstelle verbunden. Im DSLAM hat jeder Kunde einen eigenen Port, mit dem er ständig in Kontakt steht. Aus diesem Grund sollte man das ADSL-Modem nicht ausschalten.

Die eigentliche Netzwerkverbindung erfolgt nicht über Ethernet und damit auch nicht über das Netzwerk-Device der Karte. Die TCP/IP-Netzwerkverbindung wird mit dem *pppd* (Point-to-Point-Daemon) aufgebaut, also über ein PPP-Device. Daher braucht die Netzwerkkarte nicht komplett mit IP-Adresse et ce tera konfiguriert zu werden. Es reicht aus, sie zu aktivieren, damit die PPPoE-Software die Karte ansprechen kann.



Einfach: Unter SuSE-Linux können Sie einen Treiber für Ihre Netzwerkkarte komfortabel manuell installieren.

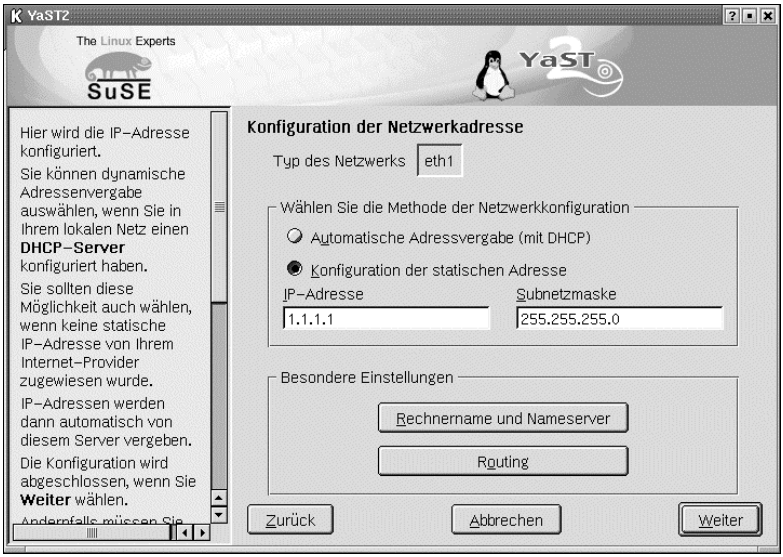
Um die Karte zu aktivieren, ist der Treiber der Netzwerkkarte erforderlich. Alle aktuellen Distributionen binden zusätzliche Treiber als Kernel-Module in das System ein, so dass Sie den Kernel nicht neu kompilieren müssen. Ferner erkennen die meisten Distributionen die Karte schon bei der Installation. Aber auch ein nachträgliches Aktivieren ist kein Problem. Am einfachsten geschieht die Treibereinbindung über das Konfigurationsprogramm der Distribution.

2.4.5 Einbindung der Netzwerkkarte

Nachdem der Treiber aktiviert wurde, ist dem System die Netzwerkkarte bekannt zu machen. Dies erfolgt mit folgendem Befehl über die Kommandozeile:

```
ifconfig eth1 up
```

Dabei wird die zweite Netzwerkkarte im System für den PPPoE-Zugang konfiguriert. Alle weiteren Beispiele beziehen sich ebenfalls immer auf die zweite Netzwerkkarte. Befindet sich nur eine Karte im System, so setzen Sie an Stelle von *eth1* immer *eth0* ein. Um die Karte dauerhaft mit dem distributionseigenen Konfigurationstool zu aktivieren, ist eine Pseudonetzwerkinstallation durchzuführen. Sonst aktiviert Linux die Karte nicht automatisch.



Wahlfrei: Wenn Sie das DSL-Netzwerk-Device einrichten, können Sie eine beliebige Netzwerkadresse eingeben, die in Ihrem Netzwerk noch nicht belegt ist.

Dabei bestehen die meisten Programme auf der Eingabe einer IP-Adresse für die Netzwerkkarte. Da ja diese Adresse für die PPPoE-Verbindung keinerlei Bedeutung hat, können Sie sie beliebig wählen. Sie sollte im eigenen Netzwerk nicht schon vergeben sein, da es sonst zu Konflikten kommt. Mögliche Beispieladressen sind 1.1.1.1 oder 10.10.10.10.

2.4.6 Test der Hardware

Im nächsten Schritt unterziehen Sie die Hardware und die Verkabelung einem ersten Test. Dazu dient der Befehl:

```
pppoe -A -I eth1
```

Er überprüft die Verbindung zum Access-Concentrator. Wenn alles richtig verkabelt und der Port auf Seiten der Telefongesellschaft aktiviert ist sowie die Treiber der Netzwerkkarte korrekt eingebunden sind, erzeugt obiger Befehl folgende Ausgabe:

```
------------------------------------------
Access-Concentrator: OSNC13-nrp3
Got a cookie: 50 64 87 59 6d bf 73 00 1d 67 98 7d c2 d3 a5 32
AC-Ethernet-Address: 00:01:96:99:50:b7
------------------------------------------
```

Erfolgreiche Verbindung: Diese Ausgabe zeigt Ihnen, dass Ihre DSL-Verbindung zum Provider steht.

Ist der Test erfolgreich, führen Sie noch das so genannte Discovery durch. Dazu dient das Kommando:

```
pppoe -T20 -I eth1 -D pppoe.log > /dev/null
```

Die erzeugte Datei *pppoe.log* sollte den Inhalt der nächsten Abbildung zeigen:

```
rp-pppoe-3.3
SENT PPPOE Discovery (8863) PADI sess-id 0 length 4
SourceAddr 00:00:1c:db:da:93 DestAddr ff:ff:ff:ff:ff:ff 01 01 00 00

RCVD PPPOE Discovery (8863) PADO sess-id 0 length 39
SourceAddr 00:01:96:99:50:b7 DestAddr 00:00:1c:db:da:93
01 01 00 00 01 02 00 0b-4f 53 4e 43 31 33 2d 6e
72 70 33 01 04 00 10 50-64 87 59 6d bf 73 00 1d
67 98 7d c2 d3 a5 32

[...]
```

Entdeckung: Wenn das Discovery erfolgreich durchgeführt wurde, sollte Ihre pppoe-Log-Datei wie im Bild dargestellt beginnen.

2.4.7 Einrichten des Zugangs

Nun folgt die Einrichtung des Zugangs. Dazu sind zunächst Benutzerkennung und Passwort festzulegen, um den Point-to-Point-Daemon zu konfigurieren. Unter Linux werden beide in der Datei */etc/ppp/pap-secrets* abgelegt. Einige wenige Provider verwenden nicht das PAP-Protokoll zur Identifizierung des Kunden, sondern das CHAP-Protokoll. Hierbei trägt man die Benutzerkennung und das Passwort in der Datei */etc/ppp/chap-secrets* ein. Um Anpassungen von */etc/ppp/pap-secrets* vorzunehmen, bearbeitet man die Einstellungen mit einem beliebigen Texteditor. Hier ein typisches Beispiel:

```
# Secrets for authentication using PAP
# client        server  secret  IP addresses

# OUTBOUND CONNECTIONS
# Here you should add your PPP Login and PPP password
# to connect to your provider via pap. The * means that the
# entry(login and passoword may be used for ANY host you
# connect to. Thus you do not have to worry about the foreign
# machine name. Just replace password with your password.
#hostname       *       password

"111111111111222222222222#0001@t-online.de" * "geHeim" *
```

Geheim: Ihr Kennwort wird in der Datei pap-secrets abgelegt. Hier ein Beispiel für den T-Online-Zugang.

Der Aufbau der Datei ist denkbar einfach: Kommentare sind mit einer vorangestellten Raute (#) gekennzeichnet. Sie werden ignoriert und dienen lediglich der Übersicht oder als Information. Jede Zeile steht für einen konfigurierten Zugang. So ist es beispielsweise möglich, durch mehrere Zeilen mehrere voneinander unabhängige Zugänge zu konfigurieren. In der ersten Spalte steht die Benutzerkennung, anschließend ein Stern (*), dann das Passwort und zum Schluss wiederum ein Stern (*).

Eine Besonderheit weist der T-Online-Zugang auf: Hier setzt sich die Userkennung aus der Anschlusskennung, der T-Online-Nummer und einer Raute gefolgt von dem Mitbenutzersuffix zusammen (die Raute kann bei einer 12-stelligen T-Online-Nummer entfallen). Bei ADSL nicht zu vergessen: Das @t-online.de am Schluss. Für eine Anschlusskennung von 111111111111, eine T-Online-Kennung von 222222222222 und einen Mitbenutzersuffix von 0001 lautet die Userkennung zum Beispiel 111111111111222222222222#0001@t-online.de.

2.4.8 Konfiguration des pppd

Anschließend erfolgt die Konfiguration des *pppd*. Die zentrale Konfigurationsdatei befindet sich unter */etc/ppp/options*. Sie gilt für alle PPP-Verbindungen. Zusätzlich erlaubt der *pppd* für jeden Zugang eine eigene Konfigurationsdatei. Somit ist es möglich, verschiedene Konfigurationen parallel anzulegen. Diese speziellen Einstellungen befinden sich im Verzeichnis */etc/ppp/peers* und können einen beliebigen Dateinamen erhalten. Kommentare erhalten eine Raute (#) zu Beginn der Zeile. Zunächst werden die Verbindungsoptionen in die Datei */etc/ppp/options* eingetragen:

```
# /etc/ppp/options
# Optionen, die für alle Verbindungen gelten sollen
# Das Passwort soll nicht in /var/log/messages
# mitgeloggt werden - Sicherheitsrisiko!
hide-password
local
noauth
# Exclusive Nutzung des Devices durch den pppd
lock
# Nur zu Testzwecken sind die Meldungen im Syslog interes-
sant
# debug
# Nicht in den Hintergrund „rutschen„ - nur zu Testzwecken
aktivieren!
# nodetach
```

Alle Einstellungen, die nur für den ADSL-Zugang gelten sollen, nehmen Sie in der Datei */etc/ppp/peers/adsl* vor. Ein Beispiel finden Sie unter www.tecChannel.de/download/833/listing_ppp.txt.

2.4.9 Der erste Verbindungstest

Nun wird es spannend. Klappt der Verbindungsaufbau? Diese Frage beantwortet ein erster Test. Dazu aktiviert man in der Datei */etc/ppp/options* die Option *nodetach*. Diese verhindert, dass beim Starten der Software der *pppd* sofort in den Hintergrund rutscht. Dadurch lässt er sich bequem in einem Terminal starten und beenden. Zudem können Sie so alle Ausgaben des *pppd* verfolgen.

Da dieses Verhalten für den späteren Betrieb nicht erwünscht ist, sollten Sie diese Option nach erfolgreichen Tests wieder deaktivieren. Für umfangreiche Statusmeldungen des *pppd* aktiviert man zudem die Option *debug*. Damit Sie während der ersten Verbindungen die Systemmeldungen ständig im Auge behalten können, starten Sie diese in einem Terminal:

```
tail -f /var/log/messages
```

Somit werden alle hinzukommenden Meldungen in der Datei */var/log/messages* sofort angezeigt. In einem weiteren Terminal wird der *pppd* und damit die erste Verbindung via

```
pppd call adsl
```

gestartet. Mit der zusätzlichen Option *call adsl* wird der *pppd* angewiesen, die Konfigurationsdatei */etc/ppp/peers/adsl* einzulesen. Der *pppd* protokolliert die Vorgänge in einer Logdatei (siehe Bild).

```
pppd[1578]: pppd 2.4.1 started by root, uid 0
pppd[1578]: Serial connection established.
pppd[1578]: Using interface ppp0
pppd[1578]: Connect: ppp0 <--> /dev/pts/2
pppoe[1579]: PADS: Service-Name: ''
pppoe[1579]: PPP session is 31959
pppd[1578]: local  IP address 62.226.75.9
pppd[1578]: remote IP address 62.225.254.169
pppd[1578]: primary   DNS address 217.5.115.7
pppd[1578]: secondary DNS address 194.25.2.129
```

Auflösung: Im pppd-Logfile sind zugewiesene IP-Adresse wie auch Nameserver-Adressen zu sehen.

Aus diesen Meldungen kann man die vom Provider zugewiesenen IP-Adressen sowie die aktuell gültigen Nameserver-Adressen ablesen. Die Verbindung steht und lässt sich mittels ping testen:

```
ping -c5 194.25.2.129
```

Der Befehl sendet fünf Testpakete an die IP-Adresse 194.25.2.129, die von dieser auch beantwortet werden sollten. Um die Verbindung zu beenden, verwendet man die Tastenkombination [Strg+C] oder alternativ das Kommando *killall pppd*.

2.4.10 Manuelle Einwahl

Beim Zugang ins Internet unterscheidet man zwischen manueller und automatischer Einwahl. Die manuelle Einwahl ist sinnvoll, wenn man beim Provider keinen Pauschaltarif besitzt. Die automatische Einwahl bietet sich beim Einsatz als Gateway für ein kleines Firmennetz an.

Für die manuelle Einwahl kann man den bereits beschriebenen *pppd*-Aufruf verwenden. Allerdings ist dieser Befehl wenig einprägsam und zudem nur Root vorbehalten. Daher ist es sinnvoll, für die manuelle Einwahl ein Bash-Script

anzulegen, das die verschiedenen, zur Einwahl berechtigten Benutzer beim Verbindungsauf- und abbau aufruft. Die folgenden Zeilen zeigen ein mögliches Script, das unter dem Dateinamen */usr/local/bin/connectadsl* abgelegt wird.

```
#! /bin/sh
# Connectadsl: Skript für die Verbindung zum
# Provider via ADSL/T-DSL
# ----- Einstellungen: -----
# Device für ADSL-Modem: eth0, eth1 etc.
DEVICE=„eth1„
# Options-Datei (muss unter /etc/ppp/peers liegen)
ADSL_FILE=„adsl„
# Pfad zu pppd
PPPD=„/usr/sbin/pppd„
# Pfad zu sudo
SUDO=„/usr/bin/sudo„
# ----- Ende der Einstellungen -----
if test -x $SUDO; then
   if test $UID -ne 0; then
            exec $SUDO $0 $*
   fi
fi
case „$1„ in
stop)
     echo Verbindung beenden
     killall pppd
     ;;
start)
     echo Verbindung starten
     $PPPD call $ADSL_FILE
     ;;
*)
 echo Falscher oder kein Parameter angegeben!
 echo Bitte starten sie $0 mit dem Parameter
</PARAGRAPH>
<PARAGRAPH> echo start oder stop
     ;;
esac
```

Hinzu kommen die Dateirechte für das Script:

```
chmod 755 /usr/local/bin/connectadsl
chown root.root /usr/local/bin/connectadsl
```

2.4.11 Internet für alle

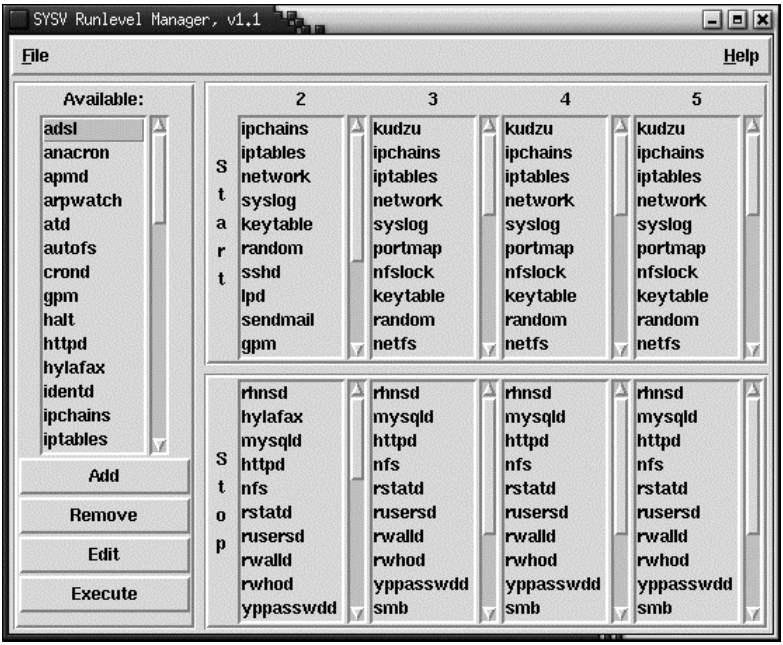
Damit auch andere Benutzer außer Root dieses Script ausführen können, greift man zu dem Utility sudo (www.courtesan.com/sudo/), das in jeder Distribution enthalten ist. Die Konfiguration von sudo erfolgt über den Befehl *root visudo*. Beispielhaft zeigt das nächste Listing, wie sudo dem Benutzer „peter“ die Ausführung des Programms */usr/local/bin/connectadsl* mit Root-Rechten erlaubt.

```
# sudoers file.
# This file MUST be edited with the ´visudo´
# command as root. See the man page for the
# details on how to write a sudoers file.
# Host alias specification

# User alias specification

# Cmnd alias specification

# User privilege specification
root    ALL=(ALL) ALL
peter   ALL=NOPASSWD:/usr/local/bin/connectadsl
```

Analog kann man beliebig viele weitere Zeilen und somit Benutzer und Programme dieser Konfigurationsdatei hinzufügen. Ist der Benutzer *peter* eingeloggt, startet er mit *connectadsl start* beziehungsweise beendet die Verbindung mit *connectadsl stopp*.

2.4.12 Einwahl mit Komfort

Komfortabel wird die Konfiguration erst, wenn man den Linux-Rechner für die automatische Einwahl konfiguriert. Um bereits beim Booten Dial-on-Demand zu aktivieren, installiert man ein Init-Script, das beim Booten beziehungsweise Wechseln in den entsprechenden Runlevel gestartet wird. Außerdem muss für Dial-on-Demand die Option *demand* in der Konfigurationsdatei des *pppd* (*/etc/ppp/peers/adsl*) aktiviert werden. Die Default-Route wird automatisch durch den pppd auf das entsprechende Device gelegt. Anfragen ins Netz lösen automatisch den Verbindungsaufbau aus. Bei den Init-Scripts gehen die verschiedenen Distributionen eigene Wege. SuSE legt diese Scripts unter */etc/init.d* ab, bei Red Hat werden die Init-Scripts unter */etc/init.d/rc.d* abgelegt.



Script-Tuning: Für das Einrichten der Init-Scripts greift man zu Bordmitteln, etwa zu tksysv.

Die Listings hinter den folgenden Links beinhalten mögliche Init-Scripts für SuSE (www.tecchannel.de/download/833/listing_suse.txt) und Red Hat (www.tecChannel.de/download/833/listing_red_hat.txt). Diese Dateien müssen Sie noch mit den korrekten Rechten versehen. Vergessen Sie auch nicht, die Links in den entsprechenden Runlevels korrekt anzulegen.

tecCHANNEL-Links zum Thema	Webcode
Linux 2.4 für den Desktop	a706
Linux als Firewall	a695
Linux als Dial-up-Router	a322

2.5 Sichere Linux-Workstation

Im Internet-Zeitalter ist Rechnersicherheit nicht länger nur ein Thema für Server- und Netzwerkadministratoren. Flatrates und DSL-Anbindungen führen dazu, dass nicht nur klassische Webserver, sondern auch immer mehr Workstations mit semipermanenter oder fester IP-Adresse im Netz der Netze erreichbar sind. Den aus dieser Situation erwachsenden Sicherheitsproblemen sind die wenigsten Client-Betriebssysteme gewachsen. Dieser Workshop zeigt, wie Sie mit den eingebauten Sicherheitsmerkmalen von Linux Ihre Linux-Workstation zuverlässig gegen unautorisierte Zugriffe schützen.

In dramatischer Weise führt das der Zwischenfall mit dem Code-Red-Wurm vor Augen. Am 19. Juli 2001 um etwa 10 Uhr UTC tauchte im Netz die Random-Seed-Variante (CRv2) von Code Red auf. Anders als ihr zwei Tage vorher aufgetretener Vorgänger CRv1 griff sie nicht nur eine begrenzte Anzahl vorgegebener IP-Adressen an. Stattdessen versuchte CRv2 von den befallenen Rechnern aus über massive Scans alle erreichbaren Systeme zu infizieren.

Innerhalb von 24 Stunden fielen dem Wurm knapp 360.000 Rechner zum Opfer, die Infektionsrate erreichte in Spitzen bis zu 2000 Hosts pro Minute. Von diesen Rechnern befanden sich rund zwei Drittel in Providernetzen, waren also zum Großteil private Clients. Von den in Deutschland betroffenen 11.700 Maschinen stammten 5.500 aus t-dialin.net - also dem Einwahlnetz der Telekom!

Diese von der CAIDA-Studiengruppe (www.caida.org) am Supercomputer Center der UCSD erhobenen Daten führen zu klaren Schlussfolgerungen: „Der Angriff zeigt, dass auch die Rechner von Privatleuten und kleinen Gewerbetreibenden das Funktionieren des Internet erheblich beeinflussen“, warnen die Autoren einer ausführlichen Code-Red-Studie. Und weiter: „Wer seinen Rechner nicht vernünftig absichert, bringt das gesamte Netz in Gefahr.“

2.5.1 Linux vs. Windows

Code Red war ein Windows-Wurm, der ein Sicherheitsloch des Microsoft IIS (die .ida Vulnerability) ausnutzte. Das bedeutet jedoch keineswegs, dass Linux-Rechner quasi „by design“ gegen solche Gefahren gefeit wären. Ebenso gut hätten die Autoren von Code Red Linux-spezifische Schwächen als Basis ihres Schädlings verwenden können, wie die in etwa parallel zur MS-.ida-Lücke entdeckte.

Der große „Erfolg“ von Code Red und ähnlicher Windows-Schadprogramme ist eher symptomatisch dafür, dass Microsoft-Produkte ihren Benutzer nicht gerade in idealer Weise bei der Abwehr von Bedrohungen unterstützen. Dies beweist nicht zuletzt die Tatsache, dass selbst windowsupdate.microsoft.com zu den Code-Red-infizierten Systemen zählte.

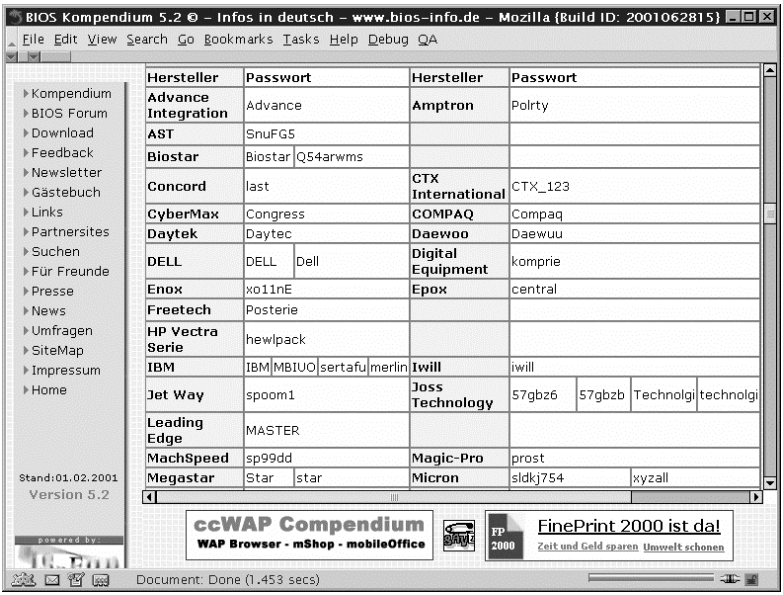
Das in bester Unix-Manier von vornherein als Netzwerk- und Multi-User-Betriebssystem konzipierte Linux glänzt mit einer ganzen Reihe eingebauter

Sicherheitsmechanismen. Wir zeigen auf den folgenden Seiten, wie sich in Netzwerken - ob LANs oder dem Web - eingesetzte Rechner mit einfachen Bordmitteln gegen unberechtigte Nutzung abschotten lassen.

2.5.2 Sichern des Bootvorgangs

Unabhängig vom Betriebssystem beginnt der Datenschutz bei jedem Rechner schon in den BIOS-Settings: Es gilt, die Maschine gegen das Booten durch Unbefugte abzusichern. Andernfalls besteht Gefahr, dass ein Angreifer mit physischem Zugang zum Computer sich Daten aus dem Filesystem abgreift oder auch welche einschmuggelt, wie etwa Trojaner. Die Tatsache, dass ein bootfähiges System samt aller Werkzeuge bequem auf eine Diskette passt, verschärft diese Gefahrenmomente im Fall von Linux zusätzlich.

Daher schalten Sie am besten im BIOS ihres Rechners die Möglichkeit zum Booten von Disketten- und CD-ROM-Laufwerk aus. Zusätzlich schützen Sie das BIOS durch Vergabe eines Passworts vor der Manipulation durch Unbefugte. In vielen BIOS-Varianten lässt sich zusätzlich auch ein User-Password vergeben, ohne dessen Eingabe ein Starten des Rechners von jeglichem Bootmedium unterbunden wird.



Hintertürchen: Selbst für die obskursten BIOS-Varianten finden sich im Web die entsprechenden Master-Passwörter.

Hundertprozentige Sicherheit erreichen Sie damit allerdings nicht. Einem potenziellen Datendieb verbleibt immer noch die Möglichkeit, den Rechner zu öffnen und das BIOS via Jumper oder schlicht durch Entnahme der Pufferbatterie zu resetten. Dagegen schützen nur Maßnahmen wie abschließbare Gehäuse. Viele BIOS-Hersteller haben zudem Master-Passwörter eingebaut, die auf einschlägigen Websites (www.bios-info.de) zur allgemeinen Verwendung bereitliegen.

2.5.3 Schwachstelle LILO

Die Beschränkung der Bootdevices und die dazugehörigen Schutzmaßnahmen am BIOS bieten allerdings unter Linux noch keinen ausreichenden Schutz gegen das Hochfahren des Rechners durch Unbefugte. Das gilt auch, wenn der Angreifer tatsächlich nur von der Festplatte booten kann. Der Grund dafür liegt in der Fähigkeit des LILO, beim Booten Systemparameter zu übergeben.

Die simpelste Möglichkeit, innerhalb weniger Sekunden vollen Zugriff auf einen Linux-Rechner zu erlangen, besteht in der Übergabe des Bootparameters init. Das am LILO-Bootprompt eingegebene Kommando

```
Linux boot: linux init=/bin/sh
```

verschafft dem Angreifer eine Shell mit kompletten Root-Rechten.

Dieser Gefahr lässt sich auf zweierlei Art begegnen: Zum einen gilt es zu verhindern, dass der Angreifer per Ctrl-Alt-Del an den Bootprompt herankommt. So können nur noch angemeldete Benutzer über *init*, *shutdown* oder *reboot* den Rechner neu starten. Zum anderen muss die unauthentifizierte Eingabe von Bootparametern unterbunden werden.

2.5.4 Ctrl-Alt-Del abschalten

Auf die finale Geierkralle Ctrl-Alt-Del lässt sich unter Linux gut verzichten. Zur Beendigung von Prozessen oder des Betriebssystems bietet sich eine Vielzahl von Möglichkeiten an, wie etwa *kill*, *top*, *init* oder *shutdown*. Diese Konsolenbefehle bieten gegenüber der aus der Microsoft-Welt gewohnten Tastenkombination einen entscheidenden Vorteil: Sie lassen sich nur von dazu autorisierten Benutzern anwenden.

Praktischerweise stellt das Abschalten der Tastenkombinationen den Anwender vor keinerlei größere Herausforderungen. In der Datei */etc/inittab* findet sich eine Zeile etwa des Inhalts (je nach Distribution):

```
ca::ctrlaltdel:/sbin/shutdown -t3 -r now
```

Hier gibt der letzte Part an, was das Betriebssystem beim Drücken der Tastenkombination veranlassen soll. Statt des per Default vorgesehenen Shutdown kann der Anwender hier auch jedes andere Kommando eintragen. Im einfachsten Fall quittiert man Ctrl-Alt-Del mit einer Konsolenmeldung:

```
ca::ctrlaltdel:/bin/echo -e ´\n\nNix da!\n´
```

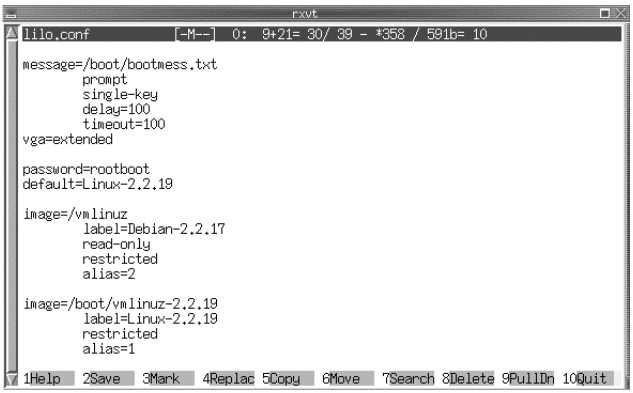
Alternativ lässt sich etwa ein vorbereitetes Script aufrufen, das eine E-Mail an root absetzt und den Reboot-Versuch in einer Logdatei protokolliert.

2.5.5 LILO absichern

Gelangt ein Angreifer trotz aller Vorsichtsmaßnahmen bis an den Bootprompt, kann man ihm auch hier noch das Handwerk legen. Dazu gilt es, die LILO-Konfigurationsdatei zu modifizieren. Sie findet sich im Verzeichnis */etc* unter dem Namen *lilo.conf*. Für die Authentifizierung beim Hochfahren des Rechners stellt LILO die beiden Schlüsselwörter *password* und *restricted* bereit:

```
password=mein_boot_passwort
restricted
```

Für *password* kann man ein beliebiges Kennwort vergeben, das der Bootloader vor dem Laden des Kernel-Image abfragt. Mit *restricted* schränkt man diesen Schutz dahingehend ein, dass LILO das Passwort nur bei der Angabe zusätzlicher Bootparameter einfordert.



Passwortschutz für LILO: Der Zugang lässt sich auch separat für jedes Image einschränken.

Um generell jeden Bootvorgang mit einem Passwortschutz zu bewehren, tragen Sie die beiden Parameter vor der Auflistung der Kernel-Images in *lilo.conf* ein:

```
(...)
password=rootboot
restricted
#
image=/vmlinuz
(...)
#
image=/boot/vmlinuz-2.2.19
(...)
```

2.5.6 Kernel-Image und lilo.conf schützen

Alternativ können Sie jedes Image einzeln absichern. Im folgenden Beispiel erfordert das Laden des Default-Kernels in jedem Fall eine Authentifizierung. Beim Booten des speziell erstellten Maintenance-Kernels verzichtet LILO auf die Kennworteingabe - es sei denn, Sie geben zusätzliche Parameter an.

```
(...)
image=/vmlinuz
password=aktuell
(...)
image=/boot/vmlinuz-2.2.19
password=wartung
restricted
(...)
```

Vergessen Sie nach der Änderung der *lilo.conf* nicht, zur Übernahme der Einstellungen LILO aufzurufen. Um die im Klartext sichtbaren LILO-Passwörter zu schützen, beschränken Sie alle Zugriffsrechte für die LILO-Konfigurationsdatei auf root. Da *lilo.conf* per Default ohnehin root als Owner und Group hat, genügt dazu in der Regel ein schlichtes

```
chmod 600 /etc/lilo.conf.
```

Zusätzlichen Schutz der Konfigurationsdatei erreichen Sie durch den Befehl

```
chattr +i /etc/lilo.conf
```

Das File kann anschließend durch niemanden mehr verändert, umbenannt, gelöscht oder verlinkt werden. Lediglich root ist in der Lage, diesen Zustand bei Bedarf per `chattr -i /etc/lilo.conf` wieder aufzuheben.

2.5.7 Passwortschutz

Die besten Schutzmaßnahmen gegen das Booten durch nicht autorisierte Benutzer helfen allerdings nicht, wenn Sie für root- und Benutzeraccounts zu schwache Passwörter vergeben. Vermutlich kennen Sie die alte Faustregel: Passwörter müssen mindestens acht Zeichen lang sein, wobei sie in guter Mischung Klein- und Großbuchstaben sowie Ziffern und Sonderzeichen enthalten sollten. Und vermutlich beherzigen Sie sie nicht.

Zugegeben, wüste Ziffern-/Zeichenkombinationen sind schwer zu merken. Dennoch taugen der Vorname der Ehefrau, die private Telefonnummer oder das Geburtsdatum nicht als Passwort. Zu leicht lassen sich derartige Merkmale schon von Fremden über Telefonbücher, Firmenunterlagen, die eigene Website oder Social Engineering herausfinden. Ihre Kollegen dagegen brauchen vermutlich nur ins firmeneigene Intranet zu schauen, um solche Informationen zu eruieren. Der Rest ist dann nur noch eine Frage der Zeit.

Einen möglichen Ausweg aus dem Dilemma zwischen gut zu merkendem und schwer zu knackendem Passwort bieten „semiprivate“ Daten. Solche also, die Sie zwar im Kopf haben, die aber für andere schwer herauszubekommen sind. Der Autor etwa verwendet eine deutlich mehr als zehnstellige, des Öfteren wechselnde Variation seiner ehemaligen Bundeswehr-Personalkennziffer, gewürzt mit Sonderzeichen.

Weniger militante Zeitgenossen können als Basis des Passworts beispielsweise auf das Datum der Französischen Revolution, die Telefonnummer der Schwiegermutter oder einen beliebigen Merksatz (Strickmuster: b00t?->0n1y_R00t!) ausweichen.

2.5.8 MD5 und Shadow

Ursprünglich benutzte Linux das Standard-Unix-Authentifizierungsverfahren, bei dem simple Hashes der Passworte direkt in */etc/passwd* gespeichert und beim Login mit der Benutzereingabe verglichen wurden. Die steigende Rechenleistung moderner Computer machte dieses Verfahren jedoch mit der Zeit allzu anfällig gegen Brute-Force-Attacken. Aus diesem Grund setzen moderne Linux-Distributionen zum Schutz der Passwörter MD5-Verschlüsselung und ein Shadow-File ein.

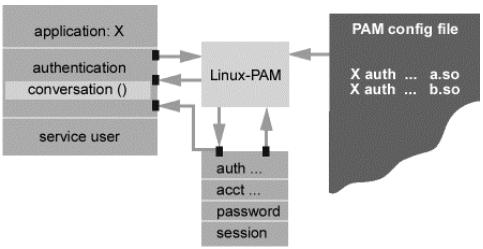
MD5 bietet nicht nur einen relativ sicheren Hashing-Algorithmus mit 128-Bit-Hashwert, sondern erlaubt auch Passwörter mit mehr als acht Zeichen Länge. Shadow verlagert die Speicherung der Passwörter aus der Datei */etc/passwd* nach /etc/shadow, das nur für root zugänglich ist. In */etc/passwd* findet sich dann statt des Passwort-Hashes lediglich ein x:

```
user:x:Default User,,,,:/home/user:/bin/bash
```

Es verweist darauf, dass das eigentliche Passwort in */etc/shadow* lagert. Die durch MD5-Hashing und das Shadow-File gesteigerte Passwortsicherheit ist durch PAM noch weiter zu steigern.

2.5.9 PAM

Das Akronym PAM (www.kernel.org/pub/linux/libs/pam/) steht für Pluggable Authentication Modules. Diese Libraries dienen als Wrapper für Sicherheitsfunktionen, die sie allen PAM-fähigen Programmen zur Verfügung stellen. Die entsprechenden Shared-Object-Files finden sich in */lib/security*. Der Hauptvorteil des PAM-Mechanismus: Um einen geänderten oder neu entwickelten Authentifizierungsmechanismus zu implementieren, genügt es, ein entsprechendes PAM bereitzustellen. Das früher notwendige Rekompilieren aller betroffenen Programme erübrigt sich damit.



PAM-Architektur: Die Bibliotheken dienen als Schnittstelle zwischen Applikation und Authentifizierungsprozess.

PAMs gliedern sich in die vier Typen *auth*, *account*, *session* und *password*. Für die eigentliche Authentifizierung zeichnen die PAMs der Kategorie *auth* verantwortlich. Sie stellen (in der Regel per Passwortabfrage) sicher, dass der Benutzer auch der ist, für den er sich ausgibt. Die account-PAMs erweitern auf Basis der Benutzerkonten die Fähigkeiten der *auth*-Module. So lässt sich etwa der Benutzerzugriff je nach Tageszeit, Systemressourcen oder Standort des Benutzers (Konsole, IP-Adresse des Rechners) beschränken.

Die Module der Gruppe *session* regeln zusätzliche Aufgaben, die rund um die eine Benutzersitzung anfallen. Dazu zählen etwa das Mounten von Laufwerken oder die Protokollierung von Dateizugriffen. Zum Ändern von Authentifizierungs-Tokens wie beispielsweise des Passworts dienen *password*-PAMs. Üblicherweise gehört zu jedem Challenge/Response-basierten *auth*-Modul auch ein entsprechendes *password*-PAM.

2.5.10 PAM-Konfiguration

Jedes PAM kann nicht nur in eine, sondern auch in mehrere der Funktionsgruppen fallen. So stellt etwa *pam_unix.so*, das Standard-Unix-Authentifizierungsmodul, Funktionen für alle vier Aufgabengruppen zur Verfügung.

Praktisch alle aktuellen Linux-Distributionen unterstützen PAM, Caldera, Debian, Red Hat und SuSE bereits seit einiger Zeit. Die dazugehörigen Konfigurationsdaten finden sich im Verzeichnis */etc/pam.d*, gelegentlich auch in der Datei */etc/pamd.conf*.

Im PAM-Verzeichnis liegt für jeden konfigurierten Dienst ein eigenes File, wie etwa *passwd* oder *login*. Die Datei *other* dient als Fallback für nicht speziell eingerichtete Services und greift in der Regel auf *pam_unix.so* zurück. Nutzt die Distribution eine */etc/pamd.conf*, steht dort der Name des zu konfigurierenden Dienstes am Beginn jeder Zeile, das Schlüsselwort *OTHER* kennzeichnet die Default-Einstellungen. In beiden Fällen gehorchen die einzelnen Einträge der Syntax:

```
Modultyp Kontrollflag Modul Parameter
```

Beim Modultyp handelt es sich um die bereits angesprochenen Kategorien *auth*, *account*, *session* und *password*. Für jeden dieser Typen lassen sich Prüfungen durch mehrere Module einrichten, die in der Reihenfolge ihres Auftretens abgearbeitet werden. Dabei gibt das Kontrollflag an, welche Gewichtung dem jeweiligen Modul zukommt. Hier kennzeichnen *required* oder *requisite*, dass beim Fehlschlagen der Prüfung die Authentifizierung insgesamt als gescheitert gilt. Nach der erfolgreichen Absolvierung eines mit *sufficient* gekennzeichneten Moduls gilt die Authentifizierung als abgeschlossen, weitere Prüfungen unterbleiben. Als *optional* gekennzeichnete Module dienen der Abarbeitung von Aufgaben, die für die eigentliche Authentifizierung nicht kritisch sind.

Module lassen sich wahlweise mit vollem Pfadnamen oder relativ zum Basisverzeichnis */usr/lib/security* angeben. Fast alle PAM verarbeiten, meist getrennt nach Modultyp, verschiedene Parameter. Eine ausführliche Beschreibung aller Werte finden Sie im Linux-PAM System Administrators Guide (www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam.html).

2.5.11 pam_cracklib

In Sachen Passwortsicherheit interessieren uns vor allen Dingen die Einträge in den Dateien */etc/pam.d/passwd* und */etc/pam.d/login*. Die erste steuert das Verhalten beim Ändern von Passwörtern, die zweite zeichnet für die Vorgänge beim Login verantwortlich. Sowohl *passwd* als auch *login* sollten folgende zwei Zeilen (in dieser Reihenfolge) enthalten:

```
# Zeile 1
password required /lib/security/pam_cracklib.so retry=
   3 minlen=8 difok=3
# Zeile 2
password required /lib/security/pam_unix.so nullok
   use_authtok md5 shadow
```

Damit erzwingen Sie über *pam_cracklib.so* bei allen Accounts die Verwendung ausreichend starker Passwörter.

Die Bibliothek überprüft alle neuen Passwörter zunächst anhand eines Dictionary auf gängige und somit leicht zu erratende. Fällt das Passwort nicht in diese Rubrik, testet *pam_cracklib* zusätzlich folgende Ausschlusspunkte:

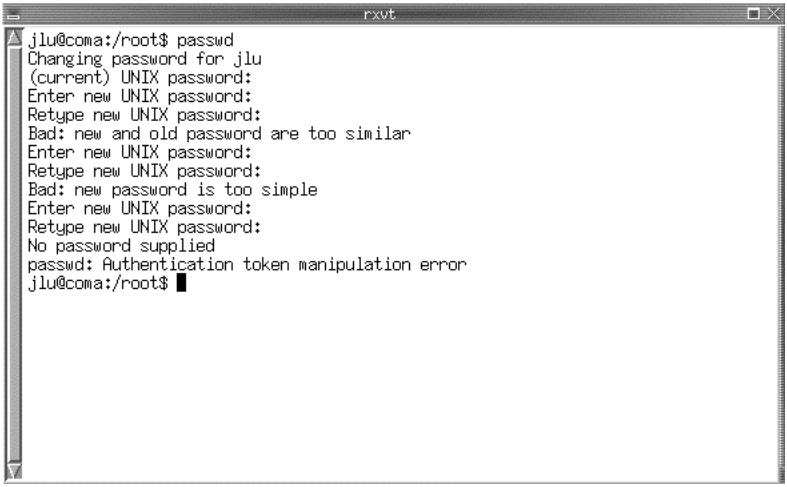
- Hat das Passwort die notwendige Mindestlänge (minlen)?
- Wurden wenigstens difok oder ersatzweise mindestens 50 Prozent der Zeichen im Passwort geändert?
- Wurde das Passwort schon einmal benutzt?
- Ist das Passwort ein Palindrom eines schon vorhandenen Passworts (beispielsweise regen_nie? statt ?ein_neger)
- Handelt es sich um ein Passwort, bei dem lediglich die Groß-/Kleinschreibung gewechselt wurde?
- Handelt es sich um die Rotation eines alten Passworts (etwa bcde statt abcd)?

Nur wenn das neue Passwort alle Hürden nimmt, reicht *pam_cracklib* es an *pam_unix* weiter. Dabei erzwingt *use_authtok* die Verwendung des von *pam_cracklib* getesteten Tokens. Andernfalls fordert *pam_cracklib* den Benutzer bis zu drei Mal (*retry=3*) zur Eingabe eines stärkeren Passworts auf.

2.5.12 cracklib-Dictionary

Beim Ausschluss zu schwacher Passwörter stützt sich *pam_cracklib.so* auf die Library *cracklib*, die ähnlich wie Brute-Force-Crackprogramme operiert: Sie stützt sich auf eine Wortliste. Diese dient hier allerdings nicht zum Abklopfen von Accounts, sondern soll allzu leicht zu erratende Passwörter ausschließen. Ein vorgefertigtes Dictionary findet sich im Verzeichnis */usr/lib* in drei Dateien namens *cracklib_dict* mit den Endungen *.hwm*, *.pwd* sowie *.pwi*. Diese Files

lassen sich über die mit *cracklib* gelieferten Dienstprogramme aus ASCII-Wortlisten mit je einem Eintrag pro Zeile generieren. Als Standardwortliste dient dabei die Datei */usr/dict/words*.



Rausgeflogen: pam_cracklib unterbindet die Vergabe zu schwacher Passwörter.

Je nach Distribution gestaltet sich die Ergänzung des Dictionary um eigene Einträge mehr oder weniger aufwendig. So erstellt Debian über ein cron-Script das Wörterbuch täglich aus den Wortlisten in den Verzeichnissen */usr/dict* und */usr/share/dict* neu. Bei Red Hat Linux oder SuSE dagegen muss der Administrator diese Arbeit manuell erledigen. SuSE liefert immerhin ein Script namens *create-cracklib-dict*, das die notwendigen Aufrufe für eine als Parameter anzugebende ASCII-Wortliste automatisch erledigt. Bei Red Hat lautet die entsprechende Kommandozeile:

```
/usr/sbin/mkdict /usr/dict/words
| /usr/sbin/packer /usr/lib/cracklib_dict
```

Dabei entfernt *mkdict* Doubletten und Kontrollzeichen aus der als Parameter angegebenen Liste und schreibt das Ergebnis sortiert auf die Standardausgabe. Das Utility *packer* erwartet eine derart formatierte Liste an der Standardeingabe und erzeugt daraus die drei von cracklib benötigten Files mit dem als Parameter angegebenen Namen und den Endungen *.hwm*, *.pwd* sowie *.pwi*

2.5.13 pam_securetty und pam_nologin

Über PAMs können Sie nicht nur die Vergabe sicherer Passwörter erzwingen, sondern auch das Login selbst zusätzlich schützen. Das gilt speziell für den kritischen root-Account. Dazu sollte */etc/pam.d/login* gleich zu Anfang die folgenden zwei Zeilen enthalten:

```
# Zeile 1
auth required /lib/security/pam_securetty.so
# Zeile 2
auth required /lib/security/pam_nologin.so
```

Das Modul pam_securetty beschränkt das Login von root auf jene Konsolen, die ausdrücklich in der Datei */etc/securetty* erfasst sind. Auf den meisten Workstations lässt sich der Zugang problemlos auf die lokalen Terminals *tty0* bis *tty12* beschränken. Serielle Terminals (ttyS) oder die devfs-Devices *vc/1* bis *vc/11* benötigt man nur in den wenigsten Konfigurationen.

Wollen Sie zu Wartungsarbeiten oder aus Sicherheitsgründen das Login vorübergehend auf root beschränken und andere Nutzer aussperren, bietet dazu *pam_nologin* eine komfortable Möglichkeit. Findet sich in */etc* das File *nologin*, weist das System die Anmeldeversuche aller User mit Ausnahme von root zurück. Dabei gibt es den Inhalt von nologin auf dem Bildschirm aus.

2.5.14 pam_wheel

Nicht nur root kann Systembefehle mit weit reichenden Folgen absetzen. Dieselbe Möglichkeit steht auch allen Benutzern offen, die das Kommando *su* ausführen. Deshalb sollten Sie diesen Anwenderkreis mit Hilfe des Moduls *pam_wheel* begrenzen. Die Bezeichnung *wheel* stammt aus der BSD-Welt, wo sich nur Benutzer der gleichnamigen Gruppe per *su* root-Rechte aneignen können. Unter Linux beschränkt *pam_wheel* den Zugriff per Default auf die Gruppe wheel oder ersatzweise jene mit der Group-ID 0 (root).

Sollen nur root und die Mitglieder einer eigens eingerichteten Gruppe suser Zugriff auf su haben, müssen die ersten zwei Zeilen von */etc/pam.d/su* folgendermaßen aussehen:

```
auth sufficient pam_rootok.so
auth required pam_wheel.so
```

Hier erlaubt die erste Zeile dem Benutzer root, ohne Angabe des Passworts die Identität anderer User zu übernehmen.

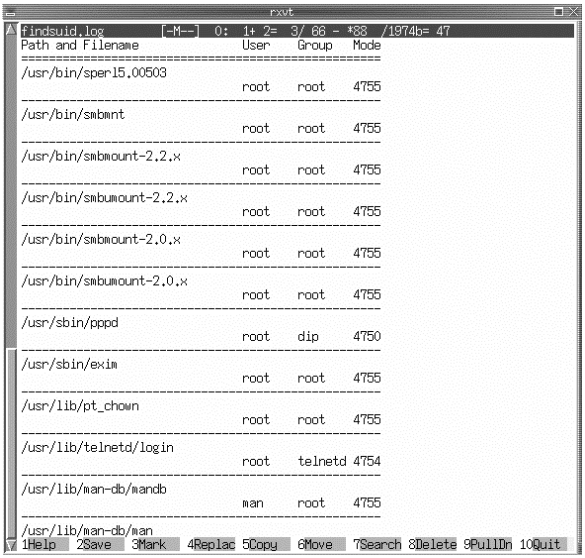
Soll statt *wheel* eine andere Benutzergruppe in den Genuss des root-Zugriffs per su kommen, ergänzen Sie die zweite Zeile um zwei Parameter:

```
auth required pam_wheel.so deny group=andere_gruppe
```

Der Parameter *deny* sperrt zunächst den Zugriff der Gruppe *wheel*. Der Ausdruck *group=andere_gruppe* räumt dieses Recht stattdessen der Gruppe mit dem Namen *andere_gruppe* ein.

2.5.15 SUID root beschränken

Linux kennt jedoch noch eine weitere Methode, nach der normale Benutzer Programme quasi als root ausführen können. Dies betrifft alle ausführbaren Dateien, die auf SUID root gesetzt sind. Der Besitzer solcher Dateien ist root, *ls -l* zeigt als Ausführungsrecht für den Besitzer ein *s* statt eines *x* an.



Gedächtnisstütze: Die Ausgabe von findsuid sollten Sie in einer Datei speichern, um später Änderungen wieder rückgängig machen zu können.

Unter Angabe des oktalen Werts finden Sie mit einem schlichten *find / -perm -4000 –xdev* schnell alle entsprechenden Files auf dem Rechner. Je nach Distribution handelt es sich dabei um mehrere Dutzend Dateien - speziell SuSE geht mit dem SUID-Flag recht freizügig um. Die wenigsten User allerdings benötigen

jemals SUID-root-Zugriff auf alle hier angebotenen Files, zudem kann man damit einigen Unsinn anstellen. Daher sollten Sie nur bei den notwendigsten Dateien das SUID-root-Flag gesetzt lassen. Dazu zählen neben *passwd*, *ping*, *traceroute* und *su* gegebenenfalls *Xwrapper* für X11 sowie einige Files aus der KDE-Suite.

Um Ihnen die Arbeit zu erleichtern, bieten wir Ihnen das kleine Script findsuid (siehe dazu auch Bild vorherige Seite) zum Download (www.tecChannel.de/download/720/findsuid). Es durchsucht das lokale Filesystem nach Dateien mit gesetztem SUID-root-Flag und schreibt eine formatierte Liste dieser Files samt Pfad, Owner, Group und Berechtigungen auf die Standardausgabe.

Speichern Sie diese Liste in einer Datei ab, bevor Sie Änderungen vornehmen. Bei Bedarf können Sie dann später jederzeit wieder den Originalzustand herstellen.

2.5.16 Serverdienste

Der Internet-Superserver *inetd* regelt die Behandlung von eingehenden Netzwerkverbindungen. Trifft eine Verbindungsanfrage an einem von *inetd* betreuten Port ein, leitet der Daemon den Request an einen Dienst namens *tcpd* weiter. Dieser entscheidet anhand der Einstellungen in den Files */etc/hosts.deny* sowie */etc/hosts.allow*, ob es sich um einen zulässigen Zugriff handelt. Ist das der Fall, startet *tcpd* den zugehörigen Serverprozess, der nun die Anfrage bedienen kann. Diesen Mechanismus bezeichnet man als TCP Wrapping.

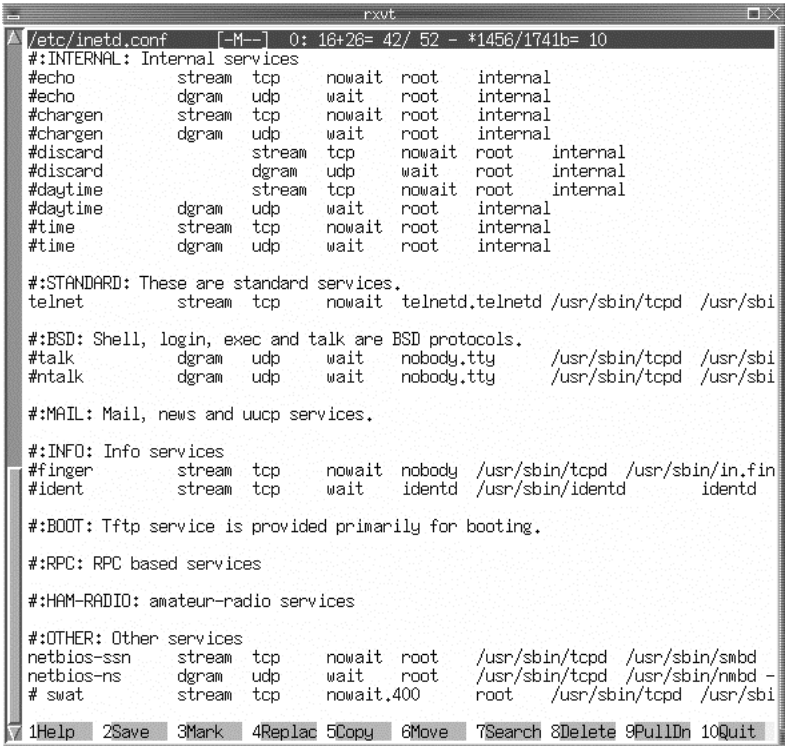
Nun benötigen typische Workstations in den seltensten Fällen einen der Serverdienste, für die *inetd* verantwortlich zeichnet. Daher bestünde eigentlich kein Grund, den Daemon auf Clients einzurichten. Dennoch installieren die meisten älteren Linux-Distributionen sowohl bei Default- als auch Workstation-Installationen nicht nur *inetd*, sondern auch Dienste wie FTP, Telnet, Finger und andere mehr.

Dies produziert unnötige Scherheitslücken: Die installierten Dienste stehen nicht nur über lokale Netzwerkverbindungen, sondern auch via Internet zur Verfügung. Für Rechner, die per konventionellem Dial-up ans Web angebunden werden, hält sich das Risiko wegen der nur zeitweilig zugeteilten IP-Adresse und der relativ kurzen Verbindungszeiten in Grenzen.

Anders verhält es sich bei Flatrate- und DSL-Verbindungen: Hier ist der Rechner stets erreichbar, die IP bleibt zumindest über mehrere Stunden konstant. Neuere Distributionen wie Red Hat 7.3, SuSE 8.0 oder Mandrake 8.2 tragen dieser Tatsache Rechnung, indem sie bei Workstation-Installationen von vornherein auf die Einrichtung des Internet-Superservers verzichten.

2.5.17 inetd.conf durchforsten

Hat das Setup auf Ihrem System *inetd* installiert, lässt sich auch dies beheben. Durch Editieren seiner Konfigurationsdatei können Sie dem Superserver auf simple Weise untersagen, unerwünschte Verbindungen nach außen zuzulassen. Die Einstellungen für *inetd* lagern, wie sich unschwer erraten lässt, in der Datei */etc/inetd.conf*.



Alles auskommentiert: Diese bearbeitete Variante der inetd.conf lässt nur die Dienste Telnet und Samba zu, alle übrigen sind nicht aktiv.

Für jeden Dienst findet sich dort eine Zeile, die mit dem Servicenamen beginnt. Kommentieren Sie mit einem Doppelkreuz eine Zeile aus, bleibt der entsprechende Dienst in Zukunft geschlossen.

Um die Änderung sofort zu übernehmen, starten Sie anschließend *inetd* über */etc/init.d/inetd restart* oder *killall -HUP inetd* neu.

Auf den meisten Linux-Workstations können Sie bedenkenlos sämtliche Einträge deaktivieren. Zu den Diensten, die Sie mit an Sicherheit grenzender Wahrscheinlichkeit nicht brauchen, zählen *chargen*, *daytime*, *discard*, *echo*, *imap*, *ntalk*, *talk*, *tftp*, *time*, *pop* und *uucp*. Als geradezu gefährlich erweisen sich *ident* und *finger*, die einen Angreifer zuvorkommenderweise mit umfangreichen Informationen über die Benutzer des Rechners bedienen.

2.5.18 Serverdienste abschalten

Vor allem unerfahrene Benutzer binden viele Daemons eher „versehentlich" ein: So etwa den DNS-Dienst named/bind, den DHCP-Server *dhcpd*, den News-Server innd oder den SMB-Server Samba.

Alle vier benötigen Sie nur, wenn Sie anderen Rechnern entsprechende Dienste zur Verfügung stellen wollen. Zur Nutzung von News, DNS und DHCP und Windows-Shares als Clients sind sie überflüssig. Ähnliches gilt für die NFS- und Yellow-Pages-Server nfsd und ypbind, die zudem fast ausschließlich in Unix-Netzwerken zum Einsatz kommen.

Bei FTP und SMTP handelt es sich um klassische Serverdienste, die auf Workstations nichts verloren haben. Sie bieten zudem funktionsbedingt derart viele Angriffspunkte, dass sie nur auf speziell gesicherten Maschinen - etwa in einer DMZ - laufen sollten.

2.5.19 Samba, Apache und Telnet

Auch auf einer Workstation möchten Sie möglicherweise gewisse Serverdienste bereitstellen. Als klassische Kandidaten fallen hier SMB- und HTTP-Services sowie Telnet für den Wartungszugriff aus der Ferne an.

Samba benötigen Sie, um anderen Maschinen im Netz SMB-File- und Printservices bereitzustellen. Die Samba-Daemons *smbd* und *nmbd* müssen Sie nicht zwangsläufig via *inetd* starten: Wollen sie ausschließlich CIFS-Dienste offerieren, lassen sich beide Services auch direkt als Daemons starten. Dies beschleunigt sogar den Zugriff. Im Regelfall wird zusammen mit Samba auch Swat installiert, das die Remote-Konfiguration des SMB-Servers per Webbrowser erlaubt. Diesen Dienst sollten Sie unbedingt deaktivieren, wenn Sie ihn nicht dringend benötigen.

Dem Webserver Apache kann man kaum etwas Schlechtes nachsagen. Im Gegensatz zu Samba läuft er nicht als root, sondern nutzt einen eigenen Account. Sollte also jemand über *httpd* Zugriff auf die Maschine erlangen, geschieht dies wenigstens nicht im Root-Kontext. Dennoch ist Vorsicht geboten, da Angriffstools gerne den HTTP-Port 80 als mögliche Einfallspforte in den Rechner abklopfen.

Bei Telnet handelt es sich um einen sehr nützlichen Dienst, allerdings läuft die Kommunikation unverschlüsselt ab - inklusive der Authentifizierung mit Username und Passwort. Daher sollten Sie auf Telnet nach Möglichkeit verzichten oder stattdessen OpenSSH (www.openssh.org) einsetzen. Mit OpenSSH erhalten Sie neben Client und Server für verschlüsselte Remote-Verbindungen mit scp und sftp auch gleich noch sichere Varianten für Remote Filecopy und FTP.

2.5.20 Dienstzugriff einschränken

Den Zugriff auf die in */etc/inetd.conf* aufgeführten Dienste können Sie über die beiden hosts_access-Dateien */etc/hosts.allow* und */etc/hosts.deny* auf Basis von IP-Adressen steuern. In beiden Dateien gehorchen die Einträge der Syntax:

```
Liste von Daemons : Liste von Clients : optionales
Shell-Kommando
```

Als Trennzeichen innerhalb der Listen dienen Kommas oder Blanks. Der Zugriff auf einen Dienst gilt als verboten, wenn sich ein passendes Daemon:Client-Päckchen in */etc/hosts.deny* findet. Umgekehrt wird er erlaubt, falls *hosts.allow* eine entsprechende Regel aufweist.

Sowohl die Daemon- als auch die Clientliste erlauben die Angabe von Wildcards. ALL lässt alle Dienste und Clients zu. *LOCAL* steht für Clients, deren Rechnername keinen Punkt enthält. *UNKNOWN* betrifft alle Clients, deren IP-Adresse oder Hostname sich nicht eruieren lässt; KNOWN bedeutet genau das Gegenteil. *PARANOID* betrifft alle Clients mit nicht übereinstimmenden Adressen und Namen. Zudem kann der Operator *EXCEPT* eingesetzt werden: *ALL EXCEPT 192.80.0.77, 192.80.0.207* betrifft alle Clients bis auf die angegebenen.

In der Client-Liste dürfen sowohl IP-Adressen als auch Rechnernamen auftauchen. Einträge, die mit einem Punkt beginnen, werden als Endstück eines *FQDN* betrachtet. So beträfe *.tecchannel.de* alle Rechner der Domain tecchannel.de. Auch IP-Adressen lassen sich generalisieren: 192.168.80. gilt für alle Rechner des entsprechenden Subnetzes. Daneben ist die Angabe per Netzmaske zulässig. 192.168.80. ließe sich also auch als 192.168.80.0/255.255.255.0 schreiben.

2.5.21 hosts.allow und hosts.deny

Eine bewährte Grundregel für alle Sicherheitsmaßnahmen lautet: „Erst einmal alles verbieten“. Daran sollte man sich tunlichst halten, so dass die *hosts.deny* genau einen Eintrag umfasst:

```
#  /etc/hosts.deny
ALL : ALL
#EOF
```

Per Default darf also kein entfernter Rechner auf irgendeinen lokalen Dienst zugreifen. Die */etc/hosts.allow* lockert anschließend diese restriktive Politik durch die Definition dedizierter Dienst-/Client-Pärchen wieder auf:

```
# /etc/hosts.allow
#
ALL EXCEPT in.telnetd : \
  192.168.80.0/255.255.255.0
in.telnetd : \
  192.168.80.207
#EOF
```

In unserem Beispiel dürfen alle Rechner im lokalen Class-C-Netz auf sämtliche Dienste mit Ausnahme von Telnet zugreifen. Letzteres bleibt ausschließlich der Maschine mit der IP-Adresse 192.168.80.207 vorbehalten.

Tragen Sie beim Editieren dafür Sorge, dass beide Dateien mit einem Zeilenumbruch enden. Andernfalls wird die letzte Zeile des Files nicht mehr ausgewertet. Am besten schließen Sie die Datei dazu mit einer Kommentarzeile ab. Da Linux die beiden *hosts_access*-Dateien bei jedem Verbindungsversuch neu auswertet, werden Änderungen an den Files auch ohne Restart des *inetd* sofort übernommen.

2.5.22 Andere Dienste

Linux startet nicht alle Dienste via *inetd*. Etliche Daemons fährt das Betriebssystem während des Bootens direkt über die Scripts in */etc/init.d* hoch. Zwei davon sollten Sie genauer unter die Lupe nehmen: den RPC-Portmapper *portmap* und den Printerdaemon *lpd*.

Ähnlich wie andere Serverdienste wird portmap auf den wenigsten Clients benötigt. Um den Start des Daemons komplett zu unterbinden, fügen Sie im Startup-Script */etc/init.d/portmap* als ersten ausführbaren Befehl ein *exit* ein. Da der RPC-Portmapper die Dienste des TCP-Wrappers nutzt, können Sie den Zugang alternativ auch über einen entsprechenden Eintrag in */etc/hosts.allow* beschränken:

```
# /etc/hosts.allow
#
ALL EXCEPT in.telnetd, portmap : \
    192.168.0.0/255.255.0.0
in.telnetd : \
    192.168.80.207
portmap : \
    192.186.80.
#EOF
```

In der Client-Liste erlaubt portmap dabei ausschließlich ALL oder die Angabe von IP-Adressen. Hostnamen werden nicht verarbeitet.

Der Druckerdaemon lpd lauscht auf Port 515 nach Connections. Per Default akzeptiert er Verbindungen zu jeder Gegenstelle, solange diese von Ports zwischen 721 und 731 stammen. Davon ist er nur durch das Einrichten der Datei */etc/hosts.lpd* abzuhalten. Den dort per Hostname oder IP-Adresse aufgeführten Maschinen gewährt der Daemon Zugriff, alle anderen lässt er abblitzen.

2.5.23 Fazit

Die geschilderten Maßnahmen verbessern den Schutz Ihrer Linux-Workstation - unangreifbar machen sie sie nicht. Das hat vor allem zwei Ursachen.

Zum einen ist Sicherheit kein Zustand, sondern ein Prozess. Das Verhältnis zwischen Systemadministrator und Cracker erinnert stark an das Rennen zwischen Hase und Igel: Der Cracker „ist schon da“. Erst wenn eine Sicherheitslücke bekannt wird, kann man sie auch stopfen. Ein Angreifer hat in der Zwischenzeit reichlich Gelegenheit, sie auszunutzen. Konsequente System-Updates senken hier allerdings das Risiko deutlich. Im Fall von Code Red beispielsweise war das Sicherheitsloch rund einen Monat vor dem Angriff bekannt, auch ein Patch lag schon vor.

Zum anderen bleiben selbst auf bestens gepflegten Systemen noch potenzielle Eingangstüren offen. Dabei handelt es sich um die Ports jener Dienste, die der Rechner nach dem Willen seines Benutzers nach außen anbieten soll. Hier kann eine Firewall helfen, die Maschine vor Angriffen zu schützen oder zumindest den Benutzer frühzeitig zu warnen. Einen umfassenden Überblick über die Möglichkeiten der bis Kernel 2.2 genutzten Ipchains bietet unsere Serie zum Thema Firewall unter Linux (webcode: a695). Einrichtung und Konfiguration der im Kernel 2.4 integrierten Iptables-Firewall sind Gegenstand eines weiteren Artikels auf tecChannel.de (webcode: a704).

tecCHANNEL-Links zum Thema	Webcode
Linux 2.4 für den Desktop	a706
Firewall-Grundlagen	a682
Linux-Bootkonfigurationen	a546
So funktioniert TCP/IP	a209
Viren unter Linux	a681

3 Linux im Servereinsatz

Linux kann Windows- und Unix-Server in fast allen Bereichen ersetzen. Die Vorteile liegen in einer offenen Architektur und geringen Kosten für das Betriebssystem. Das dritte Kapitel widmet sich daher den verschiedensten Einsatzgebieten von Linux-Servern. Mit unseren Workshops setzen Sie Basisinstallationen von Linux als Windows Server, Print-Server, Dial-up-Router und Webserver auf. Mit einem Jabber-Server unter Linux können Sie zudem Ihren Mailserver entlasten und zu einer besseren Unternehmenskommunikation beitragen.

3.1 Linux als Windows-Server

Linux ist für den Einsatz als Server geradezu prädestiniert. Sogar im Windows-Netzwerk ist Linux dank Samba eine ernst zu nehmende Alternative zu Windows NT/2000.

Wer ein Netzwerk mit Windows NT/2000 als Server aufsetzen will, muss zunächst einmal viel Geld investieren. Allein die Lizenz für fünf Benutzer kostet mehr als Tausend Euro. Dabei benötigen die meisten kleinen Netze nicht alle Funktionen, die der NT-Server bietet. Oft kommen nur die File- und Print-Dienste zum Einsatz, also die gemeinsame Nutzung von Dateien auf dem Server oder eines zentralen Netzwerkdruckers. Mit dem Samba-Server von Linux lässt sich dasselbe Ziel für deutlich weniger Geld erreichen. Aktuelle Linux-Distributionen sind für rund 50 Euro zu haben und die Hardwareanforderungen von Linux sind deutlich niedriger als von Windows NT und 2000. Für einfache File- und Print-Dienste genügt ein Rechner mit Pentium-Prozessor und 64 MByte Speicher. Das reicht aus, um beispielsweise Office-Dokumentvorlagen zentral abzulegen oder einen Drucker für alle Mitarbeiter bereitzustellen.

Samba ist ein System, das Unix-kompatible Computer in Windows-Netzwerke einbindet. Für Furore sorgt dieses System nicht zuletzt durch den Linux-Boom. Das Open-Source-Betriebssystem - ohnehin als stabil und zuverlässig bekannt - lässt sich durch Samba als waschechter Server in ein Windows-Netz einbinden. Dabei stellt sich das Gespann aus Linux und Samba im Netzwerk wie ein NT-Server dar. Somit kann das Linux-System als File- und Printserver für Windows- und andere Samba-Clients dienen. Die Clients bemerken den Unterschied zu einem NT-Server nicht.

Wenn man die Entwicklung von Linux betrachtet und neue Features wie Journaling-Dateisysteme oder Unterstützung für 4 GByte Arbeitsspeicher berücksichtigt, wird Linux auch für größere Netzwerke interessant. Es muss nicht immer der teure NT-Server sein. Dieser Beitrag gibt einen Einblick in die grundlegende Konfiguration und den Einsatz von Samba als Primary Domain Controller (PDC). Im aktuellen Update haben wir ihn an die Gegebenheiten der Samba-Version 2.2.4 angepasst.

3.1.1 Was ist Samba?

Samba ist eine Implementation des SMB-Protokolls für Unix. Über Server Message Blocks kommunizieren Rechner in einem Windows-Netzwerk miteinander und tauschen Informationen über freigegebene Verzeichnisse, Drucker oder Kommunikationsschnittstellen aus. Die ursprüngliche Version erforderte NetBEUI als Transportprotokoll. Inzwischen kommt meist TCP/IP zum Einsatz.

Undercover: Der Samba-Server taucht in der Netzwerkumgebung auf, ohne dass Windows den Unterschied bemerkt.

3.1.2 Voraussetzungen für Samba

In den meisten Linux-Distributionen ist Samba enthalten, wird jedoch nicht immer automatisch mitinstalliert. Um das nachzuholen, verwenden Sie den Paket-Manager Ihrer Distribution, beispielsweise Yast bei SuSE. Wenn Sie eine ältere Distribution haben, sollten Sie sich die aktuelle Samba-Version 2.2.4 von Samba (www.samba.org) aus dem Internet holen und installieren. Zusätzlich ist es notwendig, den SMB-Dienst im Startscript von Linux zu aktivieren. Bei der SuSE-Distribution setzen Sie in */etc/rc.config* die Variable *START_SMB* auf *yes*.

Der Samba-Server 2.2.x basiert auf zwei Daemons: *smbd*, der sich um die Share-Zuteilung kümmert und *nmbd*, der die Namensdienste bereitstellt. Bei der Verwendung von Samba müssen beide Daemons gestartet sein.

3.1.3 Was kann ein Samba-Server?

Auch die Samba-Version 2.2.4 ist nicht in der Lage, alle Eigenschaften einer proprietären Windows NT-Domain bereitzustellen. So kann ein Samba-Server zwar mittlerweile einen Primary Domain Controller (PDC) ersetzen, aber nicht

die Aufgabe eines Backup Domain Controllers übernehmen. Allerdings muss dies nicht als Fehler interpretiert werden. Die Entwickler von Samba können nur Eigenschaften implementieren, deren Funktionsweise und Protokolle vollständig bekannt sind. Leider gibt Microsoft bestimmte Protokolle nicht frei. Diese lassen sich als Konsequenz nicht implementieren. Deshalb kann Samba auch einen Exchange-Server auf Grund fehlender Protokolle nicht vollständig ersetzen. Folgende Eigenschaften eines PDC deckt ein Samba-Server der Version 2.2.4 ab:

- Verwaltung von Policies für Clients vom Typ Windows NT, 2000 und XP.
- Domain Logons für Windows 9x-, NT-, 2000 und XP-Clients. Nach erfolgreicher Authentifizierung erhält der Client Zugang zum gewünschten Share.
- Verwaltung verschiedener lokaler Profildateien der einzelnen Clients.
- Starten von entsprechenden Login-Scripts, wenn sich ein Client bei einer Domain als Mitglied anmeldet.
- Validierung von Clients, die beispielsweise PAM zur Authentifizierung verwenden.

3.1.4 NT-Domain bevorzugt

Für die Aufgabe, Linux als File- oder Printserver einzusetzen, soll der Samba-Server eine NT-Domain bereitstellen, in der sich die Windows-Clients anmelden. Eine Domain sollte es deshalb sein, weil das Peer-to-Peer-Konzept von Windows erhebliche Sicherheitslücken aufweist.

Die traditionelle Arbeitsgruppe von Windows 9x akzeptiert beim Netz-Login jeden Benutzernamen und jedes Passwort, da Windows 9x keine unautorisierten User kennt. Sobald Sie eine unbekannte Benutzerkennung eingeben, fragt Windows nach dem Passwort für diesen neuen User und legt ihn kurzerhand an. Nur wenn Sie auf einen passwortgeschützten Share im Netz zugreifen, wird das eingegebene Passwort verwendet beziehungsweise ein neues abgefragt. Öffentliche Ressourcen, wie für die gesamte Gruppe ohne Passwortschutz freigegebene Dateisysteme und Drucker, kann somit jeder Anwender oder Hacker nutzen.

Die Anmeldung an einer Domain ist hingegen sicherer. Hier wird ein Login nur akzeptiert, wenn Sie eine gültige Benutzerkennung und das entsprechende Passwort eingeben. Sollten Sie das Login abbrechen, kann kein Zugriff aufs Netz (und damit auch nicht auf öffentliche Freigaben des Domainservers) erfolgen. Die Anmeldung an einer Domain ist somit vergleichbar mit dem Login in ein Unix-System. Zudem lässt sich auf Grund des Benutzerkonzepts der Zugriff auf die Freigaben im Netz wesentlich zielgerichteter steuern.

Die Rechte auf Netzressourcen werden auf Benutzer- und Gruppenebene erteilt. Mit anderen Worten: Es erfolgt nur eine Authentifizierung - nämlich beim Login. Der Benutzer meldet sich auf dem Server an, womit alle Rechte im Netzwerk oder in der Domain festgelegt sind.

Das ist effizienter als die einzelnen Freigaben mit Passwörtern abzusichern, für die immer wieder eine neue Authentifizierung über das Netz stattfinden muss. Bei Zugriffen auf die Ressourcen kostet das unnötig Zeit und verursacht zusätzlich Netzwerkverkehr. Außerdem ist die Verwaltung der verteilten Shares und Rechte nicht nur unübersichtlich, sondern auch fehleranfällig. Wenn sich in der Firma Strukturen ändern oder Mitarbeiter ausscheiden, bestehen im schlimmsten Fall noch alte Freigaben, die ein erhebliches Sicherheitsloch darstellen.

3.1.5 Grundlegende Konfiguration

Voraussetzung für den Einsatz von Samba ist, dass im Windows-Netzwerk TCP/IP statt NetBEUI als Transportprotokoll für SMB verwendet wird. Die 32-Bit-Varianten von Windows machen hier keine Probleme. Sie liefern von Haus aus TCP/IP-Stacks mit. Lediglich Windows for Workgroups muss durch zusätzliche Software auf die Sprünge geholfen werden, die Microsoft auf seinem FTP-Server (ftp://ftp.microsoft.com/bussys/Clients/) kostenlos bereitstellt.

Des Weiteren sollten Sie sicherstellen, dass der Samba-Server der Primary Domain Controller (PDC) für die betreffende Domain ist. Die Domain, die Sie dem Samba-Server zuweisen, dürfen Sie an kein anderes Serversystem vergeben.

Unix-üblich erfolgt die Konfiguration von Samba über Konfigurationsdateien. Da nicht jeder Administrator sich mit diesen Konfigurationshürden kryptischer ASCII-Dateien auseinander setzen will, existieren eine ganze Reihe von grafischen Frontends zum Einrichten eines Samba-Servers. Die Grundlagen, die dieser Abschnitt erläutert, können Sie ohne Schwierigkeiten auf die grafischen Oberflächen übertragen.

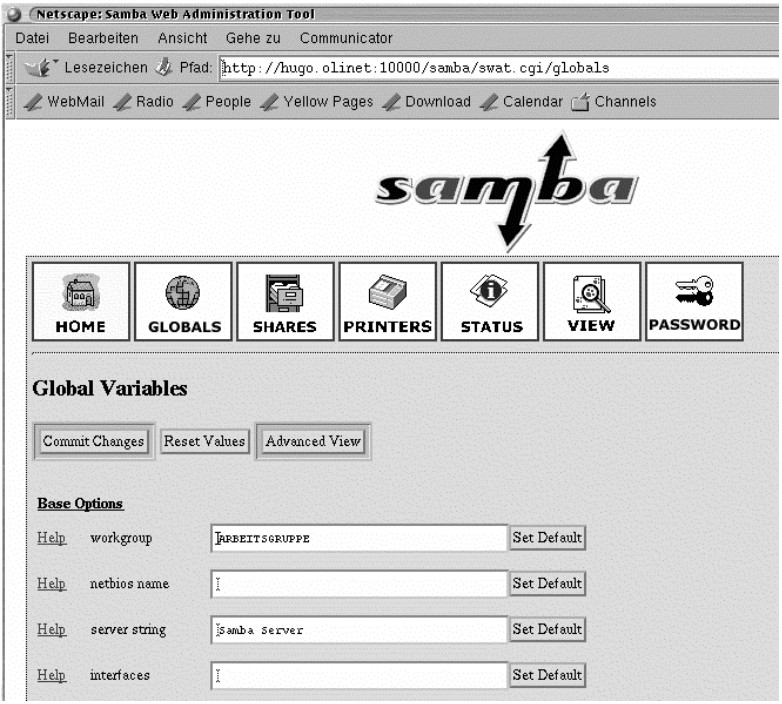
Schimpfen Sie dennoch nicht über die Textfiles, denn damit haben Sie ein mächtiges Feature zur Verfügung, das die grafischen Oberflächen und Windows nicht bieten: Suchen&Ersetzen. Globale Änderungen lassen sich damit in Sekundenschnelle ausführen, etwa Pfadänderungen für alle Shares, weil sich die Verzeichnisstruktur geändert hat.

3.1.6 smb.conf: Zentrale Konfigurationsdatei

Für die Konfiguration eines Samba-Servers ist die Datei */etc/smb.conf* die zentrale Anlaufstelle. Hier findet sich die Basiskonfiguration, wie beispielsweise Einstellungen zu Sicherheit, Domain oder Zugriff, sowie Abschnitte für die einzelnen Freigaben.

Die Datei ist ähnlich aufgebaut wie eine .ini-Datei unter Windows. Einzelne Sektionen beginnen mit in eckigen Klammern gefassten Überschriften. Im Anschluss folgen die Einstellungen paarweise in der Form *Eigenschaft = Wert*.

Im Abschnitt *[global]* finden Sie die grundlegende Konfiguration von Samba. Die restlichen Sektionen legen die einzelnen Freigaben fest. Hierbei wird der Name des Shares als Abschnittstitel in eckige Klammern gefasst. Die einzelnen



SWAT: Das grafische Setup-Tool vereinfacht die Samba-Konfiguration und ermöglicht den Remote-Zugriff per Webbrowser.

Reserviert sind die Abschnitte *[printers]* und *[homes]*. *printers* definiert die Eigenschaften für die Freigabe aller Drucker aus */etc/printcap*. Der Abschnitt *homes* legt fest, wie die Home-Verzeichnisse der Benutzer freigegeben werden sollen.

Nach jeder Änderung in der Datei */etc/smb.conf* muss Samba neu gestartet werden. Damit ermöglicht man dem Server, die Konfigurationsdatei erneut einzulesen. Unter SuSE-Linux geben Sie dazu in der Kommandozeile folgenden Befehl ein:

```
rcsmb restart
```

3.1.7 Die erste Domain

Wenn Sie die Samba-Pakete Ihrer Distribution installieren, finden Sie eine durch Kommentare (eingeleitet durch *;* oder #) gut dokumentierte */etc/smb.conf* vor. So manche Einstellung lässt sich schon auf Grund dieser Kommentare durchführen.

Um einen Samba-Server, der als Alternative zum Windows-Server fungiert, zu konfigurieren, sind ein paar Einstellungen unter *[global]* erforderlich. Zunächst ist die Eigenschaft *Workgroup* mit dem Namen der Domain zu belegen. Angenommen Ihre Domain soll den Namen „Vertrieb" erhalten, so ändern Sie die entsprechende Zeile wie folgt ab:

```
workgroup = Vertrieb
```

Im nächsten Schritt setzen Sie *domain logins* auf *yes*. Wenn dieser Parameter mit dem Wert *true* belegt ist, beantwortet Samba auch Windows-9x-Domain-Logons der Workgroup, zu welcher der Server gehört. Zusätzlich muss der Zugriff auf User-Ebene aktiviert werden.

Dies bewirkt, dass sich der Client nur mit einer gültigen Benutzername/Passwort-Kombination beim Server anmelden kann. Hierzu setzen Sie *security* auf *user*. Der Security Level ist einer der wichtigsten Einstellungen innerhalb der */etc/smb.conf*-Datei. Diese Option kann nur global festgelegt werden, da jeder Server lediglich eine Art der Sicherheit kennt.

```
domain logons = yes
security = user
```

Wie bereits erwähnt, unterstützt Samba nicht nur Domain-Logins, sondern kann auch in alter Windows-Peer-to-Peer-Manier für die Freigaben auf Share-Basis konfiguriert werden. Hierzu wäre *security* auf *share* zu setzen.

3.1.8 Verschlüsselte Passwörter

Seit Windows NT 4.0 Service-Pack 3 wird nur noch ein Login mit verschlüsseltem Passwort unterstützt. Anders gesagt: Das Passwort wird nicht mehr im Klartext über das Netz gesendet. Windows 9x-Clients (ab Version 95b) sind auf dieses Verfahren vorkonfiguriert. Auch Samba unterstützt dieses Verfahren, so dass Sie es lediglich aktivieren müssen (und sollten!):

```
encrypt passwords = yes
```

```
[netlogon]
comment = Domain Logon
path = /home/netlogon
public = no
writable = no
browseable = no
```

Durch diese Einträge wird ein schreibgeschützter, nicht öffentlicher und nicht in der Browser-Liste von Windows erscheinender File-Share angelegt. Stellen Sie sicher, dass das Verzeichnis */home/netlogon* existiert. Legen Sie es gegebenenfalls durch den Befehl *mkdir /home/netlogon* auf der Shell an.

3.1.9 Benutzerverwaltung

Die Benutzerkonzepte von Linux und Windows sind grundsätzlich verschieden. Daher kann auf der Windows/SMB-Seite des Samba-Servers nicht direkt mit Linux-Usern gearbeitet werden. Als Lösung des Problems bildet Samba daher die SMB-Benutzer auf Linux-User ab. Die Zugriffsrechte auf die freigegebenen Ressourcen legen Sie mit Usern und Groups unter Linux fest. Das Login erfolgt dagegen über SMB-Benutzer. Sie weisen lediglich den Linux-Usern einen SMB-Benutzer zu. Darüber hinaus sind die Passwörter auf Grund der Windows-Verschlüsselung zu konvertieren. Bei beiden Aufgaben helfen Ihnen die Programme *smbpasswd* und *smbadduser*.

Zunächst unterscheiden sich unter Linux und Windows festgelegte Benutzer. So ist der Name des Super-Users unter Linux „root“ und unter Windows in der Regel „Administrator“. Damit Sie nun nicht für jeden SMB-Benutzer jeweils einen Linux-Account einrichten müssen, können Sie einen oder mehrere SMB-Benutzer auf einen Linux-User abbilden. Die dafür zuständige Datei legen Sie im Abschnitt *[global]* mit

```
username map = /etc/smbusers
```

fest. In diesem Beispiel wird */etc/smbusers* als Datei für das Mapping von Linux auf SMB-User verwendet.

3.1.10 Windows-Nutzer und Linux-Nutzer

Um nun beispielsweise den NT-Administrator auf root abzubilden, geben Sie in dieser Datei die Zeile

```
root = Administrator
```

an. Wollen Sie mehrere SMB-Nutzer auf einen einzigen Linux-User festschreiben, geben Sie einfach hinter dem Gleichheitszeichen eine durch Leerzeichen getrennte Auflistung der gewünschten SMB-Benutzer an. Sollte ein Benutzername selbst Leerzeichen enthalten, ist dieser in Anführungszeichen zu fassen. Den SMB-Nutzern weisen Sie über das Programm *smbpasswd* Passwörter zu. Zum Beispiel legt der Befehl

```
smbpasswd oliver
```

das Passwort für das Windows-Netzwerk für den Benutzer „Oliver“ fest. Das hat keine Auswirkung auf das Kennwort eines möglicherweise vorhandenen Linux-Users gleichen Namens.

Um einen SMB-Nutzer festzulegen und in einem Schritt gleich das Passwort für diesen zu vergeben, können Sie das Programm *smbadduser* verwenden. Hier geben Sie als Parameter den Linux-User und den SMB-Benutzer zusammengefasst durch einen Doppelpunkt an, zum Beispiel:

```
smbadduser root:Administrator
```

Wenn Sie mit NT-Clients auf den Samba-Server zugreifen wollen, ist zusätzlich für jeden dieser Clients ein Maschinenkonto anzulegen. Wollen Sie etwa den NT-Rechner „Theodor“ an der Samba-Domain anmelden, legen Sie mit

```
smbpasswd -a -m Theodor
```

ein Maschinenkonto für diesen Rechner an.

3.1.11 File-Shares

Wollen Sie ein Verzeichnis auf dem Server freigeben, so legen Sie hierfür eine separate Sektion an. Angenommen, Sie wollen das Verzeichnis */mnt/projekte* unter dem Namen „Projekte“ freigeben, so können Sie dies durch die Sektion

```
[Projekte]
  comment = Projekte
  path = /mnt/projekte
  writable = yes
  browseable = yes
  public = no
  valid users = oliver fritzchen
  create mask = 0754
```

bewerkstelligen. *path* gibt den Pfad des Verzeichnisses an. *comment* definiert die Beschreibung, die in der Browser-Liste auf dem Client angezeigt wird. *browseable = yes* legt fest, dass der Share auch tatsächlich in der Browser-Liste erscheint. Würden Sie diese Eigenschaft auf *no* setzen, hätten Sie einen versteckten Share im Netzwerk geschaffen. Dann muss der Benutzer den Namen explizit wissen und angeben, um die Freigabe zu nutzen. *writable = yes* gibt an, dass auf die Freigabe auch schreibend zugegriffen werden kann. Mit *public = no* wird festgelegt, dass der Share nicht öffentlich ist und Gastbenutzern nicht zur Verfügung steht.

Netzbrowser: Freigaben erscheinen so, als wären sie auf einem NT-Server.

Durch Eigenschaften wie *valid users*, *invalid users*, *write list* oder *read list* lassen sich die Benutzerrechte bereits auf SMB-Ebene einschränken. Hiermit kann bestimmten SMB-Usern zum Beispiel nur Lesezugriff gestattet werden, obwohl der korrespondierende Linux-User auch Schreibzugriff hätte. Nähere Informationen finden Sie in der Manpage smb.conf(5) (http://de.samba.org/samba/docs/man/smb.conf.5.html).

create mask legt fest, mit welchen Attributen Dateien im Linux/Unix-Dateisystem angelegt werden sollen. Das ist notwendig, weil der Windows-Client die Unix-Zugriffsrechte nicht unterstützt.

3.1.12 Unix-Dateirechte

Unix speichert für jede Datei den Namen des Besitzers und der Gruppe, sowie Dateityp und die Zugriffsrechte auf die Datei. Diese Informationen legt Unix beim Erzeugen der Datei an. Dabei setzt es den Dateibesitzer auf den aktuell eingeloggten Benutzer. Die Gruppe ergibt sich aus der Gruppenzugehörigkeit des Benutzers.

Samba hat es da etwas schwerer: Der Unix-Benutzer und die Gruppe sind dem Samba-Server über die Zuordnung von Samba-Usern auf Linux-User zwar bekannt. Er kann jedoch die Zugriffsrechte nicht aus den Benutzerdaten ermitteln. Daher legen Sie die Default-Rechte fest, welche die Datei beim Erzeugen erhalten soll. Dazu geben Sie die Rechte als oktalen Zahlenwert an. Die erste Ziffer gibt den Dateityp an und sollte immer 0 sein. Die nächsten drei Ziffern legen jeweils die Zugriffsrechte für den Eigentümer, die Gruppe und sonstige Benutzer fest. Sie ergeben sich aus der Summe der einzelnen Rechte:

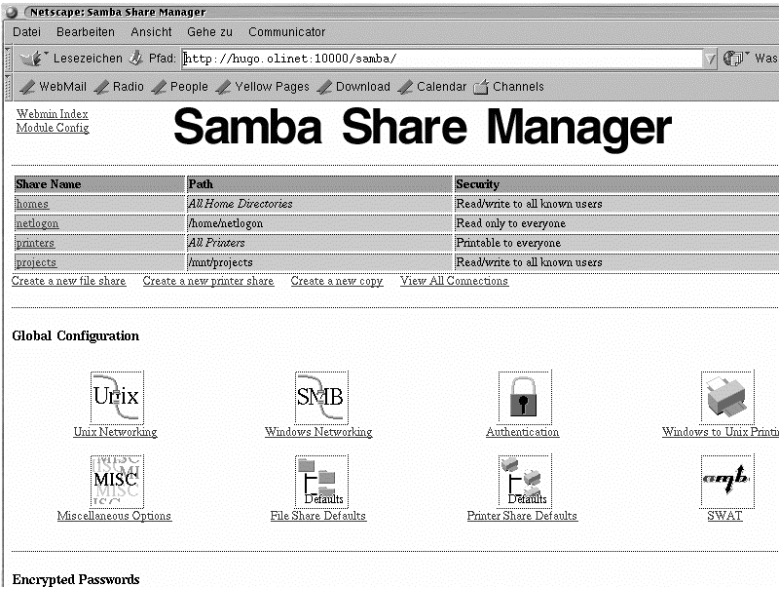
Unix-Zugriffsrechte	
Recht	**Zahlenwert**
Read	4
Write	2
Execute	1

Die Ziffernfolge 0754 besagt also, dass der Besitzer die Datei lesen, schreiben und ausführen darf. Die Gruppe darf lesen und ausführen. Andere Benutzer hingegen dürfen lediglich lesen. Einen Überblick über alle vergebenen Shares und die Rechte darauf liefert der Samba Share Manager von Webmin (siehe auch nächster Abschnitt).

3.1.13 Grafische Oberflächen

Die Administration von Samba via Konfigurationsdateien ist vom Prinzip her nicht sonderlich kompliziert. Einfacher und vor allem komfortabler geht es jedoch mit grafischen Oberflächen. Für Samba existieren eine ganze Reihe von grafischen Frontends. Bekannt sind SWAT (Samba Web Administration Tool), Webmin (www.webmin.com), KSamba (www.kneschke.de/projekte/ksamba/) und GnoSamba (http://freshmeat.net/projects/gnosamba/homepage/).

KSamba und GnoSamba sind Oberflächen für KDE (www.kde.org) beziehungsweise GNOME (www.gnome.org). Beide Programme sind recht jung und bieten noch nicht alle Konfigurationsmöglichkeiten. Außerdem fehlen hier Erfahrungswerte aus dem Administrationsalltag.

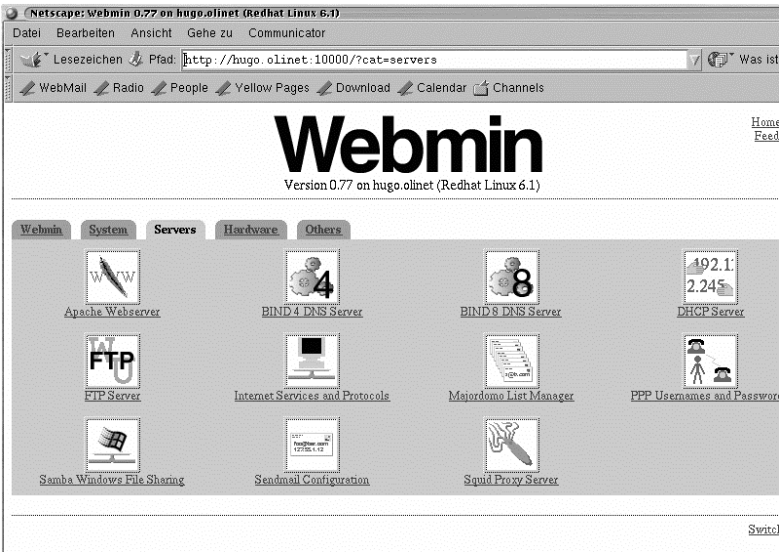


Alles was Recht ist: Der Share-Manager vereinfacht die Rechtezuordnung auf Shares.

SWAT und Webmin bieten beide einen kleinen „Mini-Webserver“, so dass die Konfiguration von Samba in einem beliebigen Webbrowser erfolgen kann. SWAT ist im Samba-Paket enthalten. Samba und SWAT greifen auf die gleiche Codebasis zu, so dass SWAT immer perfekt an Samba angepasst ist. Der eklatante Nachteil von SWAT ist jedoch, dass die Kommunikation zwischen SWAT und dem Webbrowser unverschlüsselt erfolgt. Sie sollten SWAT daher nur in einer Umgebung einsetzen, wo Sicherheit keine große Rolle spielt oder wo der Netzwerk-Traffic nicht von Unbefugten abgehört werden kann, etwa in einer voll geswitchten Umgebung.

Webmin ist ein komplettes Administrationstool, das unter anderem auch ein Modul für Samba beinhaltet. Das Sicherheitsproblem von SWAT teilt Webmin nicht, der sämtliche Kommunikation über SSL verschlüsseln kann.

Auf den Internet-Seiten des Samba-Projekts finden Sie eine Übersicht (http://de.samba.org/samba/GUI/) über zahlreiche verfügbare grafische Administrationstools.



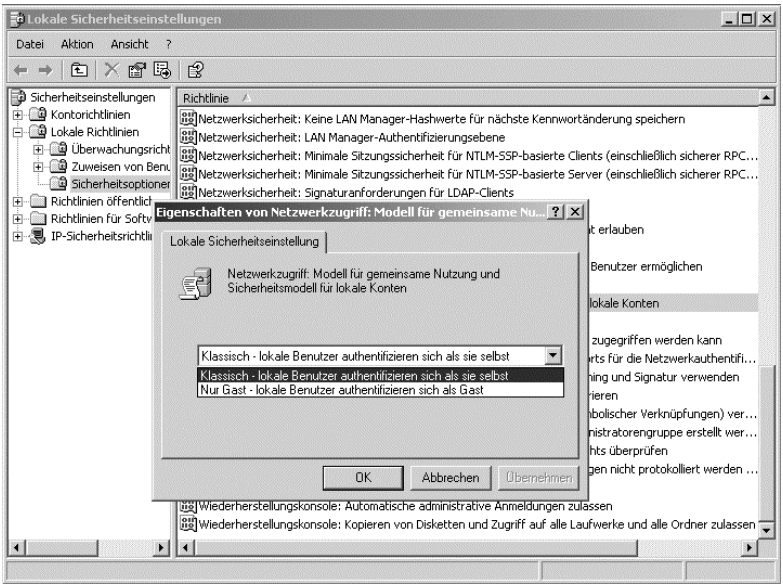
Via Browser: Webmin verwaltet nicht nur den Samba-Server, sondern eine ganze Reihe von Linux-Komponenten per Webbrowser.

3.1.14 Samba-Client an Windows XP

Eine steigende Zahl von Windows XP-Rechnern sind mit dem Internet verbunden. Aus diesem Grund erfordert Windows XP als Standardeinstellung bei allen Netzwerkverbindungen für die Benutzung des Gastkontos eine spezielle Autorisierung.

Dies soll verhindern, dass sich ein Hacker über das Internet als lokaler Administrator anmelden kann. Allerdings hat diese Sicherheitseinstellung zur Folge, dass die Anmeldung eines Linux-Rechners mit Samba an einem als Workgroup konfigurierten Windows XP-Rechner nicht möglich ist.

Mit folgender Änderung im Policy Editor von Windows XP Professional können Sie dies umgehen: Gehen Sie auf Systemsteuerung - Verwaltung - Lokale Sicherheitsrichtlinien - Lokale Richtlinien - Sicherheitsoptionen. Ändern Sie unter dem Punkt *Netzwerkzugriff: Modell für gemeinsame Nutzung und Sicherheitsmodell für lokale Konten* von *Nur Gast* auf *Klassisch*.



Policy Editor: Durch eine Änderung in den lokalen Sicherheitsrichtlinien ist die Anmeldung von einem Linux-Rechner aus mit Samba wieder möglich.

Unter Windows XP Home ist eine Änderung in der Registry nötig: Öffnen Sie den Registrierungseditor und suchen Sie nach dem Schlüssel *HKLM\System\CurrentControlSet\Control\LSA*. Wählen Sie den Eintrag ForceGuest und ändern Sie diesen auf *ForceGuest=0*. Ein Neustart des Rechners nach dieser Änderung ist nicht erforderlich.

3.1.15 Fazit

Wer in kleinen bis mittleren Netzen einen reinen File- und Printserver aufsetzen will, sollte bei der Entscheidung für das Betriebssystem Linux nicht gleich außen vor lassen. Samba bietet dieselben Funktionen wie ein NT-Server und ist praktisch kostenlos. Eine aktuelle Linux-Distribution kostet gerade mal zwischen 50 und 70 Euro, für beliebig viele Benutzer. Verzichten Sie bei der Installation auf Ressourcen-Fresser wie den X-Server, kommt Linux mit deutlich weniger Hardware aus als Windows NT und 2000. Auch Einrichtung und Konfiguration eines Samba-Servers sind dank grafischer Benutzeroberfläche für Linux-Einsteiger kein Hindernis mehr. Eine ausführliche Dokumentation zu Samba mit einer Erklärung sämtlicher Optionen finden Sie unter der Adresse http://de.samba.org/samba/docs/Samba-HOWTO-Collection.html.

Wie Sie Linux als Printserver mit automatischer Installation der Treiber auf dem Client einrichten, lesen Sie in einem anderen tecCHANNEL-Beitrag (webcode: a392). Mit etwas mehr Aufwand lässt sich Linux auch für zusätzliche Aufgaben einsetzen, etwa als Intranet-Server oder als Internet-Router mit Proxy- und Firewall-Funktionen. Das ermöglicht den Benutzern im Netz den Zugriff auf das Internet und E-Mail-Dienste. Welche Installations- und Konfigurationsschritte dazu erforderlich sind, lesen Sie in weiteren Beiträgen in unserem Linux&Unix-Channel (www.tecChannel.de/linux&unix.html).

tecCHANNEL Links zum Thema	Webcode
Linux als Webserver	a442
Linux als Printserver	a392
Linux als Dial-up-Router	a322
Linux für den Desktop	a494
Linux für den Server	a487
Linux als Firewall	a695
Proxy-Server unter Linux	a798

3.2 Linux als Printserver

Nicht jeder Windows- oder Linux-Client muss über einen eigenen Drucker verfügen. Ein zentraler Netzwerkdrucker spart Kosten und erleichtert die Administration. Seine Verwaltung kann bequem über Linux erfolgen.

Die Voraussetzung für einen funktionierenden Samba-Printserver, den auch Windows-Clients nutzen können, ist die korrekte Installation des Samba-Servers. Der Samba-Server ist in fast jeder Linux-Distribution von Haus aus enthalten. Updates, Dokumentationen und FAQs finden Sie unter www.samba.org

Um grundsätzlich alle Drucker auf dem Samba-Server unter Linux freizugeben, genügt in der Datei */etc/smb.conf* eine Sektion der Form:

```
[printers]
   comment = All Printers
   path = /var/spool/samba
   browseable = no
   writeable = no
   printable = yes
```

Die Freigabe wird als Drucker-Share *(printable = yes)* angegeben. *path* gibt hier das Spool-Verzeichnis an. Falls es noch nicht existiert, legt Samba es beim ersten Druckauftrag an.

3.2.1 Postscript oder RAW?

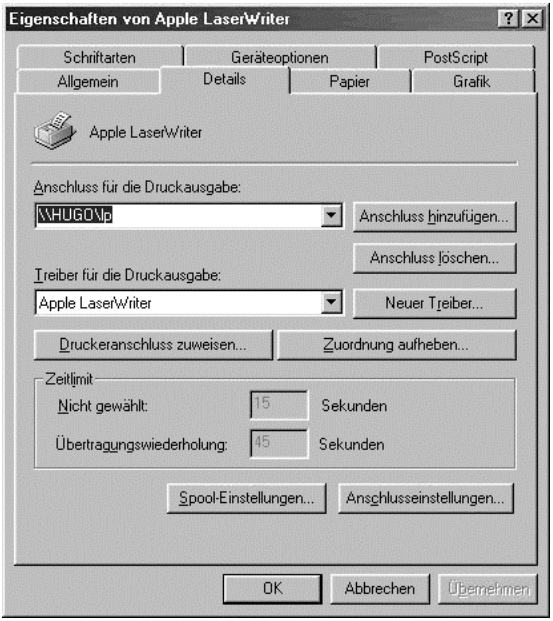
Das Problem, über das die meisten Samba-Neulinge stolpern, ist die Einrichtung des Druckers auf der Client-Seite. Drucker können Sie nämlich auf zwei Arten bereitstellen: Entweder geben Sie den Drucker als PostScript-Gerät oder als Raw-Printer frei. Letzterer erwartet in der jeweiligen Sprache des Druckers perfekt aufbereitete Druckaufträge, also eine native Druckersprache wie beispielsweise ESC/P. Linux schleust in diesem Fall die Druckdaten einfach durch den Druckerport, ohne eine weitere Aufbereitung durch Programme oder Filter wie GhostScript.

Der auf der vorigen Seite angegebene universelle Printer-Share zieht alle Drucker aus der Datei */etc/printcap* heraus und gibt diese frei. Im folgenden Beispiel einer printcap-Datei ist ein Drucker (*lp*) für die Zusammenarbeit mit GhostScript konfiguriert und somit wie ein PostScript-Drucker anzusprechen. Der andere Drucker (*raw*) ist als Raw-Printer installiert:

```
lp:\
   :sd=/var/spool/lpd/lp:\
   :mx#0:\
   :sh:\
   :lp=/dev/lp0:\
   :if=/var/spool/lpd/lp/filter:
raw:\
   :rw:sh:\
   :lp=/dev/lp0:\
   :sd=/var/spool/lpd/raw:\
   :fx=flp:
```

3.2.2 Drucker am Client

Die Printers-Sektion in */etc/smb.conf* gibt die beiden Drucker als *lp* und *raw* frei. Wenn Sie nun auf dem Windows-Client einen neuen Drucker über den Share *lp* einrichten, muss ein PostScript-Druckertreiber installiert werden. In der Praxis hat sich der Treiber für den Apple LaserWriter als gute Wahl erwiesen, da Apple schon sehr früh auf den PostScript-Standard setzte. Dabei ist es egal, welchen Drucker Sie tatsächlich angeschlossen haben.



Treiber-Selektion: Der Treiber für den Apple LaserWriter eignet sich besonders gut als PostScript-Treiber auf der Client-Seite.

3.2.3 Automatische Treiberinstallation

Die übliche Methode, den Drucker vom Anwender über den Wizard von Windows zu installieren, hat jedoch einige Schwächen. Der User muss mehrere Schritte durcharbeiten, wie etwa die Auswahl des Druckertyps oder die teils zahllosen Optionen. Als Systemadministrator können Sie dabei nicht sicher sein, dass dieser manuelle Installationsvorgang auch fehlerfrei durchgeführt wird.

Windows NT/2000-Server bieten die Möglichkeit, Druckertreiber automatisch auf dem Client-System zu installieren. Dabei wird ein Drucker per Windows NT oder Windows 2000 freigegeben. Danach klickt der Benutzer auf seinem Client nur noch doppelt auf die Netzwerkfreigabe des betreffenden Druckers, und ein Dialogfeld fragt ihn, ob der vom Server bereitgestellte Treiber verwendet werden soll. Man bejaht die Frage, und schon ist der Drucker installiert.

Ein solches Vorgehen lässt sich auch unter Samba umsetzen. Dabei sind ältere Samba-Versionen mit der Einschränkung behaftet, dass dieses Verfahren nur mit Windows-9x-Clients funktioniert. Ab der Version 2.2.0 klappt die automatische Treiberinstallation mit Samba auch auf Clients mit Windows NT/2000.

3.2.4 Referenzinstallation anlegen

Die automatische Treiberinstallation funktioniert nach einem einfachen Prinzip. Normalerweise legen Sie während der Installation die Windows-CD-ROM ein, und das System kopiert sich die erforderlichen Treiber.

Bei der automatischen Installation werden die Treiberdateien nicht von der Windows-CD, sondern vom Samba-Server auf den Client kopiert. Außerdem ist auf dem Server eindeutig festgelegt, welcher Treiber zum Einsatz kommen soll. Die Angabe eines falschen Treibers ist dadurch ausgeschlossen. Damit diese Treiberdateien auf dem Samba-System zur Verfügung stehen, gilt es zunächst, diese für Ihren Drucker relevanten Dateien zu finden. Hierzu legen Sie einfach eine Art Schablone an. Sie installieren den Treiber einmal auf einem Client und machen dann einen Abzug davon.

Diese Installation können Sie an jedem beliebigen Windows-System vornehmen. Am besten wählen Sie jedoch einen PC aus, der später ohnehin mit dem Netzwerkdrucker verbunden sein soll. Dadurch können Sie sich die Arbeit beim Einrichten des Samba-Systems mit einem kleinen Trick erleichtern.

Installieren Sie den Treiber für den an das Samba-System angeschlossenen Drucker. Denken Sie daran, dass Sie bei als PostScript (Druckername nicht raw) freigegebenen Druckern den Treiber Apple LaserWriter verwenden.

Wenn Sie den Samba-Drucker auf einem seiner künftigen Clients installieren, drucken Sie bitte eine Testseite aus und bewahren Sie diese auf. Hier finden Sie die Treiberdateien inklusive Pfadangabe. Diese Dateien kopieren Sie später auf den Samba-Server. Die Namen der Treiberdateien findet ein Script von Samba zwar selbst heraus, aber deren Pfade meldet Samba nicht.

3.2.5 Druckerdefinition für Samba

Der nächste Schritt ist das Anlegen des so genannten Printer Definition File. Diese Datei nimmt sämtliche Angaben über den Drucker auf, so wie er jetzt in der Referenzinstallation vorhanden ist, und stellt somit die Basisinformationen zur automatischen Installation für das Samba-System bereit. Die erforderlichen Informationen muss Samba sich aus den Windows-9x-Dateien MSPRINT.INF, MSPRINT2.INF, MSPRINT3.INF und MSPRINT4.INF beziehungsweise NTPRINT.INF unter Windows NT/2000 aus dem Windows-INF-Verzeichnis herausziehen. Beachten Sie dabei, dass die Dateien MSPRINT3.INF und MSPRINT4.INF bei älteren Windows-95-Systemen nicht existieren.

Kopieren Sie die Dateien in das Home-Verzeichnis Ihres Linux-/Samba-Systems. Anschließend suchen Sie mit *grep* in den *.INF-Dateien nach Ihrem Druckertreiber. Dabei ist auf die exakte Schreibweise des Windows-Druckertreibers zu achten (im Beispiel unter Windows 9x der Apple LaserWriter):

```
grep „Apple LaserWriter„ MSPRINT*.INF
```

Nun erhalten Sie eine Auflistung, in welcher Datei der Eintrag für den Treiber schlummert. Aus dieser Datei müssen Sie das Printer Definition File erzeugen. Hierzu verwenden Sie *make_printerdef*. Übergeben Sie diesem Programm als Argumente die soeben ermittelte MSPRINT*.INF-Datei (beziehungsweise das betreffende OEM*.INF-Pendant) und den Namen des Druckertreibers. Leiten Sie diese Ausgabe in eine Datei, wie beispielsweise *printers.def*, um.

3.2.6 Unbekannte Drucker installieren

Bei Druckern, die Windows nicht direkt unterstützt, finden Sie die erforderlichen INF-Daten in einer der OEM*.INF-Dateien unter *C:\WINDOWS\INF*. Dehnen Sie in diesem Fall den *grep*-Befehl einfach auf diese Dateien aus, wenn Sie mit MSPRINT*.INF keine Erfolge erzielen. In unserem Beispiel mit dem Apple LaserWriter würden Sie also das folgende Kommando eingeben:

```
make_printerdef MSPRINT.INF "Apple LaserWriter" >
printers.def
```

Als Resultat erhalten Sie eine Auflistung mit Informationen zum Drucker:

```
Found:APPLE230.SPD
End of section found
CopyFiles: @APPLE230.SPD,PSCRIPT
Datasection: PSCRIPT_DATA
Datafile: APPLE230.SPD
Driverfile: PSCRIPT.DRV
Helpfile: PSCRIPT.HLP
LanguageMonitor:

Copy the following files to your printer$ share location:
APPLE230.SPD
PSCRIPT.DRV
PSCRIPT.HLP
PSCRIPT.INI
TESTPS.TXT
APPLE380.SPD
FONTS.MFM
ICONLIB.DLL
PSMON.DLL
```

Sofern Sie beim Anlegen der Referenzinstallation keine Testseite ausgedruckt haben, notieren Sie sich bitte jetzt die unter dem Abschnitt *Copy the following...* angegebenen Dateien. Das sind die Treiberdateien, die Sie auf Ihr Samba-System übertragen müssen. Diese Dateien finden sich auf dem „Referenz-Client“ zumeist im Verzeichnis C:\WINDOWS\SYSTEM.

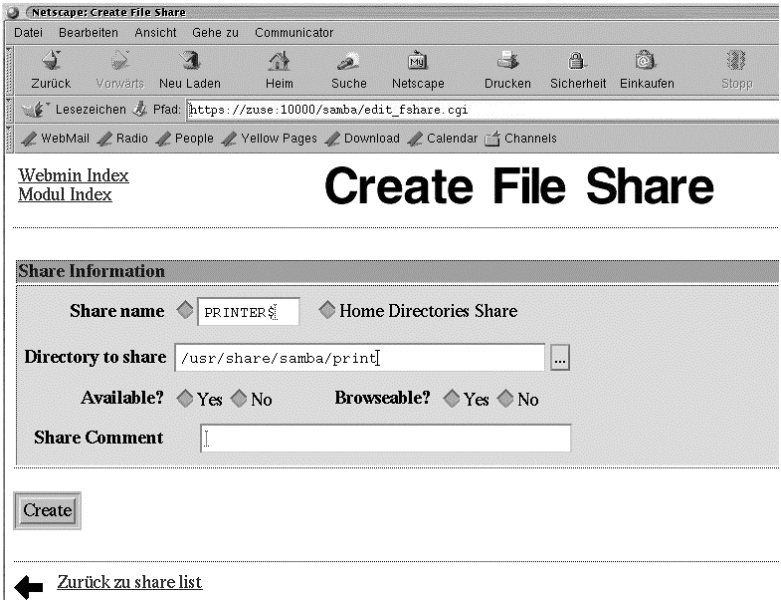
3.2.7 Treiber bereitstellen

Nachdem Sie die Druckerdefinition abgeschlossen haben, gilt es, die Treiberdateien und das Printer Definition File auf dem Samba-Server zur Verfügung zu stellen. Hierzu legen Sie zunächst ein neues Verzeichnis an, in das Sie die Dateien kopieren. Eine gute Wahl für das Verzeichnis wäre */usr/share/samba/print*:

```
mkdir -p /usr/share/samba/print
```

Dieses Verzeichnis geben Sie als Share *PRINTER$* frei. Der entsprechende Eintrag in /etc/smb.conf wäre:

```
[PRINTER$]
    path = /usr/share/samba/print
    read only = yes
    browseable = no
    guest ok = yes
```



Druckereinrichtung: Legen Sie einen Share PRINTER$ für die Treiberdateien an (hier mit Webmin).

In dieses Verzeichnis kopieren Sie die von *make_printerdef* ausgegebenen beziehungsweise auf dem Testausdruck aufgelisteten Dateien. Die Datei *msprint.cat* der Testseite benötigen Sie nicht. Anschließend befördern Sie auch die von *make_printerdef* angelegte Datei *printers.def* in das Verzeichnis des Shares *PRINTER$*.

Wollen Sie weitere Drucker mit automatischer Treiberinstallation verfügbar machen, überschreiben Sie keinesfalls die Datei *printers.def* auf dem Samba-Server. Fügen Sie stattdessen einfach die Definitionen des neuen Druckers an.

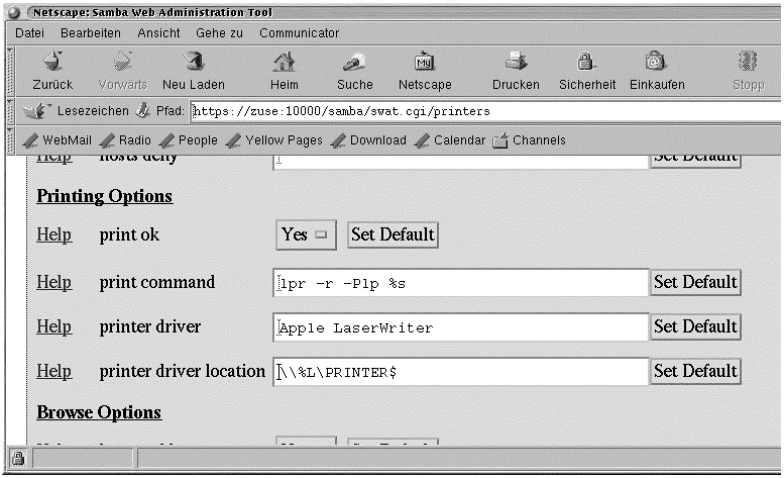
3.2.8 Drucker freigeben

Nachdem die Treiberdateien auf dem Samba-Server untergebracht sind, geben Sie den neuen Drucker frei. Dies geschieht wieder in der Datei */etc/smb.conf*. Hier geben Sie im Abschnitt *[global]* die Lage des Printer Definition File an:

```
printer driver file = /usr/share/samba/print/printers.def
```

```
[PostScript]
path = /var/spool/samba
printable = yes
print command = lpr -r -Pljet4 %s
printer driver = Apple LaserWriter
printer driver location = \\%L\PRINTER$
```

Wenn Sie Ihren Drucker-Share (hier PostScript) nicht wie die betreffende Queue in */etc/printcap* benennen, können Sie durch die Zeile *printer command* festlegen, welches Kommando die eingehenden Druckaufträge starten soll. Im Beispiel verwenden wir *lpr*, wobei die Option *-Pljet4* die Queue namens *ljet4* bestimmt. *%s* ist in diesem Fall ein Platzhalter für die Datei.

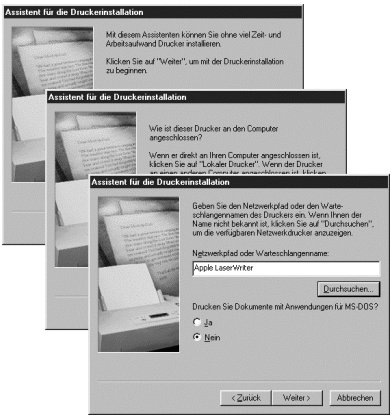


Parameterübergabe: Durch die Optionen printer driver und printer driver location legen Sie die automatische Treiberinstallation fest (hier in SWAT).

Unter *printer driver* geben Sie die Bezeichnung des Druckertreibers von Windows an. Die Option *printer driver location* gibt als UNC-Pfad an, wo sich die Treiberdateien befinden. Diese liegen im neu erzeugten Share *PRINTER$* des Samba-Servers. Daher die Angabe *\\%L\PRINTER$*, wobei *%L* durch den Servernamen Ihres Samba-Systems ersetzt wird.

3.2.9 Abschluss und Test

Jetzt ist der große Augenblick gekommen: Die automatische Druckerinstallation muss getestet werden. Dazu ist zunächst der Samba-Daemon neu zu starten.



Client-Installation: Man braucht nur noch einige Dialogfelder auszufüllen, um den Drucker automatisch auf dem Client zu installieren.

Nun können Sie sich auf einem Client-System, das den Drucker bislang noch nicht verwendet, durch einen Klick auf das Desktop-Symbol Netzwerk bis zu Ihrem Samba-Server durchklicken. Hier finden Sie den neuen Drucker-Share. Klicken Sie auf diesen doppelt. Es öffnet sich ein aus drei Dialogfeldern bestehender Wizard, durch den das Einrichten des Druckers zum Kinderspiel wird. Beim letzten Dialog geben Sie sicherheitshalber noch einmal an, dass eine Testseite gedruckt werden soll. Wenn Sie diesen Dialog verlassen haben, werden die Treiberdateien automatisch von Ihrem Samba-Server übertragen, und die Testseite sollte aus Ihrem Drucker gleiten.

tecCHANNEL-Links zum Thema	Webcode
Linux 2.4 für den Desktop	a706
Linux als Firewall	a695
Linux als Windows-Server	a248
Linux als Dial-up-Router	a322
Happy Birthday, Linux!	a758
Java Virtual Machine unter Linux	a776
Tux goes Biz	a654

3.3 Linux als Dial-up-Router

Linux kann nicht nur Aufgaben als Datei- und Druckserver wahrnehmen. Auf Grund der Wurzeln im Unix-Bereich ist es kein Hexenwerk, Linux als Dial-up-Router inklusive Dial-on-Demand für das lokale Netzwerk zu konfigurieren.

Wenn es darum geht, mehrere Rechner ins Internet zu bringen, gibt es zwei Lösungen. Die eine, jedem Rechner ein Modem und eine Einwählverbindung zu spendieren, kostet allein durch die Telefongebühren eine ganze Menge Geld. Komfortabler und billiger geht es, wenn sich nur einer der Rechner ins Internet einwählt und die anderen über diesen Rechner auf das Web zugreifen. Dieser zentrale Rechner wird im Allgemeinen als Gateway oder Router bezeichnet. Mit dem richtigen Know-how kann man Linux so konfigurieren, dass es als Gateway agiert. Zudem lässt sich seine Rolle Schritt für Schritt so ausbauen, dass es später als Proxy und sogar als Firewall dient. Damit sparen Sie nicht nur Geld, Sie sichern auch gleichzeitig Ihr lokales Netz gegenüber Eindringlingen ab.

Bevor wir uns jedoch in die Tiefe der Netzwerk- und Systemkonfiguration begeben, ein paar Begriffserläuterungen. Falls Sie in punkto Netzwerkgrundlagen schon das nötige Grundwissen über Gateways und Dial-up-Router mitbringen, können Sie sofort zur Konfiguration schreiten, indem Sie die folgenden Abschnitte überspringen und gleich bei *IP-Masquerade aktivieren* einsteigen.

3.3.1 Dial-up-Router mit Dial-on-Demand

Als Internet-Gateway (= IP-Router) stellt das Linux-System ein Tor zwischen Internet und LAN dar. Das Gateway ist - per Stand- oder Wählleitung - an das Internet angebunden. Die Clients im lokalen Netzwerk selbst haben keinen direkten Zugang zum Internet.

Die Clients werden nun so konfiguriert, dass Sie das Linux-System als Standard-Gateway verwenden. Bei jeder Netzadresse, die außerhalb des lokalen Netzes (also im Internet) liegt, wird das Linux-System kontaktiert. Dieses System leitet die Anfrage des Client ins Internet weiter.

Dieses Weiterleiten der Anfrage nennt man *IP-Forwarding*. Bei dem hier beschriebenen Verfahren wird dem Internet vorgegaukelt, dass die Anfrage vom Gateway selbst kommt. Die lokale IP-Adresse des Client-Systems wird hinter der Internet-IP-Adresse des Gateways versteckt. Für das Internet scheint es so, als ob das Gateway der einzige Computer sei, der das Internet verwendet.

Der Ausbau des Internet-Gateways zum Dial-up-Router mit *Dial-on-Demand* fügt dem System noch die Fähigkeit hinzu, bei Bedarf eine Verbindung über Wählleitung - zum Beispiel über Modem oder ISDN - herzustellen. Immer, wenn Internet-Adressen von lokalen Hosts beim Gateway angefragt werden, wählt sich das System eigenständig ins Internet ein und stellt so die Verbindung her. Erfolgen über einen gewissen Zeitraum keine Anfragen mehr, legt das Linux-System wieder auf.

3.3.2 IP-Masquerade

Der Linux-Kernel stellt ein Feature namens *IP-Masquerade* zur Verfügung, mit dem sich das System im Handumdrehen in einen IP-Router verwandeln lässt. Mit IP-Masquerade können Hosts im lokalen Netz quasi unsichtbar - verborgen hinter dem Gateway - das Internet verwenden.

Das Funktionsprinzip von IP-Masquerade basiert darauf, den Netzwerkverkehr zu „belauschen". Die ankommenden Datenpakete des lokalen Netzwerks werden dahingehend überprüft, ob der Adressat ein Host im Internet ist. Trifft dies zu, wird der IP-Header des Datenpakets so manipuliert, dass als Sender nicht mehr der Host aus dem lokalen Netzwerk in Erscheinung tritt, sondern das Gateway. Statt der IP-Adresse des Hosts aus dem Intranet wird jetzt die Internet-IP-Adresse des Gateways in den Header des Datenpakets eingesetzt. Daraufhin wird das so abgeänderte Datenpaket ins Internet entlassen. Die Folge: Rechner im Internet sehen immer nur die Adresse des Gateways.

Bei Rückantworten aus dem Internet sieht das Gateway in seinen internen Tabellen nach, für welchen Host im lokalen Netz das ankommende Datenpaket bestimmt ist. Jetzt wird der IP-Header des Pakets bearbeitet und als Adressat das Gateway eliminiert und stattdessen der lokale Host eingesetzt. Anschließend geht das Paket auf die Reise ins Intranet und findet sein Ziel.

3.3.3 IP-Masquerade aktivieren

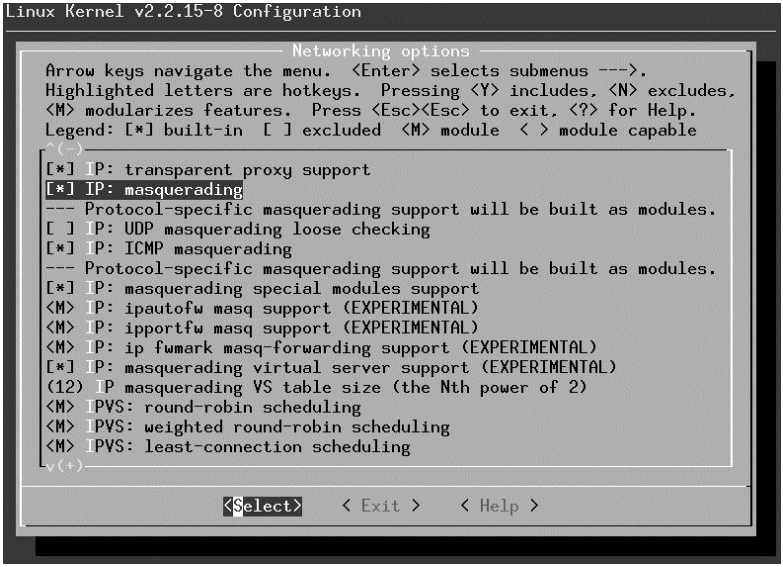
In den meisten Distributionen ist IP-Masquerade im Kernel bereits aktiviert. Wenn das bei Ihnen nicht der Fall ist, kompilieren Sie zunächst einen neuen Kernel. Dazu wechseln Sie in das Verzeichnis */usr/src/linux* und starten die Kernel-Konfiguration entweder durch

```
make menuconfig
```

oder unter X11 im Shell-Fenster durch

```
make xconfig
```

Für IP-Masquerade benötigen Sie einige Features des Kernels. Neben der grundlegenden Netzwerkfunktionalität für TCP/IP schalten Sie unter *Networking options* die Option *IP: firewalling* ein. Danach aktivieren Sie durch die Einstellung *IP: masquerading* das eigentliche IP-Masquerading.



Die Grundlage: Im Kernel-Config müssen Sie IP-Masquerade aktivieren und einen neuen Kernel kompilieren.

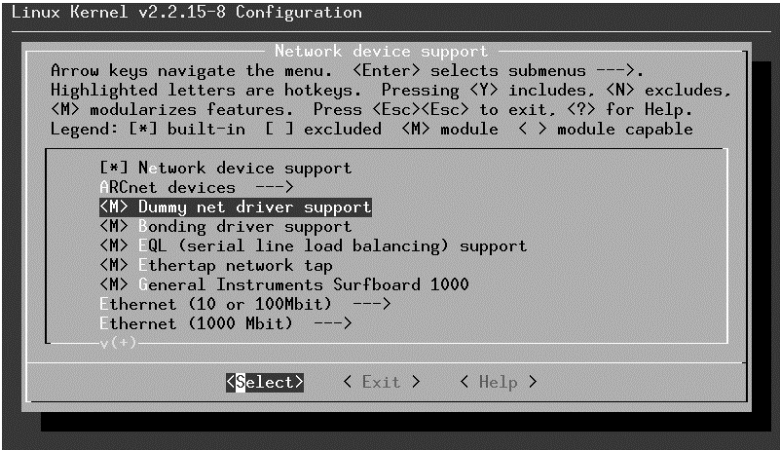
Nun folgen einige teilweise optionale Zusatz-Features von IP-Masquerading. Hier sollten Sie die beiden Punkte *IP: ICMP masquerading* und *IP: masquerading special modules support* aktivieren.

3.3.4 Zusätzliche Optionen

Anschließend binden Sie die folgenden Optionen als Kernel-Module ein:

- IP: ipautofw masq support
- IP: ipportfw masq support
- IP: ip fwmark masq-forwarding support

Hinweis: Stellen Sie zudem sicher, dass der Netzwerktreiber für Dummy-Geräte aktiviert ist. Sie finden diesen in der Kategorie *Network device support* als *Dummy net driver support*.



Wichtig: Binden Sie auf jeden Fall den Dummy Net Driver ein.

Danach verlassen Sie die Kernel-Konfiguration und speichern die Änderungen. Nun gilt es, das Kompilieren des Kernels und seiner Module vorzubereiten:

```
make dep && make clean
```

Legen Sie jetzt eine leere formatierte Diskette in Ihr Floppy-Laufwerk ein und geben Sie *make bzdisk* ein. Damit erhalten Sie eine Bootdiskette, falls irgendetwas schief gehen sollte.

Anschließend kompilieren Sie den Kernel und die Module:

```
make bzImage && make modules
```

Nach der Kaffeepause installieren Sie die Module:

```
make modules_install
```

Den neu gewonnenen Kernel kopieren Sie mit dem Befehl

```
cp -f arch/i386/boot/bzImage /boot/myKernel
```

ins Verzeichnis */boot*.

3.3.5 IP-Masquerade einrichten

Nachdem der Kernel mit IP-Masquerading nachgerüstet ist, können Sie endlich in medias res gehen. Die Konfiguration von IP-Masquerade erfolgt in mehreren Schritten: a) Aktivieren von bestimmten IP-Masquerading-Funktionen durch Laden der entsprechenden Kernel-Module. b) Aktivieren von IP-Masquerading selbst durch Einschalten von IP-Forwarding.

- Aktivieren von bestimmten IP-Masquerading-Funktionen durch Laden der entsprechenden Kernel-Module.
- Aktivieren von IP-Masquerading selbst durch Einschalten von IP-Forwarding.
- Setzen der Regeln, um festzulegen, welche Hosts über das Gateway ins Internet dürfen.

Die Konfiguration von IP-Masquerading sollten Sie in die Init-Scripts eintragen. Hier eignet sich ein Script, das als Letztes aufgerufen wird. In der Regel sollte dies */etc/rc.d/rc.local* sein. Bei einigen Distributionen liegt ein vergleichbares Script jedoch an einer anderen Position. Unter SuSE Linux finden Sie dieses Script beispielsweise unter */sbin/init.d/boot.local*.

Fügen Sie die folgenden Zeilen am Ende dieses Scripts ein. Auf diese Weise wird das IP-Masquerading automatisch beim Booten des Systems aktiviert, und der Linux-PC kann sofort nach dem Start seine Aufgabe als Gateway wahrnehmen.

3.3.6 Module laden

Um die Module beim Systemstart laden zu lassen, tragen Sie folgende Zeilen in das Start-Script ein:

```
/sbin/depmod -a
/sbin/modprobe ip_masq_ftp
/sbin/modprobe ip_masq_raudio
/sbin/modprobe ip_masq_irc
/sbin/modprobe ip_masq_cuseeme
/sbin/modprobe ip_masq_vdolive
```

Die einzelnen Module, die hier durch *modprobe* geladen werden, stellen jeweils ein bestimmtes Funktionsspektrum zur Verfügung. Falls Sie die eine oder andere Funktion nicht benötigen, können Sie das betreffende Modul auch streichen.

Modulübersicht	
Modul	**Funktion**
ip_masq_ftp	Ermöglicht, FTP über das Internet-Gateway zu nutzen.
ip_masq_raudio	Schaltet Audio-Übertragung via IP-Router ein.
ip_masq_irc	Internet Relay Chat (IRC) wird damit über IP-Masquerading möglich.
ip_masq_cuseeme	Video-Konferenzen über CU-SeeMe können durch dieses Modul genutzt werden.
ip_masq_vdolive	Aktiviert Live-Video über IP-Masquerading.

3.3.7 IP-Forwarding aktivieren

Um das IP-Masquerading einzuschalten, setzen Sie ein spezielles Signal im */proc*-Dateisystem. Verwenden Sie hierzu einfach die Befehlszeile

```
echo "1" > /proc/sys/net/ipv4/ip_forward
```

hinter den Befehlen zum Laden der Module im Start-Script.

Unter SuSE Linux verwenden Sie statt dieses Befehls die Variable *IP_FORWARD* in der Konfigurationsdatei */etc/rc.config*. Suchen Sie in dieser Datei den Eintrag *IP_FORWARD=no* und ersetzen Sie das *„no“* durch ein *„yes“*.

Auf diese Weise wird das IP-Forwarding möglich, und IP-Masquerading kann seine Aufgabe erfüllen.

3.3.8 Regeln erstellen

Beim IP-Masquerading legen Sie über die Konfiguration der Firewall explizit fest, wer das IP-Routing verwenden darf. Könnte jeder Ihr Gateway verwenden, wären Sie wahrscheinlich des Hackers bester Freund. Dieser könnte nämlich aus dem Internet über Ihr Gateway Verbindungen zu anderen Rechnern im Internet aufbauen. Die Folgen wären fatal, da nur IHR Gateway in Erscheinung tritt. Damit wären Sie der erste Verdächtige, wenn bei einem Großkonzern plötzlich die geheimen Projektdaten aus dem System verschwinden würden.

3.3.9 Sicherung des Gateways

Über ipchains können Sie entweder kompletten Subnetzen oder einzelnen Hosts den Zugriff auf das Internet über das Gateway gewähren. Zunächst sollten Sie alle Zugriffe auf das Gateway über den Befehl

```
ipchains -P forward DENY
```

sperren. Damit ist gar kein Zugriff auf das Gateway mehr gestattet, und alle Hacker sind aus Ihrem System ausgesperrt - Sie selbst jedoch auch.

Jetzt bauen Sie nach und nach die Zugriffsrechte neu auf. Wollen Sie beispielsweise allen Hosts des Subnetzes 192.168.0.0 mit der Netzmaske 255.255.255.0 den Zugriff auf das Gateway ermöglichen, so tragen Sie den Befehl

```
ipchains -A forward -s 192.168.0.0/24 -j MASQ
```

oder

```
ipchains -A forward -s 192.168.0.0/255.255.255.0 -j MASQ
```

ein. Wie Sie sehen, können Sie die Netzmaske entweder in Punktnotation oder als Bitwert (hier 24) angeben.

Wollen Sie Netzwerke nicht komplett freigeben, sondern gezielt steuern, welche Hosts das Internet verwenden dürfen, können Sie auch die IP-Adressen von Hosts bei ipchains angeben. Sie führen dann einfach die Host-IP-Adressen mit der Bitmaske 32 an. Auf diese Weise haben nur die aufgelisteten Hosts über den IP-Router Zugriff auf das Internet. Um beispielsweise den Hosts 192.168.0.13 und 192.168.0.20 den Zugriff zu gestatten, geben Sie die Befehle

```
ipchains -A forward -s 192.168.0.13/32 -j MASQ
ipchains -A forward -s 192.168.0.20/32 -j MASQ
```

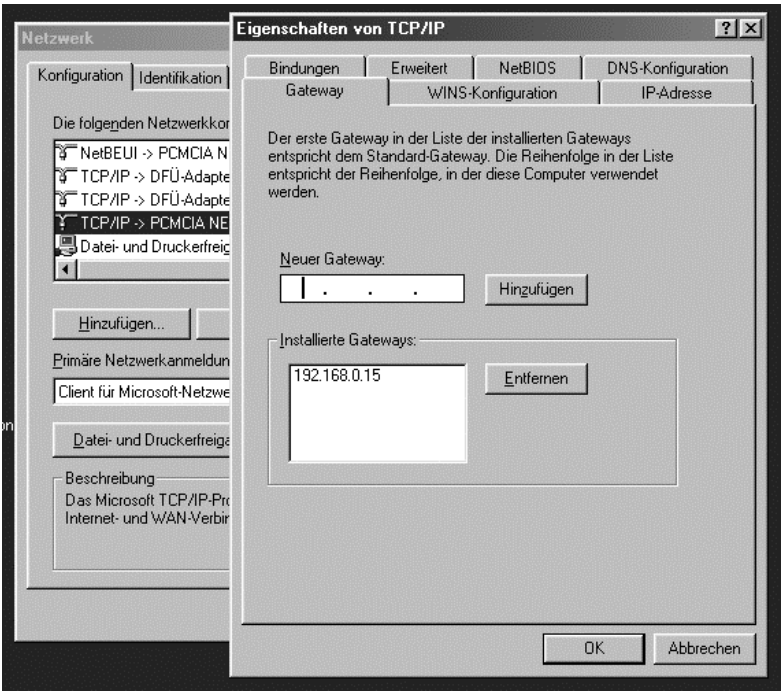
ein.

Weitere ausführliche Informationen zu ipchains finden Sie in der zugehörigen Man-Page, die Sie über *man ipchains* aufrufen.

3.3.10 Windows-Clients einrichten

Damit die Clients über das Gateway ins Internet gelangen können, muss bei diesen als Standard-Gateway die IP-Adresse des Linux-Rechners eingestellt werden.

Auf Windows-9x-Systemen öffnen Sie die Netzwerkkonfiguration in der Systemsteuerung. Wählen Sie dann den Netzwerkadapter aus (in der Regel Ihre Ethernet-Karte), über die Sie an das Gateway angebunden sind. Klicken Sie anschließend auf *Eigenschaften* und tragen Sie im Reiter *Gateway* die IP-Adresse des Linux-Rechners über die Schaltfläche *Hinzufügen* ein.

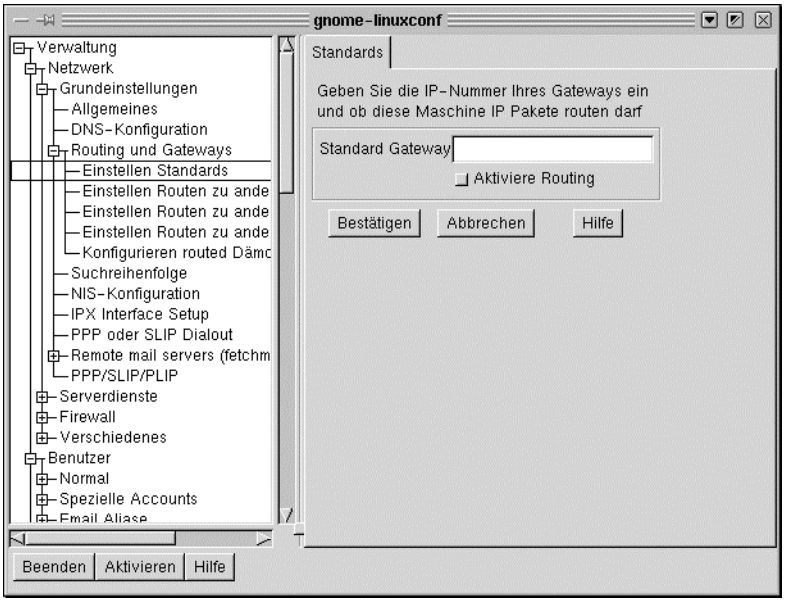


Client-Konfiguration: Tragen Sie unter Windows den IP-Router als Standard-Gateway ein.

Unter Windows NT 4.0 gehen Sie über die Eigenschaften der Netzwerkumgebung auf den Reiter *Protokolle* und wählen *TCP/IP* aus. Klicken Sie auf *Eigenschaften*. Verfügt das System über mehrere Netzwerkadapter, wählen Sie in der

3.3.11 Linux-Clients einrichten

Das Konfigurieren der Linux-Systeme ist kompliziert und einfach zugleich. Das Komplizierte daran ist schlicht, dass es keinen einheitlichen Weg für alle Distributionen gibt. Oder genauer gesagt: Der einheitliche Weg läuft über das Anlegen von Routes über das Kommando *route* oder im Falle von *routed* über die Datei */etc/route*. Dies lässt zwar das Herz eines jeden Unix-Gurus höher schlagen, den Admin hingegen schreckt es ab.



Linux-Client: Einstellen eines Standard-Gateways per linuxconf.

Konfigurationstools wie *linuxconf* oder YaST machen das Einrichten der Linux-Hosts sehr komfortabel. Der Nachteil ist, dass die Konfigurationstools nicht einheitlich aufgebaut sind. Wir zeigen am Beispiel von linuxconf und YaST, wie der generelle Ablauf vonstatten geht. Sollte Ihre Distribution ein anderes Konfigurationsprogramm bereitstellen, sehen Sie bitte in der Dokumentation nach, wie Sie hier ein Standard-Gateway einstellen.

Unter *linuxconf* wählen Sie *Verwaltung/Netzwerk/Grundeinstellungen/Routing und Gateways/Einstellen Standards*. Hier können Sie den IP-Router als Standard-Gateway eintragen.

Bei SuSEs YaST wählen Sie *Administration des Systems/Netzwerk konfigurieren/Netzwerk-Grundkonfiguration*. Wählen Sie dort den Netzwerkadapter, über den Sie mit dem Gateway verbunden sind, und drücken Sie F6. Geben Sie anschließend im entsprechenden Feld die IP-Adresse des Gateways an.

3.3.12 Wählen nach Bedarf

Die Konfiguration eines Dial-up-Routers mit Dial-on-Demand ist - zumindest was ISDN angeht - über grafische Tools wie KISDN keine Zauberei. Allerdings empfiehlt es sich, IP-Router ganz ohne X11 aufzubauen, da mit der grafischen Oberfläche nur eine unnötige Fehlerquelle hinzukommt. Wozu ein grafisches System mitschleppen, das ansonsten ohnehin nicht verwendet wird?

Die Konfiguration des Systems kann auch textbasiert über Scripts erfolgen. Vorteil: Das System verbraucht weniger Ressourcen, und etwaige Fehler im grafischen System sind ausgeschlossen. Daher lautet das Motto hier: Back to the roots. Tippakrobatik auf der Shell ist angesagt.

Hinweis: Im Folgenden wird davon ausgegangen, dass Ihr Provider die Zugangsauthentifizierung per PAP ermöglicht. Sollte Ihr Provider ein anderes Verfahren, wie beispielsweise CHAP, verwenden, ist die Konfiguration entsprechend anzupassen.

3.3.13 Verbindung über Modem oder ISDN-TA

Um eine Verbindung über ein Modem oder einen ISDN-Terminaladapter aufzubauen, benötigen Sie das Programm *wvdial* und den PPP-Daemon *pppd*. Installieren Sie hierzu die entsprechenden Pakete Ihrer Distribution.

wvdial nutzt *pppd*, um Verbindungen herzustellen. Dabei wählt *wvdial* zunächst die Nummer des Providers und erstellt dann eine PPP-Connection über *pppd*. Die Informationen bezieht *wvdial* dabei aus der Datei */etc/wvdial.conf*.

In dieser Datei können Sie Sections anlegen, über die Sie beim Aufruf von *wvdial* bestimmte Anwahlsequenzen erzeugen. So können Sie in den Sections die Verbindungsdaten für verschiedene Provider ablegen und durch die Auswahl

```
[Dialer Defaults]
Modem = /dev/ttyS1
Baud = 115200
Init1 = ATZ
Init2 = AT&F\N3
Phone = 0191011
Username = 123456789128888308387 39#0001
Password = Passwort
Idle = 120
```

Der Eintrag *Modem* legt fest, an welcher Schnittstelle Ihr Modem oder ISDN-TA angeschlossen ist. *Baud* gibt die Übertragungsgeschwindigkeit des Geräts an. Mit *Phone* legen Sie die zu wählende Telefonnummer fest. *Username* und *Password* sind die Zugangsinformationen, die Sie von Ihrem Provider erhalten haben.

Über die Zeilen *InitX* (mit X zwischen 1 und 9) können Sie Initialisierungsstrings an das Modem beziehungsweise Ihren ISDN-TA senden.

Wichtig ist hier auf jeden Fall für Dial-on-Demand die Einstellung *Idle*. Über diesen Wert legen Sie fest, nach wie vielen Sekunden ohne Datenübertragung die Verbindung getrennt werden soll.

3.3.14 PAP-Konfiguration

Obwohl Sie bereits in */etc/wvdial.conf* die Zugangsinformationen festgelegt haben, müssen Sie trotzdem unter */etc/pap-secrets* diese Daten erneut eingeben, damit die Verbindung zu Stande kommen kann. Legen Sie diese Datei mit folgenden Einträgen an:

```
# User  Server  Password
"123456789128888308387 39#0001" * "Passwort"
```

Beachten Sie, dass Sie hier Ihre Benutzerdaten eingeben müssen. Auch die Anführungszeichen dürfen Sie nicht vergessen.

Damit Sie über *wvdial* Dial-on-Demand-Funktionalität erhalten, verwenden Sie am besten das Script *wvdial.dod*. Dieses Script wurde von SuSE programmiert und steht auf dem SuSE-Server (http://sdb.suse.de/sdb/de/html/hoe_wv_dod_start.html) zur freien Verfügung. Dieses Script funktioniert nicht nur mit SuSE Linux, sondern auch mit anderen Distributionen.

Kopieren Sie das Script in die Datei */sbin/wvdial.dod*. Setzen Sie danach die Attribute dieser Datei durch das Kommando

```
chmod 744 /sbin/wvdial.dod
```

neu. Anschließend können Sie das Dial-on-Demand durch

```
wvdial.dod start
```

aktivieren und durch

```
wvdial.dod hangup
```

die Verbindung trennen. Sie beenden Dial-on-Demand durch den Befehl

```
wvdial.dod stop
```

3.3.15 Verbindung über ISDN

Die Verbindung via ISDN lässt sich einfach über Tools wie KISDN einrichten. Bei diesen Tools können Sie sogar per Knopfdruck Dial-on-Demand einschalten. Der Einsatz eines solchen Tools hat auch hier den Nachteil, dass die grafische Oberfläche X11 mitgeschleppt wird. Daher hier die Variante, mit der ein IP-Router ressourcensparend hochgezogen werden kann. Für den im Folgenden beschriebenen Weg zum Aufbau eines ISDN-Dial-up-Routers benötigen Sie nur die isdn4k-utils (www.isdn4linux.de). Für den Fall, dass Sie mit einer ISA-PnP-Karte arbeiten, installieren Sie zusätzlich die *isapnptools*.

Die Konfiguration eines ISDN-Systems von Hand ist umständlich. Deshalb beinhaltet dieser Abschnitt das Script *isdn.dod*, mit dem Sie wie bei einem Init-Script den ISDN-Dienst mit *isdn.dod start* und *isdn.dod stop* starten und beenden. So können Sie beim Booten, bevor Sie IP-Masquerading aktivieren, das Dial-on-Demand einschalten. Fügen Sie in das Start-Script den Befehl

```
isdn.dod start
```

vor die Masquerading-Konfiguration ein.

```
# User  Server  Password
"Benutzerkennung" * "Passwort"
```

Zusätzlich müssen Sie in die Datei */etc/ppp/ip-down* (oder */etc/ppp/ip-down.local*) folgende Zeile am Ende hinzufügen:

```
/sbin/route add default dev ippp0
```

3.3.16 Das Script isdn.dod

Kopieren Sie das Script in die Datei */sbin/isdn.dod*. Setzen Sie danach die Attribute dieser Datei durch das Kommando

```
chmod 744 /sbin/isdn.dod
```

neu.

```
#!/bin/bash

MSN=IhreMSN
ISPPN=ISPTelefonnummer
USERNAME=IhrBenutzername
CARDTYPE=Kartentyp
IRQ=OptionalerIRQ
IOBASE=OptionaleIOAdresse
TIMEOUT=WannAuflegen
if ["$IRQ"  != "" ] ; then
    IRQ="irq=$IRQ"
fi

if ["$IOBASE"!= "" ] ; then
    IOBASE="io=$IOBASE"
fi
case "$1" in
  start)
    modprobe hisax type=$CARDTYPE id=hisax1 protocol=2 $IRQ
$IOBASE
    hisaxctrl hisax1 1 4
    isdnctrl addif ippp0
    isdnctrl eaz ippp0 $MSN
    isdnctrl addphone ippp0 out $ISPPN
    isdnctrl secure ippp0 on
    isdnctrl l2_prot ippp0 hdlc
    isdnctrl l3_prot ippp0 trans
    isdnctrl encap ippp0 syncppp
    isdnctrl huptimeout ippp0 $TIMEOUT
    isdnctrl dialmax ippp0 5
    isdnctrl bind ippp0
    isdnctrl pppbind ippp0 0
    ifconfig ippp0 1.1.1.1 pointopoint 1.1.1.1 up
    isdnctrl verbose 3
    isdnctrl dialmode ippp0 auto
    ipppd /dev/ippp0 noipdefault user $USERNAME defaultroute
    route add default dev ippp0
    ;;
  stop)
    isdnctrl hangup ippp0
    killall ipppd
    ifconfig ippp0 down
    modprobe -r hisax
    ;;
  restart)
    $0 stop
    $0 start
    ;;
  *)
    echo ´Usage: isdn.dod {start|stop|restart}´
    exit 1
    ;;
esac

exit 0
```

3.3.17 Fazit

Damit Sie mit den Webbrowsern Ihrer Clients nun auch wirklich im Internet surfen können, müssen Sie dem System die DNS-Server für Ihren Provider mitteilen. Das erreichen Sie einfach, indem Sie in die Datei */etc/resolv.conf* Zeilen eingeleitet mit *nameserver* eintragen, hinter die Sie die IP-Adressen der DNS-Server stellen. Beispielsweise würden diese Zeilen die DNS-Server auf 194.25.2.129 und 194.25.0.125 setzen:

```
nameserver 194.25.2.129
nameserver 194.25.0.125
```

Dial-on-Demand kann sich sehr schnell als Fass ohne Boden erweisen, wenn ständig unbeabsichtigt Verbindungen ins Internet aufgebaut werden. Auf Ihren Hosts sollten Sie daher in den Webbrowsern die Startseite entweder auf eine leere Seite oder eine lokale Startseite (= auf Festplatte) setzen. Ist hier eine URL eingetragen, wird immer beim Start des Browsers das Internet kontaktiert, und Dial-on-Demand wird aktiv.

Ein weiterer Kostenfaktor kann der E-Mail-Client sein. Diesen sollten Sie so konfigurieren, dass er nicht alle zehn Minuten nachsieht, ob neue E-Mails eingetroffen sind. Dann müssen Sie jedoch von Hand nachsehen, ob Sie E-Mails erhalten haben.

Alternativ bietet es sich an, über das Gespann *sendmail/fetchmail* einen E-Mail-Server aufzubauen. Dieser E-Mail-Server lässt sich über *cron per Timer* steuern. Sie können ihn beispielsweise so konfigurieren, dass er zu jeder vollen Stunde die E-Mails aus dem Internet abholt und neu geschriebene versendet.

tecCHANNEL-Links zum Thema	Webcode
Masquerading mit Linux	a707
Linux als Firewall	a695
Firewall-Grundlagen	a682
Linux als Windows-Printserver	a248
So funktioniert TCP/IP	a209
Java Virtual Machine unter Linux	a776
Tux goes Biz	a654

3.4 Proxy-Server unter Linux

Viele Administratoren schalten zwischen Internet und Firmennetz einen Web-Proxy. Dieser bietet nicht nur die Möglichkeit, den Zugang auf bestimmte Internet-Angebote zu beschränken, sondern verringert auch das Datenvolumen. Bei Internet-Zugängen, die volumenabhängig abgerechnet werden, lassen sich so Kosten sparen. Außerdem erhöht sich die Zugriffsgeschwindigkeit auf bereits zwischengespeicherte Daten.

Ein Proxy-Server hat im Netzwerk eine Vermittlungsfunktion. Er nimmt Anfragen von den Anwendern entgegen, lädt Daten aus dem Internet und leitet diese an den User weiter. Dabei können sowohl HTTP- als auch FTP-Inhalte über einen Proxy angefordert werden.

Der Server legt alle angeforderten Daten in einem Cache ab. Bevor der Proxy Daten aus dem Internet holt, wird überprüft, ob diese bereits im Cache vorliegen und noch aktuell sind. Ist dies der Fall, liefert der Server die Internet-Inhalte aus dem lokalen Speicher. Bei Daten, die häufig von verschiedenen Benutzern abgerufen werden, ergibt sich hieraus eine deutliche Reduzierung des Übertragungsvolumens. Zudem liefert der Cache die Inhalte schneller, als wenn man diese aus dem Internet lädt.

Wir beschreiben in diesem Artikel die Installation und Konfiguration des beliebten Linux-Proxys Squid. Dazu dient folgendes Szenario: Der Proxy soll ausschließlich authentifizierten Benutzern den Zugriff auf das Internet gewähren, deren Rechner zusätzlich über die IP-Adresse auf dem Server freigegeben ist. Das Abrufen von Internet-Inhalten soll nur zu bestimmten Zeiten möglich sein und bekannte Direct-Brokering-Seiten werden gesperrt.

3.4.1 Proxy-Server: Zugriffskontrollen

Ein Proxy-Server lässt sich so einstellen, dass nur bestimmte Anwender auf diesen zugreifen können und bestimmte Internet-Inhalte gesperrt sind. Es gibt drei Arten der Zugriffskontrolle:

- Anonymous Proxy: Anfragen werden von beliebigen Rechnern und Anwendern entgegengenommen.
- Proxy-Server mit Authentifizierung: Erst nach erfolgreicher Authentifizierung des Anwenders durch Benutzername und Passwort werden Anfragen ausgeführt.
- Proxy-Server mit Zugriffsregeln (ACL / Access Control List): Diese legen beispielsweise fest, zu welchen Zeiten bestimmte Daten aus dem Internet geladen werden dürfen.

Abhängig von den Proxy-Einstellungen werden bei Zugriffsverletzungen bestimmte Aktionen ausgeführt. Eine unerlaubte Anfrage kann so das Umleiten an eine andere Adresse im internen Netzwerk zur Folge haben.

Forbidden: Versucht ein Anwender, eine gesperrte Seite aufzurufen, erscheint diese Meldung.

3.4.2 Squid: Installation

Die derzeit aktuelle Version 2.4 von Squid kann man unter www.squid-cache.org herunterladen. Nach dem Download entpacken Sie zunächst den Tarball und wechseln in das entsprechende Verzeichnis:

```
tar xzvf squid-2.4.STABLE4-src.tar.gz
cd  squid-2.4.STABLE1-src
```

Nun müssen Sie Squid konfigurieren. Dazu geben Sie folgenden Befehl ein:

```
./configure --prefix=/usr/local/squid
--sysconfdir=/etc/squid
```

Mit dem Schalter *--prefix* legen Sie das Zielverzeichnis fest, mit *--sysconfdir* den Speicherplatz für die Konfigurationsdatei. In unserem Beispiel dürfen Anwen der ausschließlich mit einem gültigen Benutzernamen und Passwort auf den Proxy-Server zugreifen. In den Unterverzeichnissen von *auth_modules* stehen dafür folgende Authentifizierungsmechanismen zur Verfügung: LDAP, MSNT, NCSA, PAM, SMB und getpwnam. Wir verwenden die NCSA-Webserver-Authentifizierung. Dazu führen Sie im entsprechenden Verzeichnis (NCSA) folgende Befehle aus:

```
make
su
make install
exit
```

Das kompilierte Authentifizierungsmodul finden Sie im Ordner *bin*. Allerdings kann Squid mit eigenen Bordmitteln nicht alle Zugriffsbeschränkungen aus

3.4.3 SquidGuard: Berkeley DB Library

Für die Installation von SquidGuard ist die Version 2.x der Berkeley-DB-Library erforderlich. Befindet sich diese nicht auf Ihrem Rechner, können Sie diese unter www.sleepycat.com herunterladen. Entpacken Sie nach dem Download das Archiv und wechseln Sie in das Unterverzeichnis *build_unix* des Berkeley-DB-Verzeichnisses. Nun wird das Script *configure* wie folgt ausgeführt:

```
../dist/configure --prefix=/usr/local
--sysconfdir=/etc/BerkeleyDB
```

Übersetzen und installieren Sie das Programm mit folgenden Befehlen:

```
cd ../dist/configure
make
su
make install
exit
```

Sämtliche erforderlichen Dateien finden Sie unter *usr/local/BerkeleyDB*, die Konfigurationsdateien unter */etc/BerkeleyDB*.

3.4.4 SquidGuard: Installation

Zur Installation von SquidGuard (www.squidguard.org) entpacken Sie den Tarball, wechseln in das neue Verzeichnis und konfigurieren das Programm:

```
tar xzvf squidGuard-1.2.0.tar.gz
cd squidGuard-1.1.4
./configure --prefix=/usr/local/squidGuard
--with-sg-config=/etc/squidGuard.conf
--with-sg-logdir=/var/squidGuard/log
```

SquidGuard wird in das Verzeichnis */usr/local/squidGuard* installiert, die Konfigurationsdatei */etc/squidGuard.conf* eingerichtet und als Log-Verzeichnis */var/squidGuard/log* festgelegt.

Nun können Sie SquidGuard kompilieren und installieren:

```
make
su
make install
exit
```

3.4.5 Konfiguration: Squid

Alle notwendigen Einstellungen wie beispielsweise den Proxy-Port (Default: 3128) oder die Zugriffsbeschränkungen nehmen Sie in einer zentralen Konfigurationsdatei vor: */etc/squid/squid.conf*. In der Regel sind jedoch wenige Änderungen nötig, eine weiter gehende Anpassung ist mit Hilfe der Erklärungen einfach zu realisieren.

Für die spätere Authentifizierung der Anwender ist es wichtig, dass Sie das dafür zuständige Programm angeben und eine ACL vom Typ *proxy_auth* festlegen. Entsprechend den in der Konfiguration angegebenen Optionen müssen Sie als User *root* einen Benutzer und entsprechende Verzeichnisse anlegen sowie die Rechte vergeben:

```
useradd -g nogroup -d /var/squid -c "Proxy-Squid"
-s /bin/bash squid
mkdir /var/squid/cache
mkdir /var/squid/logs
chown -R squid.root /var/squid
htpasswd -c /etc/squid/passwd Benutzername
```

Der letzte Befehl erstellt eine Passwortdatei. Dabei ist es wichtig, dass weitere Benutzernamen mit dem gleichen Befehl ohne die Option *-c* angelegt werden. Nun kann der Cache initialisiert werden:

```
su squid
/usr/local/squid/bin/squid -z
exit
```

Die Zugriffsbeschränkungen richten Sie mit dem Programm SquidGuard ein.

3.4.6 Konfiguration: SquidGuard

Legen Sie zunächst eine Konfigurationsdatei mit den Namen *squidGuard.conf* im Verzeichnis */etc* an. Vom tecCHANNEL-Server (www.tecchannel.de/download/798/squidguard_konfigurationsbeispiel.txt) können Sie sich eine Beispielkonfiguration herunterladen. Diese arbeitet mit einer Negativliste: „Alles was nicht verboten ist, ist erlaubt.“ Wenn Sie eine Positivliste verwenden wollen, muss die Erlaubnis vor den Verboten stehen. Alle Konfigurationsänderungen aktivieren Sie mit dem Befehl:

```
/usr/local/squid/bin/squid -k reconfigure
```

SquidGuard arbeitet anhand von Datenbanken oder Textdateien: Domain-Listen zum Sperren ganzer Domains, URL-Listen zum Sperren spezieller URLs oder reguläre Ausdrücke, mit deren Hilfe Sie bestimmte Muster in den URLs und Domains verbieten. Diese Dateien und auch die Logfiles sollte man in einem zentralen Verzeichnis speichern:

```
mkdir /var/squidGuard
mkdir /var/squidGuard/db
mkdir /var/squidGuard/log
```

Sie können verschiedene Datenquellen parallel verwenden, etwa um die Daten thematisch zu gliedern:

```
mkdir /var/squidGuard/db/finance
```

Vor einem Vergleich wandelt SquidGuard alle Zeichen in Kleinbuchstaben um. Daher muss darauf keine Rücksicht genommen werden. Legen Sie nun eine Textdatei mit dem Namen *finance.domains* unter */var/squidGuard/db/finance* an. Der Dateiname ist frei wählbar und muss nicht den hier verwendeten gleichen. In diese Datei fügen Sie nun alle Domain-Namen ein, die für den Anwender gesperrt werden sollen, für jede Domain eine separate Zeile, beispielsweise:

```
comdirect.de
dab.de
```

3.4.7 Zugriffskontrolle: Filtern von URLs

Sollen nur bestimmte URLs und keine kompletten Domains gesperrt werden, legen Sie im Datenbankordner die Datei *finance.urls* an. Verwenden Sie für jede URL eine eigene Zeile, wobei Sie das Protokoll (http, ftp, etc.) nicht mit angeben müssen. Wollen Sie zum Beispiel den Zugriff auf den Börsenticker *http://financial.domain.de:3456/applets/ticker.class* sperren, wird daraus folgender Eintrag:

```
financial.domain.de/applets/ticker.class
```

Es kann vorkommen, dass eine Webseite unter mehreren URLs erreichbar ist, beispielsweise *www.financial.domain.de* und *web.financial.domain.de*. Hier vereinfacht SquidGuard dem Administrator die Arbeit, denn das Verbot der oben

genannten URL schließt auch alle Subdomains ein. Ein Verbot aller Objekte einschließlich der Unterverzeichnisse erreichen Sie durch folgende Zeile:

```
financial.domain.de/verbotenes_Verzeichnis
```

3.4.8 Zugriffskontrolle: Filtern von Stichwörtern

Möchten Sie alle Domains oder URLs sperren, welche die Buchstabenkombination *depot* enthalten, legen Sie dazu die Datei *finance.regex* an. Verwenden Sie für jedes zu sperrende Stichwort eine eigene Zeile. Allerdings sind auch Suchanfragen bei Suchmaschinen betroffen, da in der Regel die Anfrage in der URL steht. Es besteht jedoch immer die Gefahr, dass auch erwünschte URLs gesperrt werden. Ferner reduziert sich die System-Performance, da jede URL gegen diese Stichwörter geprüft wird. Aus diesen Gründen sollten Sie reguläre Ausdrücke nur sparsam verwenden.

Bei großen Domain- und URL-Listen sollte der Zugriff auf die einzelnen Sätze nicht über eine Textdatei erfolgen. Die Zugriffsgeschwindigkeit erhöht sich durch die Verwendung von Berkeley-DB-Datenbanken erheblich. Als Vorlage für die Datenbank benutzen Sie die angelegten Textdateien, die Sie mit dem Befehl:

```
/usr/local/squidGuard/bin/squidGuard -C Dateiname
```

umwandeln. SquidGuard verwendet automatisch die Datenbankversion, wenn eine solche neben einer Textdatei vorhanden ist. Sollte SquidGuard beim Erzeugen der Datenbanken nicht zum Prompt zurückkehren, drücken Sie die Tastenkombination [Strg]+[C], um das Programm abzubrechen. Dem Fehler kommen Sie auf die Spur, indem Sie das Logfile einsehen, das nach der hier vorgestellten Konfiguration unter */var/squidGuard/log/squidGuard.log* zu finden ist. In den meisten Fällen handelt es sich um einfache Tippfehler in der Konfigurationsdatei.

3.4.9 Zugriffskontrolle: Administration

Zum Verwalten der Datensätze wird ein so genanntes Diff-File verwendet. Geben Sie darin die Datensätze in folgender Form an:

```
+domain.hinzufügen.de
-domain.aus.Datenbank.entfernen.de
```

Die Diff-Datei muss im gleichen Verzeichnis wie die betreffende Datenbank liegen und zwingend *Name-der-Datenbank.diff* heißen. Mit dem Befehl:

```
/usr/local/squidGuard/bin/squidGuard -u
```

werden alle Datenbanken aktualisiert. Die Änderungen gelten sofort, so dass Sie Squid nicht neu starten zu brauchen. Nach der Aktualisierung löschen Sie das Diff-File. Die Verzeichnisse und Dateien gehören dem Benutzer *squid,* und die Datenbankdateien sollten auch nur von diesem Benutzer und der entsprechenden Gruppe lesbar sein. Dadurch wird vermieden, dass jeder Anwender die komplette Liste der verbotenen Sites einsehen kann.

```
chown -R squid.root /var/squidGuard
chmod 640 /var/squidGuard/db/*/*
```

Zuletzt kopieren Sie das CGI-Script *squidGuard.cgi* aus dem Verzeichnis *sample* des Quellverzeichnisses in den CGI-Ordner des Webservers. Geben Sie der Datei zudem ausführbare Rechte. Ferner sind im Script wenige Einstellungen anzupassen, wie die E-Mail-Adresse, an die sich Benutzer wenden können, falls sie eine Sperrung für ungerechtfertigt halten. Sie haben die Möglichkeit, ein eigenes Script zu schreiben, das Ihren Anforderungen besser entspricht.

3.4.10 Proxy-Server ins System einbinden

Als User *root* starten Sie den Proxy-Server mit folgendem Befehl:

```
/usr/local/squid/bin/squid
```

Über den syslog-Daemon in der Datei */var/squid/logs/cache.log* wird der Start mitprotokolliert. Bei der Fehlersuche liefert die Datei hilfreiche Informationen. Den Server beenden Sie mit

```
/usr/local/squid/bin/squid -k shutdown
```

Zur Aktivierung des Proxy-Servers beim Systemstart ist ein Init-Script hilfreich. Ein Beispiel finden Sie auf unserem Server (www.tecchannel.de/download/798/init_script_proxy.txt). Unter SuSE speichern Sie dieses unter dem Namen */sbin/init.d/squid* und setzen Links des Runlevel-Verzeichnissen auf diese Datei:

```
cd /sbin/init.d/rc2.d
ln -s ../squid ./S20squid
ln -s ../squid ./K10squid
```

3.4.11 Webalizer: Überwachen des Proxys

Nachdem Sie den Proxy-Server installiert und fertig konfiguriert haben, stellt sich die Frage nach der Überwachung. Dazu gibt es verschiedene Ansätze, die jeweils ihre Vor- und Nachteile haben. Die einfachste Art der Überwachung ist, sich die neuen Einträge des Squid-Logfiles auf einer Textkonsole ausgeben zu lassen. Dazu verwenden Sie das Kommando:

```
tail -f /var/squid/logs/access.log
```

Sobald ein neuer Eintrag im Logfile erzeugt wird, erscheint er auch auf der Konsole. Dadurch erhalten Sie einen aktuellen Überblick über den Verkehr. Entdecken Sie dabei Adressen, die gesperrt werden sollten, können Sie diese umgehend in die Datenbanken aufnehmen. Diese Form der Auswertung ist allerdings bezüglich der Proxy-Nutzung nur begrenzt aussagekräftig. Abhilfe schaffen der Cachemanager von Squid oder externe Tools wie Webalizer (www.webalizer.com). Das Programm dient hauptsächlich zum Auswerten von Webserver-Logfiles. In der aktuellen Version kann es auch Logfiles von Squid auswerten. Es erstellt unter anderem Übersichten über die Anzahl der Anfragen und übertragenen Bytes sowie Tagesstatistiken.

Verwenden Sie das Programm *configure* mit den Schaltern *--enable-dns* und *--with-language=german*, um für die Statistiken DNS-Namen statt IP-Adressen zu erhalten. In der Datei */etc/webalizer.conf.sample* sind alle Konfigurationsoptionen beschrieben. Kopieren Sie die Datei nach */etc/webalizer.conf* und führen Sie Ihre Änderungen durch. Ein Beispiel steht auf dem tecCHANNEL-Server bereit (www.tecchannel.de/download/798/webalizer_konfigurationsbeispiel.txt).

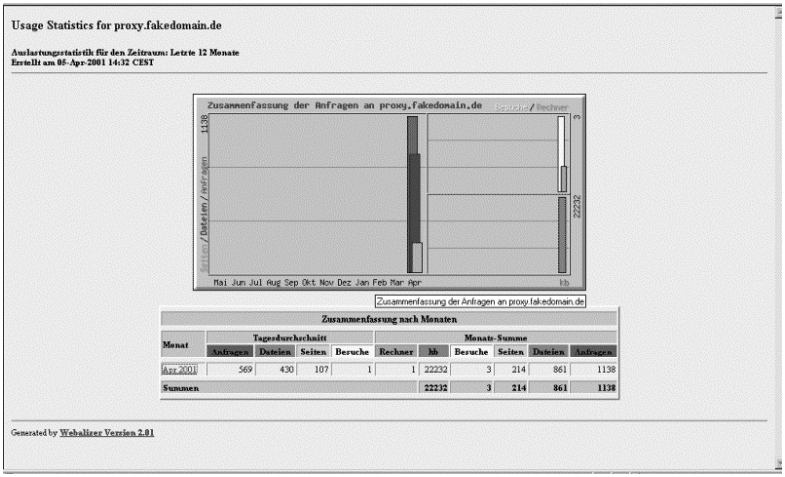
Kopieren Sie nun die Datei */var/squid/logs/access.log* nach */var/squidlogs/access_log* und lassen Sie Squid mit */usr/local/squid/bin/squid -k rotate* eine neue Logdatei anlegen. Führen Sie schließlich das Kommando *webalizer* aus, und die Auswertungen, die Sie mit einem Webbrowser ansehen können, stehen Ihnen zur Verfügung. Es ist empfehlenswert, einen Cron-Job zu erstellen, der die obigen Schritte periodisch ausführt. Dieser Schritt wird hier jedoch nicht weiter beschrieben. Um Schwierigkeiten zu vermeiden, sollten Sie in der Konfiguration von Squid die automatische Rotation mit *logfile_rotate 0* ausschalten und die Umbenennung der Logdateien selbst vornehmen.

3.4.12 Webalizer: Auswertung der Logfiles

Die Auswertung über Webalizer zählt alle Anfragen. Dabei werden auch die Anfragen berücksichtigt, die der Cache selbst ausgeliefert hat. Eine Übersicht über die aktuellen Daten von Squid liefert das CGI-Programm *Cachemanager*. Kopieren Sie die Datei */usr/local/squid/bin/cachemgr.cgi* in das CGI-Verzeichnis des Webservers und setzen Sie die Dateirechte entsprechend der Webserver-

```
http_access allow manager
cachemgr_passwd password all
```

Via Webbrowser greifen Sie über die URL *http://proxy-server/cgi-bin/cachemgr.cgi* auf die Auswertungen zu, wobei der Benutzername *cachemgr* und das Passwort im obigen Beispiel *password* lautet.



Übersichtlich: Webalizer präsentiert Ihnen die kompakten Proxy-Statistiken.

Sie können Informationen von der Auslastung des Festplatten-Cache bis hin zu den Antwortzeiten des Cache ermitteln. Interessant ist vor allem der Punkt „General/Runtime Information“. Die Punkte „Request Hit Ratios“ und „Byte Hit Ratios“ geben Aufschluss darüber, wie viele Anfragen beziehungsweise Bytes aus dem Cache geliefert worden sind. Für diese Objekte wurde keine Verbindung nach außen aufgebaut. Neben diesen Möglichkeiten finden Sie über die Webseiten von Squid verschiedene andere Programme, die jeweils ihre Vor- und Nachteile haben.

3.4.13 Einsatz eines Proxy-Servers und Datenschutz

Mit der Anleitung in diesem Artikel haben Sie einen Proxy-Server komplett konfiguriert, der weit reichende Beschränkungsmöglichkeiten bietet. Dabei ist das System einfach zu administrieren.

Neben der Verwendung Ihrer eigenen Datenbank können Sie so genannte Blacklists über die Homepage von SquidGuard herunterladen. Dabei handelt es sich um vorgefertigte Datenbanken mit Domain-Namen zu Seiten mit zweifelhaften Inhalten. Die Listen können Sie analog zu den hier erklärten Schritten in Ihr System integrieren. Allerdings sollte man beachten, dass die Blacklists von einem Programm erzeugt werden und daher Domain-Namen enthalten können, die der Administrator nicht gesperrt haben möchte. Ferner sind die Listen nicht allumfassend, bieten aber eine hervorragende Basis.

Der Datenschutz ist in diesem Beitrag außer Acht geblieben und wirft entsprechende Fragen auf. So werden die IP-Adresse jedes Clients und der Benutzername für die Authentifizierung mitprotokolliert. Diese Informationen lassen sich personenbezogen auswerten und verwenden. Vor allem der Benutzername in Verbindung mit aufgerufenen Seiten in den Auswertungen ist im Sinne des Datenschutzes bedenklich. Daher sollten Sie Ihre Anwender gegebenenfalls über Nutzungsrichtlinien darauf hinweisen.

3.4.14 Listing: Squidguard-Konfiguration

```
# Datei /etc/squidGuard.conf
logdir /var/squidGuard/log
dbhome /var/squidGuard/db

dest financial {
# Die nachstehenden Datenbanken werden unter diesem ACL-
# Namen zusammengefasst.
        domainlist     finance/finance.domains
        urllist        finance/finance.urls
        expressionlist finance/finance.regexp
}

src locals {           # hierhin alle erlaubten IP-Adressen
        ip 192.168.1.100/32   # der Proxy aus einem anderen
Subnetz darf
        ip 192.168.0.0/24     # das ganze Netzwerk
}

time working_hours {
        weekly mondays tuesdays wednesdays thursdays fridays
06:30-20:00
}
acl {
        locals within working_hours {
```

```
# Während der Bürozeiten dürfen die festgelegten lokalen
# Computer alle Seiten im Internet aufrufen, die nicht
# in den unter finacial angelegten Datenbanken stehen
                pass !financial all
# Alle abgewiesenen Anfragen werden an ein CGI-Skript
# umgeleitet, das eine Fehlermeldung erzeugt.
                redirect http://prefect.tellerrand.de/cgi-
bin/squidGuard.cgi?clientaddr=%a&clientname=%n&clientuser=%i
&clientgroup=%s&url=%u
        }
        default {
# Standardregel: "Alles was nicht erlaubt ist
# wird verboten".
                pass none
# Alle abgewiesenen Anfragen werden an ein CGI-Skript
# umgeleitet, das eine Fehlermeldung erzeugt.
                redirect http://prefect.tellerrand.de/cgi-
bin/squidGuard.cgi?clientaddr=%a&clientname=%n&clientuser=%i
&clientgroup=%s&url=%u
                log anonymous default-deny.log
        }
}
```

3.4.15 Listing: init_script_proxy

```
#!/bin/sh
# Author: Oliver Drees
# Startscript für squid
. /etc/rc.config

case "$1" in
    start)
        echo -n "Starting WWW-proxy squid: "
        startproc -l /var/squid/squid.out
/usr/local/squid/bin/squid || return="\tfailed"
        sleep 1
        echo -e "\tdone"
        ;;
    stop)
        echo -n "Shutting down WWW-proxy squid: "
        /usr/local/squid/bin/squid -k shutdown 2>/dev/null
        echo -e "\tdone"
        ;;
    status)
        echo -n "Checking for WWW-proxy squid: "
        checkproc /usr/local/squid/bin/squid && echo OK ||
echo No process
        ;;
    *)
        echo "Usage: $0 {start|stop|status}"
        exit 1
esac
exit 0
```

3.4.16 Listing: Webalizer Konfiguration

```
# Datei: webalizer.conf
LogFile        /var/squid/logs/access_log
LogType        squid
OutputDir      /home/www/htdocs/usage
HistoryName    webalizer.hist
Incremental    yes
IncrementalName       webalizer.current
ReportTitle    Usage Statistics for
HostName       proxy.fakedomain.de
HTMLExtension  html
PageType       htm*
PageType       cgi
PageType       phtml
PageType       php3
PageType       pl
DNSCache       dns_cache.db
DNSChildren    2
CountryGraph no
#      keine Benutzer-Angaben in der Auswertung zeigen
TopUsers       0
#      TopUsers        20
AllSites       yes
AllURLs        yes
AllReferrers   yes
AllAgents      yes
AllSearchStr   yes
AllUsers       yes
HideURL        *.gif
HideURL        *.GIF
HideURL        *.jpg
HideURL        *.JPG
HideURL        *.png
HideURL        *.PNG
HideURL        *.ra
HideUser       *
```

tecCHANNEL-Links zum Thema	Webcode
Linux für den Server	a487
Linux als Firewall	a695
Hypertext Transfer Protocol	a208
So funktioniert TCP/IP	a209
Linux Firewall mit ipchains	a704
Linux als Windows-Server	a248
Java Virtual Machine unter Linux	a776

3.5 Linux als Webserver

Vor dem eigenen Internet- oder Intranet-Auftritt steht zunächst die Einrichtung und Konfiguration des Servers mit Betriebssystem und des eigentlichen HTTP-Servers. Mit dem Gespann aus Linux und Apache steht dabei eine preisgünstige, sichere und leistungsfähige Plattform zur Verfügung. Auch tecChannel.de basiert aus gutem Grund auf dem Apache-Webserver unter Linux.

Wie die wichtigsten Bestandteile des Linux-Betriebssystems eingerichtet werden, erläutern Ihnen frühere Beiträge von tecChannel.de, die Sie im Online-Archiv finden. In diesem Artikel lesen Sie, wie Sie Apache auf dem Linux-Server installieren und konfigurieren, so dass dem eigenen Webauftritt eigentlich nichts mehr im Wege steht.

3.5.1 Das erste Rüstzeug

Die meisten Linux-Distributoren konfigurieren Apache schon vor, so dass Sie in aller Regel nicht dazu gezwungen sind, den Webserver von Scratch selbst einzurichten. Sie werden vielmehr Apache an Ihre Bedürfnisse anpassen. Selbst wenn Sie Apache direkt von der Homepage (www.apache.org) downloaden, finden Sie eine grundlegende Konfiguration vor.

Die Konfiguration von Apache erfolgt für Unix typisch über Textdateien. Das erste Problem, das sich dem angehenden Webadministrator stellt, ist die Frage nach dem Verzeichnis, in dem die Konfigurationsdateien zu finden sind. Diese können je nach Distribution an ganz unterschiedlichen Stellen liegen.

Lage der Konfigurationsdateien

Distribution	Position der Dateien
Original Apache	/usr/local/apache/conf
Red Hat	/etc/httpd/conf
SuSE	/etc/httpd

Falls Sie eine andere Distribution verwenden, deren Softwareverwaltung auf RPM aufbaut, können Sie über das betreffende RPM-Paket die Lage der Konfigurationsdateien herausfinden. Vorausgesetzt Apache findet sich bei Ihrem Linux-System in einem Paket apache*.rpm, erfahren Sie das Konfigurationsverzeichnis durch folgenden Shell-Befehl:

```
rpm -ql apache | grep httpd.conf
```

Die zentrale Datei zum Setup Ihres Webservers ist *httpd.conf*. Die meisten Einstellungen führen Sie über Einträge in dieser Datei durch. Weiter gibt es *access.conf*, über die Sie Zugriffsrechte und Dienste der einzelnen Verzeichnisse festlegen. Mittels *srm.conf* bestimmen Sie den Namensbereich, den die Benutzer Ihres Webservers sehen.

3.5.2 Scripts für den Webserver

Den Start des Webservers überlassen Sie am besten einem Init-Script. Unter Linux-Systemen, die zu Red Hat kompatibel sind, tragen Sie mit Tools wie *linuxconf* schlicht den httpd-Dienst zum automatischen Start ein. Bei SuSE Linux verwenden Sie entweder YaST oder Sie setzen in der Datei */etc/rc.config* den Eintrag START_HTTPD auf *yes*. In Ihren Verzeichnissen mit den Init-Scripts können Sie Apache auch direkt starten. Das betreffende Init-Script finden Sie bei Red Hat unter */etc/rc.d/init.d/httpd*. Bei SuSE liegt es unter */sbin/init.d/httpd*. Diesem Script können Sie die Argumente *start*, *stop* und *reload* übergeben, um Apache zu starten, anzuhalten oder neu zu starten.

Immer wenn Sie die Konfiguration des Apache-Servers durch Bearbeiten der betreffenden Dateien verändern, starten Sie den Webserver erneut, damit die Modifikationen wirksam werden.

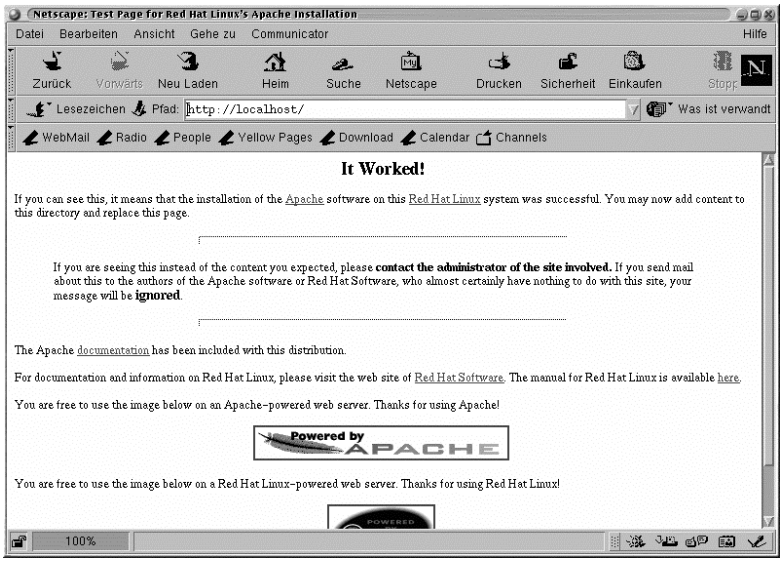
Es wäre jedoch nervenaufreibend beziehungsweise sogar absolut unpraktikabel, wenn Sie hierzu den Computer jedes Mal neu booten müssten. Stattdessen verwenden Sie einfach das betreffende Init-Script mit dem Argument *reload*. Für Red Hat sieht das dann wie folgt aus:

```
/etc/rc.d/init.d/httpd reload
```

In einigen Fällen kann das Script auch *apachectl* heißen. Die Bedienung bleibt jedoch dieselbe.

3.5.3 Der erste Start

Nach dem Start von Apache können Sie Ihren Browser aufrufen und die Seiten des lokalen Webservers ausprobieren. Geben Sie dazu als URL *http://localhost* ein und schauen Sie, was passiert. Sollte Ihre Distribution den Webserver bereits in einen funktionstüchtigen Zustand versetzt haben, sehen Sie jetzt eine Seite mit distributionsabhängigen Informationen.



Test: Wenn Apache nach dem ersten Start diese Seite anzeigt, hat es funktioniert.

Die erste anstehende Änderung ist das Einstellen der eigenen Webseiten. Dazu findet sich in der Datei *httpd.conf* oder *srm.conf* der Eintrag *DocumentRoot*. Das hier angegebene Verzeichnis verwendet Apache als Wurzelverzeichnis - und damit als Eintrittspunkt - für den HTTP-Server.

Sie können nun das hier angegebene Directory entleeren, also sämtliche Verzeichnisse und Dateien löschen. Diesem Procedere würde jedoch in den allermeisten Fällen auch die Apache-Dokumentation zum Opfer fallen. Der bessere Weg ist es, ein separates Verzeichnis für Ihre Website anzulegen und einzustellen. Darin können Sie Ihre Webseiten hochziehen.

3.5.4 Standarddokumente

Sowie Sie Ihre Domain ohne explizite Angabe eines HTML-Dokumentenpfades in einem Browser als URL eingeben, wird automatisch das Standarddokument aus dem mit DocumentRoot angegebenen Verzeichnis geöffnet. Sicherlich fragen Sie sich jetzt, woher Apache weiß, welches HTML-File das Standarddokument ist.

Hierzu existiert der Eintrag *DocumentIndex* in *httpd.conf* beziehungsweise *srm.conf*. Die Dateinamen, die Sie in diesem Eintrag auflisten, werden als Standarddokument verwendet. Dies gilt dabei nicht nur für das Wurzelverzeichnis, sondern für alle Verzeichnisse Ihrer Website. Sobald als URL ein Verzeichn-

Verzeichnis statt eines HTML-Files angegeben wird, sucht Apache eine solche über *DocumentIndex* angegebene Datei in dem betreffenden Verzeichnis. Erst wenn in diesem Verzeichnis keine solche Datei existiert, listet Apache je nach Konfiguration entweder den Verzeichnisinhalt auf oder präsentiert eine Fehlermeldung. Angenommen in Ihrer Konfiguration findet sich die Zeile

```
DocumentIndex index.html index.htm index.cgi
```

Dann sucht Apache zuerst nach einer Datei *index.html*. Ist diese nicht vorhanden, wird nach *index.htm* gesucht. Zu guter Letzt wird *index.cgi* verwendet, soweit vorhanden.

3.5.5 Andere Verzeichnisse im HTTP-Baum

Neben dem Wurzelverzeichnis können Sie auch andere Verzeichnisse in den HTTP-Baum einbinden. Diese Verzeichnisse stehen dann über einen von Ihnen festgelegten Pfad zur Verfügung.

Allgemein lassen sich über die Alias-Direktive Verzeichnisse aus dem gesamten Linux-Dateibaum in Apache einbinden. Diese Direktive kann in allen drei Konfigurationsdateien von Apache verwendet werden.

Angenommen Sie wollen das Verzeichnis */mnt/www/SuperProdukt* unter der URL *http://www.IhreDomain.de/SuperProdukt* zur Verfügung stellen. In diesem Fall verwenden Sie die Zeile

```
Alias /SuperProdukt /mnt/www/SuperProdukt
```

Eine weitere Möglichkeit, zusätzliche Verzeichnisse in den HTTP-Baum einzufügen, sind die HOME-Verzeichnisse der jeweiligen Benutzer. Wenn Sie die Zeile

```
UserDir public_html
```

in *httpd.conf* oder *srm.conf* eintragen, kann jeder Benutzer seine Homepage anlegen. Er muss lediglich seine Webseiten in dem Unterverzeichnis *public_html* des eigenen HOME-Verzeichnisses erstellen. Danach sind die Seiten über die URL http://www.IhreDomain.de/~fritz verfügbar.

Wollen Sie nicht, dass die Benutzer eigene Seiten ins Internet/Intranet/Extranet stellen, so lässt sich diese Funktion über die Zeile

```
UserDir disabled
```

```
UserDir disabled root, fritz
```

an. Auch der umgekehrte Weg ist denkbar. Über

```
UserDir enabled fritz, marina, theo
```

gestatten Sie den Benutzern fritz, marina und theo, eine eigene Homepage aufzubauen. Dazu ist zusätzlich über die Option *UserDir disabled* den anderen Anwendern die eigene Homepage zu verweigern.

3.5.6 Das Ruder in der Hand - Zugriff steuern

Die Zugriffsrechte auf die einzelnen Verzeichnisse Ihres HTTP-Baumes steuern Sie über die Konfigurationsdatei *access.conf*. Sie können zu jedem Verzeichnis über eine Directory-Struktur festlegen, welche Möglichkeiten für Zugriff und Dienste (wie beispielsweise Ausführen von CGI-Scripts) bestehen.

In dem Directory-Tag geben Sie das Verzeichnis aus dem Linux-Dateibaum an, für das Sie die Rechte festlegen wollen. Hier ist jetzt vor allem auch wichtig, dass Sie - sofern Sie vorher DocumentRoot verändert haben - die betreffende Directory-Struktur ändern. Überschreiben Sie einfach den alten Pfad mit Ihrem neuen DocumentRoot.

Eine Directory-Struktur besteht immer aus einem einleitenden und abschließenden Tag. Für das Verzeichnis */mnt/www/SuperProdukt* wird die Struktur beispielsweise mit

```
<Directory /mnt/www/SuperProdukt>
```

eingeleitet und durch

```
</Directory>
```

abgeschlossen. Alle Zeilen zwischen diesen Tags beziehen sich dann ausschließlich auf das Verzeichnis */mnt/www/SuperProdukt*.

Einschränkungen in den Diensten können Sie über den Optionseintrag innerhalb der Directory-Struktur bewirken. Fehlt ein solcher Eintrag, sind alle Dienste möglich. In der folgenden Tabelle sind die folgenden Angaben denkbar:

Konfigurationsübersicht	
Options-Argument	**Position der Dateien**
All	Alle Optionen (außer MultiViews) werden eingeschaltet (Default).
ExecCGI	Ausführen von CGI-Scripts ist erlaubt.
FollowSymLinks	Befinden sich in dem Verzeichnis symbolische Links, wird diesen gefolgt. Beachten Sie, dass diese quer durch Ihr System führen können.
Includes	Serverseitige Includes sind gestattet.
IncludesNOEXEC	Serverseitige Includes sind erlaubt, aber #exec und #include von CGI-Scripts ist ausgeschaltet.
Indexes	Wird kein Standarddokument im Verzeichnis gefunden, wird der Inhalt des Verzeichnisses aufgelistet. Wenn diese Option fehlt, kommt es stattdessen zur Ausgabe einer Fehlermeldung.
MultiViews	Durch den Inhalt ausgelöste MultiViews sind erlaubt.
SymLinksIfOwnerMatch	Der Server folgt nur symbolischen Links, wenn das Ziel die gleiche User-ID hat, wie der Link.

Wollen Sie, dass der Inhalt eines Verzeichnisses ausgegeben wird, keine Fehlermeldung erscheint und außerdem symbolischen Links gefolgt wird, so geben Sie *Options Indexes FollowSymLinks* an. Das Ausführen von CGI-Scripts ist in diesem Fall dann beispielsweise aus diesem Verzeichnis nicht möglich.

Eine Directory-Struktur bezieht auch immer gleich alle Unterverzeichnisse mit ein, sofern für diese nicht eine separate Directory-Struktur verwendet wird.

3.5.7 Wer darf und wer nicht?

Selbst in einem Intranet - und erst recht im Internet - ist es nicht allen erlaubt, alles zu sehen. Am einfachsten beschränken Sie den Zugriff auf die IP-Adressen der Mitarbeiter im Netzwerk, die beispielsweise vertrauliche Informationen verwenden sollen.

Hierzu gibt es drei wichtige Klauseln - *order*, *allow from* und *deny from*. Über *allow from* geben Sie an, welche IP-Adressen (oder Host-Namen) auf das Verzeichnis Zugriff haben. Die Option *deny from* arbeitet in umgekehrter Richtung und legt fest, welchen IP-Adressen der Zugriff untersagt ist. Via *order* geben Sie an, ob zuerst die *allow*- oder die *deny*-Regeln ausgewertet werden sollen. Wenn

```
order allow,deny
deny from all
allow from Oxygen Nitrogen Argon
```

Diese Angaben bewirken, dass zunächst die *allow*- und danach die *deny*-Regel ausgewertet werden. Die *allow*-Klausel ermöglicht den Zugriff für die drei Hosts. Greift ein anderer Host zu, findet ihn Apache in dieser *allow*-Regel nicht und wendet die *deny*-Anweisung an. Hier wird allen Hosts der Zugriff verweigert. Bei herkömmlichen Webseiten fürs Internet, die alle sehen sollen, verwenden Sie folgende Konstellation:

```
order allow,deny
allow from all
```

Eine *deny*-Klausel ist hier nicht erforderlich. Nachdem für alle Hosts der Zugriff freigegeben wurde und die *allow*- vor den deny-Regeln ausgewertet werden, würden *deny*-Regeln ohnehin niemals greifen. Hinter *deny* und *allow* können Sie auch Domains und Subnetze angeben. Eine Sperrregel für alle Hosts der Domain .warenausgang.firma.de würde wie folgt aussehen:

```
deny from .warenausgang.firma.de
```

Um das Subnetz 192.168.0.0 mit der Netzmaske 255.255.255.0 freizuschalten, geben Sie zum Beispiel

```
allow from 192.168.0.0/255.255.255.0
```

in der Directory-Struktur an.

3.5.8 Gezielter steuern

Die Angaben der Directory-Strukturen lassen sich auch gezielter setzen. Wenn Sie für eine ganze Reihe von Unterverzeichnissen die Zugriffsrechte individuell setzen müssen, kann die Bearbeitung der Datei *access.conf* sehr schnell zur Sisyphusarbeit werden. Sie müssten ja für jedes Verzeichnis eine Directory-Struktur schreiben. Wenn sich dann noch Ihr Webcontent extrem häufig ändert, werden Sie Ihres Lebens wohl kaum mehr froh.

Sie können daher die Zugriffsrechte auch in separaten Dateien in den jeweiligen Verzeichnissen verstauen. Schreiben Sie dazu eine Directory-Struktur für das übergeordnete Verzeichnis und nehmen Sie die Zeile

```
AllowOverride all
```

darin auf. Ab jetzt können Sie die Optionen der Directory-Struktur in den Verzeichnissen überschreiben, indem Sie im betreffenden Verzeichnis die Datei *.htaccess* anlegen. Darin können Sie nun Anweisungen mit order, allow from und deny from ablegen, die sich auf das entsprechende Verzeichnis auswirken.

Hinweis: Über AllowOverride sind noch wesentlich genauere Abstufungen möglich, auf die hier nicht weiter eingegangen wird. Genaueres entnehmen Sie bitte der Dokumentation von Apache.

3.5.9 Sicherheit eine Stufe höher

Den Zugriff auf Webinhalte können Sie bei Apache selbstverständlich auch durch Passwörter absichern. Hierzu müssen Sie jedoch für Apache eine eigene Passwort-/Benutzerdatei anlegen. Diese erstellen Sie mit dem Kommando

```
htpasswd -c /etc/httpd/passwd firstuser
```

Dabei wird die Datei */etc/httpd/passwd* erstellt und zugleich der erste Benutzer mit dem Namen firstuser eingerichtet. Nach Bestätigung des Befehls auf der Shell werden Sie zur Eingabe des Passworts aufgefordert. Weitere Benutzer legen Sie an durch

```
htpasswd /etc/httpd/passwd <neuerNutzer>
```

Damit die Passwortabfrage funktioniert, müssen Sie noch folgende Zeilen in die Directory-Struktur oder die .htaccess-Datei des entsprechenden Webserver-Verzeichnisses aufnehmen:

```
AuthType basic
AuthName "Das Passwort bitte"
AuthUserFile /etc/httpd/passwd
```

Über AuthType legen Sie das Autorisierungsverfahren fest. Derzeit existieren hier lediglich basic und digest. Mit AuthName setzen Sie einen Prompt, der im Eingabedialog angezeigt wird. Last but not least gibt AuthUserFile die Passwort-/Userdatei an, die Sie mit *htpasswd* angelegt haben. Um mehrere Passwortdateien für verschiedene Zwecke anzulegen, verwenden Sie als Argument für *htpasswd* und AuthUserFile einfach andere Dateinamen.

3.5.10 Fazit

Der erste Schritt zum eigenen Inter-/Intranet-Auftritt ist die Einrichtung und Konfiguration des Webservers. Unter Linux ist das mit Apache inzwischen kein großes Problem mehr. Man muss nur wissen, an welchen Knöpfen man zu drehen hat, um einerseits maximalen Komfort für die Benutzer und andererseits größtmögliche Datensicherheit zu gewährleisten.

tecCHANNEL-Links zum Thema	Webcode
Linux als Dial-up-Router	a322
Linux als Windows-Printserver	a392
Linux als Windows-Server	a248

3.6 Instant-Messaging-Server Jabber

Instant Messaging (IM) eignet sich hervorragend für eine einfache, schnelle und unkomplizierte Kommunikation zwischen mehreren Teilnehmern. Damit bietet IM nicht nur im privaten, sondern auch im Unternehmenseinsatz durch seine Einfachheit und Geschwindigkeit zahlreiche Vorteile. Speziell in der Kommunikation zwischen einzelnen Mitarbeitern kann es E-Mail in vielen Fällen ersetzen. Nicht nur Informationen lassen sich so schneller austauschen, auch die Ressourcen der E-Mail-Server werden geschont.

Eine preiswerte und leistungsfähige Lösung für ein firmeninternes IM-System ist der Jabber-Server (www.jabber.org). Selbst weniger versierte Systemadministratoren können mit Jabber ein IM-System aufbauen sowie populäre externe Dienste wie ICQ, AIM, MSN, IRC und Yahoo einbinden.

Neben verschiedenen Clients für Windows, MacOS, Linux/Unix und weiteren Plattformen, die Java unterstützen, gibt es inzwischen auch Clients für Windows CE und andere mobile Geräte wie für den Newton. Jabber-Clients sind somit für alle relevanten Plattformen verfügbar, der Server hingegen nur für Linux und Unix.

3.6.1 Jabber im Detail

Bei Jabber handelt es sich um ein Instant-Messaging- und Presence-System, das sich von anderen IM-Lösungen in verschiedenen Punkten grundlegend unterscheidet. Jabber verwendet ein eigens für dieses IM-System entwickeltes Protokoll. Wenn Jabber-Clients sich auf einem Jabber-Server (temporär) angemeldet haben, kann ein direkter Nachrichtenaustausch zwischen zwei oder mehreren Clients erfolgen. Man bezeichnet die Verfügbarkeit eines Clients auch als Presence (Anwesenheit). Diese typische IM-Funktion kombiniert Jabber mit zwei Besonderheiten: Zum einen erlaubt das offene Protokoll die Kommunikation mit anderen Messaging-Systemen, zum anderen tauschen Client und Server XML-basierte Nachrichten aus.

Dabei bietet Jabber noch eine Besonderheit: Als erstes Instant-Messaging-System unterstützt Jabber das Instant Messaging and Presence Protocol (IMPP, www.imppwg.org), das von der IMPP-Arbeitsgruppe der Internet Engineering Task Force (IETF, www.ietf.org) entwickelt wurde. IMPP stellt eine Architektur für den einfachen Nachrichtenaustausch und Hinweismeldungen bereit. Auch Fragen der Authentifizierung, Nachrichtenintegrität, Verschlüsselung und Zugriffskontrolle spielen eine Rolle.

Der Client kommuniziert in der Regel über Port 5222 mit dem Server. Die Grundfunktionen der Client-Software wie beispielsweise das Parsen von XML-Dokumenten mit den Jabber-XML-Elementen (*<message/>*, *<presence/>* oder *<iq/>*) stellen die Jabber-Client-Bibliotheken bereit.

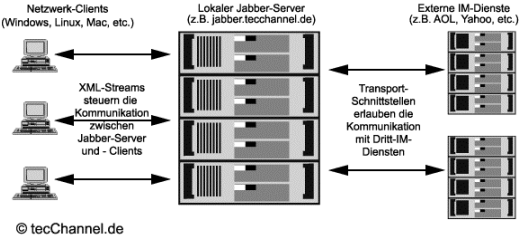
Der Vorteil für Unternehmen besteht darin, dass Entwickler ohne größeren Aufwand „eigene“ Clients realisieren können, die an die jeweiligen Anforderungen optimal angepasst sind.

XML ist integraler Bestandteil der Jabber-Architektur. Man hat sich für XML entschieden, da es sich dabei um das universellste Format handelt. So genannte XML-Streams sorgen für die Client-Server- beziehungsweise Server-Server-Kommunikation. Ein weiterer Vorzug von XML ist in diesem Zusammenhang, dass die Kommunikation mit anderen IM-Diensten oder Programmen wie AIM, ICQ oder IRC funktioniert. Man spricht diesbezüglich auch von Transportschnittstellen. Für alle wichtigen Dienste gibt es inzwischen entsprechende Interfaces.

3.6.2 Jabber-Server einrichten

Mit erstaunlich wenigen Schritten lässt sich ein funktionstüchtiges Jabber-System aufsetzen. Auf die Beschreibung der Client-Funktionen und das Handling verzichten wir an dieser Stelle. Einerseits gibt es inzwischen zahlreiche Jabber-Clients, andererseits bereitet die Handhabung, Installation und Konfiguration der Clients in aller Regel keine Probleme, da es sich um einfache Programme handelt.

Die aktuelle Version des Jabber-Servers 1.4 wurde in erster Linie für Linux entwickelt. Aber auch auf AIX, IRIX und verschiedenen anderen Unix-Plattformen ist sie erfolgreich getestet worden.



Typisches Szenario: Über einen lokal installierten Jabber-Server tauschen Clients Nachrichten aus. Transportschnittstellen sorgen für die Kommunikation mit der Außenwelt.

Für den Betrieb im Intranet oder einer kleineren Website reicht an Hardware eine typische Pentium-III-Workstation mit 256 oder 512 MByte RAM. Die einzige Datei, die Sie für die Installation eines Basissystems brauchen, ist *jabber-1.4.tar.gz*. Gegebenenfalls sind weitere Komponenten erforderlich, beispielsweise das Konferenzmodul, das Jabber User Directory sowie die Transportschnittstelle.

3.6.3 Installation

Von Version zu Version wird die Installation des Jabber-Servers zusehends einfacher. Dies ist nicht zuletzt einer Vielzahl freiwilliger Programmierer zu verdanken, die sich in den Dienst des Jabber-Projekts stellen. Für die Installation sind folgende Schritte durchzuführen: Laden Sie sich die aktuelle Jabber-Version herunter und speichern Sie das Paket beispielsweise im tmp-Verzeichnis. Öffnen Sie die Konsole und erzeugen Sie ein Verzeichnis, in das der Jabber-Server installiert werden soll, beispielsweise */pfad/zu/jabber* oder */usr/local/jabber*. Beachten Sie, dass dazu Root-Rechte erforderlich sind. Führen Sie anschließend folgende Befehle aus:

```
mv /tmp/jabber-1.4.tar.gz /pfad/zu/jabber/
cd /pfad/zu/jabber/
gzip -d jabber-1.4.tar.gz
tar -xvf jabber-1.4.tar
cd jabber-1.4/
./configure
make
```

Damit ist die Installation bereits abgeschlossen. Wenn Sie mit Red-Hat-Linux arbeiten, sollten Sie vor der Installation ein Update von gcc.2.96 installieren.

3.6.4 Konfiguration

Die Konfiguration des Servers gestaltet sich ebenfalls einfach. Die Konfiguration beschränkt sich im Wesentlichen auf Anpassungen der Konfigurationsdatei *jabber.xml*. Sie enthält selbst eine sehr detaillierte Beschreibung aller Schalter und Optionen. Für die Systemkonfiguration genügen drei Schritte. Öffnen Sie dazu *jabber.xml* mit einem Editor und passen Sie folgende Zeile an:

```
<host><jabberd:cmdline
flag="h">localhost</jabberd:cmdline></host>
```

Hier tragen Sie anstelle von *localhost* den Namen des Host-Rechners ein. Diese Änderung ist nicht erforderlich, wenn Sie den Server auf dem lokalen System zu Testzwecken einsetzen. Soll der Server allerdings mit der Außenwelt, also auch anderen Jabber-Servern kommunizieren, so ist der vollständige Domain-Name einzutragen. Im zweiten Schritt kommentieren Sie folgende Zeile aus:

```
<update><jabberd:cmdline
flag="h">localhost</jabberd:cmdline></update>
```

Die Einstellungen sorgen dafür, dass der Server automatisch das Benutzerverzeichnis aktualisiert, wenn eine vCard editiert wird. Das wiederum setzt die

Einrichtung des Jabber-User-Verzeichnisses voraus. Wenn Sie die Einstellung beibehalten wollen, ersetzen Sie localhost durch den Host-Namen Ihres Rechners.

Im dritten Schritt können Sie optional den Willkommenstext (<*welcome/*>) und die Admin-Einträge (<*admin/*>) ändern. Da Jabber die User-Roster im Dateisystem speichert, genauer in einem Jabber-Unterverzeichnis *jabber-1.4 /spool/unterverzeichnis/*, muss ein Verzeichnis mit der Bezeichnung des Host-Namens erzeugt und mit Schreibrechten versehen werden. Lautet der Host-Name beispielsweise „jabber.tecChannel.de“, so erstellen Sie das Verzeichnis *jabber-1.4/spool/jabber.tecChannel.de/*. In dieses dürfen die Jabber-User dann schreiben.

Der Administrator des Jabber-Servers sollte Root sein. Die Einstellungen lassen sich jedoch gegebenenfalls verändern:

```
chown -R username.usergruppe. /pfad/zu/jabber/
```

3.6.5 Starten des Servers

Damit ist die Basiskonfiguration des Jabber-Servers abgeschlossen, und Sie können den Server mit folgendem Befehl starten:

```
./jabberd/jabberd -h hostname
```

Um den Server im Debug-Modus zu aktivieren, verwenden Sie folgende Zeile:

```
./jabberd/jabberd -D -h hostname
```

Ein Protokoll der Serveraktivitäten erhalten Sie mit

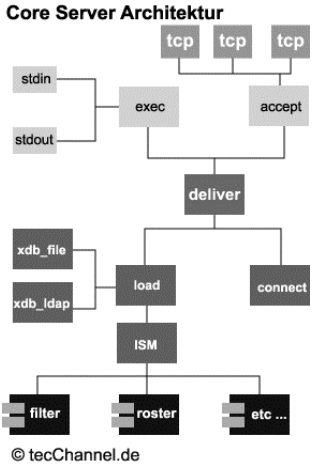
```
tail -f error.log
```

Mit einem beliebigen Jabber-Client können Sie nun eine Verbindung zum Jabber-Server aufbauen. Entsprechende Clients finden Sie unter www.jabbercentral.com/clients/.

3.6.6 Jabber aufgebohrt

Bereits in einer Grundkonfiguration genügt Jabber den Anforderungen kleiner Abteilungen und/oder Unternehmen. So richtig interessant wird es aber erst, wenn man zusätzliche Dienste hinzufügt. Gerade die Version 1.4 zeigt sich was Erweiterbarkeit betrifft flexibel. Neben Konferenzfunktionen (chat) und dem

Jabber User Directory können Sie den Jabber-Server durch die Transportschnittstellen erweitern.

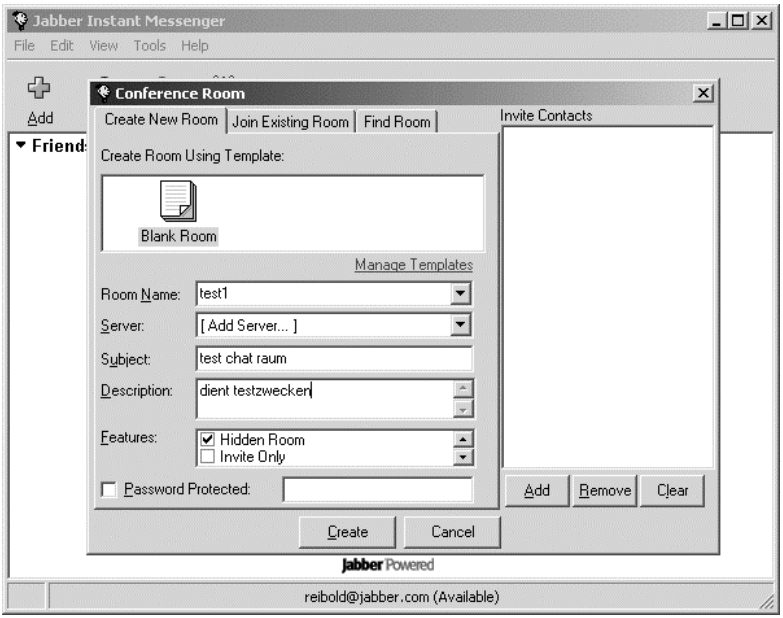


Architektur: Das Herz ist die Delivery-Komponente, die einen Austausch der Daten zwischen den Serverkomponenten ermöglicht. Von einer Basiskomponente (accept, connect, exec oder load) wird sie je nach Ziel und Einstellungen für die weitere Verarbeitung an Folgemodule übergeben.

Doch vorab eine wichtige Anmerkung: Damit die neu eingerichteten Dienste der an sie gestellten Aufgabe gerecht werden, müssen die eingerichteten Sub-Domains über einen vollständig qualifizierenden Domain-Namen verfügen. Zum Beispiel „aim.tecChannel.de“ für die AOL-Schnittstelle.

3.6.7 Installation des Konferenzmoduls

Das Konferenzmodul finden Sie ebenfalls im Download-Bereich (down load.jabber.org) der Jabber-Website. Bevor Sie sich an die Installation und Konfiguration machen, sollten Sie - wie bereits erwähnt - die zugehörige Sub-Domain (zum Beispiel „conference.tecChannel.de“) im DNS registrieren. Dazu ist die lokale DNS-Tabelle zu aktualisieren. Wenn Sie mit einem dynamischen DNS-System arbeiten, können Sie auch einen Platzhalter verwenden, damit eine Auflösung der Sub-Domains erfolgt.



Zimmereinrichtung: Ist das Konferenzmodul einmal konfiguriert, können Sie über die Jabber-Clients eigene Chat-Räume erzeugen.

Zur Installation des Konferenzmoduls laden Sie zunächst die Datei *conference-0.4.tar.gz* herunter und legen diese im Jabber-Verzeichnis ab. Anschließend führen Sie folgende Befehle aus:

```
gzip -d conference-0.4.tar.gz
tar -xvf conference-0.4.tar
cd conference-0.4/
make
```

3.6.8 Jabber-Chat-Raum einrichten

Damit ist das Konferenzmodul installiert, und es folgt die Konfiguration. Dazu öffnen Sie die Datei *jabber.xml* und nehmen verschiedene Anpassungen vor. Mit dieser ersten Erweiterung lernen Sie ein Muster kennen, das bei allen anderen Erweiterungen ebenfalls zur Verwendung kommt. Zunächst müssen Sie den Abschnitt für die Konferenz-Services in der *<browse/>*-Sektion von *jabber.xml* finden. Diese Anpassungen sind erforderlich, damit User-Clients vom Server

ablesen können, dass dieser Server Chat-Funktionen bereitstellt. Fügen Sie dazu in der <*browse/*>-Sektion folgende Zeile hinzu:

```
<conference type="private" jid="conference.localhost"
name="Conferencing"/>
```

Auch hier wird wieder *localhost* durch den Host-Namen des Servers ersetzt. Im nächsten Schritt erfolgt die Definition des neuen Dienstes. Dazu fügt man *jab ber.xml* das Service-Element hinzu. Die hierfür notwendigen Erweiterungen zeigt nachstehendes Beispiel:

```
<service id="conference.localhost">
<load><conference>./conference-
0.4/conference.so</conference></load>
<conference xmlns="jabber:config:conference">
<vCard>
<FN>Konferenz Service</FN>
<DESC>Bei diesem Dienst handelt es sich um einen privaten
Chat-Raum.</DESC>
<URL>http://localhost/</URL>
</vCard>
<history>20</history>
<notice>
<join> ist eingetreten</join>
<leave> hat verlassen</leave>
<rename> bekannt als </rename>
</notice>
</conference>
</service>
```

Stellt man nun mit einem Client eine Verbindung zum Jabber-Server her, so sollte Client-seitig Conferencing als angebotener Dienst erscheinen. Beispiele für diese und alle folgenden Erweiterungen/Anpassungen der Jabber-Konfigurationsdatei können Sie vom tecCHANNEL-Server herunterladen: www.tec Channel.de/download/834/jabber_erweiterungen.txt.

3.6.9 Jabber User Directory (JUD)

Damit ein User andere aufstöbern kann, müssen die registrierten Benutzer an einer Stelle zentral verwaltet werden. Das ist Aufgabe des Jabber User Directory, kurz JUD. Die zentrale Instanz des Verzeichnisses wird auf jabber.org betrieben. Die dort registrierten Benutzer können bei Internet-Verbindungen kontaktiert werden. Natürlich macht es auch in Unternehmen Sinn, Benutzer zentral zu verwalten. Das für die Installation erforderliche Paket *jud-0.4.tar.gz* liegt im Jabber-Download-Bereich (http://download.jabber.org). Um es zu entpacken und die Installation durchzuführen, geben Sie folgende Befehle ein:

```
gzip -d jud-0.4.tar.gz
```

```
tar -xvf jud-0.4.tar
cd jud-0.4/
make
```

Für die Konfiguration sind entsprechend wieder Anpassungen und Erweiterungen von *jabber.xml* erforderlich. Fügen Sie dazu in der <*browse/*>-Sektion folgende Zeilen hinzu. Vergessen Sie dabei nicht, localhost durch den Domain-Namen zu ersetzen:

```
<service type="jud" jid="jud.localhost" name="localhost User
Directory">
<ns>jabber:iq:search</ns>
<ns>jabber:iq:register</ns>
</service>
```

Nun folgt die Service-Definition durch das Hinzufügen eines weiteren Service-Elements:

```
<service id="jud">
<host>jud.localhost</host>
<load><jud>./jud-0.4/jud.so</jud></load>
<jud xmlns="jabber:config:jud">
<vCard>
<FN> Benutzerverzeichnis auf localhost </FN>
<DESC> Dieser Dienst stellt ein einfaches Benutzerverzeich-
nis bereit.</DESC>
<URL>http://localhost/</URL>
</vCard>
</jud>
</service>
```

Starten Sie anschließend den Jabber-Server neu und fügen Sie mit Hilfe eines Clients einen neuen User hinzu.

3.6.10 Außenanbindung

Wie bereits erwähnt wird die Außenanbindung an gängige Instant-Messaging-Systeme über die so genannten Transportschnittstellen realisiert. Die Installation und Konfiguration dieser Schnittstelle gestaltet sich ebenfalls einfach. Sie läuft nach dem bisherigen Schema ab:

Download der entsprechenden Installationsdateien sowie Installation und Bearbeiten von *jabber.xml*.

Wir beschränken uns an dieser Stelle auf die Installation und Konfiguration der AIM-Transportschnittstelle. Die erforderlichen Erweiterungen für die Transportschnittstellen für ICQ, MSN und Yahoo können Sie vom tecCHANNEL-Server herunterladen: www.tecChannel.de/download/834/jabber_erweiterungen.txt. Bei

der Inbetriebnahme der AIM-Schnittstelle kann es - im Unterschied zu anderen Schnittstellen - jedoch gelegentlich zu Problemen kommen. Der Grund: AOL nimmt ab und zu Änderungen am AIM-Protokoll vor, um die Verbindungsaufnahme von Seiten eines Jabber-Servers zu unterbinden. Für die Installation der AIM-Schnittstelle holen Sie sich das Paket *aim-transport-0.9.5.tar.gz* und führen folgende Schritte aus:

```
gzip -d aim-transport-0.9.5.tar.gz
tar -xvf aim-transport-0.9.5.tar
cd aim-transport-0.9.5/
./configure --with-jabberd=/pfad/zum/jabberd/verzeichnis
```

3.6.11 Konfiguration der AIM-Schnittstelle

Die Konfiguration erfolgt ebenfalls in der Datei *jabber.xml.* Einerseits muss die <*browse/*>-Sektion erweitert, andererseits das Service-Element hinzugefügt werden. Fügen Sie im Falle der AIM-Schnittstelle der <*browse/*>-Sektion folgende Zeilen hinzu:

```
<service type="aim" jid="aim.localhost" name="AIM Trans-
port">
<ns>jabber:iq:gateway</ns>
<ns>jabber:iq:register</ns>
</service>
```

Schließlich muss die Dienstdefinition erfolgen, indem Sie das Service-Element in *jabber.xml* ergänzen:

```
<service id=´aim.localhost´>
<aimtrans xmlns=´jabber:config:aimtrans´>
<vCard>
<FN>AIM Transport</FN>
<DESC>Bei den AIM-Transportschnittstellen gibt es leider
immer wieder Probleme.</DESC>
<URL>http://localhost/</URL>
</vCard>
</aimtrans>
<load>
<aim_transport>./aim-transport-
0.9.5/src/aimtrans.so</aim_transport>
</load>
</service>
```

Nach einem Neustart des Jabber-Server können Sie die AIM-Schnittstelle auf ihre Funktionstüchtigkeit hin überprüfen. Denken Sie dabei auch an die Umbenennung von *localhost*.

Damit ist die Installation abgeschlossen, und Sie können den Jabber-Server für Ihre Mitarbeiter bereitstellen. Wenn Sie noch andere Dienste wie ICQ, IRC, MSN oder Yahoo integrieren wollen, laden Sie sich einfach die entsprechenden Transportschnittstellen aus dem Jabber-Download-Verzeichnis (http://download.jabber.org) und verfahren analog zur Integration der AIM-Schnittstelle.

tecCHANNEL-Links zum Thema	Webcode
Jabber im Kampf der Messenger	a710
Instant Messenger im Test	a763
Zukunft XML	a452
So funktioniert E-Mail	a819
Kostenlose E-Mail-Dienste	a87

4 Linux und Sicherheit

Nicht nur im Unternehmenseinsatz, sondern auch im privaten Umfeld spielt das Thema Sicherheit eine immer wichtigere Rolle. Auch wenn Linux als relativ sicher gilt, können Sie mit einigen Vorkehrungen größere Schäden durch Viren und Hacker weiter einschränken. Das vierte Kapitel behandelt das Thema Viren unter Linux und zeigt, wie Sie Hackerangriffe unter Linux abwehren können. Wer ganz auf Nummer Sicher gehen will, dem zeigen wir, wie man eine Desktop-Firewall auf seinem System einrichtet.

4.1 Viren unter Linux

In der Linux-Community hält sich unbeirrbar die Meinung, Linux könne nicht Opfer von Viren werden. Das System sei ja technisch überlegen und hundertprozentig sicher. Auch wenn es keiner hören will: Die Realität sieht anders aus.

Viren unter Linux haben sich vom belächelten Kinderschreck zur ernsten Gefahr entwickelt. Das zunehmende Auftreten von Viren, Trojanern und Würmern unter dem freien Unix kann schon bald als Epidemie bezeichnet werden. Leider herrscht unter den Linux-Jüngern genauso viel Aufgeschlossenheit wie bei den Pestärzten im Mittelalter. Wie die drei chinesischen Affen schalten sie auf stumm, taub und blind, sobald der zum Überwesen hochstilisierte Pinguin mit den digitalen Krankheitserregern in Verbindung gebracht wird.

Dieser Beitrag soll mit folgenden Missverständnissen aufräumen:

- Die ausführbaren Dateien (ELF-Format) von Linux sind sicher gegen Infektionen aller Art.
- Würmer und Mailviren wie der Love-Letter sind unter Linux nicht möglich.
- Linux sei so überlegen sicher, dass es keine Angriffspunkte bietet.

Auf die Funktionsprinzipien der Viren geht dieser Beitrag allerdings nur soweit ein, wie es für das Erläutern der einzelnen Punkte wichtig ist. Wer hier eine Anleitung zum Bau von Viren erwartet, wird enttäuscht werden.

4.1.1 Warum die Viren auf sich warten ließen

Linux existiert nun schon zehn Jahre. Da stellt sich natürlich die berechtigte Frage, warum sich erst jetzt Viren als Gefahr für dieses System erweisen. Bei Systemen wie DOS oder Windows haben sich die ersten Viren schon nach wesentlich kürzerer Zeit eingefunden.

Die Antwort auf diese Frage ist simpel: DOS und Windows waren mit einem Schlag auf dem Desktop-PC präsent. Eine Vielzahl von Anwendern konnte da-

mit auf einen Streich erreicht werden. Linux hingegen führt auch heute noch auf dem Schreibtisch ein Schattendasein. Es hat bislang seinen Platz primär im Serverbereich erobert. Die Anwenderanzahl spielt für Virenprogrammierer jedoch eine entscheidende Rolle.

Ebenso wie es eine Hacker- beziehungsweise Cracker-Szene gibt, existiert auch ein Underground für Virenschöpfer. Innerhalb dieser Szenen existieren einzelne Gruppen, die sich auf die Fahnen geschrieben haben: „Welt pass' auf! Wir schreiben die besten Viren!" In diesen Gruppen wird reger Austausch über neue Programmiermethoden und Know-how betrieben, fachliche Probleme werden diskutiert und sogar Dokumentation und Anleitungen zum Virenbau verteilt.

Die Gruppen sind einem Ehrencodex verpflichtet. Als Qualitätsmerkmale eines Virus gelten die Anzahl der infizierten Rechner und der angerichtete Schaden. Beides schafft Aufmerksamkeit und damit Status in der eigenen Gruppe und dem gesamten Underground.

Nun ist die Chance, gigantischen Schaden auf einem Windows-System anzurichten, das auf neunzig Prozent aller Arbeitsplatzstationen läuft, wesentlich größer als bei Linux, das gerade einmal vier Prozent Marktanteil im Desktop-Bereich hat.

4.1.2 Hausgemachtes Problem?

Doch drängt sich die Frage auf: Warum wird Linux bei einem relativ geringen Marktanteil inzwischen verstärkt von Viren befallen? Das hat im Wesentlichen zwei Gründe.

Zum einen fordert eine im Aufwind befindliche Plattform wie Linux den Sportsgeist eines Virenprogrammierers heraus. Unter Windows kann schon fast jeder einen Virus programmieren, und wenn er nicht programmieren kann, verwendet er einen der vielen Virenbaukästen der Szene. Unter Linux ist noch Expertenwissen notwendig, um einen Virus ins Leben zu rufen. Virenbaukästen gibt es nicht, und Anleitungen a la „how to build a virus" sind rar.

Zum anderen ist die Linux-Community selbst schuld. Kaum gingen die ersten Systeme durch Viren in die Knie, die auf Outlook abgestimmt waren, hagelte es schon die ersten Statements, dass so etwas unter Linux nicht möglich sei. Wohlweislich: Es wurde nicht gesagt, der Love-Letter kann Linux nicht befallen, da hier kein VB-Script existiert. Stattdessen wurde mit vollem Mund verlautbart, unter Linux ist so etwas überhaupt kein Thema. Fataler Fehler!

Virenprogrammierer reagieren teilweise wie trotzige Kinder. Wer sie provoziert, erreicht genau das Gegenteil. Durch diese Stellungnahmen aus der Linux-Gemeinde fühlten sich die Schädlingserzeuger natürlich herausgefordert. Nichts schafft so viel Gurustatus im Underground, wie das Unmögliche möglich zu machen: „Viren unter Linux sind nicht möglich? Beweisen wir doch das Gegenteil. BSD rulez!"

4.1.3 Binary-Viren sind unmöglich? Falsch!

Wer sich schon einmal die Anatomie eines EXE-Virus unter DOS angesehen hat, staunt über seine Einfachheit. Ein Virus erzeugt ein neues Programmsegment mit seinem Sabotage-Code, hängt sich an das Ende der EXE an und führt sich selbst beim Start des Programms aus. Anschließend patcht er fleißig im Speicher umher, um sich dort festzusetzen und beispielsweise den DOS-Funktionen-Interrupt 21h zu infizieren. Über diesen sorgt er dann auch für seine weitere Verbreitung.

Unter Linux müssen diese beiden Schritte - Infektion und Verbreitung - ebenfalls gelöst werden. Entgegen dauerhafter Falschinformationen können ELF-Binaries ähnlich wie EXE-Dateien infiziert werden. Das Anhängen von weiteren Programmsegmenten ist kein Problem. Im Gegenteil: Viren können sich sogar in „Füllinformationen“ von ELF-Segmenten einnisten, so dass teilweise gar kein separates - und leicht zu identifizierendes - Segment notwendig ist. Lediglich der unter DOS so beliebte, sich selbst im Speicher modifizierende Code ist unter Linux nicht möglich. Doch diese Probleme lassen sich auch anders lösen.

Die Infektion durch Patchen von Binaries im Dateisystem ist also kein Problem unter Linux. Die Vermehrung hingegen auf den ersten Blick schon eher. Wie soll sich ein Linux-Virus aus einem ELF-Programm in eine zentrale Systemfunktion ähnlich dem DOS-Funktionen-Interrupt einhängen? Das ist zwar schwierig, aber nicht unmöglich.

Hierzu ist es erforderlich, die Kernel-Binaries zu patchen. Ein etwas umfangreiches Unterfangen, da kaum ein Kernel dem anderen gleicht. Je nach verwendetem Compiler, Einstellungen zur Optimierung und Kernel-Version sieht das Binary des Kernels anders aus. Prinzipiell ist dieses Problem aber lösbar.

4.1.4 Verbreitung per Daemon

Für die Verbreitung von Viren ist es jedoch gar nicht notwendig, wie beim alten DOS vorzugehen! Die Lösung heißt: Daemons und Libraries. Wenn ein Virus einen Daemon, wie zum Beispiel sendmail, infizieren kann, muss er sich über seine Verbreitung keine Gedanken mehr machen. Daemons bieten außerdem ein wahres Paradies für spätere Hacker-Attacken. Hier wären die X11-Displayma nager, wie xdm, kdm oder gdm, aber auch syslogd selbst zu nennen. Wozu Passwörter cracken, wenn sie beim Eintippen im Klartext erkannt werden können.

Außerdem ist es möglich, dass der Virus direkt die Binaries im Dateisystem infiziert. Sollte es ihm außerdem gelingen, Libraries, wie etwa die C-Library, zu infizieren, sitzt er ohnehin in jedem dynamisch gelinkten Programm.

Spätestens seitdem Win32/linux.Winux sein Unwesen parallel unter Windows und Linux treibt, ist klar: Unter Linux können ELF-Programme infiziert werden. Man darf hier jedoch nicht vergessen, dass dieser Virus höchstwahrscheinlich noch ein „proof of concept“ ist. Er soll also zeigen, dass es prinzipiell geht.

Noch ist das Zukunftsmusik, da kein Virus nach diesem Prinzip wirklich relevant geworden ist. Es ist allerdings erschreckend, dass sich in der Szene bereits „Prototypen“ solcher Viren und konkrete Anleitungen mit Beispielen finden.

4.1.5 E-Mail-Viren sind unmöglich? Falsch!

Würmer unter Linux sind keine Fiktion mehr. Einige Wellen verursachte in der Fachpresse der Ramen Worm, der Red Hat Linux befiel. Der Wurm nutzt Sicherheitslücken in rpc.statd und wu-ftpd aus. Einmal im System angelangt, sendet er E-Mails an zwei webbasierte Mail-Accounts und startet die Suche nach neuen Opfern. Bei der Suche nach weiteren Rechnern, die infiziert werden können, reduziert er die Internet-Bandbreite des infizierten Systems enorm. Darin liegt auch sein hauptsächlicher Schaden.

Der seit Ende Februar vornehmlich in den USA aufgetretene Wurm Lion ist hier schon etwas gefährlicher. Er ersetzt verschiedene Systemdienste, um seine Verbreitung zu sichern und seine Existenz zu verschleiern. Sehr problematisch ist, dass er *login* ebenfalls ersetzt. Damit werden Passwörter ausspioniert und diverse Hintertüren eingerichtet.

Von E-Mail-Viren in der Manier von „I LOVE YOU“ und „Anna Kournikowa“ wurde Linux bislang verschont. Doch: E-Mail-Viren sind nicht nur auf Outlook Express und Windows beschränkt. Wer kmail, Netscape oder auch Mozilla genauer unter die Lupe nimmt, wird dort dasselbe Prinzip wie bei Outlook Express vorfinden. Auf eine Anlage geklickt, wird diese automatisch geöffnet. Dies öffnet Viren & Co. Tür und Tor.

Wer sich schon einmal mit der Makrosprache StarBasic von StarOffice befasst hat, wird sich gefragt haben: Wo ist der Unterschied zu VBA von MS Office? Er ist nicht allzu groß! Auch hier können Makroviren programmiert werden.

4.1.6 Linux ist sicher? Falsch!

Die Meinung „Linux ist sicher, weil Linux Linux ist“ ist weit verbreitet an der philosophisch geprägten Front zwischen Windows- und Linux-Anhängern. Tatsächlich ist Linux ein System, das sehr auf Sicherheit getrimmt werden kann. Patches sind schnell verfügbar und mit dem entsprechenden Know-how sogar selbst programmierbar. Doch das bedarf eines gewissen Aufwands.

Würmer wie der „Ramen Worm“ und „Lion“ zeigen, dass auch Linux Sicherheitslücken hat und diese gezielt ausgenutzt werden können. Patches sind zwar schnell verfügbar, aber bis dahin verursachen Sabotageprogramme erst einmal Schaden.

Leider sind Systeme wie OpenBSD, die schon bei der Programmierung auf Sicherheit getrimmt werden, die Ausnahme. Bei der Programmierung von Linux wurde auf Sicherheitsaspekte ebenso wenig oder genauso viel Wert gelegt wie bei Windows, UNIX oder DOS.

Programmierer sind eben nur in den seltensten Fällen Sicherheitsexperten. Und jede Eventualität können sie schon gar nicht voraussehen. Abgesehen von Konfigurationsfehlern gibt es genügend andere Angriffspunkte. Eine beliebte Möglichkeit zum Angriff auf ein Unix-System ist der Buffer Overflow. Häufig achten Programmierer nicht darauf, dass bestimmte Pufferlängen nicht überschritten werden können (wu-ftpd und sendmail lassen grüßen).

4.1.7 Angriffspunkt Buffer Overflow

Angenommen zur Aufnahme des eingegebenen Benutzernamens für einen Netzwerkdienst wird ein Puffer von 100 Bytes eingerichtet. Werden nun mehr als 100 Bytes an das Programm geschickt und es achtet nicht darauf, kommt es zu einem Buffer Overflow. Der zuständige Daemon, zum Beispiel sendmail, stürzt daraufhin ab. Ein klassischer „Denial of Service"-Angriff (DoS).

Was hat das mit Viren zu tun? Daemons wie sendmail werden als root ausgeführt. Statt den Puffer mit unsinnigen Daten (zum Beispiel 1000-mal X) überlaufen zu lassen, kann der Hacker auch ein „Ei" ins System legen. Dabei ist ein feindseliges Assembler-Programm in die Zeichenkette, die den Buffer Overflow auslöst, eingebettet. Sowie das Programm abstürzt, tritt dieses Programm in Aktion und manipuliert das System so, dass eine Shell mit Root-Privilegien gestartet wird. Der Angreifer kann so auf das System als root zugreifen.

Dieser Mechanismus ist natürlich für Virenprogrammierer und Hacker gleichermaßen interessant. Root-Rechte sind für einen Virus Gold wert. Für ELF-Binaries sind sie die Grundvoraussetzung, um andere Dateien im System zu patchen und so zu infizieren.

Doch gerade das egg/sendmail-Beispiel zeigt noch etwas anderes. Die Grenze zwischen Hacker- und Virenangriff ist hier fließend. Es genügt in diesem Fall voll und ganz, eine E-Mail zu empfangen, um das System zu infizieren. Der Anwender muss nicht auf etwas klicken, um den Virus ins System zu bringen. Es reicht vollkommen aus, dass er seine E-Mails abruft.

4.1.8 Fazit: Die Gefahr ist real!

Zweifelsohne ist die Virengefahr auch für Linux vorhanden. Sollte Linux weiter an Popularität gewinnen und auch im Desktop-Bereich nennenswerte Marktanteile erobern, wird es um die Artenvielfalt von Linux-Viren nicht schlecht bestellt sein. Da hilft dann auch die Scheuklappentechnik mit Sätzen wie „Hacker sabotieren kein Hacker-System" nichts mehr.

Die Linux-Distributoren wären gut damit beraten, die Virengefahr ernst zu nehmen. Wer sich als Sicherheitsexperte die Techniken zum automatischen Update via Internet näher ansieht, wird vor Schrecken schaudern. Die Update-Mechanismen laden Viren und Trojaner ja förmlich ein. Die Authentifizierung des Servers ist dürftig, Zertifizierungen der Server wären angebracht.

Auch den Contrib-Verzeichnissen sollte man nicht unbedingt trauen. Zu bedenken ist, dass die darin enthaltenen Packages von anderen Benutzern hochgeladen wurden. Wer weiß schon, wer das war?

Für Linux müssen inzwischen die gleichen Grundsätze wie für Windows gelten:

- Nicht allem trauen, was im Internet zum Download angeboten wird.
- Nicht alle E-Mail-Anlagen blind öffnen.
- Fremden Dokumenten mit Makrosprachen, wie StarOffice-Files, misstrauen.

Linux ist genauso anfällig wie andere Systeme. Jetzt heißt es endlich aufwachen. Nur eine erkannte Gefahr ist eine gebannte Gefahr!

tecCHANNEL-Links zum Thema	Webcode
Virenscanner im Test	a214
Grundlagen Computerviren	a213
ILOVEYOU: Hilfe und Hintergründe	a393
Die Entwicklung der digitalen Plagegeister	a86
Windows 2000 Bugreport	a317
Aktuelle IE-Sicherheitslücken	a185

4.2 Hacker-Angriffe unter Linux

Ein modernes Serverbetriebssystem, eine gut konfigurierte Firewall und ein Administrator, der regelmäßig die Logfiles durchforstet - sie gelten als Garanten für ein sicheres lokales Netzwerk.

Um es jedoch gleich vorweg zu sagen: Ein absolut sicheres System befände sich in einem verschlossenen Raum von Fort Knox und hätte keinen Zugang zum Internet. Mit anderen Worten: Kein System oder Netzwerk ist absolut sicher. Daher ist Paranoia die beste Strategie:

- Zunächst machen Sie ein System so sicher wie möglich.
- Trauen Sie Ihren Sicherheitsmaßnahmen dennoch nicht, sondern gehen Sie stattdessen weiterhin vom „worst case“ aus - dem erfolgreichen Hacker-Einbruch.

Warum diese Paranoia? Ganz einfach: Nicht alles liegt in Ihrem Einflussbereich. Durch Schwächen oder Bugs der eingesetzten Software und durch den Einfallsreichtum von Hackern kann es, selbst bei noch so guter Sicherheitsstrategie und perfekter Konfiguration, zu einem Einbruch kommen.

Wenn die Gefahr besteht, dass Hacker-Angriffe erfolgreich sind, sollte man sich Gedanken über deren Erkennung machen. Nichts ist schlimmer, als ein erfolgreicher Angriff, der zu spät oder gar nicht bemerkt wird. Der Angreifer kann Änderungen an Dateien und Konfigurationen vornehmen und diese anschließend verwischen. Solche Änderungen zu finden, gleicht der sprichwörtlichen Suche nach der Stecknadel im Heuhaufen.

Ganz davon abgesehen, wird ein Hacker das geenterte System als Plattform für Angriffe auf weitere Systeme verwenden. Als „angreifender“ Host fungiert dann der „erbeutete“ Server. Image-Schaden und die Abmahnkosten sind oft um einiges höher als der direkte Schaden auf dem Server. Ein Administrator gerät dann schnell in Erklärungsnotstand, wenn er keinen plausiblen Grund für den mehrwöchigen Aufenthalt eines Hackers im System liefern kann.

4.2.1 Spuren im Logfile

Hacker können Sie im Wesentlichen auf zwei Arten entdecken:

- In flagranti zum Zeitpunkt des Einbruchs, sei es durch ein Intrusion Detection System (IDS) oder manuell.
- Durch Spuren in den Logfiles. Letzteres ist jedoch nicht mehr ganz so glücklich, da der Hacker dann in den meisten Fällen bereits aus dem System geflüchtet ist.

Die Protokollierungen in den Logfiles sollten Sie immer lesen. Selbst wenn Sie ein noch so ausgereiftes IDS installiert haben, sollten Sie diese Aufgabe nicht vernachlässigen.

Die Logfiles können Sie wie die Spuren am Tatort eines Verbrechens auffassen. Anhand dieser Dateien ist es möglich, den Hacker-Angriff zu rekonstruieren und den Täter gegebenenfalls zu überführen.

Wie im wirklichen Leben wird der Eindringling versuchen, seine Spuren zu verwischen. Die erste Hacker-Regel nach dem Einbruch in ein System ist das Bereinigen der Logfiles. Wenn Sie einen Hacker also dingfest machen und wissen wollen, was er überhaupt auf Ihrem System angestellt hat, müssen Sie dort für die Sicherung der Hinweise sorgen.

Konkret heißt das, dass Sie sich überlegen müssen, wie Sie die Logfiles gegen Änderungen absichern, respektive Modifikationen innerhalb kürzester Zeit erkennen. Hierzu können Sie die Logfiles auf einem anderen Host unterbringen oder durch kryptographische Methoden schützen. Ein Hacker wird sich an diesen Maßnahmen in aller Regel die Zähne ausbeißen, so dass Sie ihn entdecken können.

4.2.2 Logfiles auf Remote-Host

Auf einem Linux-Rechner protokolliert der *syslogd*-Daemon alle Meldungen in den Logfiles. Diese liegen in der Regel im Verzeichnis */var/log* auf dem System selbst. Ein Angreifer muss sich also in der Regel nur um die Dateien in diesem Verzeichnis kümmern.

Der Hacker steht jedoch vor einem Problem, wenn die Logfiles nicht mehr lokal auf dem System liegen. Die aufgezeichneten Informationen lassen sich nur mit großem Aufwand oder gar nicht mehr modifizieren, wenn sie auf einem anderen Rechner liegen. Der Eindringling müsste dann erst noch diesen anderen Host kapern.

In der Konfigurationsdatei */etc/syslog.conf* steht, wo *syslogd* die Meldungen speichert. Normalerweise sind für die entsprechenden Meldungsarten Dateien angegeben, in denen die Protokollierung erfolgen soll. So bewirkt beispielsweise das Kommando

```
kern.* -/var/log/kern.log
```

dass alle Kernel-Meldungen in der Datei */var/log/kern.log* landen.

Um alle Meldungen auf einem anderen Host zu protokollieren, geben Sie in dieser Datei die Zeile

```
*.* @host
```

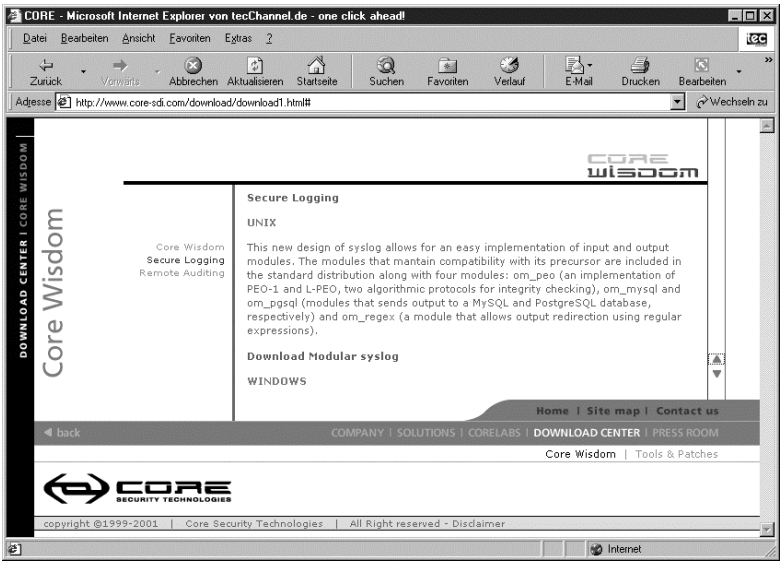
an. *host* steht hierbei für den Netzwerknamen des Computers, auf dem die Meldungen protokolliert werden sollen. Auf diesem Host muss allerdings ebenfalls ein *syslogd* laufen.

Der Remote-Host protokolliert die Meldungen nun in seinen Logfiles mit. Bedenken Sie, dass es sinnvoll ist, die Logfiles durch die standardmäßig angegebenen Aktionszeilen in */etc/syslog.conf* zusätzlich auch lokal zu halten. Damit hat der Hacker dann immerhin eine Spielwiese, und er merkt vielleicht nicht sofort, dass die Aufzeichnung seiner Aktivitäten auch remote erfolgt.

Außerdem bietet dieses redundante Führen von Logfiles einen weiteren Vorteil: Programme können die Meldungen vergleichen und somit Modifikationen erkennen. Allerdings benötigt man hierfür ein relativ intelligentes Programm, da die Zeiteinträge zwischen den jeweiligen Logfiles zumindest im Sekundenbereich differieren können. Leider hat die Remote-Protokollierung einen kleinen Haken: Kommt beim syslogd eine wahre Flut von Meldungen an, findet die Protokollierung nicht mehr korrekt statt.

4.2.3 Logfiles schützen

Sehr interessant ist es, die Logfiles durch kryptographische Methoden (PEO-1) zu schützen. Der Schutz funktioniert dabei ähnlich wie beim digitalen Signieren. Über diese Signatur lässt sich zweifelsfrei feststellen, ob ein Unberechtigter die Dateien geändert hat.



Logfiles absichern: Mit Secure Logging von Core-Wisdom können Sie Logfiles auch in mySQL-Datenbanken ablegen oder per Fingerabdruck gegen Veränderung sichern.

Linux unterstützt dieses Vorgehen ohne Probleme und Zusatzkosten. Allerdings muss man dazu den alten syslogd über Bord werfen. Und auch den Secure Syslog (*ssyslogd*) sollten Sie nach Möglichkeit nicht mehr verwenden. Dieser enthält zwar bereits PEO-1, wird aber nicht mehr weitergepflegt, was im Sicherheitsbereich ein potenzielles Risiko bedeutet.

Setzen Sie stattdessen von Anfang an auf den Nachfolger Modular Syslog *mysys logd* (www.core-sdi.com). Er ist weitest gehend zu syslogd kompatibel. So unterstützt mysyslogd das gleiche Format in */etc/syslog.conf*, erweitert dieses jedoch für seine zusätzlichen Features.

4.2.4 Modular Syslog

msyslogd arbeitet nach einem intelligenten und erweiterbaren Prinzip. Bei jeder Aktion (= Ort der Protokollierung, beispielsweise Datei oder Host) in der syslog-Konfiguration lässt sich angeben, welches Modul von msyslogd die Protokollierung übernehmen soll. Für die herkömmliche, nicht geschützte *Buchführung* kommt das Modul *classic* zum Einsatz, bei PEO-1 ist *peo* einzutragen. Standardmäßig verwendet msyslog das klassische, ungeschützte Log-Format.

Um PEO-1 zu verwenden, ändern Sie beispielsweise den Befehl

```
kern.* /var/log/kern.log
```

in

```
kern.* %peo /var/log/kern.log
```

Diese neue Zeile gibt an, dass zum Protokollieren der Kernel-Meldungen in die Datei*„/var/log/kern.log* das PEO-Modul verwendet werden soll. Dieses Logfile ist damit über PEO-1 geschützt. Das mitgelieferte Programm *peochk* testet Logfiles auf widerrechtliche Änderungen.

Selbstverständlich kann msyslogd auch remote protokollieren und zusätzlich schützen, zum Beispiel:

```
*.* %peo @host
```

Derzeit bietet msyslogd außerdem die Module *mysql* und *pgsql* zum Protokollieren in eine MySQL- oder PostgreSQL-Datenbank. Das Modul *regex* dient zum Analysieren von Log-Einträgen und zum gezielten Protokollieren.

Msyslogd hat allerdings auch Schwächen. Wenn ein Modul die Ein-/Ausgabe blockiert, kann es zum Verlust von Log-Informationen kommen. Dieser Bug ist jedoch bekannt und wird in den nächsten Versionen wahrscheinlich nicht mehr existieren.

4.2.5 Der Protokollierung letzter Schliff

Mit Hilfe der Informationen in den Logfiles können Sie einen Eindringling nachträglich überführen. Deshalb sollten Sie alles zu deren Schutz unternehmen. *logrotate* ist in diesem Zusammenhang ein nützliches Tool.

logrotate rotiert und komprimiert die Logfiles automatisch. Zudem kann es Logfiles an eine E-Mail-Adresse schicken. Machen Sie von diesem Feature Gebrauch, stehen Ihnen die Logs auch dann noch zur Verfügung, wenn Sie im System bereits überschrieben wurden. Weitere verbesserte Protokollierungsfunktionen bietet der Daemon *xinetd* (http://freshmeat.net/projects/xinetd/), der eine sichere Alternative zu *inetd* darstellt. Seine Features umfassen:

- erweiterte Zugriffskontrolle
- Schutz vor DoS-Attacken
- Beenden von Verbindungen, die nicht mehr den Sicherheitskriterien entsprechen
- Benutzerinteraktion sowie
- erweiterte Protokollierungsfunktionen

Das einzige Hindernis liegt im gänzlich anderen Format der Konfigurationsdatei. Deren Aufbau ist jedoch nicht kompliziert, so dass der Umstieg inklusive Einarbeitung in *xinetd* keine allzu große Hürde darstellen dürfte.

4.2.6 Simple Intrusion Detection

Ein Eindringling lässt sich durch seine Änderungen im System aufspüren. Doch nicht nur die Protokolldateien stehen auf der Modifikationsliste eines Hackers ganz oben. Einmal im System angelangt, wird er sich die Konfiguration des Hosts vornehmen. Will er den Rechner als Plattform für Attacken auf weitere Systeme verwenden, wird er sich zudem Hintertürchen und sonstige ungewollte Dienstleistungen einrichten.

Diese Änderungen hinterlassen natürlich auch unvermeidliche Spuren im Dateisystem. Wird eine Konfigurationsdatei bearbeitet, ändern sich ihr Zeiteintrag und meist auch ihre Größe. Richtet jemand Hintertüren oder Trojaner ein, lässt sich das häufig anhand von Änderungen in den S-Bits oder der Größe von Binaries, wie Systemprogrammen und Daemons, erkennen. Solche Modifikationen kann ein Angreifer nur durch sehr großen Aufwand verbergen.

Das Aufspüren dieser Änderungen ist recht einfach. Sie müssen sich nur einmal eine (gesicherte) Vergleichsgrundlage schaffen. Nachdem das System aufgesetzt, vollständig konfiguriert und als definitiv „untermieterfrei“ bestätigt ist, kann sich so schnell nichts mehr ändern. Was liegt also näher, als diesen Zustand durch einen „Schnappschuss“ festzuhalten. In regelmäßigen Abständen

vergleicht ein cron-Job diesen Soll-Zustand mit dem Ist-Zustand des Systems. Bei Änderungen sollten dann die Alarmglocken klingen.

4.2.7 Baselines

Einen solchen „Schnappschuss“ - im Fachjargon „Baseline“ genannt - können Sie mittels der Befehle *ls* und *find* auf der Shell anlegen. Als erstes erstellen Sie eine Baseline für die Dateien mit gesetztem S-Bit. Hier ist es sinnvoll, nur alle Verzeichnisse in *PATH* durchzuarbeiten und nicht alle Unterverzeichnisse von /. Sollten die einzelnen User nämlich in Ihren eigenen *~/bin*-Verzeichnissen selbst Programme anlegen und diese mit S-Bits versehen, würde Ihr Baseline-Audit jedes Mal Alarm schlagen, wenn ein User damit experimentiert.

Gewisse S-Bit-Änderungen bleiben dabei unerkannt. Diesen Kompromiss gilt es jedoch einzugehen. Außerdem nisten sich Trojaner, Viren und Hintertüren ohnehin meist in den PATH-Binaries ein.

Um die Baseline für die S-Bits festzulegen, verwenden Sie *ls* in Zusammenspiel mit find. Mit

```
find Verzeichnis -perm +6000
```

erhalten Sie eine Liste aller Dateien in *Verzeichnis*, bei denen entweder das SUID (4000) oder GUID (2000) gesetzt ist. Um die kompletten Verzeichnisinformationen zu den Dateien zu erhalten, übergeben Sie diese Ergebnisliste an den ls-Befehl:

```
ls -lad --full-time `find Verzeichnis -perm +6000`
```

Damit erhalten Sie pro Datei mit einem gesetzten S-Bit eine Zeile im *ls-Long*-Format, die auch den ausführlichen Zeiteintrag (*--full-time*) enthält.

Statt *Verzeichnis* sollen nacheinander die Einträge aus der Shell-Variablen PATH eingesetzt werden. Um die durch Doppelpunkte separierten Einträge aus PATH zu erhalten, verwenden Sie am besten ein AWK-Programm. Dort können Sie als Record-Separator (RS) den Doppelpunkt eintragen und so die PATH-Einträge in Zeilen aufsplitten:

```
echo $PATH | awk ´BEGIN { RS=":" }  { print }´
```

4.2.8 Baseline-Check

Wenn Sie dies alles nun in eine *for*-Schleife packen und die Ausgabe in eine Datei umleiten, erhalten Sie Ihre S-Bit-Baseline:

```
(for f in `echo $PATH | awk ´BEGIN { RS=":" }  { print }´`;
  do ls -lad --full-time `find $f -perm +6000` ;
done) > ~/s-bits.baseline
```

Die Datei *~/s-bits.baseline* enthält nun den „Schnappschuss" Ihrer S-Bits. Achten Sie bitte darauf, dass Sie die Baseline als root anlegen. Schließlich sind es die über root erreichbaren Binaries, die gefährdet sind. Außerdem sind in der Regel im PATH eines normalen Users die sbin-Verzeichnisse mit den Daemons nicht eingetragen. Ob Ihr System noch in dem Zustand ist, in dem es beim Anlegen der Baseline war, können Sie durch ein *diff* feststellen:

```
(for f in `echo $PATH | awk ´BEGIN { RS=":" }  { print }´`;
  do ls -lad --full-time `find $f -perm +6000` ;
done) | diff - ~/s-bits.baseline
```

Diesen Test kann ein cron-Job regelmäßig durchführen. Der Vorteil von *diff* ist, dass es dem alten Unix-Grundsatz „no news are good news" folgt. Es liefert nur eine Ausgabe, wenn eine Differenz beim Vergleich festgestellt wurde. Tritt dieser Fall ein, können Sie das diff-Protokoll per E-Mail direkt an root senden.

4.2.9 Weitere Baselines

Sie sollten zudem die restlichen Executables in PATH und die Konfigurationen in Baselines festhalten. Die Executables können Sie in dieselbe Baseline wie die S-Bits aufnehmen. Tragen Sie hierzu statt *-perm +6000* die Option *-perm +6111* ein. Sie können auch eine Trennung vornehmen und eine separate Baseline anlegen. Verwenden Sie dazu die Option *-perm +0111*. Diese Baseline erfasst auch nochmals die S-Bit-Dateien, da S-Bit-Files meist ausführbar sind. Sie erhalten bei einer Änderung also zwei Alarmierungen für die S-Bit-Datei.

Die meisten und wichtigsten Konfigurationen liegen im Verzeichnis */etc*. Hierfür sollten Sie daher auf jeden Fall eine Baseline und einen cron-Job anlegen:

```
ls -lRa --full-time /etc > ~/etc.baseline
```

Zum Test der Integrität dient dieses diff-Kommando:

```
ls -lRa --full-time /etc | diff - ~/etc.baseline
```

4.2.10 Wichtige Hinweise zu Baselines

Wann immer Sie neue Software installieren oder die Konfiguration ändern, müssen Sie die Baselines neu anlegen. Bevor Sie das jedoch machen, sollten Sie die Integritätstests durchführen. Ansonsten übersehen Sie eventuell einen Einbruch und nehmen ihn als gültig in das System auf!

Beachten Sie, dass Baselines nicht die schnellste Methode sind, um einen Eindringling aufzuspüren. Die Reaktionszeit hängt davon ab, wie oft der cron-Job ausgeführt wird. Starten Sie den Test beispielsweise alle zehn Minuten, kann ein Hacker im schlimmsten Fall innerhalb dieser zehn Minuten Schaden anrichten.

Noch abschließend ein Hinweis zur Sicherung der Baselines selbst. Zunächst einmal sollten Sie Baselines nur für root lesbar machen und grundsätzlich schreibschützen:

```
chmod 0400 <Baseline>
```

Statt *Baseline* setzen Sie die einzelnen Dateien ein. Sinnvoll ist es auch, die Baselines auf einen anderen Host zu legen und per Netzwerk verfügbar zu machen. Dieser Baseline-Host sollte möglichst hinter einer weiteren Firewall liegen und nicht per Internet erreichbar sein. Auch sollte das Mounten des Baseline-Hosts über ein Netzwerk-Dateisystem wie NFS oder SMB nur schreibgeschützt erfolgen. Neue Baselines sollten Sie nur per Diskette oder CD-ROM auf den Baseline-Server übertragen können.

4.2.11 Aktive Audits

Die bisher vorgestellten Methoden zur Intrusion Detection sind passiver Natur. Sie konnten einen Eindringling nur anhand seiner Auswirkungen erkennen. Es gibt allerdings auch aktivere Methoden.

Einen noch nicht eingedrungenen Hacker können Sie in den Protokollen Ihrer Firewall erkennen. Wenn ständig ein Verbindungsaufbau über die Firewall nach demselben Schema abgelehnt wird, liegt der Verdacht nahe, dass ein Hacker direkt an Ihrem System klebt oder ein Sniffer auf Ihren Server angesetzt wurde. Doch dies hat nichts mit Intrusion Detection im eigentlichen Sinne zu tun. Schließlich ist der Hacker noch nicht im System, und es liegt somit keine „Intrusion“ vor, die erkannt werden könnte. Allerdings sollten Sie hier hellhörig werden und den Webmaster des angreifenden Systems verständigen.

Der erste Weg, einen aktiven Hacker im System zu erkennen, ist eine offene Internet-Verbindung zu ihm. Wenn er in Ihrem System gelandet ist, muss ja eine Connection zu seinem eigenen System bestehen. Ideal zum Darstellen der offenen Verbindungen ist das Tool lsof (ftp://vic.cc.purdue.edu/pub/tools/unix/lsof). Es bietet wesentlich mehr Informationen als „netstat“ und ist deshalb für die Intrusion Detection besser geeignet.

4.2.12 Überwachung mit lsof

Eine Liste mit den bestehenden Internet-Verbindungen erhalten Sie über

```
lsof -i
```

```
COMMAND    PID      USER   FD   TYPE DEVICE SIZE NODE NAME
in.telnet 1114   telnetd    2u  IPv4   5858       TCP kernighan:telnet->leibniz:support (ESTABLISHED)
```

Wer spricht mit wem? lsof zeigt an, welche Verbindungen zum lokalen Rechner bestehen.

Das Beispielbild zeigt Ihnen, dass eine Telnet-Verbindung besteht. Ein User auf dem Host *leibniz* hat sich auf *kernighan* (Ihrem System) eingeloggt. Ideal ist, dass Sie hier sofort sehen, von wo aus die Verbindung hergestellt wurde. Die Host-Angaben bestehen aus *Computer:Port*. *Computer* kann hierbei ein per DNS oder */etc/hosts* aufgelöster Netzwerkname sein, oder die IP-Adresse. *Port* gibt den TCP/IP-Port an.

Sie können damit feststellen, von wo der Angriff erfolgt. Im obigen Fall müsste der Webmaster von *leibniz* kontaktiert werden, um dem Hacker auf die Spur zu kommen.

Über die Angabe *PID* erhalten Sie die ID des Prozesses, der auf Ihrem eigenen System für die Verbindung zuständig ist. Hier ist es der Wert „1114“. Sie sehen auch im Klartext, welches Programm (hier in.telnet) die Verbindung abwickelt und unter welchem User (telnetd) der Prozess läuft.

Wollen Sie den Hacker unsanft aus dem System werfen, verwenden Sie einfach einen kill-Aufruf mit der PID als Argument. Zum Beispiel beenden Sie die obige TELNET-Verbindung unsanft per

```
kill -s SIGKILL 1114
```

Bestehende Netzwerkverbindungen lassen sich per *grep* auflisten:

```
lsof -i | grep ´\>->´
```

Über das Umleiten in Dateien und *diff* über diesen Dateien können Sie in cron-Jobs die Verbindungen mitprotokollieren.

4.2.13 Gelöscht und doch offen?

Sabotage-Programme müssen sich etwas einfallen lassen, um sich selbst Daten merken zu können. Diese in Files zu protokollieren, ist eine effiziente Methode. Doch Dateien, die urplötzlich im System auftauchen, entgehen selbst einem durchschnittlichen Administrator nicht. Trotzdem müssen die digitalen Schädlinge nicht auf solchen Komfort verzichten.

Unter Linux/Unix ist es auf Grund des Mehrbenutzerkonzepts durchaus möglich, offene Dateien zu löschen, ohne dass sie ad hoc aus dem System verschwinden. Die Prozesse, die die Dateien noch geöffnet haben, können damit ungehindert weiterarbeiten, obwohl die Datei aus dem Dateisystem verschwunden ist. Dieses Phänomen machen sich ungebetene Gäste häufig zu Nutze.

Hier ein Beispiel: Legen Sie zuerst einen Prozess an, der eine Datei öffnet:

```
cat > ~/intrusion
```

Alle Eingaben von der Tastatur landen jetzt in der Datei *~/intrusion*. Schieben Sie diesen Prozess jetzt mittels [Strg+Z] in den Hintergrund. Ein

```
ls ~/intrusion
```

beweist Ihnen, dass die Datei tatsächlich existiert. Wenn Sie die Datei per

```
rm -f ~/intrusion
```

löschen, taucht sie im Verzeichnis nicht mehr auf. Holen Sie allerdings den Prozess wieder in den Vordergrund (Befehl *fg*), können Sie weiterhin Zeichen eingeben, ohne dass es einen Fehler gibt.

4.2.14 Versteckte Dateien anzeigen

Sie können jetzt spekulieren, dass die Eingaben des *cat*-Prozesses ins Leere gehen. Das ist jedoch nicht der Fall, wie die Befehlszeile

```
lsof +L1
```

zeigt. Dieser Befehl listet alle offenen Dateien mit einem Link-Counter kleiner als 1 auf. Mit anderen Worten: die gelöschten Dateien.

```
COMMAND    PID   USER   FD   TYPE DEVICE   SIZE NLINK    NODE NAME
cat       1130 oliver    1w   REG    3,6     22     0 132058 /home/oliver/intrusion (deleted)
```

Aufgespürt: lsof findet diese zwar offiziell gelöschte, aber dennoch offene und vorhandene Datei eines Trojaners.

Wie Sie sehen, ist die Datei zwar gelöscht (*deleted*), aber noch geöffnet. Wenn Sie jetzt noch ein paar Mal den cat-Prozess in den Vordergrund holen und fleißig Zeichen eintippen und wieder nach hinten schieben, werden Sie bei *lsof +L1* feststellen, dass sich die Größe der Datei ändert.

Die Daten werden also noch darin gespeichert. Für ein Sabotage-Programm ist es ein Leichtes, die in dieser Datei gespeicherten Daten wieder herauszulesen und weiter auch hineinzuschreiben. Solange die Datei geöffnet ist, kann der Schädling damit schalten und walten, wie er will. Sowie Sie jedoch den cat-Prozess in den Vordergrund holen und über [Strg+D] oder [Strg+C] beenden, verschwindet die Datei ebenfalls.

Auf offene, aber gelöschte Dateien sollten Sie ein Auge haben. Zwar verwenden auch einige Daemons dieses Prinzip, wenn jedoch plötzlich ein neuer Prozess auftaucht, sollten Sie aufhorchen.

Für diesen Fall gelten die gleichen Regeln wie für die Angaben von

```
lsof -i
```

Über Scripts und cron-Jobs können Sie solche offenen, gelöschten Dateien auch protokollieren und so schnell Unstimmigkeiten erkennen.

4.2.15 Fazit

Wer sein System mit einer Firewall abschottet und dann glaubt, sein System sei sicher, begeht einen fatalen Fehler. Administratoren mit dieser Einstellung werden sich früher oder später unangenehme Fragen anhören müssen, denn mit der einmaligen Absicherung ist die Arbeit noch lange nicht getan.

Die ständige Überwachung des Systems und das Erkennen von ungebetenen Gästen im System sind zentrale und wichtige Aufgaben des Administrators. Glücklicherweise muss er die Arbeit nicht komplett von Hand erledigen, denn die meisten Audits lassen sich gut automatisieren.

Zusätzliche Sicherheit bieten fertige Intrusion Detection Systems (IDS), die die Überwachung auf Angriffe in unterschiedlich tiefen Niveaus vornehmen. Die für Linux frei verfügbaren Systeme sollten Sie auch nutzen. Auf der nächsten Seite finden Sie eine Liste gebräuchlicher IDS.

tecCHANNEL-Links zum Thema	Webcode
Linux als Firewall	a695
Linux Firewall mit ipchains	a704
Masquerading mit Linux	a707
TCP/IP-Netze mit Linux	a562
Firewall-Grundlagen	a682

4.2.16 Intrusion Detection Systeme

Die folgende Tabelle gibt Ihnen einen Überblick über verschiedene IDS. Sie sind in steigender Komplexität angeordnet.

Frei verfügbare IDS	
IDS	**Beschreibung**
Chkwtmp	Chkwtmp analysiert wtmp und meldet gelöschte Einträge. Damit lassen sich Änderungen in Logfiles entdecken. http://sunsite.ics.forth.gr/pub/systools/chkwtmp/chkwtmp-1.0.tar.gz
tcplogd	tcplogd erkennt Stealth-Scans, die mit „halb-offenen" Verbindungen arbeiten. http://kalug.lug.net/tcplogd/
Snort	Snort ist ein Packet-Filter, Sniffer und Logger, der Intrusion Detection via Baselines ermöglicht. http://www.snort.org
HostSentry	HostSentry erkennt Login-Anomalien und meldet diese.
Shadow	Shadow ist ein freies IDS, dass auf einer Reihe von freien Tools, wie tcpdump, aufbaut. Es ist auf ca. 14.000 Hosts im militärischen und kommerziellen Bereich installiert. http://www.psionic.com/products/hostsentry.html
MOM	MOM ist ein leistungsfähiges IDS. Es arbeitet verteilt über das Netzwerk und ist nicht als 1-Host-Lösung gedacht. http://www.biostat.wisc.edu/~annis/mom3/
Humming-Bird Systems (Hummer)	Hummer ist ein IDS für große Netzwerke. Die einzelnen Teilsysteme können Sicherheitsinformationen austauschen und übers Netz verteilen. http://www.csds.uidaho.edu/~hummer/
AAFID	AAFID ist ein relativ neues Monitoring System und IDS im Alpha-Stadium. Es arbeitet verteilt über das Netzwerk. Die einzelnen Infomationen tragen relativ kleine Agenten zusammen, die auf den Hosts im Netzwerk verteilt werden. www.cerias.purdue.edu/coast/projects/aafid-announce.html

tecCHANNEL-Archiv-CD

Die tecCHANNEL-Knowledgebase bietet das tecCHANNEL-Profiwissen der Jahre 2000 und 2001. Insgesamt enthält sie über 650 MByte und mehr als 20.000 HTML-Seiten. Sie finden fast alle News, Reports, Tests und Know-how-Beiträge der vergangenen zwei Jahre im gewohnten Layout und mit der Navigation der Webseite von tecCHANNEL.DE. So können Sie in Ruhe im Archiv von tecCHANNEL stöbern, wenn Sie unterwegs oder gerade nicht online sind.

Die CD entspricht dem ISO-9660-Standard und ist auf IBM-PCs, Macintosh-Rechnern und Linux/Unix-Systemen lesbar. Voraussetzung ist lediglich ein HTML-3.2-konformer Browser. Sie können die tecCHANNEL-Archiv-CD online über den tecShop (www.tecChannel.de/tecshop) zum Preis von 19,90 Euro inklusive Versandkosten bestellen. Newsletter- und Magazin-Abonnenten erhalten die CD zum Vorzugspreis von 14,90 Euro. Daneben können Sie die CD nochmals verbilligt für 9,90 Euro downloaden und selber auf CD brennen oder auf Festplatte ablegen (http://www.tecChannel.de/tec_shop/downloads.html).

Im Download-Bereich finden Sie zudem PDF-Dokumente aller Ausgaben des tecCHANNEL-Magazins. tecCHANNEL-Newsletter-Abonnenten erhalten für die Downloads der PDFs einen Preisnachlass von 50 Prozent. Statt zwei Euro für bereits erschienene Hefte und 2,90 Euro für die aktuellen Magazine bezahlen sie nur einen beziehungsweise 1,45 Euro. Die Bestell-Links zu den Sonderangeboten finden Sie jeweils am Ende der tecCHANNEL-Newsletter. Wenn Sie noch keinen Newsletter im Abo haben, können Sie den täglichen Newsletter im Text- oder HTML-Format sowie den Wochenrückblick über diesen Link (www.tecChannel.de/news/abo/abo.htm) schnell und komfortabel anfordern.

Im Volltext: Mehr als 20.000 HTML-Seiten kompakt und komplett auf einer CD-ROM.

4.3 Desktop-Firewall mit Linux 2.4

Viele Linux-Anwender schrecken angesichts der kryptischen Konfigurationsfiles vor der Einrichtung einer Firewall zurück. Mit geeignetem grafischen Werkzeug bereitet dies aber selbst Einsteigern keine Probleme.

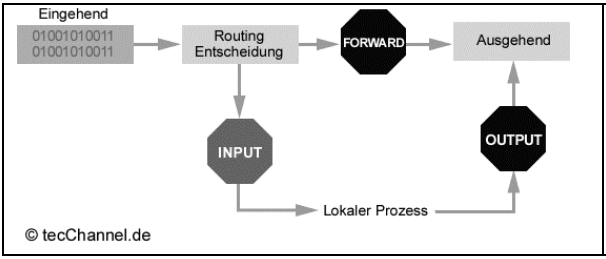
Das in bester Unix-Manier von vornherein als Netzwerk- und Multiuser-Betriebssystem konzipierte Linux glänzt mit einer Reihe eingebauter Sicherheitsmechanismen. Doch selbst auf bestens gepflegten Systemen bleiben noch potenzielle Eingangstüren offen: Dabei handelt es sich um die Ports jener Dienste, die der Rechner nach dem Willen seines Benutzers nach außen anbieten soll. Auch hier kann Linux mit eingebautem Werkzeug Abhilfe schaffen: Bereits seit Kernel 1.1 bringt das freie Unix auch Firewalling-Werkzeuge mit.

An dieser Stelle wollen wir uns mit dem Einsatz der seit Kernel 2.3 eingeführten Iptables-Firewall auf dem Desktop beschäftigen. Als Basis für die Beispiele verwenden wir Red Hat Linux 7.1.

4.3.1 Iptables

Die Iptables-Architektur arbeitet mit einer im Gegensatz zum Vorgänger ipchains wesentlich einfacheren Systemarchitektur. Der Kernel hält drei Listen von Filterregeln namens *INPUT*, *OUTPUT* und *FORWARD* vor. Man nennt sie auch Ketten, da sie jeweils aus einer Liste sequenziell abzuarbeitender Regeln bestehen.

Jede Regel bestimmt anhand des Paket-Headers, was mit anfallenden Paketen zu geschehen hat. Entspricht der Paket-Header nicht dem Regelkriterium, wird das Paket an die nächste Regel der Kette weitergereicht. Hat ein Paket die Kette ohne Zutreffen einer Regel bis zum Ende durchlaufen, greift die Policy der Kette. Sie wird ein solches nicht identifizierbares Paket im Regelfall verwerfen (*DROP* oder *REJECT*).



Geradlinige Architektur: Iptables kommt mit lediglich drei Filterlisten aus.

Geht ein Paket an der Netzwerkschnittstelle ein, wertet der Kernel zunächst dessen Zieladresse aus (Routing-Entscheidung). Ist das Paket für den lokalen

Rechner bestimmt, reicht er es an die INPUT-Kette weiter. Falls nicht, entscheidet die Forwarding-Konfiguration über das weitere Schicksal des Pakets. Bei aktivem Forwarding übernimmt die FORWARD-Kette die Weiterverarbeitung, andernfalls wird das Paket verworfen.

Auch alle von Prozessen auf dem lokalen Rechner versendeten Pakete müssen die OUTPUT-Liste durchlaufen. Falls diese das Paket nicht ausfiltert, verlässt es anschließend den Rechner über die angeforderte Schnittstelle.

4.3.2 Implementation

Jede Regel in den Filterketten besteht aus zwei Komponenten: Die eine prüft, ob die Regel überhaupt auf das Paket anzuwenden ist („matcht“). Die andere bestimmt, was in diesem Fall mit dem Paket zu geschehen hat. Beide Regelteile lassen sich über Kernel-Module flexibel konfigurieren - eine der wichtigsten Neuerungen von Iptables.

Im Fall von Red Hat 7.1 beispielsweise lagern die entsprechenden Shared Libraries im Verzeichnis */lib/iptables*, insgesamt bringt das Betriebssystem 30 davon mit. Einige stellen nützliche Paket-Matches zur Verfügung, andere legen spezielle Aktionen für durch die Regel erfasste Pakete fest. Jedes Modul muss zunächst geladen und danach über Kommandozeilenoptionen für den jeweiligen Einsatzzweck angepasst werden. Entsprechend umfangreich gestalten sich die Scripts, die eine typische Firewall-Konfiguration aufsetzen. Nun ist das manuelle Editieren seitenlanger Textdateien ohnehin nicht jedermanns Sache, zudem schleichen sich dabei nur allzu leicht Fehler ein. Gerade Linux-Einsteiger und Desktop-Nutzer geben daher meist nach einigen Versuchen klein bei und verzichten auf die Nutzung der Firewall.

4.3.3 Firewall Builder

Tatsächlich gibt es jedoch auch für Einsteiger und Mausverliebte keinen Grund, auf die Sicherheit einer Firewall zu verzichten. Im umfangreichen Fundus von Sourceforge.net findet sich ein Tool, mit dem nahezu jeder auf einfache Weise eine Iptables-Konfiguration einrichten kann: Der modular aufgebaute Firewall Builder (http://sourceforge.net/projects/fwbuilder/) besteht aus einer komfortablen GUI-Komponente und Regel-Compilern sowohl für Iptables- als auch Ipchains-Firewalls.

Während des Probeeinsatzes bei tecChannel.de zeigte der Firewall Builder schon in der Beta-Phase keine gravierenden Schwächen. Inzwischen befindet er sich in der Stable-Version 1.0.2. Die Entwickler bieten das Werkzeug sowohl in Binärvarianten für Caldera, Mandrake, Red Hat und SuSE als auch im Quellcode an. Für unseren Workshop verwenden wir die Red-Hat-Variante von Konfigurationsprogramm und Iptables-Compiler. Beide installieren wir auf einem Rechner unter Red Hat Linux 7.1 Deluxe.

4.3.4 Rahmenbedingungen

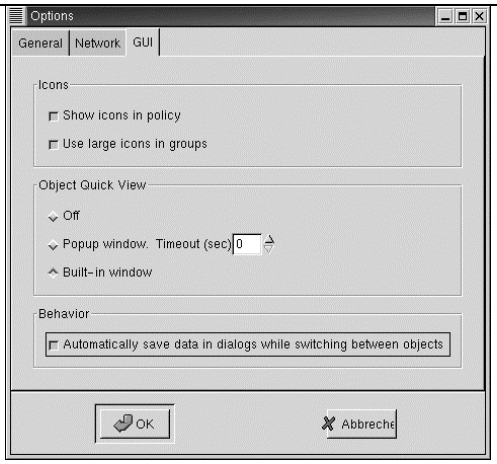
Als Aufgabenstellung wählen wir die Absicherung eines typischen Netzwerk-Client. Die Maschine soll also Windows-Shares sowohl bereitstellen als auch nutzen. Zudem wollen wir vollen Internet-Zugriff ermöglichen: Neben Web- und Mailzugriff soll der Benutzer also auch Usenet-News lesen und FTP-Zugriffe vornehmen können.

Einfache Managementaufgaben stehen ebenfalls mit im Pflichtenheft: So muss sich die geschützte Maschine von lokalen Überwachungsrechnern aus per ICMP und SNMP erreichen lassen, der Benutzer soll grundlegende Diagnosetools wie ping und traceroute einsetzen können.

„They can't crack what they can't find", so lautet die Goldene Regel der Rechnersicherheit. Daran wollen wir uns halten und den abgesicherten Rechner außer für seine dedizierten Kommunikationspartner unsichtbar machen.

4.3.5 Basiskonfiguration

Bevor wir den Firewall Builder das erste Mal starten, legen wir in */etc* als künftiges Arbeitsverzeichnis für die Applikation das Directory */etc/fwbuilder* an. Dort sollen später die Konfigurationsdatei des Tools sowie die erstellten Firewall-Regeln lagern.



Einfach: die Basisoptionen des Firewall Builder.

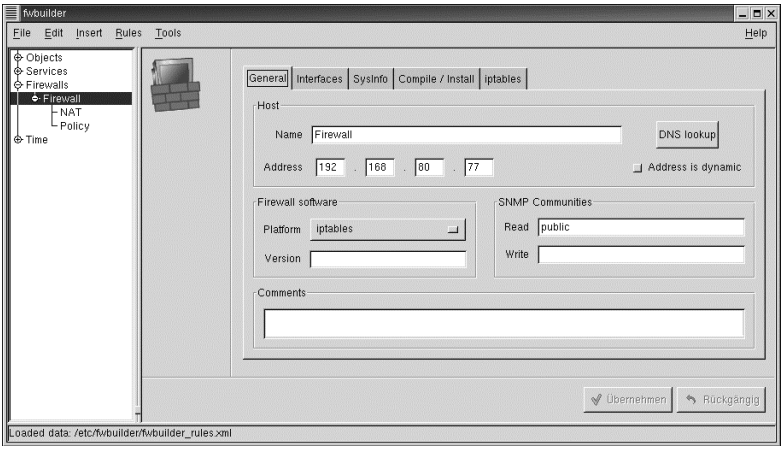
Jetzt starten wir den Firewall Builder. Als erstes gilt es, einige grundlegende Einstellungen für die Applikation selbst zu treffen. Dazu rufen wir den Menüpunkt *Edit/Options* auf. Im ersten Tab des daraufhin erscheinenden Pop-up-

Fensters tragen wir den Pfad zu unserem Arbeitsverzeichnis - also */etc/fwbuilder* - ein. Die Einstellungen des Netzwerk-Tabs können wir bei den Default-Werten belassen: Also je zehn Sekunden Timeout und einen Wiederholungsversuch.

Auf dem dritten Tab sollte auf jeden Fall unter *Behavior* die automatische Sicherung aller Einstellungen beim Wechsel zwischen den Objekten aktiviert sein. Die Anzeigeoptionen für Icons und Objekte können Sie ganz nach Geschmack variieren. Allerdings erweisen sich die eingestellten Vorgaben bei der weiteren Arbeit erfahrungsgemäß als hilfreich.

4.3.6 Das Firewall-Objekt

Ein wesentliches Stichwort beim Umgang mit dem Firewall Builder ist bereits gefallen: Objekte. Bei der weiteren Konfiguration baut das Tool auf die Definition diverser Objekte auf. Dazu zählen Netzwerke und Hosts (*Objects*), Protokolle und Ports (*Services*), sowie Zeitspannen (*Time*). Das wichtigste davon stellt das Firewall-Objekt selbst dar: Also der lokale Rechner, den es über eine zugeordnete Policy zu schützen gilt. Daher erstellen wir als erstes über den Menüpunkt *Insert/Firewall* ein entsprechendes Objekt.

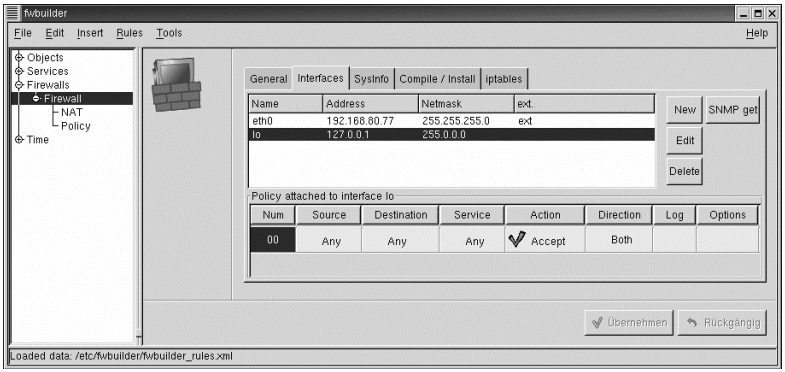


Objektorientiert: Zunächst muss das Firewall-Objekt erstellt werden.

Dieses muss über insgesamt fünf Reiter mit Einstellungen versorgt werden. Auf dem Tab *General* tragen wir eine eingängige Bezeichnung für das Objekt so wie dessen Netzwerkadresse ein. Als unterstützte Firewall-Software wählen wir *Iptables*, die anderen Werte belassen wir auf den Voreinstellungen.

4.3.7 Firewall-Interfaces

Auf dem Tab *Interfaces* legen wir mit Hilfe des *New*-Buttons die beiden Interfaces *lo* (den internen Loopback) und *eth0* (die Netzwerkkarte) an. Dabei definieren wir *eth0* als externes Interface. Anschließend definieren wir per Rechtsklick in der unteren Hälfte für beide Interfaces jeweils eine Default-Policy. Die Konfiguration sieht für beide Interfaces fast identisch aus: *Source*, *Destination* und *Service* lassen wir auf *Any* stehen, als *Direction* wählen wir *Both*. Die so erstellte Policy gilt also jeweils für alle Pakete in alle Richtungen.



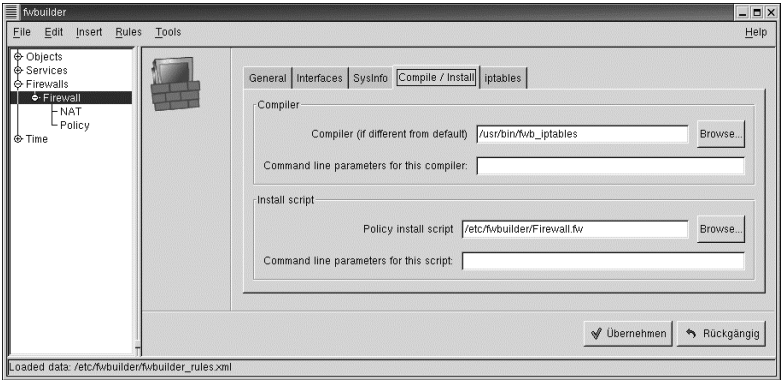
Durchlässig: Das interne Loopback muss alle Pakete akzeptieren.

Der kleine, aber wesentliche Unterschied: Für den internen Loopback tragen wir als Regel (*Action*) *Accept* ein, lassen also alle Pakete zu. Für die Netzwerkschnittstelle dagegen verbieten wir per *Deny* alles. Um die Filterregel anzuwählen, genügt jeweils ein Rechtsklick der Maus auf das entsprechende Feld. Auf ein Protokollieren der jeweiligen Aktionen können wir ebenso verzichten wie auf die Angabe von Zusatzoptionen. Die Felder *Log* und *Options* lassen wir daher leer.

Die Angaben des Reiters *Sysinfo* benötigen wir für unsere Aufgabenstellung nicht. Wir können ihn daher überspringen und uns den Angaben auf dem Tab *Compile / Install* widmen.

4.3.8 Compilierung und Installation

Der Regel-Compiler liegt nach einer Standardinstallation von Firewall Builder im Verzeichnis */usr/bin*. Da wir Regeln für die Iptables-Firewall erstellen wollen, geben wir als Pfad zum Compiler */usr/bin/fwb_iptables* an.



Schlachtentscheidend: Hier gilt es, den Pfad zum korrekten Regel-Compiler einzustellen.

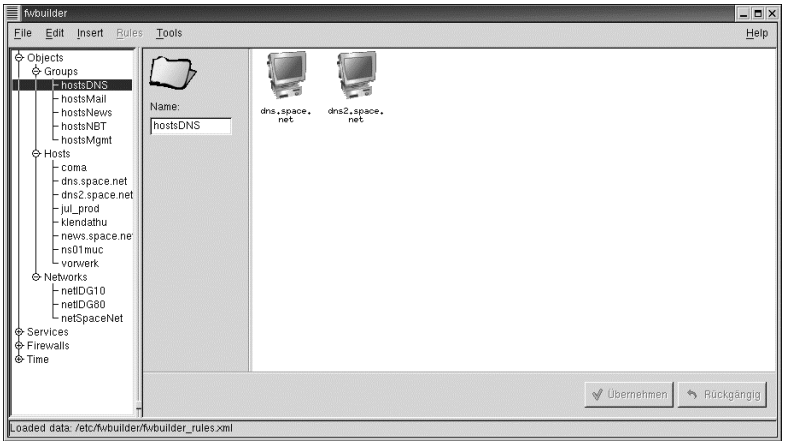
Als Installations-Script können wir, da wir lediglich eine Firewall für die lokale Maschine einrichten, direkt das von Firewall Builder später erstellte Konfigurations-Script angeben. Es liegt im Arbeitsverzeichnis der Applikation - bei uns also */etc/fwbuilder* - und trägt den Namen des Firewall-Objekts mit der Endung *.fw*.

Auf dem letzten Tab namens *iptables* benötigen wir als einzige Option die Anweisung zum Laden der passenden Kernel-Module. Sie findet sich auf der rechten Seite des Fensters in der Gruppe *Options*.

Nachdem wir diese Grundeinstellungen erledigt haben, speichern wir den momentanen Status über den Menüpunkt *File/Save As* als */etc/fwbuilder_rules.xml*.

4.3.9 Hosts und Netzwerke

Als Nächstes definieren wir über *Insert/Host* und *Insert/Network* die Rechner und Netzwerke, mit denen wir dediziert kommunizieren wollen. Im Fall der Hosts geben wir dazu jeweils Rechnername und IP-Adresse ein, für die Netzwerke vergeben wir einen eingängigen Namen und definieren die Netzwerkadresse. In unserem Beispiel verwenden wir die zwei Class-C-Subnetze netIDG10 und netIDG80 (192.168.10.0 wie auch 192.168.80.0, Netzmaske 255.255.255.0). Sie bilden unser internes Firmennetzwerk. Hinzu kommt das Class-B-Netz des Providers, netSpaceNet (195.30.0.0/255.255.0.0).



Who is who: Die zu kontaktierenden Hosts werden nach Funktion in Gruppen zusammengefasst.

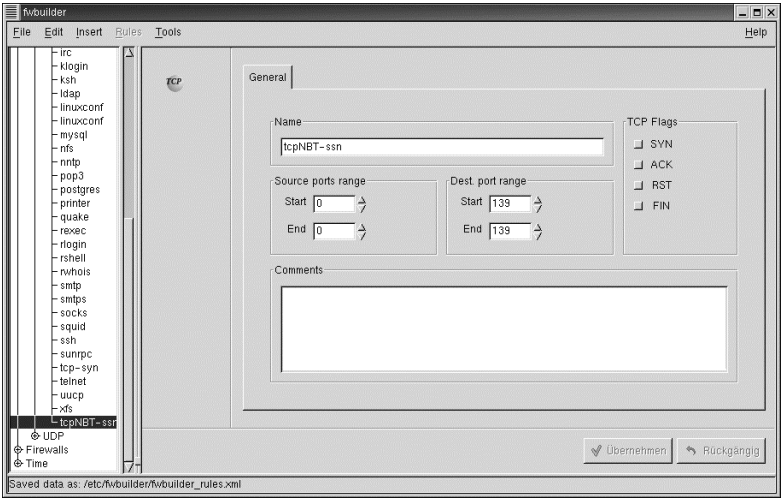
Nachdem wir unsere Änderungen sicherheitshalber gespeichert haben, gruppieren wir die Hosts nach Funktionsbereichen. Dazu bilden wir zunächst per *Insert/Group of Objects* die Gruppen

- hostsDNS (DNS-Server),
- hostsLocal (interne Subnetze)
- hostsMail (Mailserver),
- hostsNews (Newsserver),
- hostsNBT (lokales Windows-Netzwerk) und
- hostsMgmt (lokale Management-Stationen).

Diesen ordnen wir jetzt die Hosts und Netze zu, indem wir sie per rechtem Mausklick kopieren und in der entsprechenden Gruppe wieder einfügen.

4.3.10 Dienste und Protokolle

Im Abschnitt *Services* bringt Firewall Builder bereits eine reiche Auswahl an vordefinierten Diensten in den Kategorien *ICMP*, *IP*, *TCP* und *UDP* mit. Darunter finden sich mit Ausnahme eines einzigen auch schon alle, die wir für unsere Aufgabenstellung benötigen. Den fehlenden Service definieren wir über *Insert/TCP*, da es sich dabei um TCP-Pakete zum Abwickeln einer Session in Windows-Netzwerken handelt.



Selbstgestrickt: Die Definition eigener Dienste ist schnell durchgeführt.

Wir geben ihm einen aussagekräftigen Namen, in unserem Fall *tcpNBT-ssn* (TCP-Paket für Sessions via NetBIOS over TCP/IP). Als Quellport lassen wir alle Anschlüsse zu, in Firewall-Builder-Nomenklatur: 0 bis 0. Als Zielport nutzen wir Port 139 (alias *netbios-ssn*). Die TCP-Flags können wir für unseren Einsatzzweck ignorieren und belassen sie abgewählt.

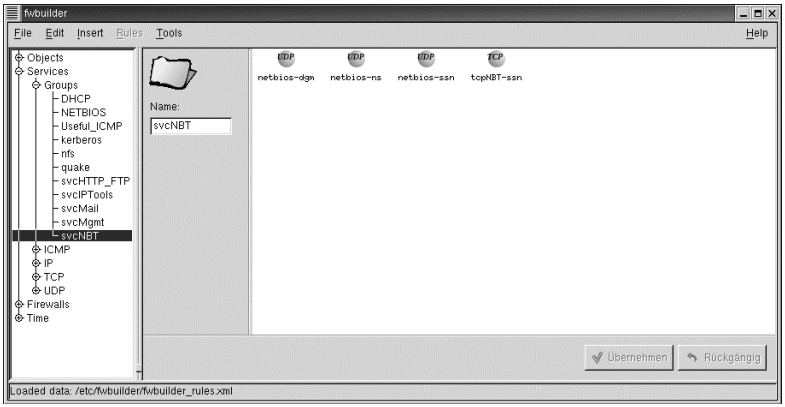
4.3.11 Dienstegruppen

Wie bei den Hosts fassen wir jetzt auch die Protokolle zu Funktionsgruppen zusammen. Für Dienste, die nur ein einziges Protokoll nutzen, können wir uns diese Arbeit allerdings sparen. Darunter fallen bei unserer Aufgabenstellung *UDP /dns* (DNS), *TCP /ftp_data* (FTP -Daten) und *TCP/nntp* (News).

Für die restlichen Verbindungstypen definieren wir über *Insert/Group of Services* die Dienstegruppen:

- svcHTTP_FTP (Web und FTP),
- svcIPTools (lokales ping und traceroute),
- svcMail (Mailversand und Empfang),
- svcMgmt (Echo und SNMP für Managementzwecke) und
- svcNBT (Windows-Netzwerk).

Auch hier speichern wir die Einstellungen nach der Definition ab. Anschließend fügen wir den Dienstegruppen per Copy-and-Paste die nötigen Protokolle hinzu.



Grüppchenbildung: Auch die Dienste sollten zu funktionsorientierten Paketen geschnürt werden.

4.3.12 Dienstefunktionen

Die Gruppe svcHTTP_FTP dient dazu, sowohl Webbrowsern als auch FTP-Clients den Betrieb zu ermöglichen. Dazu fügen wir die Dienste *TCP/http*, *TCP/https* sowie *TCP/ftp* ein. svcIPTools stellt die wichtigsten Netzdiagnose-Funktionen für unsere Workstation bereit. Dazu benötigen wir *ICMP /ping_request* und *UDP /traceroute*.

Die Dienste der Gruppe svcMail ermöglichen den Betrieb unseres Mail-Clients. Alle gehören zur TCP-Protokollfamilie. Die Services *smtp* und *smtps* dienen zum Versenden von Nachrichten. Die Abholung von Mails erfolgt je nach Mailserver über *pop3* oder *imap/imaps*.

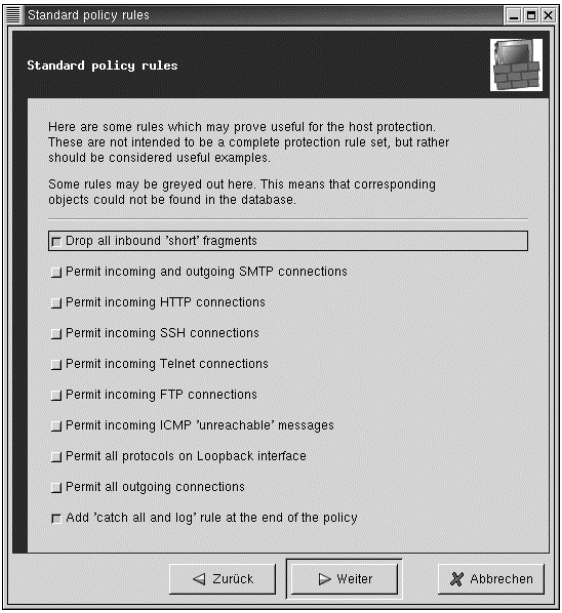
Lokale Managementrechner müssen unsere Workstation über Ping und SNMP erreichen können. Das stellt die Gruppe svcMgmt mit den Diensten *ICMP/ping_request* und *ping_reply* sowie *UDP/snmp* und *snmp-trap* sicher.

Last but not least soll unsere Maschine auch im Windows-Netzwerk Funktionen nutzen und bereitstellen können. Dazu tragen wir in die Gruppe svcNBT die Dienste *UPD/netbios-ns* (Name Service), *UDP/netbios-dgm* (Datagramme), *UDP/netbios-ssn* (Session) sowie das selbst definierte *tcpNBT-ssn* ein. Anschließend speichern wir unsere Änderungen.

4.3.13 Die Firewall-Policy

Damit haben wir alle notwendigen Vorarbeiten abgeschlossen und können nun an die Definition der eigentlichen Firewall-Policy gehen. Wir wählen dazu bei unserem Firewall-Objekt den Punkt *Policy* an. Bei den ersten Regeln lassen wir

uns vom Firewall Builder zur Hand gehen, indem wir im Menü *Rules* den Punkt *Help me build firewall policy* selektieren.



Hilfestellung:
Firewall Builder definiert etliche Regeln bei Bedarf automatisch.

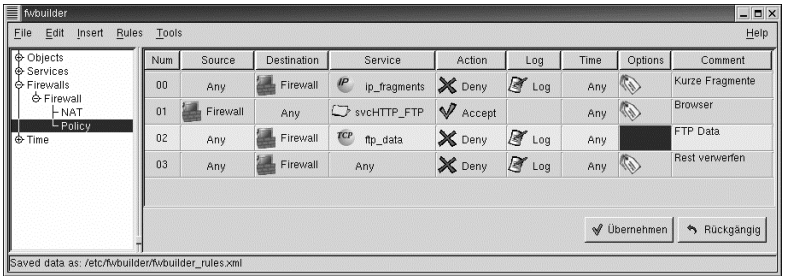
Im hier erscheinenden Fenster überspringen wir den Einführungstext und wählen auf der nächsten Seite *Firewall protects local host*. Nach einem Klick auf den *Weiter*-Button weisen wir die Firewall durch die Auswahl des ersten und letzten Punkts an, kurze IP-Fragmente stets auszufiltern und unbekannte Pakete immer zu verwerfen.

Beides dient der Sicherheit: Diverse Angriffsmethoden versuchen mit Short Fragments, Firewalls zu unterlaufen. Im lokalen Netzbetrieb dagegen kommen solche Pakete nicht vor. Das Verwerfen aller unbekannten Pakete lässt nur solche Daten die Firewall passieren, die wir im Folgenden explizit zulassen.

4.3.14 Regeln für FTP und HTTP

Über einen Rechtsklick auf das Nummernfeld der untersten Filterregel fügen wir darüber eine neue Zeile ein. Die einzelnen Felder dieser neuen Regel füllen wir aus den Objektdefinitionen per Copy-and-Paste. Als *Source* geben wir die Firewall an, als *Destination Any*. In die *Service*-Spalte tragen wir unsere Dienstegruppe *svcHTTP-FTP* ein. Die Action lautet *Accept*, auf ein Log der Pakete

können wir verzichten. Als Time lassen wir *Any* zu. Per Rechtsklick editieren wir die Optionen und markieren, wie später auch bei fast allen anderen Regeln, den Punkt *Turn off stateful inspection for this rule*. Unter *Comment* können wir einen beliebigen Erläuterungstext angeben.



Hallo, Welt: Die Definition der HTTP/FTP-Rules gestaltet sich trickreich.

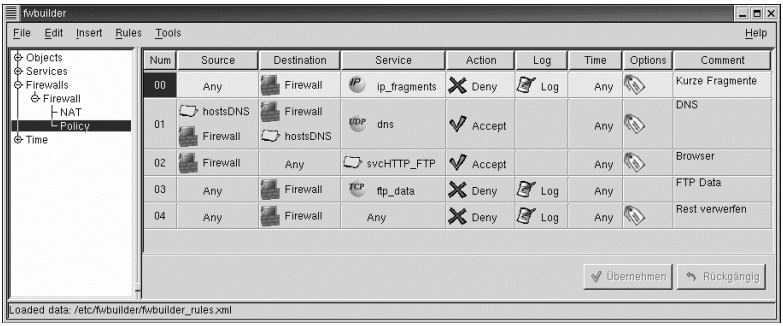
In *svcHTTP_FTP* haben wir die Pakettypen *http*, *https* und *ftp* zusammengefasst. Das erlaubt Browsern, Pages von Webservern abzurufen sowie Kontakt zu FTP-Servern aufzunehmen. Letzteres gilt ebenso für FTP-Clients. Weder Browser noch FTP-Programm können mit dieser Regel aber FTP-Daten empfangen. Dazu muss der eingehende Port 20 (*ftp-data*) geöffnet werden. Dabei handelt es sich jedoch um einen klassischen Angriffspunkt von Crackern. Daher definieren wir hier eine eigene Regel, die wir unter der ersten einfügen.

Source ist hier *Any*, *Destination* unsere Firewall. Als Dienst kommt wie erwähnt *TCP/ftp_data* zum Zuge. Als *Action* geben wir *Deny* an, lassen dafür jedoch bei *Options* die *Stateful Inspection* aktiviert. Dies hat zur Folge, dass nur solche Pakete abgelehnt werden, die unverlangt bei uns eintreffen. Daten von Verbindungen, die wir selbst per *svcHTTP_FTP* initiiert haben, lässt die Firewall dagegen durch. Sicherheitshalber schalten wir hier die Protokollierung dennoch ein.

4.3.15 Regeln für DNS

Unternähmen wir jetzt einen Versuch, die bislang definierten Regeln auszutesten, käme dennoch keine Verbindung zu Stande. Bislang fehlt unserem Rechner die Möglichkeit, die Hostnamen der Gegenstellen per DNS aufzulösen. Dazu fügen wir jetzt oberhalb der HTTP/FTP-Einstellungen eine passende Regel ein.

Hier kommt zum ersten Mal eine der von uns definierten Hostgruppen zum Einsatz: hostsDNS. Die Angabe der zugelassenen Hosts über eine Gruppe bietet zwei Vorteile: Zum einen lässt sich die Regel kürzer fassen, da statt aller Hosts nur die Gruppe eingetragen werden muss. Zum anderen kann bei Veränderung der Hosts die Regel gleich bleiben, lediglich der Inhalt der Gruppe ändert sich.

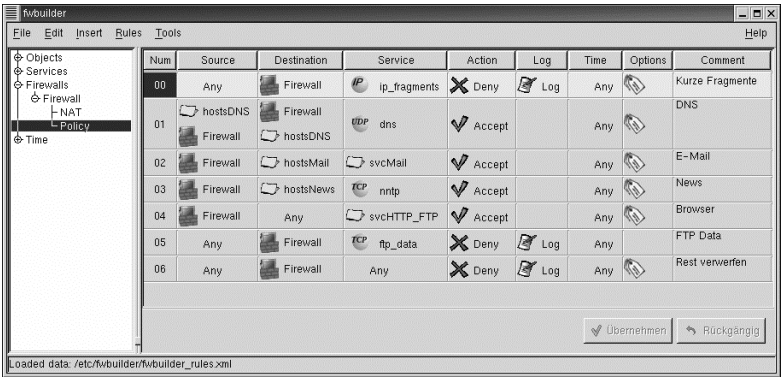


Conditio sine qua non: Ohne korrekte Namensauflösung scheitern die meisten Dienste.

hostsDNS und unsere Firewall tauchen hier sowohl als Quelle wie auch als Ziel auf, als Protokoll geben wir *TCP/dns* an. Da wir die Pakete akzeptieren, können wir uns eine Protokollierung sparen. Auch eine Prüfung per Stateful Inspection ist hier überflüssig.

4.3.16 Regeln für Mail und News

Da wir ohnehin gerade Regeln für Internet-Dienste aufstellen, können wir hier gleich noch Mail und News ergänzen. Wir fügen sie direkt unter dem DNS-Eintrag ein. Als *Source* fungiert in beiden Fällen unsere Firewall, und natürlich erlauben wir die Pakete. Auf *Logging* und *Stateful Inspection* können wir dementsprechend verzichten. Für das Protokoll *TCP /nntp*, also die News, fungiert als Gegenstelle unsere Newsserver-Gruppe *hostsNews*.

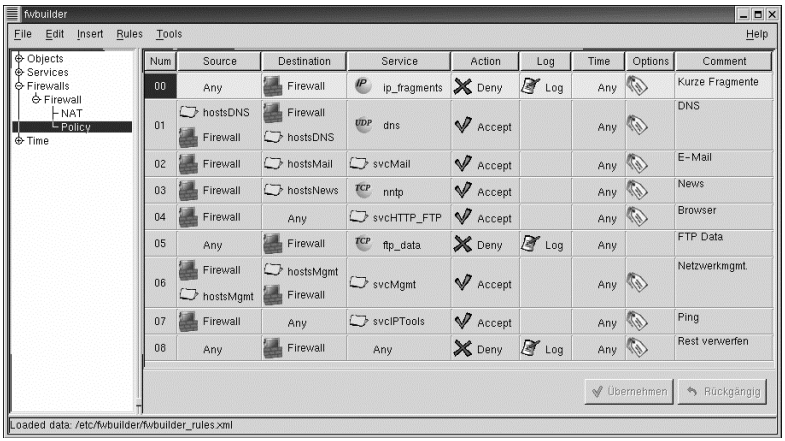


Nachrichtendienst: Die Rules für Mail und News fallen recht trivial aus.

Für die Maildienste kommt als Service unsere Gruppe svcMail zum Einsatz. Damit lassen sich via SMTP oder Secure SMTP Mails verschicken respektive Nachrichten per POP3 oder IMAP empfangen. Im Feld *Destination* setzen wir unsere Mailserver-Gruppe *hostsMail* ein.

4.3.17 Regeln für Managementtools

Im nächsten Schritt fügen wir am Ende unserer Filterkette die Regeln für interne und externe Managementtools hinzu. Zu Überwachungszwecken erlauben wir den Stationen der Gruppe hostsMgmt den Zugriff per ping und SNMP auf unseren Rechner. Dabei erfolgt der Datenfluss in beide Richtungen, weshalb alle beteiligten Rechner sowohl als *Source* wie auch als *Destination* auftauchen.



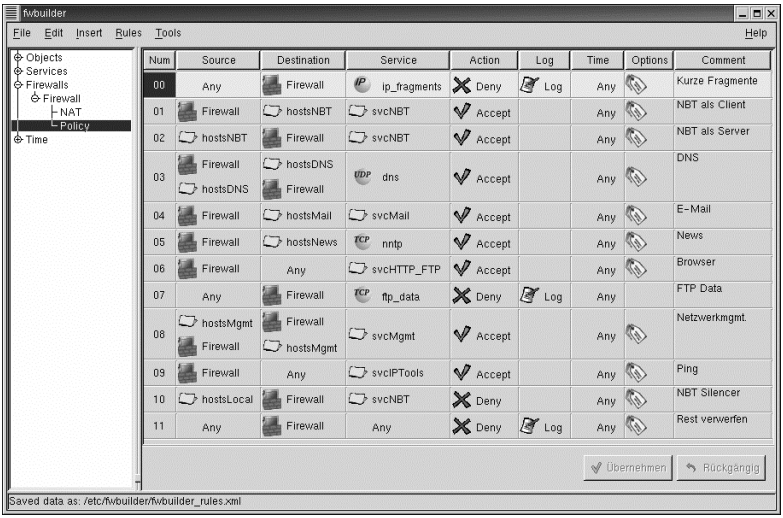
Lean Management: Ping, traceroute und SNMP müssen genügen.

Anders bei der Regel für die gängigsten IP-Diagnosetools: Hier ist nur Verkehr von der Firewall zu beliebigen anderen Rechnern gestattet. Die erlaubten Dienste umfassen *TCP/ping_request* und *UDP/traceroute*, die wir in der Servicegruppe *svcIPTools* gebündelt haben. Damit lässt sich nicht nur die Erreichbarkeit entfernter Rechner überprüfen, sondern gegebenenfalls auch der Leitweg dorthin.

4.3.18 Regeln für Windows-Netze

Zu guter Letzt definieren wir noch die Regel für das File- und Printsharing unter Windows. Dazu fügen wir unterhalb des Fragmentfilters zwei Zeilen ein. Sie unterscheiden sich nur durch die Richtung der Pakete: Um Shares im Netz zu

nutzen, dient die Firewall als *Source* und die Gruppe *hostsNBT* als *Destination*. Um Shares im Netz zur Verfügung stellen, kehren wir die Richtung der Daten in der zweiten Zeile um. In beiden Fällen nutzen wir die Dienste aus *svcNBT* als Protokoll.



Windows hin, Windows her: File- und Printsharing sollen in beiden Richtungen funktionieren.

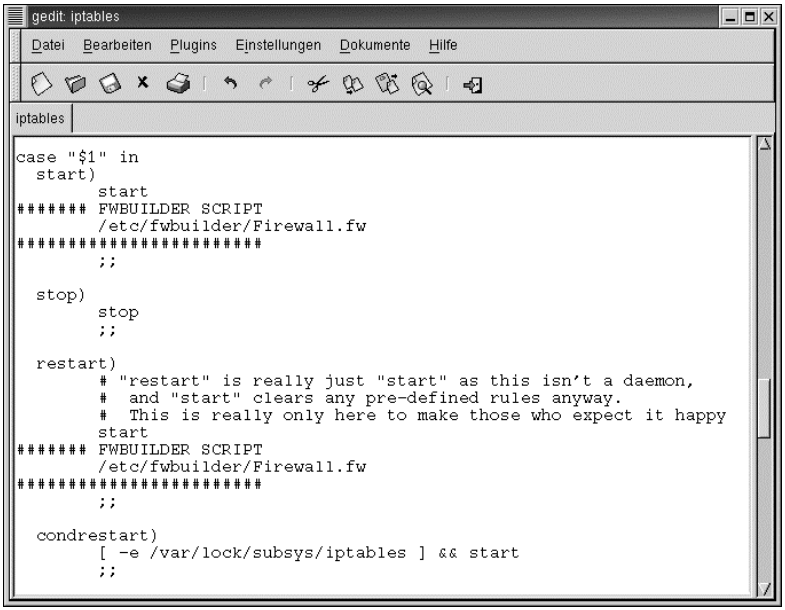
Das stellt die Verbindungen zu jenen Rechnern sicher, mit denen unsere Firewall als Server oder Client via NBT Daten austauscht. Allerdings klopfen an den entsprechenden lokalen Ports 137 bis 139 auch solche Rechner aus den lokalen Subnetzen an, die keine Verbindung erhalten sollen. Dies bläht das Log unnötig auf, da solche Pakete erst durch die Schlussregel abgefangen und dabei protokolliert werden.

Daher definieren wir als vorletzte Regel in unserer Kette einen *NBT Silencer*. Er blockt solche Pakete ohne Protokollierung ab. Als *Source* geben wir die Gruppe *hostsLocal* an, in der die lokalen Subnetze zusammengefasst sind. Als Destination fungiert die Firewall, als Dienste wiederum *svcNBT*. *Stateful Inspection* ist hier überflüssig.

4.3.19 Firewall starten

Damit haben wir die Konfiguration der Firewall abgeschlossen und speichern sie ein letztes Mal ab. Anschließend wählen wir im Menü den Punkt *Rules/Compile*

an. Firewall Builder generiert jetzt das Firewall-Script und speichert es im Arbeitsverzeichnis */etc/fwbuilder* ab. Von dort kann es über *Rules/Install* gestartet und anschließend ausgetestet werden.



Startautomatik: Die Integration in /etc/init.d/iptables sichert die Übernahme des aktuellen Regelsatzes.

Um automatisch bei jedem Systemstart das aktuelle Firewall-Script zu laden, tragen wir es in */etc/init.d/iptables*, das Startup-Script für Iptables, ein. Dazu suchen wir die Marken *start)* und *restart)* und ergänzen sie um den Befehl zur Ausführung der in */etc/fwbuilder/* gespeicherten Regeln.

Eine Überprüfung unserer Firewall mit einem Portscanner a la *nmap* (www.insecure.org/nmap/) zeigt, dass unser Rechner durch nicht autorisierte Stationen tatsächlich nicht mehr entdeckt werden kann. Potenziellen Angreifern bleibt er also künftig verborgen.

4.3.20 Fazit

Das beschriebene Konfigurationsbeispiel reizt die Fähigkeiten des Firewall Builder bei weitem nicht aus. So lässt sich durch die Definition und Einbindung von Zeitspannen die Geltungsdauer von Regeln zeitlich beschränken. Daneben

kann das Tool mehrere Firewall-Konfigurationen parallel vorhalten und bei Bedarf auf verschiedene Rechner verteilen. Viele dazu notwendige Informationen holt sich Firewall Builder bei Bedarf per Knopfdruck via DNS und SNMP.

In jedem Fall reduziert Firewall Builder den Aufwand beim Erstellen, Austesten und Verteilen von Firewall-Policies drastisch. Selbst Einsteiger mit minimalem Vorwissen konfigurieren mit diesem Tool auf Grund der fast intuitiven Bedienung innerhalb kürzester Zeit einen brauchbaren Paketfilter. Es gibt also wirklich keine Ausrede mehr, die hervorragenden Firewall-Fähigkeiten von Linux weiter brachliegen zu lassen.

tecCHANNEL-Links zum Thema	Webcode
Sichere Linux-Workstation	a720
Linux-Firewall mit Ipchains	a704
TCP/IP-Netze mit Linux	a562
So funktioniert TCP/IP	a209
Linux 2.4 für den Desktop	a706
Linux für den Desktop	a494
Linux für den Server	a487

5 Linux als Firewall

Wer nicht nur Einzelplatz-PCs, sondern ganze Netzwerke vor Angriffen schützen will, kommt um eine Firewall nicht herum. Im letzten Kapitel gehen wir daher zunächst detailliert auf die wichtigsten Firewall-Grundlagen ein. In weiteren Einzelbeiträgen lernen Sie dann die Firewall spezifisch anzupassen und sie für Spezialaufgaben zu optimieren. Auf dem Programm steht das Thema Masquerading unter Linux und Firewalls mit ipchains.

5.1 Firewall-Grundlagen

Die Sicherheit steht an erster Stelle, wenn das private Netzwerk eines Unternehmens (LAN) mit dem Internet verbunden ist. Eine zunehmende Anzahl von Mitarbeitern braucht Zugang zu Internet-Diensten wie dem WWW, E-Mail, FTP und Remote-Verbindungen (Telnet, SSH). Unternehmen wollen zudem für ihre Webseiten und FTP-Server den öffentlichen Zugang über das Internet ermöglichen. Dabei muss die Sicherheit der privaten Netze gegenüber unautorisierten Zugriffen von außen gewährleistet sein. Der Administrator muss das lokale Netzwerk gegen das große Chaos „Internet“ abschirmen, damit Daten nicht in unbefugte Hände geraten oder gar verändert werden. Für Firmen, die vom Internet-Zugang abhängig sind, stellen auch die so genannten DoS-Attacken eine große Gefahr dar.

Mit Firewalls lassen sich Netzwerke gegen unbefugte Zugriffe von außen absichern. Die verfügbaren Lösungen reichen von der Zusatzsoftware bis hin zu speziellen Geräten, die ausschließlich auf diese Aufgabe ausgelegt sind. In ihrer grundlegenden Funktionsweise unterscheiden sich die Systeme allerdings nur wenig.

5.1.1 Definition einer Firewall

Eine Firewall besteht aus einer Gruppe von Netzwerkkomponenten (Hard- und Software) an der Schnittstelle zweier Netze. Sie gewährleistet die Einhaltung von Sicherheitsrichtlinien zwischen einem privaten und einem öffentlichen (nicht sicheren) Netz, wie zum Beispiel dem Internet. An dieser „Brandschutzmauer“ entscheidet sich, auf welche Dienste innerhalb des privaten Netzes zugegriffen werden kann und welche Dienste des nicht sicheren Netzes aus dem privaten Netz heraus nutzbar sind. Damit eine Firewall effektiv arbeiten kann, muss entsprechend der gesamte Datenverkehr zwischen dem privaten Netz und dem Internet über diese Station laufen. Die Firewall untersucht alle Pakete und lässt nur die unverdächtigen passieren.

Dabei muss die Firewall ihrerseits immun gegen Eindringlinge sein. Was würde eine Firewall nutzen, wenn Hacker sie nach Belieben anpassen könnten? Daraus lässt sich eine „Schwäche“ von Firewalls ableiten: Diese Systeme bieten leider keinen Schutz, sobald es einem Angreifer gelungen ist, sie zu überwinden. Daher ist auf die eigene Sicherheit der Firewall ebenso viel Augenmerk zu legen wie auf die Sicherheit des privaten Netzes selbst, die durch die Firewall gewährleistet werden soll.

Eine Firewall ist nicht wie ein Router, ein Bastion-Host oder ein anderer Teil des Netzes. Sie ist lediglich eine logische Komponente, die ein privates Netz vor einem öffentlichen Netz schützt. Ohne eine Firewall wäre jeder Host im privaten Netz den Attacken von außen schutzlos ausgeliefert. Das bedeutet: Die Sicherheit in einem privaten Netz wäre von der Unverwundbarkeit der einzelnen Rechner abhängig und somit nur so gut wie das schwächste Glied im Netz.

5.1.2 Zentraler Sicherheitsknoten

Der Vorteil einer zentralen Firewall ist, dass sie das Sicherheitsmanagement vereinfacht. Damit gilt die von ihr hergestellte Sicherheit für das gesamte Netz und muss nicht für jeden Rechner einzeln definiert werden. Die Überwachung geschieht ebenfalls zentral über die Firewall. So kann sie gegebenenfalls auch einen Alarm auslösen, da Angriffe von außen nur über diese definierte Schnittstelle zwischen den Netzen erfolgen können. Das Erkennen eines Angriffs ist der erste Schritt zur Abwehr des Angreifers.

Als in den letzten Jahren die Internet-Adressen knapp wurden, trat auch in Unternehmen eine Verknappung von IP-Adressen ein. Eine Internet-Firewall ist in diesem Zusammenhang die geeignete Stelle zur Installation eines Network Address Translators (NAT), der die Adressenknappheit lindern kann. Und schließlich eignen sich Firewalls auch, um den gesamten Datenverkehr von und zum Internet zu überwachen. Hier kann ein Netzwerkadministrator auch Schwachstellen und Flaschenhälse erkennen.

5.1.3 Nachteile und Begrenzungen

Eine Firewall kann keine Angriffe abwehren, wenn die Pakete nicht durch sie hindurch geleitet werden. Wenn zum Beispiel eine Einwählverbindung via Modem oder ISDN aus dem geschützten Netzwerk besteht, können interne Benutzer eine direkte PPP-Verbindung zum Internet aufbauen. Benutzer, welche die zusätzliche Authentifizierung am Proxy-Server scheuen, werden schnell diesen Weg nehmen. Durch die Umgehung der Firewall erzeugen sie jedoch ein großes Risiko für eine Backdoor-Attacke.

Firewalls nützen nichts bei Angriffen aus den eigenen Reihen. Sie hindern niemanden daran, sensitive Daten auf eine Diskette zu kopieren und sie außer Haus zu schaffen. Erst recht nicht, wenn diese Person weit reichende Rechte hat oder

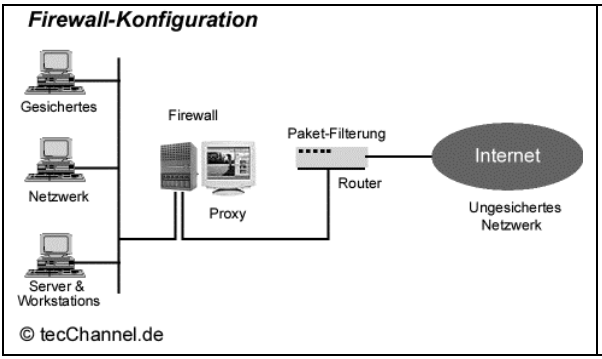
durch Diebstahl an Passwörter gelangt ist. Firewalls schützen auch nicht vor Computerviren oder Trojanern, da sie nicht jedes Datenpaket nach potenziellen Viren durchsuchen können. Auch so genannte Data-driven Attacks können Firewalls nicht verhindern. Dabei handelt es sich um scheinbar harmlose Daten mit verstecktem Code zur Änderung von Sicherheitseinstellungen.

Zudem muss die Firewall leistungsfähig genug sein, um den Datenstrom analysieren zu können. Je schneller die Internet-Anbindung, desto mehr Pakete fließen pro Sekunde in und aus dem Netzwerk. Soll die Firewall zudem noch die Datenströme - also nicht nur die einzelnen Pakete, sondern auch den logischen Datenfluss - überwachen, ist ein umso leistungsfähigeres System erforderlich.

5.1.4 Komponenten einer Firewall

Ein Firewall-System kann aus ein bis drei Komponenten bestehen:

- Paketfilterungs-Router
- Proxy-Server (Application Level Gateway)
- Verbindungs-Gateway (Circuit Level Gateway)



Firewall-Konfiguration: Hier mit Paketfilterungs-Router und Proxy-Server.

Grundsätzlich konkurrieren zwei Firewall-Konzepte: die „passive" Paketfiltertechnologie und die „aktiven" Application Level Gateways. Alle anderen Firewall-Systeme sind Varianten und Weiterentwicklungen dieser beiden Konzepte oder werden damit kombiniert. Dazu gehören etwa das Stateful Packet Filtering, Circuit Level Gateways oder so genannte Hybrid-Firewalls. Diese neueste Variante stellt eine Kombination aus Paketfilter und Application Level Gateway dar.

5.1.5 Paketfilterungs-Router

Ein Paketfilterungs-Router entscheidet bei jedem Datenpaket anhand festgelegter Filterregeln, ob er es weiterleitet oder nicht. Überprüft werden Header-Informationen wie:

- IP-Ursprungsadresse
- IP-Zieladresse
- das eingebettete Protokoll (TCP, UDP, ICMP, oder IP Tunnel)
- TCP/UDP-Absender-Port
- TCP/UDP-Ziel-Port
- ICMP message type
- Eingangsnetzwerkschnittstelle (Ethernet-Karte, Modem, etc.)
- Ausgangsnetzwerkschnittstelle

Falls das Datenpaket die Filter passiert, sorgt der Router für die Weiterleitung des Pakets, andernfalls verwirft er es. Wenn keine Regel greift, verfährt der Paketfilterungs-Router nach den Default-Einstellungen.

Anhand der Filterregeln kann ein Router auch eine reine Service-Filterung durchführen. Auch hier muss der Systemadministrator die Filterregeln vorher definieren. Service-Prozesse benutzen bestimmte Ports (Well Known Ports), wie zum Beispiel FTP den Port 21 oder SMTP den Port 25. Um beispielsweise den SMTP-Service abzublocken, sondert der Router alle Pakete aus, die im Header den Ziel-Port 25 eingetragen haben oder die nicht die Ziel-IP-Adresse eines zugelassenen Hosts besitzen.Einige typische Filterrestriktionen sind:

- Nach außen gehende Telnet-Verbindungen sind nicht erlaubt.
- Telnet-Verbindungen sind nur zu einem bestimmten internen Host erlaubt.
- Nach außen gehende FTP-Verbindungen sind nicht erlaubt.
- Pakete von bestimmten externen Netzwerken sind nicht erlaubt.

5.1.6 Abwehr von Angriffen

Bestimmte Angriffstypen verlangen eine vom Service unabhängige Filterung. Diese ist jedoch schwierig umzusetzen, da die dazu erforderlichen Header-Informationen service-unabhängig sind. Die Konfiguration von Paketfilterungs-Routern kann auch gegen diese Art von Angriffen erfolgen, für die Filterregeln sind jedoch zusätzliche Informationen notwendig. Beispiele für diese Angriffe haben wir für Sie im Folgenden zusammengefasst:

- Source IP Address Spoofing Attacke

Bei einer Spoofing-Attacke fälscht der Angreifer die IP-Absenderadresse eines Datenpakets und verwendet stattdessen die Adresse eines Rechners im internen Netz. Die Firewall kann einen solchen Angriff erkennen, indem sie überprüft, ob ein von außen kommendes Paket eine interne Adresse nutzt. Um den Angriff abzuwehren, sind solche Pakete entsprechend herauszufiltern.

- Source Routing Attacke

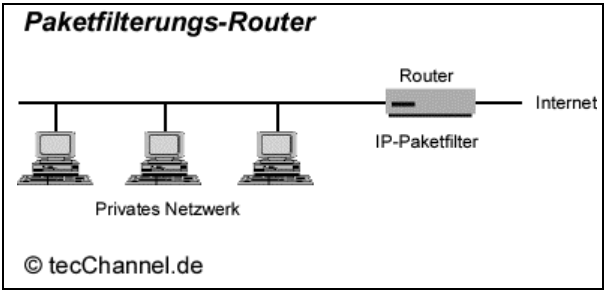
Bei einer Source Routing Attacke gibt der Angreifer die konkrete Route vor, die ein Datenpaket nehmen soll, um Sicherheitsmaßnahmen zu umgehen. Das Verfahren zum Source Routing ist zwar im TCP/IP-Standard vorgesehen, kommt jedoch kaum noch zum Einsatz. Deshalb kann die Firewall die Pakete mit diesem Flag bedenkenlos verwerfen.

- Tiny Fragment Attacke

Bei dieser Angriffsform erzeugt der Hacker extrem kleine Datenpakete, von denen nur das erste den TCP-Header enthält. Das soll den Router veranlassen, nur das erste Fragment zu prüfen und die restlichen ungeprüft durchzulassen. Dies erlaubt dem Hacker, die gewünschten Befehle ins Netz zu schmuggeln. Als Abwehr kann die Firewall alle Pakete verwerfen, bei denen das Feld Fragment-Offset auf 1 gesetzt ist.

5.1.7 Vorteile von Paketfilterungs-Routern

Die Mehrzahl der Firewall-Systeme setzen nur einen Paketfilterungs-Router ein. Außer der Zeit, die für die Planung der Konfiguration des Routers erforderlich ist, entstehen keine weiteren Kosten, denn die Filtersoftware ist Bestandteil der Router-Software. Um den Datenverkehr zwischen privatem und öffentlichem Netz nicht zu stark einzuschränken, sind von Haus aus nur sehr moderate und wenige Filter definiert. Die Paketfilterung ist im Allgemeinen durchlässig für Benutzer und Applikationen. Sie erfordert zudem kein spezielles Training und keine zusätzliche, auf den einzelnen Rechnern installierte Software.



Einfache Sicherung: Hier schützt nur ein Paketfilterungs-Router das Netzwerk.

5.1.8 Nachteile

Doch die Paketfilterung hat auch Nachteile. So ist neben detaillierten Protokollkenntnissen für eine komplexe Filterung auch eine lange Regelliste notwendig. Derartige Listen sind sehr aufwendig und daher schwer zu verwalten. Es ist zudem schwierig, die Filter auf Wirksamkeit zu testen. Auch sinkt der Router-Durchsatz, wenn zu viele Filter definiert sind.

Daneben können Hacker die Firewall durch Tunneln der Pakete überwinden, wobei ein Paket vorübergehend in einem anderen gekapselt wird. Und schließlich: Data-driven-Attacken kann der Router nicht erkennen.

5.1.9 Proxy-Server

Ein Proxy-Server (engl. Proxy: Stellvertreter, Bevollmächtigter), auch Application Level Gateway genannt, erlaubt dem Netzwerk-Administrator die Installation von strengeren Sicherheitsregeln als dies bei einem Paketfilterungs-Router möglich ist. Der Server dient als sicheres Gateway zwischen einem privaten und einem öffentlichen (ungesicherten) Netz. Als Gateway bezeichnet man entweder die Software, die eine Verbindung zwischen zwei Netzwerken herstellt, oder den Computer, auf dem diese Software ausgeführt wird.

Ein Proxy-Server dient nebenbei zur Zwischenspeicherung von Webinhalten und kann als erweiterbare Firewall verwendet werden. Das ermöglicht gleichzeitig Datensicherheit und einen schnelleren Zugriff auf Internet-Inhalte. Der Proxy hat dabei zwei Gesichter: Für den lokalen Client operiert er beim Abruf eines Webdokuments wie ein Webserver. Gegenüber dem entfernten Internet-Server tritt er wie ein Webclient auf. Proxy-Server sprechen aber nicht nur HTTP, sondern beherrschen auch Dienste wie FTP, POP3 oder IRC - allerdings abhängig vom jeweiligen Produkt. Da sie als einziger Knotenpunkt zwischen lokalem und globalem Netz geschaltet sind, schützen sie zudem die lokalen Clients. Denn nur der Proxy-Server ist Angriffen von außen ausgesetzt. Die Clients liegen „unsichtbar“ hinter ihm.

5.1.10 Vorteile eines Proxy-Servers

Das Betriebssystem auf Client-Seite spielt prinzipiell keine Rolle. Nur spezielle Funktionen wie beispielsweise eine automatische Konfiguration der Clients oder das Trennen einer Internet-Verbindung vom Client funktionieren lediglich von Windows-Clients aus.

Daneben lässt sich für jeden Dienst wie FTP oder HTTP ein separater Proxy einrichten. Unerwünschte Dienste filtert der Proxy heraus. Zudem findet kein direkter Paketfluss zwischen internen und externen Rechnern statt.

5.1.11 Bastion-Host

Unter einem Bastion-Host versteht man einen besonders gesicherten Rechner, der wie eine Festung wirken soll. Er schützt die Rechner im privaten Netz vor Angriffen von außen. Wie bei einer Festung gibt es nur einen Ein- und Ausgang, der ständig bewacht ist und bei Bedarf sofort geschlossen werden kann. Die Überwachung des Aus- und Eingangs übernimmt meist ein Router als Paketfilter. Bastion-Hosts sind von ihrer Art her damit die gefährdetsten Rechner in einer Firewall. Auch wenn sie in der Regel mit allen Mitteln geschützt sind, sind sie häufigstes Ziel eines Angriffs, da ein Bastion-Host als einziges System Kontakt zur Außenwelt unterhält.

Die Rechner im privaten Netz sind aus dem Internet nicht direkt erreichbar und dadurch unsichtbar. Andersherum ist auch das Internet nur über den Bastion-Host zugänglich. Deshalb ergibt sich für diesen Rechner die logische Grundhaltung: Je einfacher der Bastion-Host aufgebaut ist, desto leichter ist er zu schützen. Denn jeder auf dem Bastion-Host angebotene Dienst kann Software- oder Konfigurationsfehler enthalten. Bei minimalen Zugriffsrechten sollte der Bastion-Host gerade so viele Dienste anbieten, wie er für die Rolle als Firewall unbedingt braucht.

Bastion-Hosts werden in unterschiedlichen Architekturen installiert, wie etwa als Dual-Homed-Host, in Kombination mit einem Überwachungs-Router.

5.1.12 Vorteile eines Bastion-Hosts

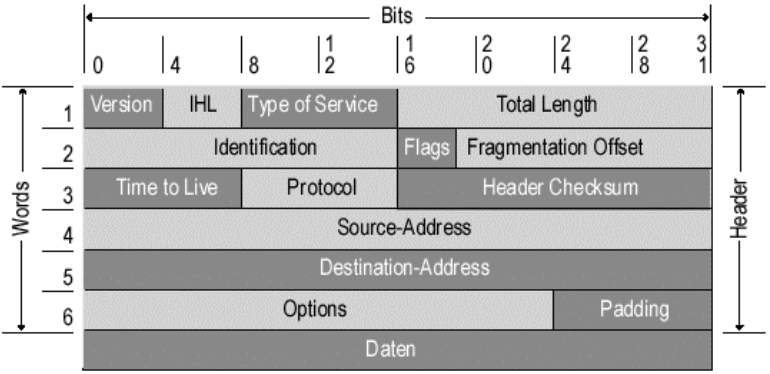
Ein Bastion-Host lässt sich so einrichten, dass Dienste nur über eine Authentifizierung abrufbar sind. Zudem kann der Administrator spezielle Bestandteile dieser Dienste komplett abschalten, etwa den PUT-Befehl für FTP-Server. Die voneinander unabhängigen Proxy-Dienste laufen unter einer unprivilegierten Benutzerkennung in separaten, gesicherten Verzeichnissen, so dass ein Angriff über diese Dienste nur schwer möglich ist. Alle anderen Dienste wie SMTP oder HTTP sind auf diesem Rechner komplett abgeschaltet und stellen somit keine Sicherheitslücke dar. Im Bedarfsfall kann der Administrator auch den kompletten Datenverkehr überwachen, um Angreifer zu erkennen.

5.1.13 Nachteile von Bastion-Hosts

Bei bestimmten Diensten, wie etwa Telnet oder FTP müssen sich die Benutzer zweimal einloggen: Einmal auf dem Proxy des Bastion-Hosts und danach auf dem eigentlichen Server. Zudem muss die Client-Software speziell an den Proxy angepasst werden.

5.1.14 Verbindungs-Gateways

Verbindungs-Gateways (Circuit Level Gateways) sind Proxy-Server mit Zusatzfunktionen. Sie beschränken sich, ähnlich wie Application Level Gateways, nicht nur auf die Kontrolle der IP- und Transportschicht-Header. Stattdessen bauen Sie die Datagramme der Transportschicht aus den IP-Paketen, die unter Umständen fragmentiert sind, zusammen. Wie bei Application Level Gateways gibt es auch hier keine direkten Verbindungen zwischen der Innen- und Außenwelt. Vielmehr findet automatisch eine Adressübersetzung statt. So lässt sich eine Benutzerauthentifizierung erzwingen. Auf der anderen Seite verstehen die Circuit Level Gateways das Anwendungsprotokoll nicht und können deshalb keine Inhaltskontrolle durchführen. Beide Gateway-Varianten verfügen zwar über gemeinsame Merkmale; aber die Fähigkeit, das Anwendungsprotokoll zu verstehen, besitzt nur das Application Level Gateway.



IP-Pakete: Ein Verbindungs-Gateway muss aus den Daten im IP-Header ersehen, welche Pakete zu einem Datenstrom gehören.

Verbindungs-Gateways vertrauen den internen Benutzern. In der Praxis werden Proxy-Server daher für die Verbindungen nach innen benutzt, während man Verbindungs-Gateways für den Datenverkehr von innen nach außen einsetzt.

5.1.15 Hybrid-Firewalls

Hybrid-Firewalls bestehen aus Paketfilter und Application Level Gateway, wobei das Gateway die Filterregeln des Paketfilters dynamisch ändern kann. Als „Stateful Inspection“ bezeichnet man einen Paketfilter „mit Gedächtnis“. Dieser speichert allerdings nur die Informationen aus den Paket-Headern.

Der Vorteil einer Hybrid-Firewall gegenüber einem alleinigen Application Level Gateway liegt in der höheren Performance. Allerdings bedingt dies auch einen gewissen Sicherheitsverlust. Der Grund liegt darin, dass bei den meisten Protokollen der Proxy keinerlei Kontrolle mehr über die Verbindung besitzt, nachdem er den Paketfilter geöffnet hat. Deshalb muss ein Angreifer den Proxy nur eine Zeit lang in Sicherheit wiegen, um anschließend durch den (für ihn geöffneten) Paketfilter freies Spiel zu haben.

Grundlage des Paketfilters mit Stateful Inspection ist die so genannte „Stateful Inspection Engine". Diese analysiert die Datenpakete während der Übertragung auf Netzwerkebene. Im gleichen Arbeitsgang erstellt die Engine dynamische Zustandstabellen, welche die Betrachtung mehrerer Pakete erlauben. Die Korrelationen zwischen zusammengehörenden ein- und ausgehenden Paketen ermöglichen ausgefeilte Analysen.

5.1.16 Hochsicherheits-Firewalls

Hochsicherheits-Firewalls können aus einem Firewall-Subnetz mit zwei Paketfilterungs-Routern und einem Proxy (Bastion-Host) bestehen. Ein solches Firewall-System sichert auf der Netzwerk- und Applikationsebene durch die Definition einer „entmilitarisierten Zone" (Englisch: demilitarized zone, kurz DMZ). Dabei befinden sich Bastion-Host, Informationsserver, Modem-Pools und andere Server im DMZ-Netz. Das DMZ-Netz agiert so als kleines isoliertes Netzwerk zwischen dem privaten Netz und dem Internet.

Dabei ist das DMZ so konfiguriert, dass Zugriffe aus dem privaten Netz und dem Internet nur auf Server im DMZ erfolgen können. Direkter Verkehr durch das DMZ-Netz hindurch ist nicht möglich - egal in welcher Richtung.

Bei den hereinkommenden Datenpaketen schützt der äußere Router gegen Standard-Angriffe wie IP-Address-Spoofing oder Routing-Attacken und überwacht gleichzeitig den Zugriff auf das DMZ-Netz. Dadurch können externe Rechner nur auf den Bastion-Host und eventuell den Information-Server zugreifen.

Durch den internen Router wird eine zweite Verteidigungslinie aufgebaut. Dieses Gerät überwacht den Zugriff vom DMZ zum privaten Netz, indem es nur Pakete akzeptiert, die vom Bastion-Host kommen. Damit kommen nur Benutzer in das interne Netz, die sich vorher am Bastion-Host authentifiziert haben.

5.1.17 Fazit

Wer sein Firmennetzwerk an das Internet anschließt, geht ein nicht unerhebliches Risiko ein. Da aber kaum noch eine Firma ohne Internet-Anschluss auskommt, gehört eine Firewall zum Pflichtprogramm. Die Paranoia lässt sich beliebig weit treiben, man muss nur genügend Zeit und Geld investieren.

Jede Firewall - egal welcher Art - ist allerdings nur so gut wie ihre Konfiguration und die Absicherung des Hosts, auf dem die Firewall läuft. Wer einfach das

Softwarepaket aufspielt oder einen fertigen Firewall-Rechner in sein Netz hängt und sich damit sicher wähnt, handelt fahrlässig. Deshalb ist es oftmals besser, sich an ein auf Netzwerkabsicherung spezialisiertes Unternehmen zu wenden.

tecCHANNEL-Links zum Thema	Webcode
Zehn Personal Firewalls im Test	a405
Linux als Firewall	a695
Desktop-Firewall mit Linux 2.4	a751
Linux Firewall mit ipchains	a704
Hackerangriffe unter Linux entdecken	a715
VPN: Daten sicher übers Internet	a306
So funktioniert TCP/IP	a209
Safer Surfen	a395
Die Netzwächter	a600

5.2 Linux als Firewall

Linux bietet von Haus aus alle notwendigen Komponenten für eine Paketfilter-Firewall. Bevor Sie jedoch eine Firewall aufsetzen können, sollten Sie das Linux-System selbst sichern. Wir zeigen wir Ihnen von Grund auf, wie Sie eine dedizierte Firewall mit Linux aufbauen.

Die Beispiele beziehen sich auf den bewährten Kernel 2.2.x. Auf Kernel 2.4.x geht diese Reihe bewusst nur im Ausblick ein. Bedenken Sie, dass die 2.4er noch recht jung sind und noch die eine oder andere Kinderkrankheit haben können. Viele der Netzwerkfunktionen wurden dort komplett neu implementiert. Gerade für eine Firewall sollten Sie jedoch bewährte Komponenten einsetzen. Da die 2.2-Reihe des Kernels noch weitergepflegt wird, sollten Sie hier für die Übergangszeit auch noch weitere Updates erhalten - wichtig für die Sicherheit Ihres Systems.

Anhand der im Kernel 2.2.x zur Verfügung stehenden Paketfilter zeigt diese Artikelreihe, wie Sie die Firewall selbst und die Server in einer DMZ bzw. Clients im Intranet schützen.

5.2.1 Hard- und Softwareauswahl

Die Auswahl der Hardware für eine Firewall ist relativ unkritisch. Allerdings hängt die Dimensionierung von CPU, Hauptspeicher und Festplattenplatz vom Einsatzzweck ab. Eine Firewall, die nur Port-Filterung durchführt, ist wesentlich anspruchsloser als eine, die zusätzlich einen Webproxy beinhaltet. Natürlich braucht eine Firewall mindestens zwei Schnittstellen, über die Netzwerkpakete ausgetauscht werden können. Hierfür ist die Auswahl groß, denn (fast) alles, das Linux als Netzwerkschnittstelle unterstützt, lässt sich verwenden. Also beispielsweise Ethernet- oder Token-Ring-Netzwerkkarten für das lokale Netzwerk und ISDN-Karten beziehungsweise Modems für WAN-Anbindungen.

Softwareseitig sollten Sie die Firewall möglichst dediziert aufbauen. Es sollen also nur die allernotwendigsten Dienste aktiv sein. Als Grundlage für diesen Artikel diente ein Red Hat Linux 6.2. Dieses System ist zwar nicht mehr das jüngste, dafür aber bekanntermaßen stabil. Das Installieren aller Updates ist hier allerdings obligatorisch. Sie stehen auf dem FTP-Server von Red Hat oder auf diversen Spiegelservern zur Verfügung. Das sollte aber nicht davor abschrecken, diese Version zu benutzen, denn vergleichbaren anderen Distributionen geht es diesbezüglich nicht besser. Red Hat bietet auch, falls Sicherheitslöcher ans Tageslicht kommen, relativ schnell neue Pakete zum Download an und veröffentlicht die entsprechenden Informationen auf einer speziellen Webseite.

5.2.2 Installation von Updates

Die Installation von Updates erfolgt in zwei Schritten. Zuerst speichern Sie die Pakete lokal, zum Beispiel mit Hilfe von *rsync*:

```
# Basisverzeichnis (Achtung, der Pfad auf dem RSYNC-Server
kann sich unter Umständen ändern)
# Remote-Verzeichnis
BASEREMOTE="ftp.tuwien.ac.at::gds/linux/ Red-
Hat.com/dist/linux"
# Lokales Verzeichnis
BASELOCAL="/home/install/ftp.redhat.com"
# 1. rsync-Befehl
rsync --delete -rv --size-only
$BASEREMOTE/updates/6.2/en/os/i386/   $BASELO-
CAL/updates/6.2/i386/
# 2. rsync-Befehl
rsync --delete -rv --size-only
$BASEREMOTE/updates/6.2/en/os/noarch/ $BASELO-
CAL/updates/6.2/noarch/
```

Anschließend führen Sie das Update der Pakete (ohne Kernel) durch:

```
find $BASELOCAL/updates/6.2/ ! -name ´kernel*.rpm´ -name
´*.i386.rpm´ -o -name ´*.noarch.rpm´ | while read paket; do
echo $paket; rpm -Fhv $paket; done
```

Wenn Sie den Kernel updaten wollen, installieren Sie zunächst das RPM. Bei SCSI-Systemen ist die initiale RAM-Disk unbedingt notwendig. Passen Sie den Eintrag in */etc/lilo.conf* an und aktivieren die Änderungen mit */sbin/lilo*. Weitere Informationen zum Kernel-Upgrade finden Sie in den How-tos.

5.2.3 Installation des Basis-Systems

Die Prämisse bei einer Firewall lautet immer: „Was nicht gebraucht wird, wird auch nicht installiert“. Darunter fallen zum Beispiel das X-Window-System und der C-Compiler. Hierauf muss natürlich schon zu Beginn geachtet werden. Das spätere Nachinstallieren von fehlenden Paketen ist einfacher als das Deinstallieren von überflüssigen. Bei Red Hat Linux 6.2 wird dazu entweder der Typ „Server“ oder „benutzerdefiniert“ ausgewählt. Der zweite Typ ist jedoch eher für Experten gedacht, da hier die einzelnen Pakete separat zu selektieren sind.

```
  PID TTY      STAT   TIME COMMAND
    1 ?        S      0:05 init [3]
    2 ?        SW     0:00 [kflushd]
    3 ?        SW     0:00 [kupdate]
    4 ?        SW     0:04 [kswapd]
 1058 ?        S      0:00 syslogd -m 0
 1067 ?        S      0:01 klogd
 1277 ?        S      0:00 gpm -t ps/2
 1377 ?        S      0:00 xfs -droppriv -daemon -port -1
 2342 tty1     S      0:00 /sbin/mingetty --noclear tty1
 2344 tty2     S      0:00 /sbin/mingetty tty2
 1463 tty3     S      0:00 /sbin/mingetty tty3
 1464 tty4     S      0:00 /sbin/mingetty tty4
 1465 tty5     S      0:00 /sbin/mingetty tty5
 1466 tty6     S      0:00 /sbin/mingetty tty6
 1466 tty6     S      0:00 /sbin/mingetty tty6
 1661 pts/2    S      0:00 login -- root
 1662 pts/2    S      0:00 -bash
 2345 pts/2    R      0:00 ps -ax
```

Minimal: Ein Beispiel für ein System, auf dem nur noch die notwendigsten Services laufen, zeigt dieses Listing des Befehls „ps –ax".

Ausgehend von diesem Minimalsystem sollten nach erfolgter Installation nur wenige lokale Dienste und überhaupt keine Netzwerkdienste aktiv sein.

5.2.4 Deaktivieren unnötiger Dienste

Sehr einfach lassen sich Dienste unter Red Hat durch das Kommando

```
chkconfig dienst off
```

deaktivieren. *dienst* ist hierbei der zu deaktivierende Service, wie zum Beispiel *httpd* für den Apache Webserver.

Zu den Diensten, die Sie unbedingt deaktivieren sollten, gehören insbesondere der unsichere Super-Daemon *inetd* (und damit alle Unterdienste, für die er zuständig ist, wie etwa *finger*, *telnet*, *ftp* et cetera). Auch den *portmapper*, der für RPC (etwa für NFS) zuständige ist, sollten Sie deaktivieren. Ein solches abgespecktes System ist trotzdem in der Lage, nach einer Netzwerkkonfiguration Pakete weiterzuleiten und damit auch zu filtern.

Falls Sie die Firewall nicht nur direkt an der Konsole administrieren wollen, ist die SSH (Secure Shell) nützlich. Fertige Binärpakete finden Sie unter www.openssh.com/portable.html. Weil telnet die Authentifizierungsdaten im Klartext überträgt, ist dieser Dienst absolut tabu! Vergessen Sie diesen Veteranen der Remote-Administration schnell wieder.

5.2.5 Wichtige Proxies

Zur Basisinstallation gehören gegebenenfalls noch Proxies. Hier eine Auswahl für die wichtigsten Internet-Dienste:

Wichtige Proxydienste	
Dienst	**Applikation**
(Caching) Domain Name System	named www.isc.org/products/BIND/
HTTP, HTTPS, FTP-over-http	Squid www.squid-cache.org
HTTP, inkl. Junk-Filter	Junkbuster www.junkbuster.com
FTP	jftpgw www.mcknight.de/jftpgw/

Weitere Proxies findet man am schnellsten durch eine Suche bei Freshmeat.net. Die Proxies und die eventuell zusätzlichen Pakete zur Auflösung der Abhängigkeiten sind natürlich separat zu installieren und konfigurieren.

Falls die Pakete nicht in der Distribution enthalten sind, empfiehlt sich die Neukompilierung. Beachten Sie hier allerdings, dass auf der Firewall aus Sicherheitsgründen kein Compiler installiert sein darf. Erledigen Sie deshalb diese Schritte auf einem ähnlichen Testsystem. Dann übertragen Sie nur die Binärdateien (vorzugsweise als RPM oder bei Debian als DEB) auf die Firewall mit Diskette, CD-ROM oder SCP (Secure Copy).

Generell sollten Sie versuchen, benötigte Proxy- und Netzwerkdienste auf der Firewall immer fest an IPv4-Adressen zu binden: Somit können Unberechtigte diese bei Fehlkonfiguration nicht nutzen. Wenn der Proxy nicht selbst als Serverdienst laufen kann, so ist unbedingt *xinetd* dem *inetd* vorzuziehen, da hier entsprechende Konfigurationsmöglichkeiten (Option *bind*) bestehen.

5.2.6 Konfiguration und Verbindungen

Bevor Sie den ersten Firewall-Regelsatz erstellen können, muss die grundlegende Netzwerkfunktionalität gewährleistet sein. Dazu gibt es verschiedene einfache Werkzeuge, die in der Linux-Distribution enthalten, jedoch eventuell noch nicht installiert sind. Mit

```
ifconfig
```

überprüfen Sie die Konfiguration der Netzwerkadressen und mit

```
route -n
```

die Routing-Tabelle. Die Option *n* deaktiviert die Zuordnung von IPv4-Adressen oder -Netzwerken zu Namen. Dies vermeidet Verzögerungen bei der Anzeige der Einträge, falls DNS noch nicht eingerichtet oder voll funktionsfähig ist.

```
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address     Foreign Address   State   PID/Program name
tcp        0      0 192.168.1.1:3128  0.0.0.0:*         LISTEN  1344/(squid)
tcp        0      0 0.0.0.0:22        0.0.0.0:*         LISTEN  1133/sshd
tcp        0      0 192.168.1.1:53    0.0.0.0:*         LISTEN  1123/named
tcp        0      0 127.0.0.1:53      0.0.0.0:*         LISTEN  1123/named
udp        0      0 0.0.0.0:1024      0.0.0.0:*                 1123/named
udp        0      0 192.168.1.1:53    0.0.0.0:*                 1123/named
udp        0      0 127.0.0.1:53      0.0.0.0:*                 1123/named
```

Netstat: Die Dienste named, ssh und squid sind bereits aktiv.

Mit dem Tool netstat lassen sich die verschiedensten Informationen über die IPv4-Schnittstelle anzeigen. Der Befehl

```
netstat -n --ip
```

gibt aktuell bestehende Verbindungen aus. Netzwerkdienste, die auf eingehende Verbindungen warten und sich deshalb im so genannten LISTEN-Status befinden, zeigt

```
netstat -nlptu
```

an. Dabei stehen die Option *-l* für Ports im LISTEN-Zustand, *-p* zeigt (meist) die zugehörigen Dienste mit Prozess-ID. *-t* und *-u* spezifizieren jeweils TCP und UDP für die Anzeige.

5.2.7 Weitere Tools zur Netzwerkkontrolle

Um offene oder aktuell benutzte Ports einem Prozess zuzuordnen, nutzen Sie das Werkzeug *lsof* (list open files). Mit dem Befehl

```
lsof -i :53
```

sehen Sie beispielsweise alle Prozesse, die Port 53 benutzen.

Weitere Information liefert auch die Liste aller so genannter Well Known Ports: STD2 (ftp://ftp.isi.edu/in-notes/std/std2.txt) beziehungsweise RFC 1700 (ftp://ftp.isi.edu/in-notes/rfc1700.txt). Zudem gibt es eine Suchmaschine für Ports (http://www.snort.org/ports.html).

Hilfreich für die Fehlersuche ist auch *tcpdump*. Das Tool zeigt alle Pakete an, die an den Schnittstellen eintreffen. Die Paketbeschreibungssprache ist zwar nicht sehr komfortabel, dafür aber ist das Programm über die Kommandozeile und damit auch von der Ferne über SSH oder eine serielle Leitung zu bedienen. Wichtige Schalter sind *-n* zum Deaktivieren der Namensauflösung und *-i interface* zur Auswahl einer speziellen Netzwerkschnittstelle. Das Tool gibt es übrigens auch für Windows (http://netgroup-serv.polito.it/winpcap/), um etwa festzustellen, ob Pakete auch an diesen Clients ankommen. Wer lieber grafische Werkzeuge benutzen will, für den ist ethereal (www.ethereal.com) das Mittel der Wahl. Ethereal ist für Windows und Linux verfügbar.

5.2.8 Grundschutz durch den Linux-Kernel

Neben der im Folgenden beschriebenen Paketfilterung sind im Linux-Kernel Möglichkeiten zum Grundschutz schon eingebaut und sollten auch aktiviert werden. Zu finden sind diese im proc-Dateisystem. Sie lassen sich normalerweise mit dem Befehl *sysctl* kontrollieren. Eine Beschreibung der wichtigsten Parameter von *sysctl* finden Sie auf der nächsten Seite. Vollständige Beschreibungen aller vorhandenen Schalter stehen in den Kernel-Quellen unter *documentation/proc.txt* und *documentation/networking/ip-sysctl.txt* zur Verfügung.

Ein Beispiel für das Lesen einer Variable ist der Befehl

```
sysctl net.ipv4.icmp_destunreach_rate
```

Gesetzt wird ein Wert mit dem Befehl

```
sysctl -w net.ipv4.icmp_destunreach_rate=100
```

Besonders hervorzuheben ist der Eintrag */proc/sys/net/ipv4/ip_local_port_range*. Dieser gibt an, aus welchem Bereich lokale Ports für Verbindungen stammen dürfen.

Der Bereich ist im Kernel (*net/ipv4/tcp_ipv4.c*) standardmäßig auf 1024-4999 festgelegt, lässt sich aber während des Betriebs verändern. Aus Sicherheitsgründen sollten Sie den Bereich unbedingt auf 32768-60999 setzen, denn dieser Portbereich wird sehr selten von einem Dienst statisch benutzt. Zudem stehen damit auch mehr Ports für gleichzeitige Verbindungen zur Verfügung, was für einen lokalen installierten Webcache von Vorteil sein kann. Der Befehl für die Änderung lautet:

```
sysctl -w net.ipv4.ip_local_port_range="32768 60999"
```

Zusammen mit dem für IP-Maskierung festgelegten Portbereich von 61000-65095 lässt sich der Portbereich für eingehende Antwortpakete später in den Paketfilterregeln genau festlegen. Die Ports zur IP-Maskierung sind nur durch Patchen des Quellcodes (include/net/ip_masq) und Neukompilieren des Kernels änderbar.

Ausnahmen sind allerdings die Clients der r-Dienste (rlogin, rcmd, rexec) und ssh, falls das SUID-Bit gesetzt ist oder diese von root benutzt werden. Denn in diesem Fall werden Ports zwischen 512 und 1023 für abgehende Verbindungen benutzt.

Grund dafür ist die Authentifizierungsmethode der r-Dienste auf Vertrauensbasis mit der Annahme, dass Ports zwischen 1 und 1023 nur von „root“ benutzt werden können. Diese Methode sollte jedoch nur in abgeschotteten Netzwerken zum Einsatz kommen, deren teilnehmende Server ganz sicher unter der Kontrolle des Administrators sind.

Auf den folgenden Seiten finden Sie die wichtigsten Grundschutzeinstellungen mit ihrer Wirkung tabellarisch zusammengestellt.

5.2.9 Grundschutz im Detail I

In dieser Tabelle finden Sie die wichtigsten Parameter für das Kommando *sysctl*. Die Tabelle führt Parameter auf, die alle Schnittstellen betreffen.

Global einstellbarer Grundschutz im Linux-Kernel		
Eintrag in "/proc/sys/"	**Wofür**	**Hilft gegen**
net/ipv4/icmp_destunreach_rate net/ipv4/icmp_echoreply_rate net/ipv4/icmp_paramprob_rate net/ipv4/icmp_timeexceed_rate	Limitieren der Rate an auszusendenden speziellen ICMP-Paketen, je höher der Wert, desto niedriger die Rate	DoS-Attacken
net/ipv4/tcp_syncookies	Aktivieren der TCP-Syncookie-Unterstützung	SYN-Flooding
net/ipv4/icmp_echo_ignore_broadcasts	Abschalten von ICMP echo-reply auf Broadcast-Adressen	Scanning
net/ipv4/icmp_echo_ignore_all	Abschalten von ICMP echo-reply generell	Scanning
net/ipv4/icmp_ignore_bogus_error_responses	Protokollieren von fehlerhaften ICMP-Paketen	Attacken
net/ipv4/ip_always_defrag	Defragmentieren von allen eingehenden IPv4-Paketen	Attacken
net/ipv4/ip_forward	Globales Deaktivieren der Weiterleitung (routing) von IPv4-Paketen	unerwünschtes Routing
net/ipv4/ip_local_port_range	Festlegen des Quellportbereichs bei ausgehenden Verbindungen	Filterprobleme beim statischen Portfilter

5.2.10 Grundschutz im Detail II

In dieser Tabelle finden Sie die wichtigsten Parameter für das Kommando *sysctl*, die für jede Schnittstelle im Linux-Kernel dediziert angegeben werden müssen.

Für jede Schnittstelle einzeln einstellbarer Grundschutz		
Eintrag in "/proc/sys/"	**Wofür**	**Hilft gegen**
net/ipv4/conf/*/ accept_source_ route	Verwerfen von IPv4-Paketen mit Source-Route-Option	Attacken
net/ipv4/conf/*/ secure_redirects	Erlauben von ICMP-redirect, die vom Standard-Gateway eintreffen	Attacken
net/ipv4/conf/*/ accept_redirects	Verwerfen von ICMP-redirect-Paketen	Attacken
net/ipv4/conf/*/ log_martians	Protokollieren von Paketen mit unmöglichen IP-Adressen	Scanning
net/ipv4/conf/*/ forwarding	Deaktivieren von IP-Forwarding per Interface	uner-wünschtes Routing
net/ipv4/conf/*/rp_filter	Überprüfen des Absenders anhand der Routing-Tabelle	IP-Spoofing

Und hier noch die lokal benutzten Ports bei ausgehenden Verbindungen:

Lokale Ports bei ausgehenden Verbindungen	
Art der Verbindung	**Quellportbereich**
Firewall-Verbindungen mit r-Clients, oder ssh mit gesetztem SUID-Bit bzw. von root aufgerufen	512-1023
von Firewall ausgehend (Standard)	1024-4999
von Firewall ausgehend (empfohlen, modifiziert via sysctl)	32768-60999
von Firewall maskiert ausgehend	61000-65095

Eingehende Pakete, die andere lokale Zielports ansprechen, werden an der externen Schnittstelle verworfen. Damit erledigt sich das Problem, diverse Ports ab

1024 dediziert zu sperren. Beispiele hierfür wären Webcache (3128, 8080 oder 8000), X11 (6000-6063) und ISDN-Kontrolldienste (6105, 6106).

5.2.11 Kontrolle der Paketweiterleitung

Auch die Schalter für das Routing (*/proc/sys/net/ipv4/*Schnittstelle*/forwarding*) sind bei der Konfiguration einer Firewall zu berücksichtigen. Linux benutzt diese Schalter, um beim Eingang des Pakets zu entscheiden, ob es an eine andere Schnittstelle gehen darf. Dies gilt auch für Pakete, die demaskiert werden müssen. Steht der Wert für eine Schnittstelle auf 0, bleibt das Paket in der Firewall. Damit lässt sich erreichen, dass Pakete von intern nicht nach außen geroutet werden, auch wenn die Firewall das Standard-Gateway ist.

Der aus Kompatibilitätsgründen zu Kernel 2.0.x noch vorhandene globale Schalter */proc/sys/net/ipv4/ip_forward* ist mit Vorsicht zu genießen. Dieser Schalter verändert automatisch alle schnittstellenspezifischen Schalter für das Forwarding. Dies kann eine vorhandene Feinkonfiguration zunichte machen. Die meisten Netzwerkkonfigurations-Scripts benutzen nur diesen globalen Schalter, so dass Sie diese unter Umständen bearbeiten müssen. Zum Schluss sollten Sie den für das Defragmentieren von IPv4-Paketen zuständigen Schalter aktivieren:

```
sysctl -w net.ipv4.ip_always_defrag=1
```

Sonst können eventuell die Filterlisten der Firewall nicht immer greifen. Außerdem verhindert diese Einstellung Angriffe mit fragmentierten Paketen. Für die IP-Maskierung ist dies zudem obligatorisch, da sonst der Mechanismus nicht zuverlässig funktioniert.

5.2.12 Fazit

Mit den vorgestellten Schritten haben Sie das grundsätzliche System aufgebaut und soweit abgesichert, dass der Einsatz als Firewall Sinn macht. In dem folgenden Kapitel stellen wir Ihnen vor, wie Sie Ihre Client-Rechner per Masquerading über die Firewall mit dem Internet verbinden.

tecCHANNEL-Links zum Thema	Webcode
Linux Firewall mit ipchains	a704
Firewall Grundlagen	a682
So funktioniert TCP/IP	a209
TCP/IP Netze mit Linux	a562
IPv6	a189

5.3 Masquerading mit Linux

Firewalls mit Proxies bieten einen guten Schutz vor Angriffen. Für manche Anwendungsbereiche sind sie jedoch eher ein Hemmschuh. IP-Masquerading hilft, diese Probleme zu umgehen.

Nicht alles, was über die Weiten des Internets möglich ist, kann heute über Proxies laufen. Direkte Verbindungen ins Internet sind daher für manche Anwendungen zwingend erforderlich. Meistens werden im Intranet private IPv4-Adressen aus den Bereichen 10.0.0.0/8, 172.16.0.0/12 und 192.168.0.0/16 (nach RFC 1918) beziehungsweise 169.254.0.0/16 (bei DHCP link local) verwendet, da die Zuteilung von offiziellen IPv4-Adressen in größeren Mengen schwierig geworden ist. Mit solchen Absenderadressen können Clients allerdings nicht direkt mit dem Internet in Kontakt treten, da die Antwortpakete den Weg zurück nicht finden.

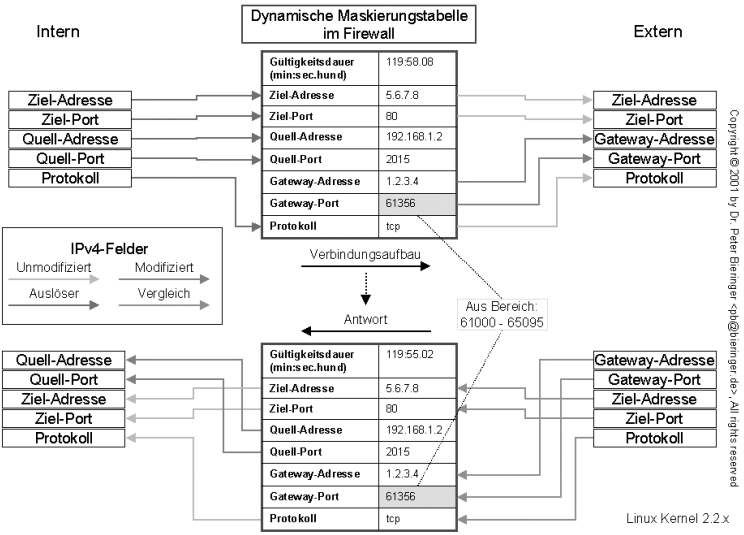
Direkte Verbindungen sind ebenfalls nicht möglich, wenn interne Adressen nach außen nicht bekannt werden sollen oder bereits vergeben sind. Hier muss dann Masquerading in Aktion treten. Im Falle statischer Portfilter-Firewalls erhöht dieser Mechanismus auch ein wenig die Sicherheit, denn Antwortpakete werden nur so lange durchgelassen, wie das Masquerading für diesen Port aktiv ist.

Wer allerdings an den Einsatz von IPsec von extern über die Firewall zu den internen Clients denkt, muss sich um einen Pool offizieller IPv4-Adressen kümmern oder auf IPv6 hoffen. Denn IPsec und Masquerading schließen sich aus. Andernfalls können Sie aber auf Masquerading setzen. Die Firewall aus den vergangenen Beiträgen unserer Reihe lässt sich entsprechend konfigurieren.

Die Listings aus diesem Kapitel haben wir für Sie auf dem tecCHANNEL-Server als Textfiles in einem Tarball unter der Adresse www.tecChannel.de/download/707/masquerading_listings.tar.gz zum Download bereitgestellt. Zum Entpacken speichern Sie die Datei ins Zielverzeichnis und rufen *tar -xzvf masquerading_listings.tar.gz* auf.

5.3.1 Masquerading

Bei Masquerading handelt es sich streng genommen um eine spezielle Art von Adressumsetzung. Diese wird auch SNAT (Source Network Address Translation) genannt. Bei Paketen von intern, die durch die Firewall nach extern gelangen wollen, wird die Original-Quelladresse durch die der Firewall und der ursprüngliche Quellport durch einen neuen ersetzt. Diese Daten hinterlegt die Software in einer Tabelle, damit sie die Antwortpakete entsprechend wieder umsetzen kann.



Aufwendig: Um interne Adressen beim Masquerading zu verstecken, hat die Firewall einiges zu tun.

Jeder Tabelleneintrag erhält zusätzlich eine Gültigkeitsdauer, die bei jeder Benutzung des Eintrags auf den Ausgangswert zurückgesetzt wird. Bei TCP-Verbindungen ist das nicht so sehr von Bedeutung, da bei erfolgreichem Verbindungsabbau der Eintrag meist wieder gelöscht wird. Bei UDP oder ICMP dagegen gibt es keine entsprechenden Antwortsignale. Der Zeitzähler soll verhindern, dass die Tabelle im Laufe der Zeit unnötig wächst und die zur Verfügung stehenden Ports nicht zur Neige gehen.

5.3.2 Konfiguration von Masquerading

Filterregeln, die Pakete zum Maskieren schicken, lassen sich nur in der Forward-Liste einfügen. Deren Aktion lautet hierfür MASQ. Die Parameter für die Gültigkeitsdauer setzen Sie mit dem Befehl

```
ipchains -M -S tcp tcpfin udp
```

Dabei gibt der erste Wert *tcp* den Zeitraum an, wie lange sich das nächste Paket bei einer bestehenden TCP-Verbindung verzögern darf, ehe der Eintrag gelöscht wird. Zu empfehlen ist hier je nach Anwendung 900 bis 7200 Sekunden (15 Minuten bis 2 Stunden). Ein zu niedriger Eintrag macht sich dadurch bemerkbar,

dass bestehende Telnet- oder SSH-Sitzungen ohne oder mit zu groß eingestelltem Keep-Alive-Intervall durch die Firewall plötzlich nicht mehr funktionieren.

tcpfin ist die Zeitspanne nach dem ersten FIN-Paket. Einseitig ist die Verbindung bereits abgebaut, es wird jedoch noch auf das FIN-Paket für den entgegengesetzten TCP-Kanal gewartet. Brauchbare Werte liegen hier zwischen 10 und 180 Sekunden.

udp ist entsprechend für UDP-Verbindungen zuständig. Eine Zeitspanne zwischen 60 und 300 Sekunden ist hier sinnvoll. Für ICMP kann der Wert nur durch Neukompilieren des Kernels (net/ipv4/ip_masq.c) geändert werden. Ansonsten gelten 60 Sekunden als Standard. Aktuell bestehende maskierte Verbindungen lassen sich mit

```
netstat -M -n
```

oder

```
ipchains -M -L
```

auflisten.

5.3.3 Masquerading Proxies

Einfache TCP- oder UDP-Verbindungen sowie ICMP-Pakete sind ohne Aufwand zu maskieren. Für Applikationsprotokolle, die mehr als eine Verbindung aufbauen, müssen so genannte Masquerading-Proxies zum Einsatz kommen, die den Datenfluss im Kontrollkanal mitlesen und bei Bedarf modifizieren.

Das Standardbeispiel ist aktives FTP. Hier schickt der Client dem Server im Kontrollkanal den Port des Datenkanals und seine IPv4-Adresse, zu welcher der FTP-Server eine Verbindung aufbauen kann. Bei Nutzung privater Adressen hinter einer Firewall ist diese Information für den Server wertlos, da er die vom Client erhaltene Adresse nie erreichen wird und der Port auf der Firewall dafür sicher nicht freigeschaltet ist. Hier tritt nun der FTP-Masquerading-Proxy in Aktion, modifiziert on the fly die übertragenen Daten im Kommandokanal und merkt sich die Werte in einer Tabelle.

Auch andere, standardmäßig von Linux unterstützte Protokolle sind ähnlich abzuarbeiten: beispielsweise CUSeeMe, IRC, Quake, RealAudio und VDOLive. Weitere Module stehen im Internet zur Verfügung und können eingebunden werden. Ein guter Ausgangspunkt für weitere Informationen über Masquerading im Linux-Kernel sind wie immer die HOWTOs (www.linuxdoc.org/HOWTO/).

5.3.4 Masquerading von innen nach außen

Wann wird nun Masquerading wirklich benötigt? Zu den klassischen Einsatzszenarien zählen vornehmlich direktes POP, IMAP oder SMTP vom Intranet zum externen, beim Provider im Internet platzierten E-Mail-Server.

Dies ließe sich zwar auch über dedizierte Proxies auf der Firewall abwickeln, bedeutet jedoch einen zusätzlichen Aufwand und ist auch nicht immer möglich. Also müssen entsprechende Pakete freigeschaltet werden. Dazu nimmt ipchains das Paket erst einmal an (ACCEPT), schickt es dann zum Maskieren (MASQ) und versendet es schließlich. Folgende Regelsätze erlauben dies:

```
## POP via Masquerading
ipchains -A input -i eth0 -p tcp -s 192.168.1.0/24 1024: -d
! 192.168.1.0/24 110 -j ACCEPT
ipchains -A forward -i ppp0 -p tcp -s 192.168.1.0/24  1024:
--dport 110 -j MASQ
ipchains -A output -i ppp0 -p tcp --sport 61000:65095 --
dport 110 -j ACCEPT
# Antwortpakete (werden demaskiert, treffen nie auf die for-
ward-Liste)
ipchains -A input -i ppp0 -p tcp --sport 110 --dport
61000:65095 ! -y -j ACCEPT
ipchains -A output -i eth0 -p tcp -s ! 192.168.1.0/24 110  -
d 192.168.1.0/24 1024: ! -y -j ACCEPT
## SMTP via Masquerading
ipchains -A input -i eth0 -p tcp -s 192.168.1.0/24 1024: -d
! 192.168.1.0/24 25 -j ACCEPT
ipchains -A forward -i ppp0 -p tcp -s 192.168.1.0/24  1024:
--dport 25 -j MASQ
ipchains -A output -i ppp0 -p tcp --sport 61000:65095 --
dport 25 -j ACCEPT
# Antwortpakete (werden demaskiert, treffen nie auf die for-
ward-Liste)
ipchains -A input -i ppp0 -p tcp --sport 25 --dport
61000:65095! -y -j ACCEPT
ipchains -A output -i eth0 -p tcp -s ! 192.168.1.0/24 25 -d
192.168.1.0/24 1024: ! -y -j ACCEPT
```

5.3.5 Verbindung zu PGP-Keyservern

Die Verbindung zu PGP-Keyservern muss zumindest bei Verwendung der aktuellen PGP-Software für Windows direkt erfolgen, da kein Proxy konfiguriert werden kann. Als Beispiel hier der Regelsatz für die Benutzung eines Keyservers:

```
## Verbindung zu PGP-Keyservers via Masquerading
ipchains -A input -i eth0 -p tcp -s 192.168.1.0/24 1024: -d
europe.keys.pgp.com 11370 -j ACCEPT
```

```
ipchains -A forward -i ppp0 -p tcp -s 192.168.1.0/24  1024:
-d europe.keys.pgp.com 11370 -j MASQ
ipchains -A output -i ppp0 -p tcp --sport 61000:65095 -d
europe.keys.pgp.com 11370 -j ACCEPT
# Antwortpakete (werden demaskiert, treffen nie auf die for-
ward-Liste)
ipchains -A input -i ppp0 -p tcp -s europe.keys.pgp.com
11370 --dport 61000:65095 ! -y -j ACCEPT
ipchains -A output -i eth0 -p tcp -s europe.keys.pgp.com
11370 -d 192.168.1.0/24 1024: ! -y -j ACCEPT
```

Hier ist allerdings zu bedenken, dass sich die IPv4-Adresse des PGP-Keyservers ändern kann oder gar mehrere Adressen im DNS konfiguriert sind. Abhilfe schafft ein kleines Script, das alle Adressen in Erfahrung bringt und die Regeln für jede mögliche IPv4-Adresse automatisch erstellt:

```
hostname="europe.keys.pgp.com"
LC_ALL=C dig $hostname A IN +pfmin | grep "IN A" | awk ´{
print $5 }´ | while read ipv4addr; do
  ## Verbindung zu PGP-Keyservers via Masquerading
  ipchains -A input -i eth0 -p tcp -s 192.168.1.0/24 1024: -
d $ipv4addr 11370 -j ACCEPT
  ipchains -A forward -i ppp0 -p tcp -s 192.168.1.0/24 1024:
-d $ipv4addr 11370 -j MASQ
  ipchains -A output -i ppp0 -p tcp --sport 61000:65095 -d
$ipv4addr 11370 -j ACCEPT
  # Antwortpakete (werden demaskiert, treffen nie auf die
forward-Liste)
  ipchains -A input -i ppp0 -p tcp -s $ipv4addr 11370 --
dport 61000:65095 ! -y -j ACCEPT
  ipchains -A output -i eth0 -p tcp -s $ipv4addr 11370 -d
192.168.1.0/24  1024: ! -y -j ACCEPT
done
```

Zudem sollten Sie das Script jede Woche einmal neu starten, damit die Regeln nur für aktuelle Adressen gelten, falls der Server netzwerktechnisch gesehen umzieht. Eine Alternative zur direkten Weiterleitung solcher Pakete wäre die Installation eines lokalen PGP-Relays.

5.3.6 Sonderfall T-Online

Wenn Sie den Internet-Zugang über einen Privatkundenzugang von T-Online abwickeln, sollten alle Warnglocken läuten: T-Online überprüft eine Kennung und ein dazugehöriges Passwort für den Zugriff auf T-Online-Server nicht! Die einzige Authentifizierung ist die IPv4-Adresse der ausgehenden Pakete. Und diese wird bei Masquerading vom kompletten Intranet benutzt:

Das hat zur Folge, dass eine Freischaltung für POP (wenn nicht weiter eingeschränkt) zum T-Online-Postfach automatisch für alle internen Clients gilt und diese somit Zugriff auf das Postfach erhalten. Dasselbe gilt für FTP zum T-

Online-Upload-Server (home-up.t-online.de), für private Webseiten und die Administration der T-Online-Einstellungen über Webbrowser. Wer das nicht glaubt, sollte von einem beliebigen internen Client bei T-Online (http://www.t-online.de) nach *Service* und dann bei *Administration für T-Online Kunden* auf *Kennwort ändern* klicken.

Abhilfe schafft für Webzugriffe zum einen eine Blockliste auf dem Webproxy sowie für POP und SMTP, zum anderen das Sperren in entsprechenden Filterlisten. Da die jeweiligen DNS-Namen mehrere IPv4-Adressen als Ergebnis zurückliefern, müssen wie schon beim PGP-Keyserver alle eingetragen werden.

5.3.7 Absicherung bei T-Online

Folgendes Miniscript erledigt das Sperren der IPv4-Adressen von T-Online automatisch, indem es die Resultate einer DNS-Abfrage für das Erstellen von Filterregeln benutzt. Zu beachten ist der Schalter *-I*, welcher sicherstellt, dass die Regeln an den Anfang der jeweiligen Filterliste kommen. Wären sie am Ende der Liste, könnten sie unter Umständen keine Beachtung mehr finden. Sie werden jeweils in den Listen input und output eingefügt, damit zum einen der Client ein ICMP-Paket des Typs *port-unreachable* zurückbekommt. Zum anderen soll ein ungenügend konfigurierter Proxy nicht aus Versehen eine Verbindung gestatten.

```
# Blockiere POP-Zugang zu T-Online
hostname="pop.t-online.de"
LC_ALL=C dig $hostname A IN +pfmin | grep "IN A" | awk ´{
print $5 }´ | while read ipv4addr; do
  ipchains -I input -i eth0 -p tcp -d $ipv4addr 110 -j RE-
JECT -l
  ipchains -I output  -i ppp0 -p tcp -d $ipv4addr 110 -j
REJECT -l
done
```

SMTP zu T-Online lässt sich auf die gleiche Art und Weise sperren:

```
# Blockiere SMTP-Zugang zu T-Online
hostname="smtp.t-online.de"
LC_ALL=C dig $hostname A IN +pfmin | grep "IN A" | awk ´{
print $5 }´ | while read ipv4addr; do
  ipchains -I input -i eth0 -p tcp -d $ipv4addr 25 -j REJECT
-l
  ipchains -I output -i ppp0 -p tcp -d $ipv4addr 25 -j RE-
JECT -l
done
```

Gegen FTP zum T-Online-Upload-Server hilft folgende Regel, die einfach den Kontrollkanal blockiert:

```
# Blockiere FTP-Zugang zu home-up.t-online.de
hostname="home-up.t-online.de"
LC_ALL=C dig $hostname A IN +pfmin | grep "IN A" | awk ´{
print $5 }´ | while read ipv4addr; do
  ipchains -I input  -i eth0 -p tcp -d $ipv4addr 21 -j RE-
JECT -l
  ipchains -I output -i ppp0 -p tcp -d $ipv4addr 21 -j RE-
JECT -l
done
```

Eventuell müssen Sie in Zukunft noch mehr Ports sperren, falls T-Online auch Secure POP (Port 995), IMAP (Port 143) oder Secure IMAP (Port 993) anbietet. Wer ganz sicher gehen will, blockiert gleich von vornherein alle Ports.

5.3.8 Transparente Proxies

Die gezeigten Regeln für transparentes Maskieren reichen manchmal nicht aus. Hin und wieder wollen Administratoren die Erreichbarkeit eines externen Servers per Ping oder ICMP-Traceroute testen. Der tracert.exe bei Windows benutzt automatisch ICMP, unter Linux hilft meist (distributionsabhängig) der Schalter *-I*. Folgende Regeln erlauben dies:

```
# Ping und Traceroute von intern via Masquerading
ipchains -A input -i eth0 -p icmp -s 192.168.1.0/24 -d !
192.168.1.0/24 --icmp-type echo-request -j ACCEPT
ipchains -A forward -i ppp0 -p icmp -s 192.168.1.0/24 --
icmp-type echo-request -j MASQ
ipchains -A output -i ppp0 -p icmp --icmp-type echo-request
-j ACCEPT
# Antwortpakete
ipchains -A input -i ppp0 -p icmp --icmp-type echo-reply -j
ACCEPT
ipchains -A input -i ppp0 -p icmp --icmp-type time-exceeded
-j ACCEPT
ipchains -A input -i ppp0 -p icmp --icmp-type host-
unreachable -j ACCEPT
ipchains -A input -i ppp0 -p icmp --icmp-type network-
unreachable -j ACCEPT
ipchains -A output  -i eth0 -p icmp -d 192.168.1.0/24 --
icmp-type echo-reply -j ACCEPT
ipchains -A output -i eth0 -p icmp -d 192.168.1.0/24 --icmp-
type time-exceeded -j ACCEPT
ipchains -A output -i eth0 -p icmp -d 192.168.1.0/24 --icmp-
type host-unreachable -j ACCEPT
ipchains -A output -i eth0 -p icmp -d 192.168.1.0/24 --icmp-
type network-unreachable -j ACCEPT
```

Den Traceroute via UDP sollten Sie aus Sicherheitsgründen nicht freischalten, da solche Pakete schwerer mit statischen Paketfiltern zu kontrollieren sind.

5.3.9 Transparente Proxies: FTP

Da bei FTP der transparente Proxy wegen des Informationsaustauschs auf dem Kommandokanal etwas komplexer ausfällt, wurde er in das separate Masquerading-Modul *ip_masq_ftp* ausgelagert, das Sie mit diesem Script laden:

```
# Modul für FTP-Masquerading laden
modprobe ip_masq_ftp
```

Für direktes passives FTP finden folgende Regeln Anwendung:

```
# Passives FTP von intern via Masquerading
## FTP:Kommandokanal
ipchains -A input -i eth0 -p tcp -s 192.168.1.0/24 1024: -d
! 192.168.1.0/24 21 -j ACCEPT
ipchains -A forward -i ppp0 -p tcp -s 192.168.1.0/24 1024: -
-dport 21 -j MASQ
ipchains -A output -i ppp0 -p tcp --sport 61000:65095 --
dport 21 -j ACCEPT
# Antwortpakete
ipchains -A input -i ppp0 -p tcp --sport 21 --dport
61000:65095 ! -y -j ACCEPT
ipchains -A output -i eth0 -p tcp -s ! 192.168.1.0/24 21 -d
192.168.1.0/24 1024: ! -y -j ACCEPT
## FTP:passiver Datenkanal
ipchains -A input -i eth0 -p tcp -s 192.168.1.0/24 1024: -d
! 192.168.1.0/24 1024: -j ACCEPT
ipchains -A forward -i ppp0 -p tcp -s 192.168.1.0/24 1024: -
-dport 1024: -j MASQ
ipchains -A output -i ppp0 -p tcp --sport 61000:65095 --
dport 1024: -j ACCEPT
# Antwortpakete
ipchains -A input -i ppp0 -p tcp --sport 1024: --dport
61000:65095 ! -y -j ACCEPT
ipchains -A output -i eth0 -p tcp -s ! 192.168.1.0/24 1024:
-d 192.168.1.0/24 1024: ! -y -j ACCEPT
```

5.3.10 Fazit

Portfilter für andere komplexe Protokolle können Sie analog zu den erläuterten Beispielen erstellen. Hilfestellung geben HOWTOs (http://www.linuxdoc.org), RFCs (http://www.rfc.editor.org) und die Einträge von abgelehnten Paketen im Kernel-Protokoll. Die bislang aufgezeigten Regeln bieten bereits einen guten Grundschutz. Wer Ideen für weitere Filterregeln beispielsweise auch gegen IPv4-Adress-Spoofing oder für das Blocken aller nicht im Internet benutzten Adressen sucht, für den gibt es Anregungen im generischen Firewall-Script Bastille (http://bastille-linux.sourceforge.net).

tecCHANNEL-Links zum Thema	Webcode
Linux Firewall mit ipchains	a704
Firewall Grundlagen	a682
Linux für den Server	a487
So funktioniert TCP/IP	a209
TCP/IP Netze mit Linux	a562
IPv6	a189
Netzwerktechnik	
http://www.tecchannel.de/hardware/netzwerktechnik.html	

Glossar

ADSL

Asymmetric Digital Subscriber Line. Zugangstechnologie, die breitbandige Angebote über die normale Telefonleitung ermöglicht. Die maximale Datenrate im Downstream (von der Vermittlungsstelle zum Teilnehmer) beträgt 8 MBit/s. Im Upstream, der Gegenrichtung, sind bis zu 768 KBit/s erreichbar.

ANSI-C

American National Standards Institute. Institut in den USA zur Normierung von technischen Spezifikationen. Vergleichbar mit DIN in Deutschland.

ASCII

American Standard Code for Information Interchange. Standardisierte Zuordnung von Steuer- und Sonderzeichen sowie Buchstaben und Ziffern auf Bytewerte von 0 (0 x 00) bis 127 (0 x7F). Im übertragenen Sinn auch: unformatierter Text.

Backdoor

Hintertür in Form eines offenen Ports, den sich Trojaner-Programme für das Eindringen in einen Rechner offenhalten.

BIND

Berkeley Internet Name Domain, Implementation der Domain-Name-System-Protokolle. Der vom Internet Software Consortium (ISC) gepflegte BIND-DNS-Server stellt den Standarddienst für die Namensauflösung im Internet dar. BIND umfasst den eigentlichen DNS-Serverdienst, eine Resolver-Bibliothek sowie Werkzeuge zur Wartung und Pflege des Dienstes.

BIOS

Basic Input Output System: Programmroutine, die im ROM eines Rechners untergebracht ist und beim Booten die angeschlossene Hardware überprüft.

Bluetooth

ist eine standardisierte Funktechnik, die mobile und stationäre Geräte mit Sendeleistungen zwischen 1 und 100 mW zu individuellen Piconets verbindet.

Brute-Force

Brute-Force-Angriff: Der Versuch, ein Kryptosystem durch das Ausprobieren aller möglichen Schlüssel-Kombinationen zu brechen. Der Aufwand, durch einen Brute-Force-Angriff ein System zu knacken, stellt eine obere Grenze für die Stärke eines Algorithmus dar. Die Stärke eines Kryptosystems gilt als optimal, wenn es keinen möglichen Angriff auf das System gibt, der weniger aufwendig als ein Brute-Force-Angriff wäre.

CGI

Common Gateway Interface. Definierte Schnittstelle, über die der Webserver externe Programme aufrufen und deren Ergebnisse als Webseiten zur Verfügung stellen kann.

CHAP

Challenge Handshake Authentication. Verfahren zur Authentifizierung bei Einwählverbindungen. Der Server sendet zunächst eine spezielle Codesequenz (Challenge), auf die der Client richtig antworten muss (Handshake).

CIFS

Common Internet File System. Verbesserte Version des Server Message Block (SMB) Protokolls zum Datenaustausch über Inter- oder Intranet. CIFS unterstützt kein Printer-Sharing.

Daemon

Ein Daemon lässt sich mit einem Dienst unter Windows NT vergleichen. Er stellt beispielsweise einen FTP- oder HTTP-Server zur Verfügung.

DHCP

Dynamic Host Configuration Protocol. Bei DHCP bezieht ein Arbeitsrechner seine Konfigura tion des IP-Netzwerks von einem Server.

DMZ

Entmilitarisierte Zone, demilitarized zone: Dabei befinden sich Bastion-Host, Informationsserver, Modem-Pools und andere Server im DMZ-Netz. Das DMZ-Netz agiert so als kleines isoliertes Netzwerk zwischen dem privaten Netz und dem Internet.

DNS

Domain Name System (oder Service). Ein Internet-Dienst, der Domain-Namen wie etwa www. tecChannel.de in die zugehörigen IP-Adressen umsetzt. Weiß ein DNS-Server die IP-Adresse eines Namens nicht, fragt er bei einem anderen Server nach.

Domain

Um Benutzer und Rechner zu einer logischen Struktur zu organisieren, bietet Windows NT das Domänenkonzept. Ein Server pro Domain ist für die zentrale Sicherheitsverwaltung zuständig (Primary Domain Controller).

DoS

Denial of Service. Hacker-Angriff auf einen Rechner, der nur ein Ziel hat: den angegriffenen Computer lahm zu legen, so dass er auf Anfragen nicht mehr reagieren kann.

DSL

Digital Subscriber Line. Die Standleitung ins Internet für kleine Firmen und Privatpersonen. DSL arbeitet mit denselben Kupferkabeln wie analoge Telefone und ISDN-Anschlüsse. Die Übertragungsgeräte (Splitter und DSL-Modem) sind jedoch aufwendiger.

Dual-Homed

Bei dieser Variante befindet sich der Firewall-Rechner, bestehend entweder aus einem Paketfilter-Router oder einem Application Level Gateway, zwischen dem Firmennetz und dem Internet. Dieser Aufbau erleichtert zwar die Implementierung, der potenzielle Angreifer muss aber auch nur eine einzige Hürde überwinden. Ihren Namen hat die Variante von der Notwendigkeit, der Firewall zwei Netzschnittstellen zu geben, so

dass sie in zwei Netzen zu Hause ist (dual-homed).

ECC

Error Correcting Code: Verfahren zum Erkennen und Korrigieren von Bitfehlern.

EIDE

Enhanced Integrated Disc Electronic: Obwohl der Begriff EIDE in den ATA-Spezifikationen nicht explizit auftaucht, hat er sich im Sprachgebrauch eingebürgert. Somit ist EIDE auch kein Standard, sondern vielmehr ein Oberbegriff für eine Vielzahl neuer Features, die in den einzelnen ATA-Spezifikationen verabschiedet wurden. Ursprünglich nannte der Festplattenhersteller Western Digital seine Vision einer schnelleren IDE-Schnittstelle Enhanced IDE und behielt ihn als Marketingnamen bei.

ESC/P

Epson Standard Code for Printers. Von Epson definierte Steuerungssprache für Drucker.

Ethernet

Die am weitesten verbreitete Methode zur Vernetzung in einem LAN. Verbindungen lassen sich über Twisted-Pair-Kabel, Glasfaser oder Koaxial-Kabel herstellen. Es sind Geschwindigkeiten von 10, 100 und 1000 MBit/s möglich.

FAQs

Frequently Asked Questions. Eine Liste von häufig gestellten Fragen mit den dazugehörigen Antworten. FAQs werden von Herstellern und Anwendern zu zahlreichen Hard- und Softwarethemen angeboten.

Finger

Der Finger-Dienst ermöglicht die Abfrage von Benutzerdaten bei entfernten Rechnern.

Firewall

Software zur Sicherung des LAN vor Angriffen aus dem Internet. Eine Firewall kann auf verschiedenen Ebenen arbeiten: Als Paketfilter erlaubt sie nur Zugriffe auf bestimmte lokale IP-Adressen und Ports. Als Proxy-Server agiert sie als Kommunikationsschnittstelle. Der Client im LAN leitet seine Anfragen nicht direkt an den Zielserver, sondern über den Proxy. Mit Stateful Inspection überwacht sie nicht nur den reinen Datenverkehr, sondern auch die Anwendungsebene des OSI-Schichtenmodells.

FQDN

Fully Qualified Domain Name. Der komplette Hostname eines Rechners inklusive sämtlicher Subdomain- und Domain-Namen (Bsp.: search.support.microsoft.com).

FTP

File Transfer Protocol. Spezielles IP-Protokoll zur Dateiübertragung.

GUI

Graphical User Interface. Oberbegriff für grafische Benutzeroberflächen wie Windows oder X-Windows.

Hashes

Hash von Hashing-Algorithmus. Ausgehend von einer Datenmenge wird ein eindeutiger nummerischer Wert erzeugt. Jede Veränderung der Datenbasis führt zu einer Veränderung des Hash-Wertes.

HTML

HyperText Markup Language. Diese Seitenbeschreibungssprache ist die Grundlage jeder Webseite. Der HTML-Standard wird vom W3C verwaltet.

http

HyperText Transport Protocol. Dienst zur Übertragung von Webseiten zwischen Webserver und Browser.

https

Protokollkennzeichner für über SSL gesicherte http-Verbindungen.

Hub

Netzwerkkonzentrator. Schließt mehrere Netzwerkendgeräte zu einem Netz zusammen. Dabei teilen sich die angeschlossenen Geräte die verfügbare Bandbreite.

IANA

Internet Assigned Numbers Authority. Zeichnet für die Administration des Domain Name System (DNS) verantwortlich. Regelt über regionale Registrare wie APNIC, ARIN oder RIPE die Vergabe von IP-Adressen und Top Level Domains (TLDs).

ICMP

Internet Control Message Protocol. TCP/IP-Protokoll, über das Fehler- und Statusmeldungen ausgetauscht werden.

IM

Instant-Messaging-Systeme erlauben den sofortigen Austausch von Informationen oder Daten. Dafür ist eine entsprechende Software nötig, beispielsweise ICQ oder der Yahoo! Messenger (beide kostenlos verfügbar). Benutzer der Software können sehen, ob ein anderer Nutzer online ist und direkt mit ihm chatten oder ihm Nachrichten oder Dateien schicken. Ermöglicht wird dies über die Anmeldung an einem zentralen Server. Dieser zeigt daraufhin anderen (festzulegenden) Anwendern an, ob Anwender X online oder offline ist.

IMAP

Internet Mail Access Protocol. Standardprotokoll zur Zustellung von E-Mails. E-Mail-Clients wie Outlook, Netscape Messenger und Eudora verwenden das Protokoll zur Kommunikation mit einem E-Mail-Server. Im Gegensatz zu POP3 bleiben bei IMAP4 die Nachrichten standardmäßig auf dem Server.

Init-Script

Über Init-Scripts startet Linux beim Systemstart abhängig vom Runlevel die benötigten Dienste. Die Scripts sind mit der autoexec.bat von DOS vergleichbar, jedoch deutlich flexibler.

IP

Internet-Protokoll. Bestandteil der TCP/IP-Suite. IP sendet die Daten in Paketen (auch Datagramme genannt) an die Empfängeradresse. Es kümmert sich jedoch nicht darum, ob die Daten wirklich ankommen. Dafür ist der TCP-Bestandteil zuständig. Derzeit ist IP Version 4 im Einsatz. IP Version 6 (auch IP Next Generation) ist bereits in Arbeit.

IP-Adressen

Ein TCP-Port dient als Kommunikationskanal für den Zugriff auf einen Internet-Rechner über das TCP/IP-Protokoll, ähnlich den Nebenstellen eines Telefonanschlusses. Jedes TCP/IP-Programm verwendet einen TCP-Port für die Kommunikation mit anderen Rechnern.

IP-Maskierung

Network Address Translation. NAT ist ein Verfahren zur Abschottung des LAN gegenüber dem Internet. Dabei wird zum Internet hin immer nur eine Adresse gemeldet, unabhängig von der tatsächlichen IP-Adresse im LAN. Der NAT-Router übernimmt dabei die Verteilung der IP-Pakete zu den richtigen Empfängern.

IPX

Internetwork Packet Exchange. Von Novell für Netware entwickeltes Netzwerkprotokoll zur Übertragung von Daten. Inzwischen verwendet jedoch auch Netware TCP/IP.

IRC

Internet Relay Chat. Chatsystem zur Echtzeitkommunikation.

IrDA

The Infrared Data Association. Standard zur Datenübertragung per Infrarot. In der Version 1.0 mit Geschwindigkeiten bis zu 115 KBit/s, seit der Version 1.1 (Fast IrDA) mit bis zu 4 MBit/s.

IRQ

Eine per Kontrollleitung gesteuer te Datenflussunterbrechung. Diese gewollte Unterbrechung startet andere Funktionen.

ISDN

Integrated Services Digital Network. Digitales Übermittlungsverfahren, das verschiedene Dienste wie Telefonie oder Datenaustausch im Verbund ermöglicht.

Java

Von Sun entwickelte, plattformunabhängige Programmiersprache. Dies gelingt durch die Java Virtual Machine genannte Umgebung, in der Java-Programme ausgeführt werden.

KDE

K Desktop Environment. Grafische Benutzeroberfläche unter Linux.

LAN

Local Area Network. Netzwerk aus Computern und Geräten an einem Standort.

LDAP

Lightweight Directory Access Protocol. Standardisiertes Netzwerkprotokoll zum Zugriff auf

Verzeichnisdienste, über die sich Ressourcen wie E-Mail-Adressen finden lassen.

LILO

Linux Loader. Standard-Bootmanager unter Linux.

Manpage

„Hilfedatei" unter Unix. Wird normalerweise durch das Kommandozeilentool man und die gewünschte Hilfeseite aufgerufen.

Masquerading / NAT

Network Address Translation. NAT ist ein Verfahren zur Abschottung des LAN gegenüber dem Internet. Dabei wird zum Internet hin immer nur eine Adresse gemeldet, unabhängig von der tatsächlichen IP-Adresse im LAN. Der NAT-Router übernimmt dabei die Verteilung der IP-Pakete zu den richtigen Empfängern.

MBR

Der Master Boot Record ist der erste physikalische Sektor einer Festplatte. In diesen 512 Byte sind der Bootloader und die Partitionstabelle untergebracht.

MD5

Message-Digest-Algorithmus:
Version 5 ist ein Verschlüsselungsalgorithmus, der zur Erzeugung digitaler Signaturen verwendet wird.

Minix

Mini-Unix von Andrew S. Tanenbaum, das er in seinem Buch Operating Systems, Design and Implementation als Studienobjekt für Vorlesungen entwickelt hat.

NAT

Siehe Masquerading / NAT.

NBT

NetBIOS over TCP/IP. Windows nutzt als Netzwerkprotokoll kein natives TCP/IP, sondern transportiert NetBIOS über TCP als Low-Level-Transportprotokoll. Hauptvorteil: Bei TCP/IP handelt es sich im Gegensatz zu NetBIOS/NetBEUI um ein routingfähiges Protokoll. Hauptnachteil: Während NetBIOS Namen zur Rechneridentifikation einsetzt, benutzt TCP/IP dazu Nummern. Dies erfordert einen Zusatzdienst zur Namensauflösung, der beide Varianten aufeinander abbildet (WINS, Windows Internet Naming Service).

NetBEUI

Netzwerkprotokoll zur Kommunikation zwischen Rechnern. Wurde primär in Windows-Netzen eingesetzt. Inzwischen verwendet Windows auch TCP/IP.

NetBIOS

Network Basic Input Output System: Protokoll in DOS- und Windows-Netzwerken. NetBIOS stellt eine Programmierschnittstelle für Applikationen zur Verfügung, die auf Schicht 5 des OSI-Modells arbeiten. NetBIOS kann auf dem nicht routingfähigen Transportprotokoll NetBEUI sowie den routingfähigen Protokollen TCP/IP und IPX/SPX aufsetzen.

NFS

Network Filesystem. Spezielles Dateisystem, das in Unix-Umgebungen den Zugriff auf entfernte

Verzeichnisse und Dateien ermöglicht.

NICs

Network Interface Card. Netzwerkadapterkarte.

NIS

Network Information Services. Früher als Yellow Pages (YP) bezeichnet. Dient der Verteilung von wichtigen Daten (Passwörter, Adressen, Schlüsselcodes) vom Server an den Client. Setzt wie NFS auf RCP auf.

NOS

Network Operating System. Netzwerkbetriebssystem für Server, wie zum Beispiel Novell Netware oder Windows NT Server.

NUMA

Non-Uniform Memory Architecture. Architekturansatz für Multiprozessorsysteme mit hoher Anzahl von Prozessoren. Bei SMP (Symmetrical Multiprocessing) teilen sich alle Prozessoren den Speicher (uniform). Bei steigender Prozessorenanzahl erweist sich dies als nachteilig für die Performance, da häufiger auf nicht lokalen Speicher zugegriffen werden muss. NUMA stellt die Synchronisierung zwischen den CPUs über die Sicherung eines kohärenten Zustands der Prozessor-Caches her (ccNUMA: Cache coherent NUMA). So lassen sich die meisten Speicherzugriffe sehr schnell über das lokale Memory abwickeln.

PAM

Pluggable Authentication Modules. Eine Reihe von Bibliotheken, die Unix-Anwendungen gemeinsame Methoden zur Benutzerauthentifizierung zur Verfügung stellen. Neben der eigentlichen Authentifizierung stellen PAMs auch Methoden zur Behandlung von Konten- und Sitzungsdaten zur Verfügung.

PAP

Password Authentication Protocol. Bei PAP authentifiziert sich ein Einwahl-Client per Benutzernamen und Passwort beim Server.

Peer-to-Peer

Netzwerkkonzept, bei dem die einzelnen Arbeitsrechner direkt miteinander kommunizieren können. Jeder Rechner kann gleichzeitig Server und Client sein.

POP / POP3

Post Office Protocol. Standardprotokoll zur Zustellung von E-Mails. E-Mail-Clients wie Outlook, Netscape Messenger und Eudo ra verwenden das Protokoll zur Kommunikation mit einem E-Mail-Server.

Port

Ein TCP-Port dient als Kommunikationskanal für den Zugriff auf einen Internet-Rechner über das TCP/IP-Protokoll, ähnlich den Nebenstellen eines Telefonanschlusses. Jedes TCP/IP-Programm verwendet einen TCP-Port für die Kommunikation mit anderen Rechnern.

Posix

Portable Operating System Interface. Unter der Schirmherrschaft der IEEE entwickelter Interface-Standard für Betriebssysteme. Er bietet Programmierern eine Abstraktionsebene, auf deren Basis sie portable Programme entwickeln können. POSIX lehnt sich eng an Unix-Gepflogenheiten an, kann jedoch auch von Nicht-Unix-OS implementiert werden.

POST

Power-On Self Test. Einer der vier Funktionsbereiche des PC-BIOS. Umfasst eine Reihe von Diagnose- und Initialisierungsprogrammen, die sofort nach dem Einschalten des Rechners abgearbeitet werden.

PPP

Point-to-Point-Protocol. Gebräuchlichstes Einwahlverfahren für den Zugriff auf entfernte Netze, wie etwa das Internet.

PPPoE

Point-to-Point-Protocol over Ethernet. Spezielles Protokoll, das Punkt-zu-Punkt-Verbindungen über das Ethernet ermöglicht.

Proxy

Meist als Kurzform für Proxy-Cache verwendet. Dabei handelt es sich um eine Komponente des Proxy-Servers einer Firewall. Der Cache speichert beispielsweise Internet-Seiten lokal zwischen, so dass sie beim nächsten Abruf nicht vom Internet-Server geholt werden müssen, sondern schneller und kostengünstiger aus dem lokalen Cache.

RISC

Reduced Instruction Set Computer. Prozessor, der einen relativ kleinen Befehlssatz mit Kommandos fester Länge besitzt. Durch die feste Länge und die kleine Anzahl werden die einzelnen Befehle besonders schnell ausgeführt.

Root

(Engl.: Wurzel) Wurzel des Verzeichnisbaums auf Unix-Systemen; auch: Benutzername für den Systemadministrator unter Unix.

Roster

(Engl.: Liste) Roster bezeichnet eine Liste von Jabber-Usern, mit denen Sie kommunizieren wollen, vergleichbar mit der Buddy-Liste im AIM. Im Jabber-Client ist anhand der Roster-Liste zu sehen, wer gerade online ist. Klickt man auf einen Namen der Liste, kann man sofort mit diesem in Verbindung treten.

Router

Router vermitteln die Daten zwischen zwei oder mehreren Subnetzen, die beispielsweise durch Weitverkehrsleitungen wie ISDN verbunden sind. Auch ein Einsatz im LAN ist möglich, um die Datensicherheit zu erhöhen.

RPC

Remote Procedure Call. Programmierschnittstelle, mit der Funktionen auf entfernten Rechnern ausgeführt werden können. RPC wurde von Sun für die „Open Network Computing"-Architektur entwickelt. Eine weitere RPC-Architektur ist Microsofts DCOM.

RPM

Spezielles Dateiformat, über das der Red Hat Package Manager installierte Programme und Zusatzbibliotheken für Linux verwaltet.

Samba

Linux-Paket, das auf dem Linux-Rechner das Server-Message-Block-Protokoll bereitstellt. SMB ist die Grundlage für Windows-Netzwerke.

SCSI

Small Computer System Interface. Allgemeine Bezeichnung für SCSI-1 bis -3 und CCS (Common Command Set). SCSI ist ein Bus (Kanal) vorwiegend zum Anschluss von Peripheriegeräten an Rechner/Server.

SMB

Server Message Block. Netzwerkprotokoll auf Schicht 6 und 7 des OSI-Modells. Es bietet Mechanismen zur Freigabe von Dateien, Druckern und Kommunikations-Schnittstellen wie seriellen Ports. Es definiert auch abstrakte Kommunikation über named pipes und mail slots. Zur Datenübertragung ist ein Transportprotokoll wie TCP/IP oder NetBEUI notwendig.

SNMP

Simple Network Management Protocol. Netzwerkprotokoll, das die Verwaltung von Netzwerkgeräten definiert.

SSH

Secure Shell. Ein von der Firma SSH Communications Security entwickeltes Programm, das eine sichere Kommunikation und Authentifizierung ermöglicht. Dazu verschlüsselt SSH den kompletten Login-Prozess einschließlich der Passwortübermittlung. SSH steht unter anderem für Windows und Unix zur Verfügung.

SSL

Secure Sockets Layer. Von Netscape eingeführtes Protokoll zur Übermittlung von privaten Informationen. Verwendet ein Public-Key-Verfahren für die Verschlüsselung.

Switches

Intelligenter Netzwerk-Konzentrator. Statt wie ein Hub alle Datenpakete auf alle Ports zu schicken, merkt sich der Switch anhand der MAC-Adresse der Netzwerkkarte, welche Rechner an welchem Port zu finden sind. Anhand der Adresseinträge leitet der Switch das Datenpaket nur an den Port weiter, an dem der Zielrechner hängt. Im Optimalfall ist jeder Rechner an einen eigenen Switchport angeschlossen und hat damit die volle Bandbreite des Netzwerks zur Verfügung.

TCP

Transmission Control Protocol. Verbindungsorientiertes Transportprotokoll aus der TCP/IP-Suite. Umfasst Verbindungsauf- und -abbau, Reihenfolgegarantie, Verlustsicherung, Flusskontrolle und anderes mehr.

Telnet

Telnet ermöglicht den Zugriff auf die Kommandoebene eines entfernten Rechners.

Token-Ring

Inzwischen kaum noch benutztes Verfahren zur Vernetzung im LAN nach IEEE802.5. Die Geschwindigkeit kann 4 oder 16 MBit/s betragen. Als Medium dient Twisted Pair.

UDP

Das User Datagram Protocol erlaubt das Versenden von Datagrammen zwischen zwei Systemen. Das Protokoll garantiert weder die Zustellungen, noch die korrekte Reihenfolge der zugestellten Datenpakete.

URL

Uniform Resource Locator. Eindeutiger Bezeichner für den Zugriff auf webbasierte Inhalte wie beispielsweise HTML-Dokumente. Beispiel: http://www.tecchannel.de

USB

Universal Serial Bus, einfacher Bus für externe Peripherie mit pyramidenförmiger Topologie. In der Spezifikation 1.1 sind zwei Modi vorgesehen. Der schnelle Modus bietet 12 MBit/s, der langsamere 1,5 MBit/s. Seit 2001 ist die Erweiterung USB 2.0 auf dem Markt. Diese arbeitet mit einer Datenübertragungsrate von 480 MBit/s.

WAN

Wide Area Network. Computernetzwerk, das über Telefon-, Funk- oder andere Weitverkehrsverbindungen kommuniziert. Das größte WAN ist das Internet.

Webproxy

Software zur Sicherung des LAN vor Angriffen aus dem Internet. Eine Firewall kann auf verschiedenen Ebenen arbeiten: Als Paketfilter erlaubt sie nur Zugriffe auf bestimmte lokale IP-Adressen und Ports. Als Proxy-Server agiert sie als Kommunikationsschnittstelle. Der Client im LAN leitet seine Anfragen nicht direkt an den Zielserver, sondern über den Proxy. Mit Stateful Inspection überwacht sie nicht nur den reinen Datenverkehr, sondern auch die Anwendungsebene des OSI-Schichtenmodells.

X11

Aktuelle Version des X-Window-Systems (X11 Release 6 oder kurz X11R6).

XML

Die eXtensible Markup Language ist eine Meta-Sprache zur Beschreibung strukturierter Daten. XML ist über Tags erweiterbar und auf vielen verschiedenen Plattformen darstellbar.

X-Windows

Grafische Oberfläche für Unix-Systeme.

Index

tecCHANNEL-Preview

Vorschau auf das tecCHANNEL-Magazin im Juli 2002

Reports: tecCHANNEL berichtet von den Highlights der diesjährigen „Computex“, Lebensdauerproblemen bei E-IDE-Festplattten sowie der neuen Nomenklatur für die Angaben von Kapazität und Speichergrößen.

Specials: Zwei große Specials bilden den Schwerpunkt der nächsten Ausgabe. Im Grafikkarten-Special testen wir aktuelle High-End-Grafikkarten und haben über 50 Modelle auf ihre Signalqualität überprüft.

Das zweite Special widmet sich dem Thema Netzwerke. High-Speed-WLAN-Netze und WLAN-Sicherheit, GBit-Ethernet und die Funktion von Verzeichnisdiensten werden ausführlich behandelt.

Tests: Zahlreiche Mainboards sind bereits ab Werk übertaktet und laufen außerhalb der Spezifikationen. tecCHANNEL hat über 50 Mainboards auf Mogeleien überprüft.

Daneben zeigt unser aktueller Test von Virenscannern, mit welchen Produkten Sie Arbeitsplatz-PCs sicher machen.

Praxis: In unserer Praxis-Rubrik zeigen wir, wie Sie mehr aus Ihrem DSL-Anschluss herausholen, ein VPN-Netzwerk unter Linux aufsetzen sowie Microsoft- Office-Dokumente signieren und von heiklen, firmeninternen Informationen befreien.

Know-How: PNG soll sich als leistungsfähiges und universelles Bildformat im Internet etablieren. Wir stellen Ihnen die Technologie und das Verfahren detailliert vor.

tecCHANNEL-Magazin - ab 1. Juli am Kiosk oder jetzt keine Ausgabe mehr verpassen und online das Abo bestellen unter:

www.tecChannel.de/tecshop