Debian Sicherheitshandbuch

Javier Fernández-Sanguino Peña <jfs@debian.org>

Debian Sicherheitshandbuch

von Javier Fernández-Sanguino Peña

Zusammenfassung

Dieses Dokument handelt von der Sicherheit im Debian-Projekt und im Betriebssystem Debian. Es beginnt mit dem Prozess, eine Standardinstallation der Debian GNU/Linux-Distribution abzusichern und zu härten. Es behandelt auch die normalen Aufgaben, um eine sichere Netzwerkumgebung mit Debian GNU/Linux zu schaffen, und liefert zusätzliche Informationen über die verfügbaren Sicherheitswerkzeuge. Es befasst sich auch damit, wie die Sicherheit in Debian vom Sicherheits- und Auditteam gewährleistet wird.

Inhaltsverzeichnis

Liste der Beispiele

Kapitel 1. Einleitung

Eines der schwierigsten Dinge beim Schreiben über Sicherheit besteht darin, dass jeder Fall einzigartig ist. Sie müssen zwei Dinge beachten: Die Gefahrenlage und das Bedürfnis auf Sicherheit bei Ihnen, Ihres Rechners oder Ihres Netzwerkes. So unterscheiden sich zum Beispiel die Sicherheitsbedürfnisse eines Heimanwenders grundlegend von den Sicherheitsbedürfnissen des Netzwerkes einer Bank. Während die Hauptgefahr eines Heimanwenders von »Script-Kiddies« ausgeht, muss sich das Netzwerk einer Bank wegen direkter Angriffe Sorgen machen. Zusätzlich muss eine Bank die Daten ihrer Kunden mit mathematischer Präzision beschützen. Um es kurz zu machen: Jeder Benutzer muss selbst zwischen Benutzerfreundlichkeit und Sicherheit/Paranoia abwägen.

Beachten Sie bitte, dass diese Anleitung nur Software-Themen behandelt. Die beste Software der Welt kann Sie nicht schützen, wenn jemand direkten Zugang zu Ihrem Rechner hat. Sie können ihn unter Ihren Schreibtisch stellen oder Sie können ihn in einen starken Bunker mit einer ganzen Armee davor stellen. Trotzdem kann der Rechner unter Ihrem Schreibtisch weitaus sicherer sein – von der Software-Seite aus gesehen – als der eingebunkerte, wenn Ihr Schreibtisch-Rechner richtig konfiguriert und die Software des eingebunkerten Rechners voller Sicherheitslöcher ist. Sie müssen beide Möglichkeiten betrachten.

Dieses Dokument gibt Ihnen lediglich einen kleinen Überblick, was Sie tun können, um die Sicherheit Ihres Debian GNU/Linux Systems zu erhöhen. Wenn Sie bereits andere Dokumente über Sicherheit unter Linux gelesen haben, werden Sie feststellen, dass es einige Überschneidungen gibt. Dieses Dokument soll aber auch nicht die allumfassende Informationsquelle sein, es versucht nur, die gleichen Informationen so aufzubereiten, dass sie gut zu einem Debian GNU/Linux System passen. Unterschiedliche Distributionen erledigen manche Dinge auf unterschiedliche Weise (zum Beispiel den Aufruf von Daemons); hier finden Sie Material, das zu Debians Prozeduren und Werkzeugen passt.

Autoren

Der aktuelle Betreuer dieses Dokuments ist Javier Fernández-Sanguino Peña. Falls Sie Kommentare, Ergänzungen oder Vorschläge haben, schicken Sie ihm diese bitte. Sie werden dann in künftigen Ausgaben dieses Handbuchs berücksichtigt werden. Bei Fehlern in dieser Übersetzung wenden Sie sich bitte an den aktuellen deutschen Übersetzer mailto:sbrandmair@gmx.net oder (wenn dieser nicht erreichbar ist) an die mailto:debian-l10n-german@lists.debian.org (d.Ü.).

Dieses Handbuch wurde als *HOWTO* von mailto:ar@rhwd.de ins Leben gerufen. Nachdem es im Internet veröffentlicht wurde, gliederte es mailto:jfs@debian.org in das http://www.debian.org/doc ein. Zahlreiche Menschen haben etwas zu diesem Handbuch beigesteuert (alle Beiträge sind im Changelog aufgeführt), aber die folgenden haben gesonderte Erwähnung verdient, da sie bedeutende Beiträge geleistet haben (ganze Abschnitte, Kapitel oder Anhänge):

- Stefano Canepa
- Era Eriksson
- Carlo Perassi
- Alexandre Ratti
- Jaime Robles
- Yotam Rubin
- Frederic Schutz
- Pedro Zorzenon Neto

- Oohara Yuuma
- Davor Ocelic

Wo Sie diese Anleitung bekommen (und verfügbare Formate)

You can download or view the latest version of the Securing Debian Manual from the Debian Documentation Project [https://www.debian.org/doc/user-manuals#securing]. If you are reading a copy from another site, please check the primary copy in case it provides new information. If you are reading a translation, please review the version the translation refers to to the latest version available. If you find that the version is behind please consider using the original copy or review the to see what has changed.

If you want a full copy of the manual you can either download the text version [https://www.debian.org/doc/manuals/securing-debian-manual/securing-debian-manual.en.txt] or the PDF version [https://www.debian.org/doc/manuals/securing-debian-manual/securing-debian-manual.en.pdf] from the Debian Documentation Project's site. These versions might be more useful if you intend to copy the document over to a portable device for offline reading or you want to print it out. Be forewarned, the manual is over two hundred pages long and some of the code fragments, due to the formatting tools used, are not wrapped in the PDF version and might be printed incomplete.

Das Dokument ist auch in den Formaten Text, HTML und PDF im Paket http://packages.debian.org/harden-doc enthalten. Beachten Sie allerdings, dass das Paket nicht genauso aktuell sein muss wie das Dokument, das Sie auf der Debian-Seite finden (Sie können sich aber immer eine aktuelle Version aus dem Quellpaket bauen).

Dieses Dokument gehört zu den Dokumenten, die vom https://www.debian.org/doc/ddp bereit gestellt werden. Sie können die an dem Dokument vorgenommenen Änderungen mit einem Webbrowser über https://salsa.debian.org/ddp-team/securing-debian-manual verfolgen. Sie können auch den Code mit Git mit folgendem Befehl herunterladen:

```
$ git clone https://salsa.debian.org/ddp-team/securing-debian-manual.git
```

Organisatorisches / Feedback

Nun kommt der offizielle Teil. Derzeit sind die meisten Teile dieser Anleitung noch von mir (Alexander Reelsen) geschrieben, aber meiner Meinung nach sollte dies nicht so bleiben. Ich wuchs mit freier Software auf und lebe mit ihr, sie ist ein Teil meiner alltäglichen Arbeit und ich denke, auch von Ihrer. Ich ermutige jedermann, mir Feedback, Tipps für Ergänzungen oder andere Vorschläge, die Sie haben, zuzuschicken.

Wenn Sie glauben, dass Sie einen bestimmten Abschnitt oder Absatz besser pflegen können, dann schreiben Sie dem Dokumenten-Betreuer und Sie dürfen es gerne erledigen. Insbesondere, wenn Sie einen Abschnitt finden, der mit »FIXME« markiert wurde – was bedeutet, dass die Autoren noch nicht die Zeit hatten oder sich noch Wissen über das Thema aneignen müssen – schicken Sie ihnen sofort eine E-Mail.

Der Titel dieser Anleitung macht es sehr deutlich, dass es wichtig ist, dass sie auf dem neusten Stand gehalten wird. Auch Sie können Ihren Teil dazu beitragen. Bitte unterstützen Sie uns.

Vorwissen

Die Installation von Debian GNU/Linux ist nicht sehr schwer und Sie sollten in der Lage gewesen sein, es zu installieren. Wenn Ihnen andere Linux-Distributionen, Unixe oder grundsätzliche Sicherheitskonzepte

ein wenig vertraut sind, wird es Ihnen leichter fallen, diese Anleitung zu verstehen, da nicht auf jedes einzelne Detail eingegangen werden kann (oder dies wäre ein Buch geworden und keine Anleitung). Wenn Sie jedoch mit diesen Dingen noch nicht so vertraut sind, sollten Sie vielleicht einen Blick in „Vorwissen“ für tiefer gehende Informationen werfen.

Dinge, die noch geschrieben werden müssen (FIXME/TODO)

Dieses Kapitel beschreibt die Dinge, welche in diesem Handbuch noch verbessert werden müssen. Einige Abschnitte beinhalten *FIXME*- oder *TODO*-Markierungen, in denen beschrieben wird, welche Dinge fehlen (oder welche Aufgaben erledigt werden müssen). Der Zweck dieses Kapitels ist es, die Dinge, die zukünftig in dieses Handbuch aufgenommen werden könnten, und die Verbesserungen, die durchgeführt werden müssen (oder bei denen es interessant wäre, sie einzufügen) zu beschreiben.

Wenn Sie glauben, dass Sie Hilfe leisten könnten, den auf dieser Liste aufgeführten Punkten (oder solchen im Text) abzuhelfen, setzen Sie sich mit dem Hauptautor (siehe „Autoren“) in Verbindung.

- Dieses Dokument muss noch auf Grundlage der aktuellen Debian-Veröffentlichung aktualisiert werden. Die Standardkonfiguration einiger Pakete muss angepasst werden, da sie verändert wurden, seitdem dieses Dokument geschrieben wurde.

- Expand the incident response information, maybe add some ideas derived from Red Hat's Security Guide's chapter on incident response [https://web.archive.org/web/20100412191348/http://www.redhat.com/docs/manuals/linux/RHL-9-Manual/security-guide/ch-response.html].

- Write about remote monitoring tools (to check for system availability) such as monit, daemontools and mon. See Sysamin Guide [https://web.archive.org/web/20100110040204/http://linuxdevcenter.com/pub/a/linux/2002/05/09/sysadminguide.html].

- Prüfen, ob ein Abschnitt zu schreiben ist, wie man auf Debian basierende Netzgeräte erstellt (mit Informationen etwa zum Basissystem, equivs und FAI)

- Check if this site [https://web.archive.org/web/20040731082209/http://www.giac.org/practical/gsec/Chris_Koutras_GSEC.pdf] has relevant info not yet covered here.

- Add information on how to set up a laptop with Debian, look here [https://web.archive.org/web/20040725013857/http://www.giac.org/practical/gcux/Stephanie_Thomas_GCUX.pdf].

- Informationen, wie man unter Debian GNU/Linux eine Firewall aufsetzt, hinzufügen. Der Firewalls betreffende Abschnitt orientiert sich derzeit an Einzelplatz-Systemen (die keine anderen Systeme schützen müssen); auch auf das Testen der Installation eingehen.

- Hinzufügen, wie man eine Proxy-Firewall unter Debian GNU/Linux aufsetzt, unter der Angabe, welche Pakete Proxy-Dienste anbieten (zum Beispiel xfwp, ftp-proxy, redir, smtpd, dnrd, jftpgw, oops, pdnsd, perdition, transproxy, tsocks). Sollte zu der Anleitung mit weiteren Informationen verweisen. Erwähnenswert ist, dass zorp jetzt Teil von Debian ist und eine Proxy-Firewall *ist* (und auch der Programmautor Debian-Pakete zur Verfügung stellt).

- Informationen über die Service-Konfiguration mit file-rc

- Alle Referenzen und URLs prüfen und die nicht mehr verfügbaren aktualisieren oder entfernen

- Informationen über möglichen Ersatz (unter Debian) für häufig eingesetzte Server, die bei eingeschränktem Funktionsumfang nützlich sind, hinzufügen. Beispiele:

  - lokaler Lpr mit Cups (Paket)?
  - Lrp in der Ferne mit Lpr
  - bind mit dnrd/maradns
  - apache mit dhttpd/thttpd/wn (tux?)
  - exim/sendmail mit ssmtpd/smtpd/postfix
  - squid mit tinyproxy
  - ftpd mit oftpd/vsftp
  - …
- Mehr Informationen über sicherheitsrelevante Patches des Kernels unter Debian einschließlich der oben aufgeführten und insbesondere, wie man diese Patches unter einem Debian-System benutzt
  - Erkennung von Eindringlingen (Linux Intrusion Detection kernel-patch-2.4-lids)
  - Linux Trustees (im Paket trustees)
  - NSA Enhanced Linux [http://wiki.debian.org/SELinux]
  - linux-patch-openswan
  - …
- Details, wie man unnötige Netzwerkdienste deaktiviert (abgesehen von **inetd**), dies ist teilweise Teil des Härtungsprozesses, könnte aber etwas ausgeweitet werden
- Informationen über Passwort-Rotation, was sehr nah mit Sicherheitsrichtlinien (Policies) zusammenhängt
- Sicherheitsrichtlinie und die Aufklärung der Benutzer über die Sicherheitsrichtlinie
- Mehr über tcpwrapper und Wrapper im Allgemeinen?
- `hosts.equiv` und andere wichtige Sicherheitslöcher
- Probleme bei der Dateifreigabe wie z.B. mit Samba und NFS?
- suidmanager/dpkg-statoverrides
- lpr und lprng
- Abschalten der GNOME-IP-Dinge
- Talk about pam_chroot (see http://lists.debian.org/debian-security/2002/05/msg00011.html) and its usefulness to limit users. Introduce information related to https://web.archive.org/web/20031204060940/http://www.securityfocus.com/infocus/1575. pdmenu, for example is available in Debian (whereas flash is not).
- Darüber reden, Dienste mit einer chroot-Umgebung zu versehen, mehr Informationen dazu unter http://www.linuxfocus.org/English/January2002/article225.shtml

- Programme erwähnen, die Chroot-Gefängnisse (chroot jails) herstellen. compartment und chrootuid warten noch in Incoming. Einige andere (makejail, jailer) könnten ebenfalls eingeführt werden.
- Mehr Informationen über Software zur Analyse von Protokoll-Dateien (log-Dateien, logs; zum Beispiel logcheck und logcolorise)
- »Fortgeschrittenes« Routing (Traffic-Regelungen sind sicherheitsrelevant)
- Zugang über **ssh** so einschränken, dass man nur bestimmte Befehle ausführen kann
- Benutzung von dpkg-statoverride
- Sichere Möglichkeiten, um mehreren Benutzern den Zugriff auf einen CD-Brenner zu erlauben
- Sichere Wege, um Töne zusammen mit graphischer Darstellung über ein Netzwerk zu leiten (so dass die Töne eines X-Clients über die Sound-Hardware eines X-Servers abgespielt werden)
- Absichern von Web-Browsern
- Aufsetzen von ftp über **ssh**
- Benutzung von verschlüsselten Loopback-Dateisystemen
- encrypting the entire file system.
- Steganographie-Werkzeuge
- Aufsetzen einer PKA für eine Organisation
- Einsatz von LDAP zur Verwaltung der Benutzer. Es gibt ein HOWTO zu ldap+kerberos für Debian auf http://www.bayour.com von Turbo Fredrikson.
- Wie man Informationen mit begrenztem Nutzen wie z.B. `/usr/share/doc` oder `/usr/share/man` auf Produktivsystemen entfernt (jawohl, security by obscurity)
- Mehr Informationen über lcap, die sich auf die README-Datei des Pakets stützen (gut, die Datei ist noch nicht vorhanden, vergleiche http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=169465) und diesen Artikel der LWN: http://lwn.net/1999/1202/kernel.php3
- Add Colin's article on how to setup a chroot environment for a full sid system (https://web.archive.org/web/20030204012846/https://people.debian.org/~walters/chroot.html).
- Informationen darüber hinzufügen, wie man mehrere **snort**-Sensoren in einem System betreibt (prüfe die Fehlerberichte zu snort)
- Informationen hinzufügen, wie man einen Honigtopf (honeypot) einrichtet (honeyd)
- Darstellung der Situation von FreeSwan (verwaist) und OpenSwan. Der Abschnitt über VPN muss überarbeitet werden.
- Einen gesonderten Abschnitt über Datenbanken hinzufügen, ihre Standardwerte und, wie man den Zugriff absichert
- Einen Abschnitt über den Nutzen von virtuellen Servern (wie Xen u.a.) hinzufügen
- Erklären, wie Programme zur Überprüfung der Integrität verwendet werden (AIDE, integrit oder samhain). Die Grundlagen sind einfach und könnten sogar einige Verbesserungen der Konfiguration erklären.

Danksagungen

- Alexander Reelsen schrieb die ursprüngliche Version.
- fügte der Originalversion einiges an Informationen hinzu.
- Robert van der Meulen stellte den Abschnitt über Quota und viele seiner guten Ideen zur Verfügung.
- Ethan Benson korrigierte den PAM-Abschnitt und hatte einige gute Ideen.
- Dariusz Puchalak trug Informationen zu verschiedenen Kapiteln bei.
- Gaby Schilders trug eine nette Genius/Paranoia-Idee bei.
- Era Eriksson gab dem Ganzen an vielen Stellen den sprachlichen Feinschliff und trug zur Checkliste im Anhang bei.
- Philipe Gaspar schrieb die LKM-Informationen.
- Yotam Rubin trug sowohl Korrekturen für viele Tippfehler bei als auch Informationen über die Versionen von Bind und MD5-Passwörter.
- Francois Bayart stellte den Anhang zur Verfügung, in dem beschrieben wird, wie man eine Bridge-Firewall aufsetzt.
- Joey Hess schrieb im http://wiki.debian.org/SecureApt den Abschnitt, der erklärt, wie Secure Apt funktioniert.
- Martin F. Krafft schrieb in seinem Blog etwas darüber, wie die Verifizierung von Fingerabdrücken funktioniert. Dies wurde im Abschnitt über Secure Apt verwendet.
- Francesco Poli sah dieses Dokument umfassend durch und stellte eine große Anzahl von Fehlerberichten und Ausbesserungen von Tippfehlern zur Verfügung, mit denen dieses Dokument verbessert und aktualisiert werden konnte.
- All den Leuten, die Verbesserungen vorschlugen, die (letzten Endes) eingeflossen sind (siehe „Wo Sie diese Anleitung bekommen (und verfügbare Formate)“).
- (Alexander) All den Leuten, die mich ermutigten, dieses HOWTO zu schreiben (das später zu einer ganzen Anleitung wurde).
- Dem ganzen Debian-Projekt

Kapitel 2. Bevor Sie beginnen

Wofür möchten Sie dieses System benutzen?

Das Absichern von Debian ist nicht viel anders als das Absichern von irgendeinem anderen System. Um es richtig zu machen, müssen Sie zunächst entscheiden, was Sie mit Ihrem System machen möchten. Anschließend müssen Sie sich klarmachen, dass Sie die folgenden Schritte sorgfältig ausführen müssen, um ein wirklich sicheres System zu erhalten.

Sie werden feststellen, dass diese Anleitung von der Pike auf geschrieben ist. Sie werden die Informationen zu einer Aufgabe, die Sie vor, während und nach der Debian-Installation ausführen sollten, in der entsprechenden Reihenfolge vorgestellt bekommen. Die einzelnen Aufgaben können wie folgt beschrieben werden:

- Entscheiden Sie, welche Dienste Sie benötigen, und beschränken Sie Ihr System auf diese. Das schließt das Deaktivieren oder Deinstallieren von überflüssigen Diensten und das Installieren von firewall-ähnlichen Filtern oder TCP-Wrappern ein.
- Einschränken der Nutzer- und Zugriffsrechte auf Ihrem System
- Abhärten der angebotenen Dienste, damit der Einfluss auf Ihr System im Falle einer Kompromittierung möglichst gering ist
- Benutzen Sie die passenden Werkzeuge, um sicherzustellen, dass ein unautorisierter Zugriff auf Ihr System entdeckt wird, so dass Sie geeignete Gegenmaßnahmen ergreifen können

Seien Sie wachsam gegenüber generellen Sicherheitsproblemen!

Diese Anleitung geht (normalerweise) nicht im Detail darauf ein, warum bestimmte Sachen als Sicherheitsrisiko betrachtet werden. Es wäre aber sicherlich von Vorteil, wenn Sie mehr Hintergrundwissen von der Sicherheit in Unix im Allgemeinen und von der in Linux im Besonderen haben. Nehmen Sie sich die Zeit, um sicherheitsrelevante Dokumente zu lesen, um Entscheidungen informiert treffen zu können, wenn Sie eine Auswahl treffen müssen. Debian GNU/Linux basiert auf dem Linux-Kernel, so dass viele Informationen über Linux, und sogar über andere Distributionen und allgemeine UNIX-Sicherheit, auch hierauf zutreffen (sogar wenn sich die benutzten Werkzeuge oder die verfügbaren Programme unterscheiden).

Ein paar nützliche Dokumente sind:

- The http://www.tldp.org/HOWTO/Security-HOWTO/ is one of the best references regarding general Linux security.
- Das http://www.tldp.org/HOWTO/Security-Quickstart-HOWTO/ ist auch ein sehr guter Einstieg für unerfahrene Benutzer (sowohl bezüglich Linux als auch bezüglich Sicherheit).
- Der http://seifried.org/lasg/ ist eine komplette Anleitung, die alle Sicherheitsangelegenheiten von Linux behandelt, von Sicherheit im Kernel bis hin zu VPNs. Beachten Sie bitte, dass er seit 2001 nicht mehr aktualisiert wurde, trotzdem sind einige Informationen immer noch sachdienlich. [1]

[1] Irgendwann wurde er von der »Linux Security Knowledge Base« abgelöst. Dieses Dokument wird ebenfalls durch das Paket lakb zur Verfügung gestellt. Jetzt wird der Guide wieder unter dem Namen *Lasg* verbreitet.

- Kurt Seifrieds http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html.
- In http://www.tldp.org/links/p_books.html#securing_linux finden Sie eine Dokumentation ähnlich zu dieser, bezogen auf Red Hat. Manche behandelten Sachen sind nicht distributionsspezifisch, passen also auch auf Debian.
- Another Red Hat related document is https://web.archive.org/web/20050520170309/https://ltp.sourceforge.net/docs/RHEL-EAL3-Configuration-Guide.pdf.
- IntersectAlliance has published some documents that can be used as reference cards on how to harden Linux servers (and their services), the documents are available at https://web.archive.org/web/20030210231943/http://www.intersectalliance.com/projects/index.html.
- For network administrators, a good reference for building a secure network is the https://web.archive.org/web/20030418093551/http://www.linuxsecurity.com/docs/LDP/Securing-Domain-HOWTO/.
- Wenn Sie die Programme, die Sie benutzen möchten (oder die Sie neu schreiben wollen), bezüglich der Sicherheit evaluieren wollen, sollten Sie das http://www.tldp.org/HOWTO/Secure-Programs-HOWTO/ durchlesen (das Originaldokument ist unter http://www.dwheeler.com/secure-programs/ verfügbar. Es beinhaltet Präsentationen und Kommentare des Autors David Wheeler).
- Wenn Sie erwägen, eine Firewall zu installieren, sollten Sie das http://www.tldp.org/HOWTO/Firewall-HOWTO.html und das http://www.tldp.org/HOWTO/IPCHAINS-HOWTO.html (bei Kerneln vor Version 2.4) lesen.
- Finally, a good card to keep handy is the https://web.archive.org/web/20030308013020/http://www.linuxsecurity.com/docs/QuickRefCard.pdf.

Auf jedem Fall gibt es mehr Informationen über die hier behandelten Dienste (NFS, NIS, SMB, ...) in den vielen HOWTOs, die Sie beim http://www.tldp.org/ finden. Manche dieser Dokumente gehen auf die Sicherheitsaspekte von bestimmten Diensten ein. Sie sollten auch hierauf einen Blick werfen.

Die HOWTO-Dokumente des Linux-Dokumentations-Projekts sind unter Debian GNU/Linux durch Installation der Pakete doc-linux-text (englische Text-Version) oder doc-linux-de (HTML-Version) verfügbar. Nach der Installation sind diese Dokumente in den Verzeichnissen `/usr/share/doc/HOWTO/en-txt` bzw. `/usr/share/doc/HOWTO/de-html` vorhanden.

Andere empfohlene Linux-Bücher:

- Maximum Linux Security: A Hacker's Guide to Protecting Your Linux Server and Network. Anonymous. Paperback - 829 pages. Sams Publishing. ISBN: 0672313413. Juli 1999.
- Linux Security von John S. Flowers. New Riders; ISBN: 0735700354. März 1999.
- https://web.archive.org/web/20030202131658/https://www.linux.org/books/ISBN_0072127732.html By Brian Hatch. McGraw-Hill Higher Education. ISBN 0072127732. April, 2001

Andere Bücher (auch über allgemeine Aspekte von Sicherheit unter Unix, nicht nur linuxspezifisch):

- https://web.archive.org/web/20030206231652/http://www.oreilly.com/catalog/puis/ Garfinkel, Simpson, and Spafford, Gene; O'Reilly Associates; ISBN 0-56592-148-8; 1004pp; 1996.
- Firewalls and Internet Security von Cheswick, William R. and Bellovin, Steven M.; Addison-Wesley; 1994; ISBN 0-201-63357-4; 320pp.

Andere nützliche Websites, um sich bezüglich Sicherheit auf dem Laufenden zu halten:

- http://csrc.nist.gov/.
- https://cve.mitre.org/data/refs/refmap/source-BUGTRAQ.html CVE Reference Map for Source BUGTRAQ
- http://www.linuxsecurity.com/. General information regarding Linux security (tools, news...). Most useful is the https://linuxsecurity.com/howtos page.

Wie geht Debian mit der Sicherheit um?

Um einen allgemeinen Überblick über die Sicherheit unter Debian GNU/Linux zu bekommen, sollten Sie sich ansehen, was Debian unternimmt, um ein insgesamt sicheres System bereitzustellen.

- Debians Probleme werden immer öffentlich behandelt, sogar wenn sie die Sicherheit betreffen. Sicherheitsfragen werden öffentlich auf der debian-security-Mailingliste diskutiert. Debian-Sicherheits-Ankündigungen (DSA) werden an öffentliche Mailinglisten (sowohl intern als auch extern) versendet und auf dem öffentlichen Server bekannt gegeben. Wie der http://www.debian.org/social_contract:*Wir werden Probleme nicht verbergen.*.
- Debian verfolgt Sicherheitsangelegenheiten sehr aufmerksam. Das Sicherheits-Team prüft viele sicherheitsrelevante Quellen, die wichtigste davon http://www.securityfocus.com/cgi-bin/vulns.pl, wobei es Pakete mit Sicherheitsproblemen sucht, die ein Teil von Debian sein können.
- Sicherheitsaktualisierungen genießen höchste Priorität. Wenn ein Sicherheitsproblem in einem Debian-Paket entdeckt wird, wird eine Sicherheitsaktualisierung so schnell wie möglich vorbereitet und für den Stable-, Testing- und Unstable-Zweig, einschließlich aller Architekturen, veröffentlicht.
- Alle Informationen über Sicherheit sind an einer zentralen Stelle zu finden: http://security.debian.org/.
- Debian versucht immer, die gesamte Sicherheit seiner Distribution mittels neuer Projekte zu verbessern, beispielsweise durch automatische Paket-Signierungs- und Verifikations-Mechanismen.
- Debian stellt eine Reihe von brauchbaren sicherheitsrelevanten Werkzeugen für die Systemadministration und -überwachung zur Verfügung. Entwickler versuchen, diese Werkzeuge fest mit der Distribution zu verbinden, um sie zu einer besseren Einheit zur Durchsetzung lokaler Sicherheitsrichtlinien zu machen. Diese Werkzeuge schließen Folgendes mit ein: integritätsprüfende Programme, Überwachungswerkzeuge, Werkzeuge zum Abhärten, Werkzeuge für Firewalls, Eindringlings-Erkennungs-Werkzeuge und vieles andere.
- Paketbetreuer sind sich der Sicherheitsprobleme bewusst. Dies führt oft zur »voreingestellt sicheren« Installation von Diensten, die sie manchmal in ihrer normalen Benutzung etwas einschränken können. Dennoch versucht Debian, Sicherheitsaspekte und Einfachheit der Administration abzuwägen, zum Beispiel werden Dienste nicht inaktiv installiert (wie es bei den Betriebssystemen der BSD-Familie üblich ist). Auf jeden Fall sind bedeutende Sicherheitsaspekte, wie zum Beispiel `setuid`-Programme, Teil der http://www.debian.org/doc/debian-policy/.

Dieses Dokument versucht, eine bessere Installation von Computersystemen hinsichtlich der Sicherheit zu erzielen, indem es Informationen über Sicherheit veröffentlicht, die auf Debian zugeschnitten sind, und diese durch andere Dokumente ergänzt, die sicherheitsspezifische Angelegenheiten im Zusammenhang mit Debian behandeln (vergleiche „Vorwissen“).

Kapitel 3. Vor und während der Installation

Setzen Sie ein Passwort im BIOS

Bevor Sie irgendein Betriebssystem auf Ihrem Computer installieren, setzen Sie ein Passwort im BIOS. Nach der Installation (sobald Sie von der Festplatte booten können) sollten Sie zurück ins BIOS gehen und die Boot-Reihenfolge ändern, so dass Sie nicht von Diskette, CD-ROM oder sonstigen Geräten booten können, von denen dies nicht gehen sollte. Andernfalls benötigt ein Cracker nur physischen Zugang und eine Bootdiskette, um Zugriff auf Ihr ganzes System zu bekommen.

Es ist noch besser, wenn das System beim Booten immer ein Passwort verlangt. Dies kann sehr effektiv sein, wenn Sie einen Server laufen lassen, der selten neu gestartet wird. Der Nachteil dieser Vorgehensweise ist, dass das Neustarten einen menschlichen Eingriff benötigt, was zu Problemen führen kann, wenn das System nicht leicht zugänglich ist.

Hinweis: Viele BIOS-Varianten haben bekannte Master-Passwörter und es gibt sogar Programme, um Passwörter aus dem BIOS wieder auszulesen. Folglich können Sie sich nicht auf diese Maßnahme verlassen, um den Zugriff auf das System zu beschränken.

Partitionieren des Systems

Wählen Sie eine intelligente Partitionierung

Was eine sinnvolle Partitionierung ist, hängt davon ab, wie die Maschine benutzt wird. Eine gute Faustregel ist, mit Ihren Partitionen eher großzügig zu sein und die folgenden Faktoren zu berücksichtigen:

- Jeder Verzeichnisbaum, auf den ein Benutzer Schreibzugriff hat (wie zum Beispiel `/home`, `/tmp` und `/var/tmp/`) sollte auf einer separaten Partition liegen. Dies reduziert das Risiko eines DoS (Denial of Service, »Dienstverweigerung«) durch einen Benutzer, indem er Ihren »/«-Einhängepunkt vollschreibt und so das gesamte System unbenutzbar macht. [1] Außerdem verhindert dieses Vorgehen Hardlink-Angriffe.[2]

- Außerdem sollte jeder Verzeichnisbaum, dessen Größe schwanken kann, zum Beispiel `/var` (insbesondere `/var/log`) eine separate Partition bekommen. Auf einem Debian-System sollten Sie der /var-Partition etwas mehr Platz als auf anderen Systemen geben, da heruntergeladene Pakete (der Zwischenspeicher von apt) unter `/var/cache/apt/archives` gespeichert werden.

- Jede Partition, in der Sie Nicht-Distributions-Software installieren wollen, sollte separat sein. Nach dem File-Hierarchy-Standard wären dies `/opt` oder `/usr/local`. Wenn dies separate Partitionen sind, werden sie nicht gelöscht, falls Sie einmal Ihr Debian neu installieren (müssen).

- Rein sicherheitstechnisch ist es sinnvoll, zu versuchen, statische Daten auf eine eigene Partition zu legen und diese dann als nur-lesbar einzuhängen (mounten). Oder noch besser: Legen Sie diese Daten auf einem rein-lesbaren Medium ab. Lesen Sie dazu die Ausführungen weiter unten.

[1] Eigentlich ist das so nicht ganz richtig, da immer etwas Platz für Root reserviert wird, den ein normaler Benutzer nicht belegen kann.

[2] Ein sehr gutes Beispiel dieser Art von Angriff, der das /tmp-Verzeichnis benutzt, ist ausführlich auf http://www.hackinglinuxexposed.com/articles/20031111.html und auf http://www.hackinglinuxexposed.com/articles/20031214.html beschrieben (beachten Sie, dass dieser Vorfall in einem Zusammenhang mit Debian steht). Im Prinzip ist das ein Angriff, bei dem ein lokaler Benutzer eine angreifbare Setuid-Anwendung *versteckt*, indem er einen harten Link zu ihr einrichtet. So kann er wirksam verhindern, dass diese Anwendung vom Systemadministrator aktualisiert (oder entfernt) wird. Dpkg wurde kürzlich verbessert, um das zu verhindern (vergleiche http://bugs.debian.org/225692). Aber andere Setuid-Anwendungen, die nicht vom Paketverwaltungsprogramm gesteuert werden, bleiben ein Risiko, wenn Partitionen nicht richtig eingerichtet werden.

Im Falle eines Mailservers ist es wichtig, eine separate Partition für die Mail-Warteschlange (mail spool) anzulegen. Nicht-Lokale Benutzer können (wissentlich oder unwissentlich) diese Verzeichnisse (`/var/mail` oder `/var/spool/mail`) füllen. Liegt dieses Verzeichnis auf einer separaten Partition, würde dies das System nicht sofort unbenutzbar machen. Anderenfalls (wenn das Verzeichnis auch auf der `/var`-Partition liegt) hat das System ein großes Problem: Protokoll-Einträge (logs) können nicht erstellt werden, Pakete können nicht installiert werden und es könnten sogar ein paar Programme Probleme mit dem Starten haben (wenn sie `/var/run` benutzen).

Außerdem sollten Sie für Partitionen, deren Platzbedarf Sie noch nicht abschätzen können, den Logical-Volume-Manager (lvm-common und die benötigten ausführbaren Programme, entweder lvm10 oder lvm2) installieren. Durch Benutzen von `lvm` können Sie Datenträger-Gruppen erstellen, die über mehrere Festplatten verteilt sind.

Auswahl der passenden Dateisysteme

Während der Partitionierung des Systems müssen Sie sich ebenfalls entscheiden, welche Dateisysteme Sie benutzen möchten. Als Standard-Dateisystem[3] wird während der Installation für Linux-Partitionen `ext3` ausgewählt, das ein »Journaling Dateisystem« ist. Es ist empfehlenswert, immer ein solches Dateisystem zu verwenden, wie zum Beispiel `ext3`, `reiserfs`, `jfs` oder `xfs`. Dadurch verringern Sie Probleme nach einen Absturz des Systems in folgenden Fällen:

- Auf Laptops auf allen Dateisystemen. Auf diese Art reduzieren Sie die Wahrscheinlichkeit eines Datenverlustes, wenn beispielsweise unerwartet Ihr Akku leer wird oder das System aufgrund eines Hardware-Problems (etwa durch die X-Konfiguration, was relativ häufig auftritt) neu gestartet werden muss.

- Auf produktiven Systemen, die große Mengen von Daten speichern (zum Beispiel Mail-Server, FTP-Server, Netzwerk-Dateiserver, ...), ist es empfehlenswert, ein Journaling-Dateisystem auf diesen Partitionen einzusetzen. Wenn das System abstürzt, benötigt der Server so weniger Zeit, um das Dateisystem wieder herzustellen und zu prüfen, und die Wahrscheinlichkeit eines Datenverlustes wird verringert.

Lassen wir mal die Betrachtung der Leistung von Journaling-Dateisystemen beiseite (da dies oft in quasi-religiöse Glaubenskriege ausartet). In der Regel ist es besser, das `ext3`-Dateisystem zu benutzen. Der Grund dafür ist die Abwärtskompatibilität zu `ext2`. So können Sie, wenn es Probleme mit dem Journal gibt, dieses einfach abschalten und haben immer noch ein funktionierendes Dateisystem. Außerdem müssen Sie, wenn Sie das System mal mit einer Boot-Diskette (oder CD-ROM) wiederherstellen müssen, keinen speziellen Kernel benutzen. Wenn es sich um einen 2.4er oder 2.6er Kernel handelt, ist Unterstützung für `ext3` bereits vorhanden. Wenn es sich um einen 2.2er-Kernel handelt, können Sie trotzdem Ihr Dateisystem booten, auch wenn Sie die Journaling-Fähigkeiten einbüßen. Wenn Sie ein anderes Journaling-Dateisystem benutzen, werden Sie feststellen, dass eine Wiederherstellung nicht möglich ist, bis Sie einen 2.4er oder 2.6er Kernel mit den benötigten Modulen haben. Wenn Sie einen 2.2er Kernel auf der Rettungsdiskette verwenden müssen, kann es sich als noch schwerer erweisen, auf `reiserfs` oder `xfs` zuzugreifen.

Auf jeden Fall ist die Datenintegrität unter `ext3` besser, da es auch Datei-Daten protokolliert, während andere Dateisysteme lediglich Meta-Daten protokollieren (siehe auch http://lwn.net/2001/0802/a/ext3-modes.php3).

Beachten Sie aber, dass es auch einige Partitionen gibt, die von einem Journaling-Dateisystem nicht profitieren könnten. Wenn Sie beispielsweise eine eigene Partition für `/tmp/` verwenden, könnte ein übliches `ext2`-Dateisystem besser sein, weil es bei einem Neustart des Systems ohnehin geleert wird.

[3] Seit Debian GNU/Linux 4.0 mit dem Codenamen `Etch`.

Gehen Sie nicht ins Internet, bevor Sie nicht bereit sind

Während der Installation sollten Sie das System nicht sofort mit dem Internet verbinden. Dies hört sich vielleicht komisch an, aber die Installation über das Netzwerk ist eine gängige Methode. Da das System einige Dienste installiert und diese sofort aktiviert werden, könnten Sie Ihr System für Angriffe öffnen, wenn das System mit dem Internet verbunden ist und die Dienste nicht geeignet konfiguriert sind.

Außerdem sollten Sie beachten, dass manche Pakete noch Sicherheitsprobleme haben können, weil das Installationsmedium nicht auf dem aktuellen Stand ist. Dies ist für gewöhnlich dann der Fall, wenn Sie von älteren Medien (wie CD-ROMs) installieren. In diesem Fall könnte Ihr System bereits kompromittiert sein, bevor Sie mit der Installation fertig sind!

Da die Debian-Installation und die Upgrades über das Internet durchgeführt werden können, denken Sie vielleicht, es sei eine gute Idee, dies gleich während der Installation zu nutzen. Wenn das System direkt mit dem Internet verbunden ist (und nicht von einer Firewall oder NAT geschützt wird), ist es besser, das System ohne Internet-Verbindung zu installieren. Benutzen Sie sowohl für die zu installierenden Pakete als auch für die Sicherheitsaktualisierungen eine lokale Quelle (Spiegel). Sie können einen Paket-Spiegel aufsetzen, indem Sie ein anderes System nutzen, dass mit dem Internet verbunden ist und für Debian spezifische Werkzeuge (falls es sich um ein Debian-System handelt) wie apt-move oder apt-proxy oder andere gebräuchliche Werkzeuge zur Erstellung von Spiegeln verwendet. Damit kann das Archiv für das installierte System zur Verfügung gestellt werden. Sollte dies nicht möglich sein, sollten Sie Firewall-Regeln aufsetzen, die den Zugriff auf Ihr System beschränken, während Sie die Aktualisierung durchführen (siehe „Schutz der Sicherheitsaktualisierung durch eine Firewall“).

Setzen Sie ein Passwort für Root

Die wichtigste Grundlage für ein sicheres System ist ein gutes Root-Passwort. Siehe passwd(1) für einige Tipps, wie man gute Passwörter auswählt. Sie können auch automatische Passwort-Generatoren verwenden (siehe „Erstellen von Benutzerpasswörtern“).

Plenty of information on choosing good passwords can be found on the Internet; two that provide a decent summary and rationale are Eric Wolfram's http://wolfram.org/writing/howto/password.html and Walter Belgers' https://web.archive.org/web/20030218000949/http://www.belgers.com/write/pwseceng.txt

Lassen Sie so wenige Dienste wie möglich laufen

Dienste sind Programme wie FTP- und Web-Server. Da sie auf eingehende Verbindungsanfragen, die den Dienst anfordern, *warten* müssen, können sich externe Computer mit Ihrem Computer verbinden. Dienste sind manchmal verwundbar (das heißt, durch einen bestimmten Angriff kompromittierbar) und stellen dadurch ein Sicherheitsrisiko dar.

Sie sollten keine Dienste installieren, die Sie nicht unbedingt auf dem System brauchen. Jeder installierte Dienst könnte neue, vielleicht nicht gerade offensichtliche (oder bekannte) Sicherheitslöcher auf Ihrem Computer öffnen.

Wie Sie vielleicht schon wissen, wird ein Dienst, sobald er installiert wird, auch gleich automatisch aktiviert. Bei einer Standardinstallation ohne weitere installierte Dienste ist die Anzahl der laufenden Dienste ziemlich gering. Und die Anzahl der Dienste, die im Netzwerk angeboten werden, ist noch niedriger. In

einer Standardinstallation von Debian 3.1 werden Sie mit OpenSSH, Exim (abhängig davon, wie Sie ihn konfiguriert haben) und dem RPC-Portmapper als Netzwerkdienste auskommen.[4] Wenn Sie nicht eine Standard-, sondern eine Experten-Installation durchgeführt haben, kann es sein, dass Sie überhaupt keine aktiven Netzwerkdienste haben. Der RPC-Portmapper ist standardmäßig installiert, da er für viele Dienste wie zum Beispiel NFS benötigt wird. Er kann allerdings sehr leicht entfernt werden. Weitere Informationen, wie Sie RPC-Dienste absichern oder abschalten, finden Sie unter „Absichern von RPC-Diensten“.

Wenn Sie einen neuen Netzwerkdienst (Daemon) auf Ihrem Debian GNU/Linux System installieren, kann er auf zwei Arten gestartet werden: durch den Superdaemon **inetd** (d. h. eine Zeile wird zu `/etc/inetd.conf` hinzugefügt) oder durch ein eigenständiges Programm, das sich selbst an die Netzwerkschnittstelle bindet. Eigenständige Programme werden durch `/etc/init.d` gesteuert. Sie werden beim Hochfahren durch den Sys-V-Mechanismus gestartet, der die symbolischen Links in `/etc/rc?.d/*` benutzt. Weitere Informationen dazu finden Sie in `/usr/share/doc/sysvinit/README.runlevels.gz`.

Wenn Sie Dienste installieren möchten, diese aber selten benutzen, entfernen Sie sie mit den update-Befehlen wie **update-inetd** oder **update-rc.d** aus dem Startvorgang. Weitere Informationen, wie Sie Netzwerkdienste abschalten, finden Sie unter „Daemons abschalten“. Wenn Sie das Standardverhalten des Startens von Diensten nach der Installation von ihren Paketen ändern wollen[5], lesen Sie bitte für weiterführende Informationen `/usr/share/doc/sysv-rc/README.policy-rc.d.gz`.

Die Unterstützung von **invoke-rc.d** ist bei Debian nun zwingend. Dies bedeutet, dass Sie seit Debian 4.0 *Etch* eine **policy-rc.d**-Datei anlegen können, die das Starten von Daemons verbietet, bevor Sie sie konfiguriert haben. Zwar sind derartige Skripte noch nicht in Paketen enthalten, sie sind aber ziemlich leicht zu schreiben. Sehen Sie sich auch policyrcd-script-zg2 an.

Daemons abschalten

Das Abschalten eines Daemons ist sehr einfach. Entweder Sie entfernen das Paket, welches das Programm für diesen Dienst anbietet, oder Sie entfernen oder benennen die Startlinks unter `/etc/rc${runlevel}.d/` um. Wenn Sie sie umbenennen, stellen Sie sicher, dass sie nicht mehr mit einem »S« beginnen, damit sie nicht von **/etc/init.d/rc** ausgeführt werden. Entfernen Sie nicht alle verfügbaren Links, denn sonst wird das Paketverwaltungssystem sie bei dem nächsten Upgrade des Pakets wieder herstellen. Gehen Sie also sicher, dass zumindest ein Link übrig bleibt (typischerweise ein »K«-Link, »K« steht für »kill«). Zusätzliche Informationen finden Sie im Abschnitt http://www.debian.org/doc/manuals/reference/ch-system#s-custombootscripts der Debian-Referenz (2. Kapitel - Debian-Grundlagen).

Sie können diese Links manuell entfernen oder Sie benutzen `update-rc.d` (siehe auch update-rc.d(8)). So können Sie zum Beispiel einen Dienst in den Multi-User-Runleveln abschalten:

```
# update-rc.d name stop XX 2 3 4 5 .
```

Wobei *XX* eine Zahl ist, die bestimmt, wann die Stop-Aktion für diesen Dienst ausgeführt wird. Bitte beachten Sie, dass `update-rc.d -f Dienst remove` nicht korrekt arbeiten wird, wenn Sie *nicht* file-rc benutzen, da *alle* Verknüpfungen entfernt werden. Nach einer Neuinstallation oder einem Upgrade dieses Paketes werden diese Verknüpfungen neu angelegt (was Sie vermutlich nicht wollen). Wenn Sie denken, dass dies nicht sehr intuitiv ist, haben Sie wahrscheinlich recht (siehe http://bugs.debian.org/67095). Aus der Handbuchseite:

[4] Die Zahl war bei Debian 3.0 und davor nicht so niedrig, da einige **inetd**-Dienste standardmäßig aktiviert waren. Außerdem war in Debian 2.2 der NFS-Server wie auch der Telnet-Server Bestandteil der Standardinstallation.
[5] Dies ist z.B. wünschenswert, wenn Sie eine Chroot-Umgebung zur Entwicklung einrichten.

```
If any files /etc/rcrunlevel.d/[SK]??name already exist then
update-rc.d does nothing.  This is so that the system administrator
can rearrange the links, provided that they leave at least one
link remaining, without having their configuration overwritten.
```

Wenn Sie file-rc benutzen, werden alle Informationen über das Starten von Diensten durch eine gemeinsame Konfigurationsdatei verarbeitet und sogar nach der Deinstallation von Paketen beibehalten.

Sie können das TUI (Text User Interface, textbasierte Benutzungsoberfläche) des Paketes sysv-rc-conf benutzen, um all diese Änderungen einfach zu erledigen (sysv-rc-conf arbeitet sowohl mit file-rc als auch mit normalen System-V-Runleveln). Es gibt auch vergleichbare GUIs für Desktop-Systeme. Sie können auch die Befehlszeile von sysv-rc-conf verwenden:

```
# sysv-rc-conf foobar off
```

Der Vorteil dieses Werkzeugs ist, dass die rc.d-Links wieder auf den Status zurückgesetzt werden, die sie vor dem Aufruf von »off« hatten, wenn Sie den Dienst wieder aktivieren mit:

```
# sysv-rc-conf foobar on
```

Andere (weniger empfohlene) Methoden zum Abschalten eines Dienstes sind:

- Löschen Sie das Skript `/etc/init.d/service_name` und entfernen Sie die Start-Links mit:

  ```
  # update-rc.d name remove
  ```

- Benennen Sie die Skriptdatei (`/etc/init.d/Dienst`) um (zum Beispiel in `/etc/init.d/OFF.Dienst`). Da das zu Verweisen führt, die kein Ziel mehr haben (dangling symlinks), werden beim Systemstart Fehlermeldungen erzeugt.

- Entfernen Sie das Ausführungsrecht von der Datei `/etc/init.d/Dienst`. Auch das wird beim Booten Fehlermeldungen verursachen.

- Editieren Sie die Datei `/etc/init.d/Dienst`, so dass sich das Skript sofort beendet, sobald es gestartet wird, indem Sie die Zeile **exit 0** am Anfang einfügen oder den `start-stop-daemon`-Abschnitt auskommentieren. Falls Sie dies tun, können Sie das Skript nicht später dazu verwenden, um den Dienst von Hand zu starten.

Jedoch handelt es sich bei allen Dateien unter `/etc/init.d` um Konfigurationsdateien und sollten daher bei einem Upgrade des Pakets nicht überschrieben werden.

Sie können im Gegensatz zu anderen (UNIX-)Betriebssystemen Dienste unter Debian nicht abschalten, indem Sie die Dateien unter `/etc/default/Dienst` modifizieren.

FIXME: Add more information on handling daemons using file-rc.

Abschalten von Inetd oder seinen Diensten

Sie sollten überprüfen, ob Sie heutzutage den **inetd**-Daemon überhaupt brauchen. Inetd war früher eine Möglichkeit, Unzulänglichkeiten des Kernels auszugleichen. Diese sind aber in modernen Linux-Kerneln nicht mehr vorhanden. Gegen **inetd** gibt es die Möglichkeit von Angriffen, die zur Dienstverweigerung führen (Denial of Service), welche die Last des Rechners unglaublich erhöhen. Viele Leute ziehen es vor, einzelne Daemonen zu benutzen, anstatt einen Dienst über **inetd** zu starten. Wenn Sie immer noch einen

inetd-Dienst laufen lassen wollen, wechseln Sie wenigstens zu einem besser zu konfigurierenden Inet-Daemonen wie **xinetd**, **rlinetd** oder **openbsd-inetd**.

Sie sollten alle nicht benötigten Inetd-Dienste auf Ihrem System abschalten, wie zum Beispiel **Echo**, **Chargen**, **Discard**, **Daytime**, **Time**, **Talk**, **Ntalk** und die r-Dienste (**Rsh**, **Rlogin** und **Rcp**), die als SEHR unsicher gelten (benutzen Sie stattdessen **Ssh**).

Sie können Dienste abschalten, indem Sie direkt `/etc/inetd.conf` editieren, aber Debian stellt Ihnen einen besseren Weg zur Verfügung: `update-inetd` (womit die Dienste auf eine Art auskommentiert werden, in der sie leicht wieder aktiviert werden können). Sie können den **Telnet**-Daemon sehr leicht mit dem folgenden Kommando abschalten, so dass die Konfigurationsdateien angepasst und der Daemon neu gestartet wird:

```
/usr/sbin/update-inetd --disable telnet
```

Wenn Sie Dienste starten wollen, aber nur auf bestimmten IP-Adressen Ihres Systems, können Sie auf eine undokumentierte Funktion des inetd zurückgreifen (Austausch des Namens des Dienstes durch dienst@ip). Alternativ können Sie einen Daemon wie **xinetd** benutzen.

Installieren Sie möglichst wenig Software

Debian bietet *sehr viel* Software an. Debian 3.0 (*Woody*) enthält sechs oder sieben (je nach Architektur) CDs mit Software und tausenden Paketen. Debian 3.1 *Sarge* wird mit etwa 13 CD-ROMs ausgeliefert. Bei so viel Software, selbst wenn Sie die Installation auf das Basis-System reduzieren [6], könnten Sie auf Abwege geraten und mehr installieren, als Sie wirklich benötigen.

Da Sie bereits wissen, was Sie mit Ihrem System machen wollen (oder etwa nicht?), sollten Sie nur Software installieren, die Sie wirklich für den Betrieb benötigen. Jedes unnötig installierte Programm könnte von einem Benutzer, der Ihr System kompromittieren will, genutzt werden – oder von einem externen Eindringling, der Shell-Zugriff bekommen hat (oder der Code von außerhalb durch einen fehlerhaften Dienst ausführen kann).

Zum Beispiel kann das Vorhandensein von Hilfsprogrammen für Programmierer (ein C-Compiler) oder Interpretern (wie **Perl** siehe allerdings unten, **Python**, **tcl**, ...) einem Angreifer helfen, das System weiter zu kompromittieren:

- Der Angreifer kann seine Privilegien auf dem System erweitern. Es ist beispielsweise leichter, eine lokale Sicherheitslücke des Systems auszunutzen, wenn man einen Debugger und Compiler zur Verfügung hat, um den eigenen Exploit (ein Programm, das eine Sicherheitslücke ausnutzt) zu kompilieren und zu testen.
- Man könnte dem Angreifer Werkzeuge zur Verfügung stellen, die ihm helfen könnten, das kompromittierte System als *Basis für Angriffe* auf andere Systeme zu benutzen. [7]

Natürlich kann ein Eindringling mit lokalem Shell-Zugriff seine eigenen Programme herunterladen und ausführen. Und sogar die Shell selbst kann benutzt werden, um komplexere Programme zu schreiben. Das

[6] Unter Debian-Woody ist das Basis-System etwa 400-500MB groß. Probieren Sie Folgendes:

```
$ size=0 $ for i in `grep -A 1 -B 1 "^Section: base" /var/lib/dpkg/available | grep -A 2 "^Priority: required" |
```

;

[7] Häufig werden fremde Systeme nur deshalb gehackt, weil sie zu weiteren illegitimen Aktivitäten benutzt werden sollen (DoS-Angriffe, Spam, geheime FTP-Server, DNS-Schweinereien, ...). Der Angreifer möchte meist gar nicht an die vertraulichen Daten auf dem kompromittierten System herankommen.

Entfernen unnötiger Programme wird also nicht helfen, das Problem zu *verhindern*. Jedoch wird es für den Angreifer etwas schwieriger, das System zu kompromittieren (und manchmal wird er in dieser Situation aufgeben und sich ein leichteres Ziel suchen). Wenn Sie also auf einem produktivem System Werkzeuge lassen, die benutzt werden können, um andere Systeme anzugreifen (siehe „Programme zur Fernprüfung der Verwundbarkeit“), müssen Sie davon ausgehen, dass ein Angreifer sie auch benutzen wird.

Beachten Sie bitte, dass eine Standardinstallation von Debian *Sarge* (d.h. eine Installation, bei der nicht individuell Pakete ausgewählt werden) eine Reihe von Paketen zur Softwareentwicklung installiert, die normalerweise nicht benötigt werden. Das liegt daran, dass einige Pakete zur Softwareentwicklung die Priorität *Standard* haben. Wenn Sie keine Software entwickeln, können Sie ohne Bedenken die folgenden Pakete von Ihrem System entfernen, was nebenbei auch etwas Platz schafft:

```
Paket                     Größe
------------------------+--------
gdb                     2,766,822
gcc-3.3                 1,570,284
dpkg-dev                  166,800
libc6-dev               2,531,564
cpp-3.3                 1,391,346
manpages-dev            1,081,408
flex                      257,678
g++                         1,384 (Hinweis: virtuelles Paket)
linux-kernel-headers    1,377,022
bin86                      82,090
cpp                        29,446
gcc                         4,896 (Hinweis: virtuelles Paket)
g++-3.3                 1,778,880
bison                     702,830
make                      366,138
libstdc++5-3.3-dev        774,982
```

Dieses Verhalten wurde in den Veröffentlichungen nach Sarge verändert. Für weitere Informationen sehen Sie sich http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=301273 und http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=301138 an. Wegen eines Fehlers im Installationssystem ist dies nicht geschehen, wenn mit dem Installationssystem von Debian 3.0 *Woody* installiert wird.

Entfernen von Perl

Sie müssen bedenken, dass es nicht gerade einfach ist, **Perl** von einem Debian-System zu entfernen (in der Tat kann es ziemlich schwierig werden), da es von vielen Dienstprogrammen benutzt wird. perl-base hat außerdem *Priority: required* (und das sagt eigentlich schon alles). Es ist aber trotzdem machbar. Allerdings können Sie auf diesem System keine **Perl**-Anwendung mehr laufen lassen. Außerdem müssen Sie auch das Paketverwaltungssystem hereinlegen, damit es weiterhin denkt, dass perl-base installiert ist, auch wenn es das nicht mehr ist. [8]

Welche Dienstprogramme benutzen **Perl**? Sie können es selbst herausfinden:

```
$ for i in /bin/* /sbin/* /usr/bin/* /usr/sbin/*; do [ -f $i ] && {
type=`file $i | grep -il perl`; [ -n "$type" ] && echo $i; }; done
```

Diese Liste schließt die folgenden Dienstprogramme mit der Priorität *required* oder *important* ein:

[8] Sie können (auf einem anderen System) eine Paket-Attrappe mit equivs erstellen.

- `/usr/bin/chkdupexe` aus dem Paket util-linux
- `/usr/bin/replay` aus dem Paket bsdutils
- `/usr/sbin/cleanup-info` aus dem Paket dpkg
- `/usr/sbin/dpkg-divert` aus dem Paket dpkg
- `/usr/sbin/dpkg-statoverride` aus dem Paket dpkg
- `/usr/sbin/install-info` aus dem Paket dpkg
- `/usr/sbin/update-alternatives` aus dem Paket dpkg
- `/usr/sbin/update-rc.d` aus dem Paket sysvinit
- `/usr/bin/grog` aus dem Paket groff-base
- `/usr/sbin/adduser` aus dem Paket adduser
- `/usr/sbin/debconf-show` aus dem Paket debconf
- `/usr/sbin/deluser` aus dem Paket adduser
- `/usr/sbin/dpkg-preconfigure` aus dem Paket debconf
- `/usr/sbin/dpkg-reconfigure` aus dem Paket debconf
- `/usr/sbin/exigrep` aus dem Paket exim
- `/usr/sbin/eximconfig` aus dem Paket exim
- `/usr/sbin/eximstats` aus dem Paket exim
- `/usr/sbin/exim-upgrade-to-r3` aus dem Paket exim
- `/usr/sbin/exiqsumm` aus dem Paket exim
- `/usr/sbin/keytab-lilo` aus dem Paket lilo
- `/usr/sbin/liloconfig` aus dem Paket lilo
- `/usr/sbin/lilo_find_mbr` aus dem Paket lilo
- `/usr/sbin/syslogd-listfiles` aus dem Paket sysklogd
- `/usr/sbin/syslog-facility` aus dem Paket sysklogd
- `/usr/sbin/update-inetd` aus dem Paket netbase

Ohne Perl und solange Sie diese Dienstprogramme nicht in einem Shell-Skript neu schreiben, werden Sie also wahrscheinlich keine Pakete mehr verwalten können (und so kein Upgrade des Systems durchführen können, was *keine gute Idee* ist).

Wenn Sie fest dazu entschlossen sind, Perl aus dem Debian-Basissystem zu entfernen und ein wenig Freizeit haben, schicken Sie uns doch Fehlerberichte zu den aufgezählten Paketen, die (als ein Patch) einen Ersatz dieser Dienstprogramme als Shell-Skript enthalten.

Wenn Sie wissen wollen, welche Debian-Pakete von Perl abhängen, können Sie Folgendes verwenden:

```
$ grep-available -s Package,Priority -F Depends perl
```

oder

```
$ apt-cache rdepends perl
```

Lesen Sie Debians Sicherheits-Mailinglisten

Es ist niemals falsch, einen Blick in die Mailingliste debian-security-announce zu werfen, auf der Anleitungen und Problemlösungen durch das Debian-Sicherheits-Team bekannt gemacht werden, oder sich an mailto:debian-security@lists.debian.org zu beteiligen, wo Sie an Diskussionen zu sicherheitsrelevanten Fragen teilnehmen können.

Um wichtige Warnungen zu Sicherheitsaktualisierungen zu erhalten, senden Sie eine E-Mail an mailto:debian-security-announce-request@lists.debian.org mit dem Wort »subscribe« in der Betreffzeile. Sie können diese moderierte E-Mail-Liste unter http://www.de.debian.org/MailingLists/subscribe auch über das Web abonnieren.

Diese Mailingliste hat ein sehr geringes Aufkommen und, indem Sie sie abonnieren, werden Sie sofort über Sicherheitsaktualisierungen der Debian-Distribution informiert. Dies erlaubt Ihnen sehr schnell, neue Pakete mit Sicherheitsaktualisierungen herunterzuladen, was sehr wichtig ist, um ein sicheres System zu verwalten (siehe „Ausführen von Sicherheitsaktualisierungen“ für weitere Details, wie Sie dies durchführen).

Kapitel 4. Nach der Installation

Wenn das System installiert ist, können Sie es noch weiter absichern, indem Sie einige der in diesem Kapitel beschriebenen Schritte ausführen. Natürlich hängt dies vor allem von Ihrer Einrichtung ab, aber um physischen Zugriff zu verhindern, sollten Sie „Änderen Sie das BIOS (noch einmal)“, „Ein Passwort für LILO oder GRUB einstellen“, „Entfernen des Root-Promptes aus dem Kernel“, „Einschränkung der Anmeldemöglichkeiten an der Konsole“ und „Einschränkung des System-Neustarts von der Konsole aus“ lesen.

Bevor Sie sich mit einem Netzwerk verbinden, insbesondere wenn es sich um ein öffentliches Netzwerk handelt, sollten Sie wenigstens eine Sicherheitsaktualisierung (siehe „Ausführen von Sicherheitsaktualisierungen“) durchführen. Daneben können Sie auch einen Schnappschuss Ihres Systems machen (siehe „Einen Schnappschuss des Systems erstellen“).

Abonnement der Security-Announce-Mailingliste von Debian

Um Informationen zu verfügbaren Sicherheitsaktualisierungen und die Debian-Sicherheits-Ankündigungen (DSA) zu erhalten, sollten Sie Debians Security-Announce-Mailingliste abonnieren. Lesen Sie „Das Sicherheitsteam von Debian“ für weitere Informationen, wie das Sicherheitsteam von Debian arbeitet. Hinweise, wie Sie die Mailinglisten von Debian abonnieren, finden Sie unter http://lists.debian.org.

DSAs werden mit der Signatur des Sicherheitsteams von Debian unterschrieben, die unter http://security.debian.org erhältlich ist.

Sie sollten in Betracht ziehen, auch die http://lists.debian.org/debian-security zu abonnieren. Dort finden allgemeine Diskussionen zu Sicherheitsthemen im Betriebssystem Debian statt. Sie können auf der Liste sowohl mit gleichgesinnten Systemadministratoren als auch mit Entwicklern von Debian und Programmautoren in Kontakt treten. Diese werden Ihre Fragen beantworten und Ihnen Ratschläge geben.

FIXME: Add the key here too?

Ausführen von Sicherheitsaktualisierungen

Sobald neue Sicherheitslöcher in einem Paket entdeckt wurden, reparieren sie Debians Paketbetreuer und Originalautoren im Allgemeinen innerhalb von Tagen oder sogar Stunden. Nachdem das Loch gestopft wurde, werden neue Pakete unter http://security.debian.org bereit gestellt.

Wenn Sie eine Debian-Veröffentlichung installieren, müssen Sie berücksichtigen, dass es seit der Veröffentlichung Sicherheitsaktualisierungen gegeben haben könnte, nachdem entdeckt wurde, dass ein bestimmtes Paket verwundbar ist. Ebenso könnte es Zwischenveröffentlichungen gegeben haben, die diese Paketaktualisierungen enthalten. Für Debian 3.1 *Sarge* gab es vier Zwischenveröffentlichungen.

Während der Installation werden Sicherheitsaktualisierungen für Ihr System eingerichtet, offene Sicherheitsaktualisierungen heruntergeladen und Ihrem System hinzugefügt, sofern Sie sich nicht explizit dagegen entscheiden oder keine Internetverbindung besteht. Die Aktualisierungen werden noch vor dem ersten Systemstart eingespielt, damit das neue System sein Leben so aktuell wie möglich beginnt.

Um Ihr System manuell zu aktualisieren, fügen Sie die folgende Zeile in Ihre `/etc/apt/sources.list` ein. So werden Sie Sicherheitsaktualisierungen automatisch erhalten, wann immer Sie Ihr System aktualisieren. Ersetzen Sie *`[CODENAME]`* mit dem Namen der Veröffentlichung, z.B. mit *squeeze*.

```
deb http://security.debian.org/ [CODENAME]/updates main contrib non-free
```

Hinweis: Falls Sie den *Testing*-Zweig einsetzen, sollten Sie die Sicherheitsspiegel für Testing verwenden. Das wird unter „Sicherheitsunterstützung für den Testing-Zweig“ beschrieben.

Wenn Sie dies erledigt haben, stehen Ihnen zahlreiche Werkzeuge zur Verfügung, mit denen Sie Ihr System aktualisieren können. Wenn Sie ein Desktop-System einsetzen, können Sie eine Anwendung mit dem Namen **Update-notifier** verwenden[1], mit der Sie leicht prüfen können, ob neue Aktualisierungen verfügbar sind. Damit können Sie Ihr System auch über den Desktop auf den neusten Stand bringen (mit **update-manager**). Weitere Informationen finden Sie unter „Überprüfung von Aktualisierungen auf dem Desktop“. Für den Desktop können Sie auch Synaptic (GNOME), Kpackage oder Adept (KDE) einsetzen, die einen größeren Funktionsumfang aufweisen. Wenn Sie auf einem textbasierten Terminal arbeiten, stehen Ihnen Aptitude, Apt und Dselect, wobei letzteres veraltet ist, zur Verfügung:

- Falls Sie die textbasierte Oberfläche von Aptitude verwenden wollen, müssen Sie zunächst *u* (für Update) und dann *g* (für Upgrade) eingeben. Oder Sie führen auf der Befehlszeile Folgendes als Root aus:

```
# aptitude update
# aptitude upgrade
```

- Falls Sie Apt einsetzen möchten, müssen Sie obige Zeilen von **Aptitude** nur mit **apt-get** ersetzen.
- Falls Sie dselect verwenden wollen, müssen Sie zuerst aktualisieren ([U] für Update), dann installieren ([I] für Install) und schließlich die installieren/aktualisierten Pakete konfigurieren ([C] für Configure).

Wenn Sie möchten, können Sie der Datei `/etc/apt/sources.list` die Zeilen mit deb-src hinzufügen. Weitere Details finden Sie unter apt(8).

Sicherheitsaktualisierungen für Bibliotheken

Wenn Sie eine Sicherheitsaktualisierung durchgeführt haben, müssen Sie gegebenenfalls einige Dienste des Systems neu starten. Wenn Sie das nicht tun, könnten Dienste auch nach der Sicherheitsaktualisierung immer noch verwundbar sein. Das liegt daran, dass Daemonen, die schon vor einem Upgrade liefen, immer noch die alten Bibliotheken vor dem Upgrade verwenden könnten.[2] Um herauszufinden, welche Daemonen neu gestartet werden müssen, können Sie das Programm **Checkrestart** (ist im Paket debian-goodies enthalten) oder diesen Einzeiler (als Root) verwenden:[3]

From Debian *Jessie* and up, you can install the needrestart package, which will run automatically after each APT upgrade and prompt you to restart services that are affected by the just-installed updates. In earlier releases, you can run the **checkrestart** program (available in the debian-goodies package) manually after your APT upgrade.

Some packages (like libc6) will do this check in the postinst phase for a limited set of services specially since an upgrade of essential libraries might break some applications (until restarted)[4].

Indem das System auf Runlevel 1 (Single User) und dann zurück auf Runlevel 3 (Multi User) gebracht wird, sollten die meisten (wenn nicht alle) Systemdienste neu gestartet werden. Dies ist aber keine Option,

[1] Ab *Etch* und den folgenden Veröffentlichungen.
[2] Selbst wenn die Bibliotheken aus dem Dateisystem entfernt wurden, werden die Inodes nicht beseitigt, bis kein Programm mehr einen offenen Dateideskriptor mit Verweis auf sie hat.
[3] Je nach der Version von Lsof müssen Sie $8 statt $9 verwenden.
[4] This happened, for example, in the upgrade from libc6 2.2.x to 2.3.x due to NSS authentication issues, see http://lists.debian.org/debian-glibc/2003/03/msg00276.html.

wenn Sie die Sicherheitsaktualisierung über eine Verbindung aus der Ferne (z.B. mit Ssh) vornehmen, da diese getrennt werden würde.

Lassen Sie Vorsicht walten, wenn Sie es mit Sicherheitsaktualisierungen über eine Verbindung aus der Ferne wie mit SSH zu tun haben. Die empfohlene Vorgehensweise für Sicherheits-Upgrades, die Dienste betreffen, ist, den SSH-Daemon neu zu starten und sofort zu versuchen, eine neue SSH-Verbindung herzustellen, ohne die alte zu beenden. Falls der Verbindungsversuch scheitern sollte, machen Sie das Upgrade rückgängig und untersuchen Sie das Problem.

Sicherheitsaktualisierung des Kernels

Stellen Sie zunächst sicher, dass Ihr Kernel durch das Paketsystem verwaltet wird. Wenn Sie die Installation mit dem Installationssystem von Debian 3.0 oder früher durchgeführt haben, ist Ihr Kernel *nicht* in das Paketsystem integriert und könnte veraltet sein. Sie können das leicht überprüfen, indem Sie Folgendes ausführen:

```
$ dpkg -S `readlink -f /vmlinuz`
linux-image-2.6.18-4-686: /boot/vmlinuz-2.6.18-4-686
```

Wenn Ihr Kernel nicht vom Paketsystem verwaltet wird, werden Sie anstatt der obigen Nachricht die Rückmeldung bekommen, dass das Paketverwaltungsprogramm kein Paket finden konnte, das mit der Datei verbunden ist. Die obige Meldung besagt, dass die Datei, die mit dem laufenden Kernel verbunden ist, vom Paket linux-image-2.6.18-4-686 zur Verfügung gestellt wird. Sie müssen also zuerst ein Paket mit einem Kernel-Image von Hand installieren. Das genaue Kernel-Image, das Sie installieren sollten, hängt von Ihrer Architektur und Ihrer bevorzugten Kernelversion ab. Wenn Sie das einmal erledigt haben, können Sie die Sicherheitsaktualisierungen des Kernels wie die jedes anderen Pakets durchführen. Beachten Sie allerdings, dass Kernelaktualisierungen *nur* für Aktualisierungen der gleichen Kernelversion wie der Ihrigen durchgeführt werden. D.h. **apt** wird nicht automatisch Ihren Kernel von 2.4 auf 2.6 aktualisieren (oder von 2.4.26 auf 2.4.27 [5]).

Das Installationssystem von aktuellen Debian-Veröffentlichungen wird den gewählten Kernel als Teil des Paketsystems behandeln. So können Sie überprüfen, welche Kernel Sie installiert haben:

```
$ COLUMNS=150 dpkg -l 'linux-image*' | awk '$1 ~ /ii/ { print $0 }'
```

Um festzustellen, ob Ihr Kernel aktualisiert werden muss, führen Sie Folgendes aus:

```
$ kernfile=`readlink -f /vmlinuz`
$ kernel=`dpkg -S $kernfile | awk -F : '{print $1}'`
$ apt-cache policy $kernel
linux-image-2.6.18-4-686:
  Installiert: 2.6.18.dfsg.1-12
  Installationskandidat: 2.6.18.dfsg.1-12
  Versionstabelle:
 *** 2.6.18.dfsg.1-12 0
        100 /var/lib/dpkg/status
```

Wenn Sie eine Sicherheitsaktualisierung durchführen, die auch das Kernel-Image umfasst, *müssen* Sie das System neu starten, damit die Sicherheitsaktualisierung Wirkung zeigen kann. Anderenfalls lassen Sie immer noch das alte (und verwundbare) Kernel-Image laufen.

[5] Es sei denn, Sie haben ein Kernel-Metapaket wie linux-image-2.6-686 installiert, welches immer die neueste Minor-Version des Kernels einer Architektur installieren wird.

Wenn Sie das System neu starten müssen (wegen eines Kernel-Upgrades), sollten Sie sicherstellen, dass der Kernel fehlerfrei booten wird und die Netzwerkverbindungen hergestellt werden, besonders wenn die Sicherheitsaktuali sierung über eine Verbindung aus der Ferne wie mit SSH durchgeführt wird. Für den ersten Fall können Sie Ihren Boot-Loader so konfigurieren, dass er den Originalkernel lädt, wenn ein Fehler auftritt (für weiterführende Informationen sollten Sie http://www.debian-administration.org/?article=70 lesen). Im zweiten Fall müssen Sie ein Skript verwenden, das die Netzwerkverbindungen testen kann und überprüft, ob der Kernel das Netzwerksystem korrekt gestartet hat, und, wenn das nicht geschehen ist, das System neu startet [6]. Dies sollte böse Überraschungen verhindern, wie wenn Sie den Kernel aktualisieren und dann nach einem Reboot merken, dass die Netzwerkhardware nicht richtig erkannt oder konfiguriert wurde, und Sie daher eine weite Strecke reisen müssen, um das System wieder zum Laufen zu bringen. Natürlich hilft es beim Debuggen von Reboot-Problemen aus der Ferne, wenn die serielle Konsole des Systems [7] mit einem Konsolen- oder Terminalserver verbunden ist.

Änderen Sie das BIOS (noch einmal)

Erinnern Sie sich an „Setzen Sie ein Passwort im BIOS“? Nun, jetzt sollten Sie, nachdem Sie nicht mehr von Wechseldatenträgern booten müssen, die Standard-BIOS-Einstellung ändern, so dass das System *ausschließlich* von der Festplatte bootet. Gehen Sie sicher, dass Sie Ihr BIOS-Passwort nicht verlieren, oder Sie werden nicht mehr ins BIOS zurückkehren können, um die Einstellung wieder zu ändern, damit Sie im Falle eines Festplattenfehlers Ihr System wiederherstellen können, indem Sie zum Beispiel eine CD-ROM benutzen.

Eine andere, weniger sichere, aber bequemere Möglichkeit ist es, das BIOS so einzustellen, dass es von der Festplatte bootet, und nur falls dies fehlschlägt zu versuchen, von austauschbaren Datenträgern zu booten. Übrigens wird dies oft so gemacht, weil viele Leute ihr BIOS-Passwort nur selten benutzen, so dass sie es leicht vergessen.

Ein Passwort für LILO oder GRUB einstellen

Jeder kann sehr einfach eine Root-Shell auf Ihrem System bekommen, indem er einfach

`<name-of-your-bootimage> init=/bin/sh`

am Bootprompt eingibt. Nachdem die Passwörter geändert und das System neu gestartet wurde, hat die Person uneingeschränkten Root-Zugang und kann nach Belieben alles auf Ihrem System machen. Nach dieser Prozedur haben Sie keinen Root-Zugang mehr zu Ihrem System, weil Sie das Root-Passwort nicht kennen.

Um sicher zu stellen, dass dies nicht passieren kann, sollten Sie den Boot-Loader mit einem Passwort schützen. Sie können zwischen einem globalen Passwort und Passwörtern für bestimmte Images wählen.

Für LILO müssen Sie die Konfigurationsdatei `/etc/lilo.conf` bearbeiten und eine **`password`**- und **`restricted`**-Zeile, wie im folgenden Beispiel, einfügen:

```
image=/boot/2.2.14-vmlinuz
   label=Linux
   read-only
   password=hackmich
   restricted
```

[6] Ein Beispielskript mit dem Namen http://www.debian-administration.org/articles/70/testnet ist im Artikel http://www.debian-administration.org/?article=70 enthalten. Ein ausgereifteres Testskript befindet sich im Artikel http://www.debian-administration.org/?article=128.

[7] Das Einrichten einer seriellen Konsole würde den Rahmen dieses Dokuments sprengen. Informationen dazu finden Sie im http://www.tldp.org/HOWTO/Serial-HOWTO.html und im http://www.tldp.org/HOWTO/Remote-Serial-Console-HOWTO/index.html.

Stellen Sie danach sicher, dass die Konfigurationsdatei nicht für alle lesbar ist, um zu verhindern, dass lokale Benutzer das Passwort lesen können. Haben Sie dies getan, rufen Sie lilo auf. Wenn Sie die `restricted`-Zeile weglassen, wird LILO immer nach dem Passwort fragen, egal ob LILO Parameter übergeben wurden oder nicht. Die Standard-Zugriffsrechte auf `/etc/lilo.conf` erlauben Root das Lesen und Schreiben und der Gruppe von lilo.conf, ebenfalls Root, das Lesen.

Wenn Sie GRUB anstelle von LILO verwenden, bearbeiten Sie `/boot/grub/menu.lst` und fügen Sie die folgenden zwei Zeilen am Anfang ein (dabei ersetzen Sie natürlich `hackmich` mit dem vorgesehenen Passwort). Dies verhindert, dass Benutzer die Booteinträge verändern können. `timeout 3` legt eine Wartedauer von 3 Sekunden fest, bevor **Grub** den Standard-Eintrag bootet.

```
timeout 3
password hackmich
```

Um die Integrität Ihres Passwortes zusätzlich abzusichern, können Sie Ihr Passwort verschlüsselt ablegen. Das Dienstprogramm **grub-md5-crypt** erzeugt ein gehashtes Passwort, das mit GRUBs Verschlüsselungsalgorithmus (MD5) kompatibel ist. Um **Grub** mitzuteilen, dass ein Passwort im MD5-Format verwendet wird, benutzen Sie die folgende Anweisung:

```
timeout 3
password --md5 $1$bw0ez$tljnxxKLfMzmnDVaQWgjP0
```

Der Parameter --md5 wurde hinzugefügt, um bei **Grub** einen MD5-Authentifizierungsprozess zu erzwingen. Das angegebene Passwort ist die mit MD5 verschlüsselte Version von »hackmich«. MD5-Passwörter sind Klartext-Passwörtern vorzuziehen. Weitere Informationen über **Grub**-Passwörter können Sie im Paket grub-doc finden.

Entfernen des Root-Prompts von Initramfs

Hinweis: Dies betrifft alle Standard-Kernel, die nach Debian 3.1 veröffentlicht wurden.

Die Linux-Kernel 2.6 enthalten die Möglichkeit, während des Bootvorgangs auf eine Root-Shell zuzugreifen. Dies geschieht, wenn beim Laden von Initramfs ein Fehler auftritt. Dadurch kann der Administrator auf eine Rettungs-Shell mit Root-Rechten zugreifen. Mit dieser Shell können von Hand Module geladen werden, falls eine automatische Erkennung scheitern sollte. Dieses Verhalten ist Standard für ein von **Initramfs-tools** erzeugtes Initramfs. Folgende Fehlermeldung wird auftreten:

```
"ALERT!  /dev/sda1 does not exist.  Dropping to a shell!
```

Um dieses Verhalten abzuschalten, müssen Sie folgenden Boot-Parameter setzen: *panic=0*. Sie können ihn entweder in den Abschnitt »kopt« in `/boot/grub/menu.lst` eintragen und **update-grub** ausführen oder ihn dem Abschnitt »append« von `/etc/lilo.conf` hinzufügen.

Entfernen des Root-Promptes aus dem Kernel

Hinweis: Dies trifft nicht auf Kernel zu, die in Debian 3.1 enthalten sind, da die Wartezeit auf Null verändert wurde.

Linux 2.4-Kernel bieten kurz nach dem Laden des Cramfs einen Weg, Zugriff auf eine Root-Shell zu bekommen, also während das System bootet. Es erscheint eine Meldung, die dem Administrator erlaubt, eine ausführbare Shell mit Root-Privilegien zu öffnen. Diese Shell kann dazu benutzt werden, manuell Module zu laden, falls die automatische Erkennung fehlschlägt. Dies ist das Standard-Verhalten bei **initrd**'s `linuxrc`. Die folgende Meldung wird erscheinen:

```
Press ENTER to obtain a shell (waits 5 seconds)
```

Um dieses Verhalten zu entfernen, müssen Sie `/etc/mkinitrd/mkinitrd.conf` bearbeiten und folgenden Eintrag setzen:

```
# DELAY  Anzahl Sekunden, die das linuxrc Skript warten soll,
# um den Benutzer Eingriffe zu erlauben, bevor das System hochgefahren
# wird
DELAY=0
```

Erstellen Sie anschließend Ihr Ramdisk-Image neu. Dies können Sie zum Beispiel so tun:

```
# cd /boot
# mkinitrd -o initrd.img-2.4.18-k7 /lib/modules/2.4.18-k7
```

oder (vorzugsweise) so:

```
# dpkg-reconfigure -plow kernel-image-2.4.x-yz
```

Einschränkung der Anmeldemöglichkeiten an der Konsole

Manche Sicherheitsrichtlinien können Administratoren dazu zwingen, sich erst als Benutzer mit ihrem Passwort auf dem System einzuloggen und dann Superuser zu werden (mit **Su** oder **Sudo**). Eine solche Richtlinie wird in Debian durch Bearbeitung der Dateien `/etc/pam.d/login` und `/etc/securetty` (falls Sie PAM verwenden) implementiert.

Die Datei `/etc/pam.d/login` In älteren Debian-Veröffentlichungen müssen Sie in der Datei `login.defs` die CONSOLE-Variable ändern, die eine Datei oder eine Liste von Terminals definiert, an denen sich Root anmelden darf. aktiviert das Modul pam_securetty.so. Wenn es richtig konfiguriert ist, wird Root, wenn er sich auf einer unsicheren Konsole anmelden will, nicht nach einem Passwort gefragt, sondern sein Anmeldeversuch wird abgelehnt.

In `securetty`[8] entfernen Sie oder fügen Sie Terminals hinzu, auf denen sich Root anmelden darf. Falls Sie nur lokalen Zugang zur Konsole erlauben wollen, benötigen Sie *console*, *ttyX* [9] und *vc/X* (falls Sie die *devfs*-Schnittstelle verwenden). Sie sollten auch *ttySX* [10] hinzufügen, wenn Sie eine serielle Konsole für den lokalen Zugang verwenden (wobei X eine ganze Zahl ist; es kann wünschenswert sein, mehrere Instanzen zu verwenden). Die Standardeinstellung in *Wheezy* [11] beinhaltet viele tty-Konsolen, serielle Schnittstellen und virtuelle Konsolen sowie den X-Server und das *console*-Gerät. Sie können das ohne Probleme anpassen, wenn Sie nicht derartige viele Konsolen benutzen. Sie können die Anzahl der Konsolen und Schnittstellen in `/etc/inittab` überprüfen [12]. Weiterführende Informationen zu Terminal-Schnittstellen finden Sie im http://tldp.org/HOWTO/Text-Terminal-HOWTO-6.html.

Wenn Sie PAM benutzen, können Sie auch andere Änderungen am Login-Prozess, die auch Einschränkungen für einzelne Benutzer oder Gruppen zu bestimmten Zeiten enthalten können, durch Konfigurati-

[8] Die Datei `/etc/securetty` ist eine Konfigurationsdatei, die zum Paket login gehört.
[9] Oder *ttyvX* unter GNU/FreeBSD und *ttyE0* unter GNU/KNetBSD.
[10] Oder *comX* unter GNU/Hurd, *cuaaX* unter GNU/FreeBSD und *ttyXX* unter GNU/KNetBSD.
[11] Die Standardeinstellung in *Woody* beinhaltet zwölf lokale tty- und virtuelle Konsolen und die *console*-Schnittstelle. Anmeldungen aus der Ferne sind nicht erlaubt. In *Sarge* stellt die Standardeinstellung 64 Konsolen für tty- und virtuelle Konsolen zu Verfügung.
[12] Achten Sie auf die *getty* Einträge.

on der Datei `/etc/pam.d/login` vornehmen. Eine interessante Eigenschaft, die man auch abschalten kann, ist die Möglichkeit, sich mit einem leeren Passwort (Null-Passwort) anzumelden. Diese Eigenschaft kann eingeschränkt werden, indem Sie *nullok* aus folgender Zeile entfernen:

```
auth       required   pam_unix.so nullok
```

Einschränkung des System-Neustarts von der Konsole aus

Wenn eine Tastatur an Ihr System angeschlossen ist, kann es jeder (ja, wirklich *jeder*) mit physischem Zugang zu Ihrem System neu starten, ohne sich an Ihrem System anmelden zu müssen, einfach indem er die Tastenkombination *Strg+Alt+Entf* drückt (auch als *Affengriff* bekannt). Dies könnte gegen Ihre Sicherheitsrichtlinien verstoßen (oder auch nicht).

Dies ist schwerwiegender, wenn das Betriebssystem in einer virtuellen Umgebung läuft. Dann erstreckt sich diese Fähigkeit auch auf Benutzer, die Zugriff auf die virtuelle Konsole haben (was auch über das Netzwerk geschehen könnte). Beachten Sie zudem, dass in einer solchen Umgebung diese Tastenkombination ständig verwendet wird (um in einigen grafischen Benutzeroberflächen eine Login-Shell zu öffnen), so dass ein Systemadministrator sie *virtuell* auslösen kann und das System neu startet.

Es gibt zwei Möglichkeiten, dies einzuschränken:

- mit einer Konfiguration, mit der nur *bestimmte* Benutzer das System neu starten dürfen,
- diese Eigenschaft vollständig zu deaktivieren.

Wenn Sie dies einschränken wollen, müssen Sie in `/etc/inittab` sicherstellen, dass die Zeile, die **`ctrlaltdel`** enthält, s**hutdown** mit der Option **`-a`** aufruft.

Standardmäßig enthält Debian diese Optionen:

```
ca:12345:ctrlaltdel:/sbin/shutdown -t1 -a -r now
```

Die Option **`-a`** ermöglicht es, *einigen* Benutzern zu erlauben, das System neu zu starten (vgl. die Handbuchseite shutdown(8)). Dazu müssen Sie die Datei `/etc/shutdown.allow` erstellen und die Namen der Benutzer, die das System neu booten dürfen, eintragen. Wenn der *Affengriff* in einer Konsole ausgeführt wird, wird geprüft, ob irgendeiner der Benutzer, die in der Datei aufgelistet sind, angemeldet ist. Wenn es keiner von ihnen ist, wird **Shutdown** das System *nicht* neu starten.

Wenn Sie die Tastenkombination Strg+Alt+Entf deaktivieren möchten, müssen Sie nur die Zeile mit *ctrlaltdel* in `/etc/inittab` auskommentieren.

Remember to run **`init q`** after making any changes to the `/etc/inittab` file for the changes to take effect.

Einschränkung der Tastenkombination Magische S-Abf

Die Tastenkombination *Magische S-Abf* (Magic SysRq) erlaubt es Benutzern einer Konsole eines Linux-Systems, bestimmte systemnahe Befehle auszuführen, indem gleichzeitig *Alt+S-Abf* und eine bestimmte Taste gedrückt wird. Die Taste S-Abf wird auf vielen Tastaturen mit *Druck* bezeichnet.

Seit der Veröffentlichung von Etch ist die Tastenkombination Magische S-Abf im Linux-Kernel aktiviert, damit die Benutzer einer Konsole bestimmte Privilegien erhalten können. Ob dies auf Ihr System zutrifft, erkennen Sie daran, ob `/proc/sys/kernel/sysrq` existiert, und an dessen Wert:

```
$ cat /proc/sys/kernel/sysrq
438
```

Der oben gezeigte Standardwert erlaubt alle S-Abf-Funktionen mit Ausnahme der Möglichkeit, Signale an Prozesse zu senden. Zum Beispiel können Benutzer, die an der Konsole angemeldet sind, alle Systeme nur-lesend neu einhängen, das System neu starten oder eine Kernelpanik auslösen. Wenn alle Fähigkeit aktiviert sind oder in älteren Kernel-Versionen (früher als 2.6.12), wird der Wert einfach 1 sein.

Sie sollten diese Fähigkeit deaktivieren, wenn der Zugang zur Konsole nicht auf angemeldete Benutzer beschränkt ist, nämlich wenn die Konsole an ein Modem angebunden ist, es leichten physischen Zugang zum System gibt oder es in einer virtuellen Umgebung läuft und andere Benutzer auf die Konsole zugreifen können. Dafür müssen Sie `/etc/sysctl.conf` bearbeiten und folgende Zeile einfügen:

```
# Schaltet die Magische S-Abf-Taste ab
kernel.sysrq = 0
```

For more information, read security chapter in the Remote Serial Console HOWTO [http://tldp.org/HOWTO/Remote-Serial-Console-HOWTO/security-sysrq.html], Kernel SysRQ documentation [https://www.kernel.org/doc/Documentation/admin-guide/sysrq.rst]. and the Magic_SysRq_key wikipedia entry [http://en.wikipedia.org/wiki/Magic_SysRq_key].

Partitionen auf die richtige Art einhängen

Wenn Sie ein `Ext`-Dateisystem (`ext2`, `ext3` oder `ext4`) einhängen, können Sie verschiedene Optionen mit dem mount-Befehl oder in `/etc/fstab` verwenden. Dies ist zum Beispiel mein fstab-Eintrag für meine `/tmp`-Partition:

```
/dev/hda7    /tmp    ext2    defaults,nosuid,noexec,nodev    0    2
```

Achten Sie auf den Abschnitt mit den Optionen. Die Option `nosuid` ignoriert komplett alle setuid- und setgid-Bits, während `noexec` das Ausführen von Programmen unterhalb des Einhängepunkts verbietet und `nodev` Gerätedateien ignoriert. Das hört sich toll an, aber:

- ist nur auf `ext2` oder `ext3`-Dateisysteme anwendbar,
- kann leicht umgangen werden.

Die Option `noexec`, die verhindert, dass Programme ausgeführt werden können, ließ sich in früheren Kernelversionen leicht umgehen:

```
alex@joker:/tmp# mount | grep tmp
/dev/hda7 on /tmp type ext2 (rw,noexec,nosuid,nodev)
alex@joker:/tmp# ./date
bash: ./date: Keine Berechtigung
alex@joker:/tmp# /lib/ld-linux.so.2 ./date
```

```
So 3. Dec 17:49:23 CET 2000
```

Neuere Versionen des Kernels verarbeiten aber die Option `noexec` richtig:

```
angrist:/tmp# mount | grep /tmp
/dev/hda3 on /tmp type ext3 (rw,noexec,nosuid,nodev)
angrist:/tmp# ./date
bash: ./tmp: Keine Berechtigung
angrist:/tmp# /lib/ld-linux.so.2 ./date
./date: error while loading shared libraries: ./date: failed to map segment
from shared object: Operation not permitted
```

Wie auch immer, viele Skript-Kiddies haben Exploits, die versuchen, eine Datei in `/tmp` zu erstellen und auszuführen. Falls sie keine Ahnung haben, werden sie in dieser Grube hängen bleiben. Mit anderen Worten: Ein Benutzer kann nicht hereingelegt werden, einen ausführbaren Trojaner in `/tmp` laufen zu lassen, zum Beispiel, indem er zufällig `/tmp` in seinen Suchpfad (PATH) aufnimmt.

Seien Sie sich auch bewusst, dass manche Skripte darauf aufbauen, dass `/tmp` ausführbare Rechte hat. Bemerkenswerterweise hatte (oder hat?) Debconf Probleme bei dieser Sache, weitere Informationen enthält Fehler http://bugs.debian.org/116448.

Nachfolgend ein gründlicheres Beispiel. Eine Anmerkung dazu: `/var` könnte auch noexec enthalten, aber manche Software [13] verwahrt ihre Programme unterhalb von `/var`. Dasselbe gilt für die Option nosuid.

```
/dev/sda6   /usr          ext3    defaults,ro,nodev       0       2
/dev/sda12  /usr/share    ext3    defaults,ro,nodev,nosuid        0       2
/dev/sda7   /var          ext3    defaults,nodev,usrquota,grpquota 0      2
/dev/sda8   /tmp          ext3    defaults,nodev,nosuid,noexec,usrquota,grpquota
/dev/sda9   /var/tmp      ext3    defaults,nodev,nosuid,noexec,usrquota,grpquota
/dev/sda10  /var/log      ext3    defaults,nodev,nosuid,noexec    0       2
/dev/sda11  /var/account  ext3    defaults,nodev,nosuid,noexec    0       2
/dev/sda13  /home         ext3    rw,nosuid,nodev,exec,auto,nouser,async,usrquota,
/dev/fd0    /mnt/fd0      ext3    defaults,users,nodev,nosuid,noexec      0
/dev/fd0    /mnt/floppy   vfat    defaults,users,nodev,nosuid,noexec      0
/dev/hda    /mnt/cdrom    iso9660 ro,users,nodev,nosuid,noexec            0
```

`/tmp` noexec setzen

Be careful if setting `/tmp` noexec when you want to install new software, since some programs might use it for installation. apt is one such program (see http://bugs.debian.org/116448) if not configured properly `APT::ExtractTemplates::TempDir` (see apt-extracttemplates(1)). You can set this variable in `/etc/apt/apt.conf` to another directory with exec privileges other than `/tmp`.

/usr auf nur-lesend setzen

Wenn Sie auf `/usr` nur lesenden Zugriff erlauben, werden Sie nicht in der Lage sein, neue Pakete auf Ihrem Debian-GNU/Linux-System zu installieren. Sie werden es erst mit Schreibzugriff erneut einhängen müssen, die Pakete installieren und dann wieder nur mit lesendem Zugriff einhängen. Apt kann so konfiguriert werden, dass Befehle vor und nach dem Installieren von Paketen ausgeführt werden. Daher müssen Sie es passend konfigurieren.

[13] Einiges davon trifft auf den Paketverwalter Dpkg zu, da die Installations- oder Deinstallationsanweisungen (post, pre) unter `/var/lib/dpkg/` liegen, und auch auf Smartlist.

Dafür müssen Sie `/etc/apt/apt.conf` bearbeiten und Folgendes einfügen:

```
DPkg
{
    Pre-Invoke  { "mount /usr -o remount,rw" };
    Post-Invoke { "mount /usr -o remount,ro" };
};
```

Beachten Sie, dass das Post-Invoke mit der Fehlermeldung »/usr ist belegt« scheitern kann. Dies passiert vorwiegend, wenn Sie eine Datei benutzen, die aktualisiert wurde. Sie können diese Programme finden, indem Sie Folgendes ausführen:

```
# lsof +L1
```

Halten Sie diese Programme an oder starten Sie sie erneut und rufen dann Post-Invoke manuell auf. *Achtung!* Das bedeutet, dass Sie wahrscheinlich jedes Mal Ihre Sitzung von X (falls Sie eine laufen haben) neu starten müssen, wenn Sie ein größeres Upgrade Ihres Systems durchführen. Sie müssen entscheiden, ob ein nur lesbares `/usr` zu Ihrem System passt. Vergleichen Sie auch diese http://lists.debian.org/debian-devel/2001/11/threads.html#00212.

Den Benutzerzugang absichern

Benutzerauthentifizierung: PAM

PAM (Pluggable Authentication Modules) erlaubt Systemadministratoren, auszuwählen, wie Anwendungen Benutzer authentifizieren. Beachten Sie, dass PAM nichts machen kann, solange die Anwendung nicht mit Unterstützung für PAM kompiliert wurde. Die meisten Anwendungen, die mit Debian geliefert werden, haben diese Unterstützung eingebaut. Vor Version 2.2 hatte Debian keine Unterstützung für PAM. Die derzeitige Standardkonfiguration für jeden Dienst, der PAM benutzt, ist es, UNIX-Authentifizierung zu emulieren (lesen Sie `/usr/share/doc/libpam0g/Debian-PAM-MiniPolicy.gz`, um mehr darüber zu erfahren, wie PAM-Dienste unter Debian arbeiten *sollten*).

Jede Anwendung mit PAM-Unterstützung stellt eine Konfigurationsdatei unter `/etc/pam.d/` zur Verfügung, in welcher Sie ihr Verhalten einstellen können:

- welches Verfahren zur Authentifizierung benutzt wird
- welches Verfahren innerhalb einer Sitzung benutzt wird
- wie Passwörter überprüft werden

The following description is far from complete, for more information you might want to read the Linux-PAM Guides [https://packages.debian.org/sid/libpam-doc] as a reference. This documentation is available in the system if you install the libpam-doc at `/usr/share/doc/libpam-doc/html/`.

PAM bieten Ihnen die Möglichkeit, durch mehrere Authentifizierungsschritte auf einmal zu gehen, ohne dass der Benutzer es weiß. Sie können gegen eine Berkeley-Datenbank und gegen die normale `passwd`-Datei authentifizieren, und der Benutzer kann sich nur anmelden, wenn er beide Male korrekt authentifiziert wurde. Sie können mit PAM viel einschränken, genauso wie Sie Ihr System weit öffnen können. Seien Sie also vorsichtig. Eine typische Konfigurationszeile hat ein Steuerfeld als zweites Element. Generell sollte es auf `requisite` gesetzt werden, so wird ein Anmeldefehler erzeugt, wenn eines der Module versagt.

Passwortsicherheit in PAM

Sehen Sie sich `/etc/pam.d/common-password` an, die von `/etc/pam.d/passwd` eingebunden wird. [14] Andere Dateien in `/etc/pam.d/` lesen diese Datei ein, um die Verwendung eines Passworts durch Programme, die einen Zugriff auf das System erlauben, wie etwa das Konsolen-Login (Login), grafische Login-Manager (z.B. `Gdm` oder `Lightdm`) und Login aus der Ferne (etwa mit Sshd), zu definieren.

Sie müssen sicherstellen, dass das Modul pam_unix.so die Option »sha512« verwendet, damit die Passwörter verschlüsselt werden. In Debian Squeeze ist dies standardmäßig eingerichtet.

Die Zeile mit der Konfiguration des Moduls pam_unix sollte etwa so aussehen:

```
password   [success=1 default=ignore]      pam_unix.so nullok obscure minlen=8 s
```

Dieser Ausdruck

- erzwingt die Verschlüsselung von Passwörtern mit der Hashfunktion SHA-512, wenn sie gespeichert werden (Option *sha512*),
- aktiviert die Überprüfung der Komplexität eines Passworts, wie sie in der Handbuchseite pam_unix(8) beschrieben wird (Option *obscure*),
- erfordert, dass das Passwort mindestens acht Zeichen lang ist (Option *min*).

Sie müssen sicherstellen, dass in PAM-Anwendungen verschlüsselte Passwörter verwendet werden, weil dies Wörterbuchangriffe erschwert. Zugleich wird es dadurch möglich, Passwörter mit mehr als acht Zeichen einzusetzen.

Da mit diesem Modul auch definiert wird, wie Passwörter geändert werden, weil es von **Chpasswd** eingebunden wird, können Sie die Passwortsicherheit Ihres Systems erhöhen, indem sie libpam-cracklib installieren und folgenden Ausdruck in die Konfigurationsdatei `/etc/pam.d/common-password` eintragen:

```
# Gehen Sie sicher, dass Sie libpam-cracklib zuerst installiert haben,
# sonst werden Sie sich nicht einloggen können
password   required     pam_cracklib.so retry=3 minlen=12 difok=3
password   [success=1 default=ignore]      pam_unix.so obscure minlen=8 sha512 u
```

Also, was macht diese Beschwörungsformel nun genau? Die erste Zeile lädt das PAM-Modul cracklib, welches einen Passwort-Sicherheitscheck bereitstellt. Es fragt nach einem neuen Passwort mit mindestens zwölf Zeichen [15], einer Differenz von mindestens drei Zeichen zum alten Passwort und erlaubt drei Versuche. Cracklib benötigt ein Paket mit Wörterlisten (wie wngerman, wenglish, wspanish, ...). Stellen Sie also sicher, dass Sie ein passendes Paket für Ihre Sprache installiert haben. Ansonsten ist Cracklib nicht verwendbar.

Die zweite Zeile (mit dem Module pam_unix.so) ist – wie oben beschrieben – der Standard in Debian mit Ausnahme der Option *use_authok*. Diese Option ist notwendig, wenn pam_unix.so nach pam_cracklib.so aufgerufen wird, damit das Passwort vom zuerst aufgerufenen Modul weitergereicht wird. Anderenfalls muss der Benutzer sein Passwort zweimal eingegeben.

[14] In früheren Debian-Veröffentlichungen befand sich die Konfiguration der Module direkt in `/etc/pam.d/passwd`.

[15] Die Option *minlen* ist nicht auf Anhieb völlig verständlich, weil sie nicht die genaue Anzahl der Zeichen eines Passworts ist. Ein Kompromiss zwischen Komplexität und Länge eines Passworts kann mit dem Parameter »credit« für verschiedene Arten von Zeichen erreicht werden. Weitere Informationen finden Sie in der Handbuchseite pam_cracklib(8).

Weitere Informationen über die Konfiguration von Cracklib finden Sie in der Handbuchseite pam_cracklib(8) und dem Artikel http://www.deer-run.com/~hal/sysadmin/pam_cracklib.html von Hal Pomeranz.

Mit dem PAM-Modul Cracklib richten Sie eine Richtlinie ein, welche die Verwendung guter Passwörter erzwingt.

Als Alternative können Sie auch PAM-Module einsetzen, die eine Zwei-Faktor-Authentifizierung verwenden, wie z.B. libpam-barada, libpam-google-authenticator, libpam-oath, libpam-otpw, libpam-poldi, libpam-usb oder libpam-yubico. Diese Module ermöglichen es, sich mit einer externen Authentifizierungsmethode am System anzumelden, etwa mit einer Chipkarte, einem USB-Stick oder Einmal-Passwörtern, die mit einer externen Anwendung, z.B. auf einem Mobiltelefon, erzeugt wurden.

Denken Sie daran, dass diese Einschränkungen alle Benutzer betreffen *außer* Änderungen von Passwörtern, die der Benutzer Root vornimmt. Dieser kann unabhängig von den eingerichteten Beschränkungen jedes Passwort (in Hinblick auf Länge oder Komplexität) für sich oder andere Benutzer vergeben.

Steuerung des Benutzerzugangs in PAM

Um sicher zu stellen, dass sich der Benutzer Root nur an lokalen Terminals anmelden kann, sollten Sie die folgende Zeile in `/etc/pam.d/login` einfügen:

```
auth     requisite  pam_securetty.so
```

Danach sollten Sie die Liste der Terminals in `/etc/securetty` ändern, auf denen sich Root unmittelbar anmelden darf (wie in „Einschränkung der Anmeldemöglichkeiten an der Konsole“ beschrieben). Alternativ dazu können Sie auch das pam_access-Modul aktivieren und `/etc/security/access.conf` bearbeiten. Dieses Vorgehen erlaubt eine allgemeinere und feiner abgestimmte Zugangssteuerung, leider fehlen aber vernünftige Protokollmeldungen (diese sind in PAM nicht standardisiert und sind ein besonders unbefriedigendes Problem). Wir werden zu `access.conf` in Kürze zurückkehren.

Höchstgrenzen für Benutzer in PAM

Die folgende Zeile sollte in `/etc/pam.d/login` aktiviert werden, um den Benutzern Grenzen ihrer Systemressourcen zu setzen.

```
session  required   pam_limits.so
```

Dies schränkt die Systemressourcen ein, die ein Benutzer nutzen darf (siehe „Ressourcen-Nutzung begrenzen: Die Datei `limits.conf`“ weiter unten). Sie können zum Beispiel die Anzahl der Logins, die man haben kann, einschränken (für eine Gruppe von Benutzern oder systemweit), die Anzahl der Prozesse, den belegten Speicher etc.

Steuerung von su in PAM

Wenn Sie **Su** schützen möchten, so dass nur manche Leute es benutzen können, um Root auf Ihrem System zu werden, müssen Sie eine neue Gruppe »wheel« zu Ihrem System hinzufügen (das ist der sauberste Weg, da keine Datei solche Gruppenrechte bisher benutzt). Fügen Sie Root und die anderen Benutzer, die zu Root **su**en können sollen, zu dieser Gruppe hinzu. Ergänzen Sie anschließend `/etc/pam.d/su/` um die folgende Zeile:

```
auth        requisite   pam_wheel.so group=wheel debug
```

Dies stellt sicher, dass nur Personen aus der Gruppe »wheel« **su** benutzen können, um Root zu werden. Andere Benutzer wird es nicht möglich sein, Root zu werden. Tatsächlich werden sie eine ablehnende Nachricht bekommen, wenn sie versuchen, Root zu werden.

Wenn Sie es nur bestimmten Benutzern erlauben wollen, sich bei einem PAM-Dienst zu authentifizieren, ist dies sehr leicht zu erreichen, indem Sie Dateien benutzen, in denen die Benutzer, denen es erlaubt ist, sich anzumelden (oder nicht), gespeichert sind. Stellen Sie sich vor, Sie möchten lediglich dem Benutzer »ref« erlauben, sich mittels **ssh** anzumelden. Sie schreiben ihn also in eine Datei `/etc/ssh-users-allowed` und schreiben das Folgende in `/etc/pam.d/ssh`:

```
auth        required    pam_listfile.so item=user sense=allow file=/etc/sshusers
```

Temporäre Verzeichnisse in PAM

Da es eine Reihe von Sicherheitslücken mit so genannten unsicheren temporären Dateien zum Beispiel in Thttpd (vgl. http://www.debian.org/security/2005/dsa-883) gab, lohnt es sich, das Paket libpam-tmpdir zu installieren. Sie müssen dann lediglich Folgendes zu `/etc/pam.d/common-session` hinzuzufügen:

```
session    optional     pam_tmpdir.so
```

Es gab auch eine Diskussion, dies standardmäßig in Debian einzufügen. Sehen Sie sich http://lists.debian.org/debian-devel/2005/11/msg00297.html für weitere Informationen an.

Konfiguration für nicht definierte PAM-Anwendungen

Zuletzt, aber nicht am unwichtigsten, erstellen Sie `/etc/pam.d/other` mit den folgenden Zeilen:

```
auth     required       pam_securetty.so
auth     required       pam_unix_auth.so
auth     required       pam_warn.so
auth     required       pam_deny.so
account  required       pam_unix_acct.so
account  required       pam_warn.so
account  required       pam_deny.so
password required       pam_unix_passwd.so
password required       pam_warn.so
password required       pam_deny.so
session  required       pam_unix_session.so
session  required       pam_warn.so
session  required       pam_deny.so
```

Diese Zeilen stellen für alle Anwendungen, die PAM unterstützen, eine gute Standardkonfiguration dar (Zugriff wird standardmäßig verweigert).

Ressourcen-Nutzung begrenzen: Die Datei `limit-s.conf`

Sie sollten sich wirklich ernsthaft mit dieser Datei beschäftigen. Hier können Sie Ihren Benutzern Ressourcengrenzen vorgeben. In alten Veröffentlichungen war die Konfigurationsdatei `/etc/limit-`

`s.conf`. Aber in neueren Versionen (mit PAM) sollte stattdessen die Konfigurationsdatei `/etc/security/limits.conf` benutzt werden.

Wenn Sie die Ressourcennutzung nicht einschränken, kann *jeder* Benutzer mit einer gültigen Shell auf Ihrem System (oder sogar ein Einbrecher, der das System durch einen Dienst kompromittierte, oder ein außer Kontrolle geratener Daemon) so viel CPU, Speicher, Stack etc. benutzen, wie das System zur Verfügung stellen kann. Dieses Problem der *Überbeanspruchung von Ressourcen* kann mit der Nutzung von PAM gelöst werden.

Es gibt einen Weg, Ressourcengrenzen zu manchen Shells hinzuzufügen (zum Beispiel hat **Bash ulimit**, siehe bash(1)). Aber da nicht alle die gleichen Höchstgrenzen zur Verfügung stellen und der Benutzer seine Shell ändern kann (siehe chsh(1)), ist es besser, die Höchstgrenzen in den PAM-Modulen zu platzieren, da diese unabhängig von der verwendeten Shell Anwendung finden und auch PAM-Module betreffen, die nicht shellorientiert sind.

Ressourcengrenzen werden vom Kernel verhängt, aber sie müssen durch `limits.conf` konfiguriert werden und die PAM-Konfiguration der verschiedenen Dienste muss das passende PAM laden. Sie können herausfinden, welche Dienste Höchstgrenzen durchsetzen, indem Sie Folgendes ausführen:

```
$ find /etc/pam.d/ \! -name "*.dpkg*" | xargs -- grep limits |grep -v ":#"
```

Für gewöhnlich setzen `Login`, `Ssh` und die grafischen Sitzungsmanager (`Gdm`, `Kdm` und `Xdm`) Benutzerhöchstgrenzen durch, aber Sie sollte dies auch in anderen Konfigurationsdateien für PAM wie für `Cron` vornehmen, um zu verhindern, dass System-Daemons alle Systemressourcen aufbrauchen.

Die konkreten Begrenzungen, die Sie festlegen wollen, hängt von den Ressourcen Ihres Systems ab. Das ist einer der Hauptgründe, warum keine Höchstgrenzen in der Standardinstallation enthalten sind.

So setzt die Konfiguration im Beispiel unten eine Begrenzung von 100 Prozessen für alle Benutzer (um *Fork-Bomben* [16] zu vermeiden), eine Begrenzung auf 10 MB Speicher pro Prozess und ein Höchstgrenze von 10 gleichzeitigen Logins durch. Benutzer in der Gruppe `adm` haben höhere Begrenzungen und können Dateien mit einem Speicherabbild schreiben, wenn sie das wollen (es gibt also nur eine *weiche* Begrenzung).

```
*               soft    core            0
*               hard    core            0
*               hard    rss             1000
*               hard    memlock         1000
*               hard    nproc           100
*               -       maxlogins       1
*               hard    data            102400
*               hard    fsize           2048
@adm            hard    core            100000
@adm            hard    rss             100000
@adm            soft    nproc           2000
@adm            hard    nproc           3000
@adm            hard    fsize           100000
@adm            -       maxlogins       10
```

Dies könnten die Höchstgrenzen eines Standardbenutzers (einschließlich der System-Daemons) sein:

[16] Programme, die immer mehr Prozesse erzeugen, um so das System zum Absturz zu bringen, d.Ü.

```
$ ulimit -a
core file size        (blocks, -c) 0
data seg size         (kbytes, -d) 102400
file size             (blocks, -f) 2048
max locked memory     (kbytes, -l) 10000
max memory size       (kbytes, -m) 10000
open files                    (-n) 1024
pipe size          (512 bytes, -p) 8
stack size            (kbytes, -s) 8192
cpu time             (seconds, -t) unlimited
max user processes            (-u) 100
virtual memory        (kbytes, -v) unlimited
```

Und dies die Höchstgrenzen für einen Administrator:

```
$ ulimit -a
core file size        (blocks, -c) 0
data seg size         (kbytes, -d) 102400
file size             (blocks, -f) 100000
max locked memory     (kbytes, -l) 100000
max memory size       (kbytes, -m) 100000
open files                    (-n) 1024
pipe size          (512 bytes, -p) 8
stack size            (kbytes, -s) 8192
cpu time             (seconds, -t) unlimited
max user processes            (-u) 2000
virtual memory        (kbytes, -v) unlimited
```

Lesen Sie für weitere Informationen:

- PAM reference guide for available modules [https://web.archive.org/web/20030601112932/http://www.kernel.org/pub/linux/libs/pam/Linux-PAM-html/pam-6.html]
- PAM configuration article [https://web.archive.org/web/20030217012148/http://www.sama-g.com/documents/s=1161/sam0009a/0009a.htm].
- http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html in dem Abschnitt *Limiting users overview*
- http://seifried.org/lasg/users/ in dem Abschnitt *Limiting and monitoring users*

Aktionen bei der Benutzeranmeldung: Bearbeiten von `/etc/login.defs`

Der nächste Schritt ist es, die grundlegende Konfiguration und die Aktionen bei der Benutzeranmeldung zu bearbeiten. Beachten Sie, dass diese Datei kein Bestandteil der PAM-Konfiguration ist. Sie ist eine Konfigurationsdatei, die von den Programmen **Login** und **Su** berücksichtigt wird. Es ist also wenig sinnvoll, sie auf Fälle abzustimmen, in denen keines der beiden Programme wenigstens indirekt aufgerufen wird (Das Programm Getty, welches auf der Konsole läuft und die anfängliche Anmeldeaufforderung zu Verfügung stellt, *ruft* Login auf).

```
FAILLOG_ENAB        yes
```

Wenn Sie diese Variable einschalten, werden fehlgeschlagene Anmeldeversuche protokolliert. Es ist wichtig, hier auf dem Laufendem zu bleiben, um jemanden zu ermitteln, der einen Brute-Force-Angriff versucht.

```
LOG_UNKFAIL_ENAB    no
```

Wenn Sie diese Variable auf »yes« setzen, werden unbekannte Benutzernamen protokolliert, wenn eine Anmeldung scheitert. Es ist zu empfehlen, sie auf »no« (den Standard) zu belassen, da anderenfalls das Passwort eines Benutzers aufgezeichnet werden könnte (falls er nämlich versehentlich anstatt seines Benutzernames sein Passwort eingibt). Falls Sie sie dennoch auf »yes« setzen, müssen Sie sicherstellen, dass die Protokolldateien angemessene Zugriffsrechte haben (zum Beispiel 640, mit einer passenden Gruppenzugehörigkeit wie adm).

```
SYSLOG_SU_ENAB      yes
```

Dies schaltet das Mitprotokollieren von **su**-Versuchen im `Syslog` ein. Sehr wichtig auf ernsthaft betriebenen Maschinen, aber beachten Sie, dass dies auch die Privatsphäre verletzen kann.

```
SYSLOG_SG_ENAB      yes
```

Das gleiche wie bei `SYSLOG_SU_ENAB`, aber für das Programm **Sg**.

```
ENCRYPT_METHOD  SHA512
```

Wie bereits erklärt, reduziert eine Verschlüsselung von Passwörtern die Gefahr von Wörterbuchangriffen erheblich, da Sie längere Passwörter benutzen können. Diese Definition muss mit dem Wert in `/etc/pam.d/common-password` übereinstimmen.

Aktionen bei der Benutzeranmeldung: `/etc/pam.d/login` bearbeiten

Sie können die Datei zur Konfiguration des Anmeldevorgangs anpassen, um eine strengere Richtlinie festzuschreiben. Zum Beispiel können Sie die Wartezeit zwischen zwei Anmeldeversuchen im Vergleich zur Standardkonfiguration erhöhen. Diese Standardvorgabe setzt eine Wartezeit von drei Sekunden:

```
auth       optional   pam_faildelay.so  delay=3000000
```

Wenn Sie den Wert von *delay* erhöhen, wird es schwieriger, sich durch bloßes Ausprobieren von Passwörtern (brute force) erfolgreich am Terminal anzumelden. Wenn ein falsches Passwort eingegeben wird, muss ein möglicher Angreifer (oder ein normaler Benutzer!) viele Sekunden warten, bis er wieder eine Eingabeaufforderung erhält, wodurch das Durchprobieren von Passwörtern sehr zeitaufwendig werden kann. So müssen etwa Benutzer bei *delay=10000000* zehn Sekunden warten, wenn sie das falsche Passwort eingeben.

In dieser Datei können Sie auch einrichten, dass das System dem Benutzer vor einer Anmeldung eine Nachricht anzeigt. Standardmäßig ist dies deaktiviert, wie Sie hier sehen können:

```
# auth       required   pam_issue.so issue=/etc/issue
```

Falls es Ihre Sicherheitsrichtlinie erfordert, können Sie mit dieser Datei eine Standardnachricht, dass der Zugang zum System beschränkt und der Benutzerzugang protokolliert wird, anzeigen lassen. Ein solcher Hinweis kann in bestimmten Regionen und nach der jeweiligen Rechtsprechung notwendig sein. Um dies zu aktivieren, müssen Sie nur die entsprechende Mitteilung in die Datei `/etc/issue` [17] eintragen und das Kommentarzeichen in der Zeile in `/etc/pam.d/login` entfernen, um das Modul pam_issue.so zu aktivieren. In dieser Datei können Sie weitere Einstellungen vornehmen, die für Ihre Sicherheit relevant sein könnten, wie zum Beispiel:

- Regeln erstellen, welcher Benutzer zu welchen Zeiten auf das System zugreifen kann, indem Sie das Modul *pam_time.so* aktivieren und `/etc/security/time.conf` entsprechend konfigurieren (standardmäßig deaktiviert),
- den Anmeldevorgang so einrichten, dass Benutzerbegrenzungen, die in `/etc/security/limits.conf` definiert sind, verwendet werden (standardmäßig aktiviert),
- dem Benutzer Informationen über die vorangegangene Anmeldung anzeigen (standardmäßig aktiviert),
- nach erfolgter Anmeldung den Benutzern eine Nachricht (`/etc/motd` und `/run/motd.dynamic`) anzeigen (standardmäßig aktiviert).

Ftp einschränken: bearbeiten von `/etc/ftpusers`

Die Datei `/etc/ftpusers` enthält eine Liste von allen Benutzern, denen es *nicht* erlaubt ist, sich auf dem Rechner mit Ftp einzuloggen. Benutzen Sie diese Datei nur, wenn Sie wirklich Ftp erlauben wollen (wozu im Allgemeinen nicht geraten wird, da es Klartext-Passwörter benutzt). Wenn Ihr Ftp-Daemon PAM unterstützt, können Sie dies ebenfalls benutzen, um Benutzern bestimmte Dienste zu erlauben oder zu verbieten.

FIXME (FEHLER): Ist es ein Fehler, dass `ftpusers` in Debian standardmäßig *nicht* die Benutzer mit Administratorenrecht (in base-passwd) beinhaltet?

Folgender Befehl ist ein bequemer Weg, alle Systemkonten zu `/etc/ftpusers` hinzuzufügen:

```
$ awk -F : '{if ($3<1000) print $1}' /etc/passwd > /etc/ftpusers
```

Verwendung von Su

Wenn es wirklich benötigt wird, dass Benutzer der Super-User (also Root, d.Ü.) auf Ihrem System werden, zum Beispiel um Pakete zu installieren oder neue Benutzer anzulegen, können Sie das Programm **Su** benutzen, um Ihre Identität zu wechseln. Sie sollten jeden Login als Benutzer Root vermeiden und stattdessen das Programm **Su** benutzen. Eigentlich ist die beste Lösung, **Su** zu entfernen und zu **Sudo** zu wechseln, da es eine feinere Steuerung und mehr Möglichkeiten bietet als **Su**. Wie auch immer, **Su** ist verbreiteter und wird auf vielen Unices eingesetzt.

Verwendung von Sudo

Sudo erlaubt es dem Benutzer, bestimmte Befehle unter einer anderen Benutzeridentität auszuführen, sogar als Root. Wenn der Benutzer zu `/etc/sudoers` hinzugefügt ist und sich korrekt authentifiziert, ist er in der Lage, Befehle, die in `/etc/sudoers` definiert wurden, auszuführen. Sicherheitsverletzun-

[17] Der Standardinhalt dieser Datei enthält Informationen über das Betriebssystem und dessen Version, die Sie möglicherweise unbekannten Benutzern nicht mitteilen möchten.

gen, wie ein inkorrektes Passwort oder der Versuch ein Programm auszuführen, für das die Rechte nicht ausreichen, werden protokolliert und an Root gemailt.

Administrativen Fernzugriff verweigern

Sie sollten `/etc/security/access.conf` ebenfalls so verändern, dass ein Login aus der Ferne in ein administratives Konto nicht erlaubt wird. Auf diese Weise müssen Benutzer das Programm **Su** (oder **Sudo**) aufrufen, um Administratorenrechte zu bekommen, so dass es immer eine nachprüfbare Spur gibt.

Sie müssen die folgende Zeile zu Ihrer `/etc/security/access.conf` hinzufügen, in Debians Standardkonfigurationsdatei ist ein Beispiel auskommentiert:

```
-:wheel:ALL EXCEPT LOCAL
```

Vergessen Sie nicht, in `/etc/pam.d/` das `pam_access`-Module für jeden Dienst (oder die Standardkonfiguration) anzuschalten, wenn Sie wollen, dass Ihre Änderungen an `/etc/security/access.conf` berücksichtigt werden.

Den Benutzerzugang einschränken

Manchmal werden Sie denken, dass Sie einen Benutzer auf Ihrem System erstellen müssen, um einen bestimmten Dienst (Pop3-E-Mail-Server oder Ftp) anzubieten. Bevor Sie dies tun, denken Sie zuerst daran, dass die PAM-Implementierung in Debian GNU/Linux Ihnen erlaubt, Benutzer mit einer breiten Auswahl von externen Verzeichnisdiensten (Radius, LDAP etc.) zu überprüfen. Dies wird vom Paket libpam bewerkstelligt.

Wenn Sie einen Benutzer anlegen müssen und auf Ihr System aus der Ferne zugegriffen werden kann, beachten Sie, dass es Benutzern möglich sein wird, sich anzumelden. Sie können dies beheben, indem Sie diesen Benutzern Null (`/dev/null`) als Shell (sie muss in `/etc/shells` aufgelistet sein) zuweisen. Wenn Sie den Benutzern erlauben wollen, auf das System zuzugreifen, aber ihre Bewegungen einschränken wollen, können Sie `/bin/rbash` benutzen. Dies hat das gleiche Ergebnis, wie wenn Sie die Option `-r` der **Bash** (*RESTRICTED SHELL*, siehe bash(1)) verwendet hätten. Beachten Sie bitte, dass sogar mit einer beschränkten Shell ein Benutzer, der auf ein interaktives Programm zugreifen kann (das ihm erlaubt, eine Subshell auszuführen), diese Limitierung der Shell umgehen kann.

Debian bietet zurzeit in seiner Unstable-Veröffentlichung (und wird es vielleicht der nächsten Stable-Veröffentlichung hinzufügen) das Modul `pam_chroot` (in libpam-chroot) an. Eine Alternative hierzu ist es, die Dienste, die eine Fernanmeldung ermöglichen (**Ssh** und **Telnet**), in einer **Chroot**-Umgebung laufen zu lassen. [18]

Wenn Sie einschränken wollen, *wann* ein Benutzer auf das System zugreifen kann, müssen Sie `/etc/security/access.conf` an Ihre Bedürfnisse anpassen.

Informationen, wie man Benutzer, die auf das System mittels des Dienstes **Ssh** zugreifen, in eine **Chroot**-Umgebung einsperrt, wird in „Chroot-Umgebung für SSH“ beschrieben.

Überprüfen der Benutzer

Wenn Sie wirklich paranoid sind, sollten Sie eine systemweite Einrichtung verwenden, um zu überwachen, was die Benutzer auf Ihrem System tun. In diesem Abschnitt werden einige Tipps vorgestellt, wie Sie verschiedene Werkzeuge verwenden.

[18] libpam-chroot wurden noch nicht vollständig getestet. Es funktioniert mit **Login**, aber es dürfte nicht leicht sein, diese Umgebung für andere Programme einzurichten.

Überwachung von Ein- und Ausgabe mittels eines Skripts

Um sowohl die von den Benutzern ausgeführten Programme als auch deren Ergebnisse zu überwachen, können Sie den Befehl **script** verwenden. Sie können **script** nicht als eine Shell einsetzen (auch dann nicht, wenn Sie es zu `/etc/shells` hinzufügen). Aber Sie können in die Datei, welche den Startvorgang der Shell steuert, folgendes eintragen:

```
umask 077
exec script -q -a "/var/log/sessions/$USER"
```

Wenn Sie dies systemweit vornehmen, bedeutet dies natürlich, dass die Shell die weiteren persönlichen Startdateien nicht abarbeitet (weil die Shell von **script** überschrieben wird). Eine Alternative ist, dies in den Startdateien des Benutzers vorzunehmen (dann kann der Benutzer aber dies entfernen, vgl. dazu die Anmerkungen unten).

Sie müssen auch die Dateien im Überwachungsverzeichnis (im Beispiel `/var/log/sessions/`) so einrichten, dass die Benutzer in sie schreiben, sie aber nicht löschen können. Dies kann zum Beispiel bewerkstelligt werden, indem die Sitzungsdateien der Benutzer vorab erstellt und mit **chattr** auf *append-only* (nur anfügen) gesetzt werden.

Eine sinnvolle Alternative für Systemadministratoren, die auch Zeitinformationen enthält, ist:

```
umask 077
exec script -q -a "/var/log/sessions/$USER-`date +%Y%m%d`"
```

Die Chronikdatei der Shell benutzen

Wenn Sie auswerten wollen, was die Benutzer in die Shell eingeben (aber nicht was das Ergebnis ist), können Sie eine systemweite `/etc/profile` so einrichten, dass alle Befehle in einer Chronikdatei gespeichert werden. Die systemweite Einstellung muss so eingerichtet werden, dass Benutzer die Überwachungsfähigkeit nicht aus ihrer Shell entfernen können. Ob dies möglich ist, hängt von der Art der Shell ab. Sie müssen also sicherstellen, dass alle Benutzer eine Shell verwenden, die das unterstützt.

Für die Bash zum Beispiel könnte `/etc/profile` folgendermaßen aufgebaut werden [19]:

```
HISTFILE=~/.bash_history
HISTSIZE=10000
HISTFILESIZE=999999
# Verhindert, dass Benutzer Befehle eintragen,
# die in die Verlaufsdatei ignoriert werden
HISTIGNORE=""
HISTCONTROL=""
readonly HISTFILE
readonly HISTSIZE
readonly HISTFILESIZE
readonly HISTIGNORE
readonly HISTCONTROL
export HISTFILE HISTSIZE HISTFILESIZE HISTIGNORE HISTCONTROL
```

[19] Wenn HISTSIZE eine sehr große Zahl zugewiesen wird, kann dies bei einigen Shells zu Problemen führen, da der Verlauf für jede Sitzung eines Benutzers im Speicher abgelegt wird. Sie sind auf der sichereren Seite, wenn Sie HISTSIZE auf einen ausreichend großen Wert setzen und eine Kopie der Chronikdatei des Benutzers anlegen (falls Sie aus irgendwelchen Gründen den ganzen Verlauf von einem Benutzer benötigen).

Damit dies funktioniert, dürfen die Benutzer nur Informationen zur `.bash_history`-Datei hinzufügen. Sie müssen daher *zusätzlich* die Option *append-only* (nur anfügen) mittels des Programms **Chattr** für die `.bash_history` aller Benutzer setzen [20].

Beachten Sie, dass Sie obige Konfiguration auch in `.profile` des Benutzers eintragen können. Dann müssten Sie aber die Rechte korrekt vergeben, so dass der Benutzer daran gehindert ist, diese Datei zu verändern. Dies schließt ein, dass das Home-Verzeichnis des Benutzers diesem *nicht* gehört (sonst könnte er die Datei einfach löschen). Gleichzeitig müsste ihm ermöglicht werden, die Konfigurationsdatei `.profile` zu lesen und in `.bash_history` zu schreiben. Falls Sie diese Variante wählen wollen, wäre es auch gut, den Schalter *immutable* (unveränderbar) für `.profile` zu setzen (auch dazu verwenden Sie **Chattr**).

Vervollständigung der Benutzerüberwachung durch Accounting-Werkzeuge

Die vorherigen Beispiele stellen eine einfache Art dar, um die Überwachung von Benutzern einzurichten. Sie eignen sich aber nicht unbedingt für komplexe Systeme oder für solche, auf denen die Benutzer überhaupt keine (oder ausschließlich) Shells am Laufen haben. Sollte dies der Fall sein, schauen Sie sich das Paket acct an, das Werkzeuge zur Auswertung (accounting utilities) enthält. Diese werden alle Befehle, die ein Benutzer oder ein Prozess auf dem System ausführt, – auf die Kosten von Plattenplatz – aufzeichnen.

Wenn Sie diese Auswertung aktivieren, werden alle Informationen über Prozesse und Benutzer unter `/var/account/` gespeichert, genauer gesagt in `pacct`. Das Accounting-Paket enthält einige Werkzeuge (**Sa**, **Ac** und **Lastcomm**) zur Analyse dieser Daten.

Andere Methoden zur Benutzerüberwachung

If you are completely paranoid and want to audit every user's command, you could take **bash** source code, edit it and have it send all that the user typed into another file. Or have ttysnoop constantly monitor any new ttys [21] and dump the output into a file. Other useful program is snoopy (see also github: https://github.com/a2o/snoopy) which is a user-transparent program that hooks in as a library providing a wrapper around `execve()` calls, any command executed is logged to **syslogd** using the `authpriv` facility (usually stored at `/var/log/auth.log`).

Nachprüfung der Benutzerprofile

Wenn Sie *sehen* wollen, was Benutzer tatsächlich tun, wenn sie sich am System anmelden, können Sie die `wtmp`-Datenbank benutzen, die alle Anmeldeinformationen enthält. Diese Datei kann mit verschiedenen Werkzeugen weiterverarbeitet werden, unter ihnen **Sac**, das ein Profil für jeden Benutzer ausgeben kann und zeigt, in welchem Zeitfenster sie sich für gewöhnlich auf dem System anmelden.

Für den Fall, dass Sie Accounting aktiviert haben, können Sie auch die mitgelieferten Werkzeuge verwenden, um festzustellen, wann Benutzer auf das System zugreifen und was sie ausführen.

Umasks der Benutzer einstellen

Abhängig von Ihren Benutzerrichtlinien möchten Sie ändern, wie Benutzer Informationen gemeinsam benutzen können. Dabei geht es um die Standardrechte von neu erstellten Dateien.

Das Standardwert von `Umask` ist in Debian *022*. Das bedeutet, dass die Gruppe des Benutzers und alle anderen Benutzers auf dem System die Dateien (und Verzeichnisse) lesen und darauf zugreifen kann.

[20] Ohne das Append-Only-Flag wäre es den Benutzern möglich, den Inhalt des Verlaufs zu löschen, indem sie `> .bash_history` ausführen.
[21] Ttys are spawned for local logins and remote logins through ssh and telnet

Dieser Wert wird in der Standardkonfigurationsdatei `/etc/profile` gesetzt, die von allen Shells verwendet wird.

Wenn die Standardwerte von Debian für Ihr System zu großzügig sind, müssen Sie die Umask-Einstellungen für alle Shells ändern. Strengere Umask-Einstellungen sind 027 (kein Zugriff der Gruppe *other* auf neue Dateien, dazu zählen andere Benutzer auf dem System) oder 077 (kein Zugriff der Mitglieder der Gruppe des Benutzers). Debian erzeugt (standardmäßig[22]) für jeden Benutzer eine eigene Gruppe, so dass das einzige Gruppenmitglied der Benutzer selbst ist. Daher ergibt sich zwischen 027 und 077 kein Unterschied, da die Benutzergruppe nur den Benutzer selbst enthält.

Dies ändern Sie, indem Sie eine passende `Umask` für alle Benutzer einstellen. Dazu müssen Sie einen **umask**-Aufruf in den Konfigurationsdateien aller Shells einfügen: `/etc/profile` (wird von allen Shells beachtet, die kompatibel mit Bourne sind), `/etc/csh.cshrc`, `/etc/csh.login`, `/etc/zshrc` und wahrscheinlich noch ein paar andere (je nachdem, welche Shells Sie auf Ihrem System installiert haben). Sie können auch die `UMASK`-Einstellung in `/etc/login.defs` verändern. Von all diesen Dateien erlangt die letzte, die von der Shell geladen wird, Vorrang. Die Reihenfolge lautet: die Standard-Systemkonfiguration für die Shell des Benutzers (d.h. `/etc/profile` und andere systemweite Konfigurationsdateien), dann die Shell des Benutzers (seine `~/.profile`) und `~/.bash_profile` etc.). Allerdings können einige Shells mit dem *nologin*-Wert ausgeführt werden, was verhindern kann, dass einige dieser Dateien ausgewertet werden. Sehen Sie in der Handbuchseite Ihrer Shell für weitere Informationen nach.

Bei Anmeldungen, die von **Login** Gebrauch machen, erhält die UMASK-Festlegung in `/etc/login.defs` Vorrang vor allen anderen Einstellungen. Dieser Wert wird aber nicht von Anwendungen des Benutzers beachtet, die nicht **Login** verwenden, wie z.B. solche, die durch **Su**, **Cron** oder **Ssh** ausgeführt werden.

Vergessen Sie nicht, die Dateien unter `/etc/skel/` zu überprüfen und gegebenenfalls anzupassen, da dort die Standards für Benutzer festgelegt werden, die mit dem Befehl **adduser** erstellt werden. Standardmäßig enthalten die Dateien in Debian keinen Aufruf von **umask**. Wenn sich aber ein solcher in Konfigurationsdateien befindet, sind neue Benutzer eher geneigt, ihn ihren Bedürfnissen anzupassen.

Beachten Sie allerdings, dass ein Benutzer seine **Umask**-Einstellung ändern kann, wenn er es möchte, um sie großzügiger oder einschränkender zu machen, indem er seine Konfigurationsdateien verändert.

Das Paket libpam-umask passt die Standard-`Umask` eines Benutzers mit Hilfe von PAM an. Nachdem Sie das Paket installiert haben, tragen Sie Folgendes in `/etc/pam.d/common-session` ein:

```
session    optional     pam_umask.so umask=077
```

Zu guter Letzt sollte Sie in Betracht ziehen, die Standard-Umask von Root (022, wird in `/root/.bashrc` festgelegt) auf einen strengeren Wert zu verändern. Damit kann verhindert werden, dass der Systemadministrator als Root sensible Dateien in von allen lesbaren Verzeichnissen (wie z.B. `/tmp`) ablegt und sie so dem Durchschnittsbenutzer zugänglich macht.

Beschränken, was Benutzer sehen und worauf sie zugreifen können

FIXME: Inhalt benötigt. Aufzeigen der Folgen beim Upgraden, wenn die Paketrechte verändert werden, falls nicht **dpkg-statoverride** verwendet wird (übrigens sollte ein derartig paranoider Administrator seine Benutzer in eine **Chroot**-Umgebung einsperren).

[22] Wird in `/etc/adduser.conf` festgelegt (USERGROUPS=yes). Sie ändern dieses Verhalten, wenn Sie den Wert auf »no« setzen. Dies wird aber nicht empfohlen.

Wenn Sie einem Benutzer Zugriff auf das System mit einer Shell gewähren müssen, sollten Sie vorsichtig sein. Ein Benutzer kann normalerweise, wenn er sich nicht in einer streng abgeschirmten Umgebung befindet (z.B. in einem `Chroot`-Gefängnis), ziemlich viel Informationen über Ihr System sammeln. Darunter fallen:

- Einige Konfigurationsdateien unter `/etc`. Jedoch werden Debians Standardrechte für sensible Dateien (die zum Beispiel Passwörter enthalten könnten) den Zugriff auf kritische Informationen verhindern. Um zu sehen, auf welche Dateien nur der Root-Benutzer zugreifen kann, führen Sie zum Beispiel `find /etc -type f -a -perm 600 -a -uid 0` als Superuser aus.

  `find /etc -type f -a -perm 600 -a -uid 0`

  als Superuser aus.

- Ihre installierten Pakete. Indem entweder die Paketdatenbank und das Verzeichnis `/usr/share/doc/` angesehen wird oder indem versucht wird, dies durch Anschauen der auf Ihrem System installierten Programme und Bibliotheken zu raten.

- Einige Protokolle unter `/var/log`. Beachten Sie, dass auf einige Protokolle nur Root und die **adm**-Gruppe zugreifen kann (versuchen Sie

  `find /var/log -type f -a -perm 640`

) Manche sind sogar ausschließlich für Root verfügbar (sehen Sie sich

  `find /var/log -type f -a -perm`
  `600 -a -uid 0` an

).

Was kann ein Benutzer von Ihrem System sehen? Wahrscheinlich ziemlich viele Sachen, versuchen Sie mal Folgendes (und jetzt tief durchatmen):

```
find / -type f -a -perm +006 2>/dev/null
find / -type d -a -perm +007 2>/dev/null
```

Was Sie sehen, ist eine Liste von allen Dateien, die ein Benutzer *einsehen* kann, und von den Verzeichnissen, auf die er Zugriff hat.

Begrenzung des Zugangs zu Informationen anderer Benutzer

Wenn Sie immer noch Benutzern einen Shellzugang zur Verfügung stellen wollen, sollten Sie die Informationen begrenzen, die man über anderen Benutzern einholen kann. Benutzer mit einer Shell haben die Neigung, eine ziemlich große Anzahl von Dateien in ihrem $HOME zu erstellen: Mailboxen, persönliche Daten, Konfigurationen für X/GNOME/KDE-Anwendungen ...

Unter Debian wird jeder Benutzer mit einer zugehörigen Gruppe erstellt. Verschiedene Benutzer gehören dabei nie zur selben Gruppe. Folgendes ist das Standardverhalten: Wenn ein Benutzerkonto angelegt wird, wird auch eine Gruppe mit dem gleichen Namen erstellt. Dieser Gruppe wird der Benutzer zugewiesen. Damit wird die Idee einer allgemeinen *users*-Gruppe überflüssig, die es Benutzern erschweren könnte, Informationen vor anderen Benutzern zu verstecken.

Allerdings wird das `$HOME`-Verzeichnis der Benutzer mit 0755-Rechten (lesbar von der Gruppe, lesbar von der Welt) erstellt. Die Rechte für die Gruppe sind kein Thema, da nur der Benutzer zu dieser Gruppe gehört. Allerdings könnten die Rechte für die Welt ein Problem darstellen, wobei dies von Ihren lokalen Richtlinien abhängt.

Sie können dieses Verhalten so abändern, dass das Erstellen eines Benutzers andere Rechte für `$HOME` liefert. Um dieses Verhalten für *neue* Benutzer zu ändern, wenn sie erstellt werden, ändern Sie in der Konfigurationsdatei `/etc/adduser.conf` *DIR_MODE* auf 0750 (nicht lesbar für die Welt) ab.

Benutzer können immer noch Informationen austauschen, aber nicht mehr unmittelbar in ihrem `$HOME`-Verzeichnis, es sei denn, dass sie dessen Recht verändert haben.

Wenn Sie den Lesezugriff auf die Home-Verzeichnisse für die Welt verhindert, sollten Sie beachten, dass dann Benutzer ihre persönlichen Webseiten nicht unter `~/public_html` erstellen können, da der Webserver einen Teil des Pfads nicht lesen kann – und zwar das `$HOME`-Verzeichnis. Wenn Sie es Benutzern erlauben wollen, ihre HTML-Seiten in ihrem `~/public_html` zu veröffentlichen, sollten Sie *DIR_MODE* auf 0751 setzen. Das ermöglicht dem Webserver Zugriff auf das eigentliche `public_html`-Verzeichnis (welches selbst die Rechte 0755 haben sollte). So kann er den von den Benutzern veröffentlichten Inhalt anbieten. Natürlich sprechen wir hier nur über die Standardeinstellung. Benutzer können grundsätzlich die Rechte für ihre eigenen Dateien nach ihrem Gutdünken vergeben. Oder Sie können die Dinge, die für das Web bestimmt sind, in einem getrennten Ort ablegen, der kein Unterverzeichnis vom `$HOME`-Verzeichnis des Benutzers ist.

Erstellen von Benutzerpasswörtern

In vielen Fällen muss ein Administrator viele Benutzerkonten erstellen und alle mit Passwörtern ausstatten. Der Administrator könnte natürlich einfach als Passwort den Namen des Benutzerkontos vergeben. Dies wäre aber unter Sicherheitsgesichtspunkten nicht sehr klug. Ein besseres Vorgehen ist es, ein Programm zur Erzeugung von Passwörtern zu verwenden. Debian stellt die Pakete makepasswd, apg und pwgen zur Verfügung, die Programme liefern (deren Name ist der gleiche wie der des Pakets), die zu diesem Zweck verwendet werden können. **Makepasswd** erzeugt wirklich zufällige Passwörter, gibt also der Sicherheit gegenüber der Aussprechbarkeit den Vorzug. Dagegen versucht **pwgen**, bedeutungslose, aber aussprechbare Passwörter herzustellen (dies hängt natürlich auch von Ihrer Muttersprache ab). **Apg** liefert Algorithmen für beide Möglichkeiten (Es gibt auch eine Client/Server-Version dieses Programms. Diese befindet sich aber nicht im Debian-Paket).

Passwd erlaubt nur die interaktive Zuweisung von Passwörtern (da es direkt den tty-Zugang benutzt). Wenn Sie Passwörter ändern wollen, wenn Sie eine große Anzahl von Benutzern erstellen, können Sie diese unter der Verwendung von **adduser** mit der `--disabled-login`-Option erstellen, und danach **usermod** oder **chpasswd** [23] benutzen (beide Programme stammen aus dem passwd-Paket. Sie haben sie also schon installiert). Wenn Sie lieber eine Datei verwenden, die alle Informationen zur Erstellung von Benutzern als Batch-Prozess enthält, sind Sie vielleicht mit **newusers** besser dran.

Überprüfung der Benutzerpasswörter

Die Passwörter der Benutzer sind manchmal die *schwächste Stelle* der Sicherheit eines Systems. Das liegt daran, dass manche Benutzer schwache Passwörter für ihr Konto wählen (und je mehr Benutzer Zugang zum System haben, umso größer die Chance, dass das passiert). Selbst wenn Sie Überprüfungen mit dem PAM-Module cracklib und Grenzen für Passwörter einsetzen, wie in „Benutzerauthentifizierung: PAM“ beschrieben wird, ist es Benutzern immer noch möglich, schwache Passwörter zu verwenden. Da der Zugang der Benutzer auch den Zugang aus der Ferne (hoffentlich über **Ssh**) umfassen kann, ist es wichtig, dass das Erraten von Passwörtern für Angreifer aus der Ferne so schwierig wie möglich ist. Dies gilt insbesondere dann, wenn es ihnen gelungen sein sollte, Zugriff auf wichtigen Informationen wie den Benutzernamen oder sogar den Dateien `passwd` und `shadow` selbst zu bekommen.

[23] **Chpasswd** kann keine MD5-Passwörter erzeugen. Daher muss ihm das Passwort in verschlüsselter Form mit der Option

`-e`

übergeben werden.

Ein Systemadministrator muss bei einer großen Anzahl von Benutzern überprüfen, ob deren Passwörter mit den lokalen Sicherheitsrichtlinien in Einklang stehen. Und wie überprüft man das? Indem man versucht, sie wie ein Angreifer zu knacken, der Zugriff auf die gehashten Passwörter hat (also auf die Datei `/etc/shadow`).

An administrator can use john or crack (both are brute force password crackers) together with an appropriate wordlist to check users' passwords and take appropriate action when a weak password is detected. You can search for Debian GNU packages that contain word lists using **apt-cache search wordlist**, or visit some Internet wordlist sites.

Abmelden von untätigen Benutzern

Untätige (idle) Benutzer stellen für gewöhnlich ein Sicherheitsproblem dar. Ein Benutzer kann untätig sein, da er Mittagessen ist, oder weil eine Verbindung aus der Ferne hängen blieb und nicht wieder hergestellt wurde. Unabhängig von den Gründen können untätige Benutzer zu einer Kompromittierung führen:

- weil die Konsole des Benutzers vielleicht nicht verriegelt ist und damit ein Eindringling darauf zugreifen kann.
- weil ein Angreifer an eine schon beendete Netzwerkverbindung anknüpfen und Befehle an die Shell in der Ferne schicken kann (das ist ziemlich einfach, wenn die Shell in der Ferne, wie bei **Telnet**, nicht verschlüsselt ist).

In einige Systeme in der Ferne wurde sogar schon durch ein untätiges (und abgelöstes) **Screen** eingedrungen.

Die automatische Trennung von untätigen Benutzern ist gewöhnlich ein Teil der lokalen Sicherheitsrichtlinie, die durchgesetzt werden muss. Es gibt mehrere Arten, dies zu erreichen:

- Wenn die Shell des Benutzers die Bash ist, kann ein Systemadministrator **TMOUT** einen Standardwert zuweisen (vergleich bash(1)). Das hat zur Folge, dass die Shell automatisch untätige Benutzer aus der Ferne abmeldet. Beachten Sie, dass der Wert mit der Option **-o** gesetzt werden muss. Ansonsten ist es den Benutzern möglich, ihn zu verändern (oder zu löschen).
- Installieren Sie Timeoutd und konfigurieren Sie `/etc/timeouts` passend zu Ihren lokalen Sicherheitsrichtlinien. Der Daemon achtet auf untätige Benutzer und beendet entsprechend ihre Shells.
- Installieren Sie Autolog und richten Sie es so ein, dass es untätige Benutzer entfernt.

Vorzugswürdige Methoden sind die Daemonen **Timeoutd** oder **Autolog**, da letzten Endes die Benutzer ihre Standardshell ändern können oder zu einer anderen (unbeschränkten) Shell wechseln können, nachdem sie ihre Standardshell gestartet haben.

Die Nutzung von Tcpwrappers STOPP

TCP-Wrapper (Schutzumschläge für TCP) wurden entwickelt, als es noch keine echten Paketfilter gab, aber Zugangskontrollen notwendig waren. Trotzdem sind sie immer noch hoch interessant und nützlich. Ein TCP-Wrapper erlaubt Ihnen, einem Host oder einer Domain einen Dienst anzubieten oder zu verweigern, und standardmäßig Zugriff zu erlauben oder zu verweigern (das alles wird auf der Anwendungsebene durchgeführt). Wenn Sie mehr Informationen haben möchten, sehen Sie sich hosts_access(5) an.

Viele der unter Debian installierten Dienste

- werden entweder durch den TCP-Wrapper Service (`tcpd`) aufgerufen,
- oder wurden mit Unterstützung für libwrapper (Bibliothek für TCP-Wrapper) kompiliert.

Einerseits werden Sie bei manchen Diensten (einschließlich **telnet**, **ftp**, **netbios**, **swat** und **finger**), die in `/etc/inetd.conf` konfiguriert werden, sehen, dass die Konfigurationsdatei zuerst **/usr/sbin/tcpd** aufruft. Andererseits, selbst wenn ein Dienst nicht über den **inetd**-Superdaemon ausgeführt wird, kann die Unterstützung von TCP-Wrapper einkompiliert werden. Dienste, die unter Debian mit TCP-Wrappern kompiliert wurden, sind **ssh**, **portmap**, **in.talk**, **rpc.statd**, **rpc.mountd**, **gdm**, **oaf** (der GNOME-Aktivierungs-Daemon), **nessus** und viele andere.

Um herauszufinden, welche Pakete TCP-Wrapper benutzen[24], geben Sie Folgendes ein:

```
$ apt-cache rdepends libwrap0
```

Beachten Sie bitte Folgendes, wenn Sie **tcpchk** (ein sehr nützliches Programm zur Überprüfung der TCP-Wrapper-Konfiguration und -Syntax) laufen lassen. Wenn Sie Stand-Alone-Dienste (alleinstehende Dienste, also solche, die direkt mit der Wrapper-Bibliothek verbunden sind) der `host.deny`- oder `host.allow`-Datei hinzufügen, wird **tcpchk** Sie warnen, dass er sie nicht finden kann, da er sie nur in `/etc/inetd.conf` sucht (die Handbuchseite ist an dieser Stelle nicht sehr genau).

Jetzt kommt ein kleiner Trick und vielleicht die kleinste Alarmanlage zur Erkennung von Eindringlingen: Im Allgemeinen sollten Sie eine anständige Firewall als erste und TCP-Wrapper als zweite Verteidigungslinie haben. Der Trick besteht nun darin, ein `SPAWN`-Kommando [25] in `/etc/hosts.deny` einzutragen, das immer dann eine Mail an Root schickt, wenn ein Dienst abgewiesen wurde:

```
ALL: ALL: SPAWN ( \
  echo -e "\n\
  TCP Wrappers\: Verbindungsaufbau abgelehnt\n\
  Von\: $(uname -n)\n\
  Prozess\: %d (pid %p)\n\
  Benutzer\: %u\n\
  Host\: %c\n\
  Datum\: $(date)\n\
" | /usr/bin/mail -s "Verbindung zu %d blockiert" root) &
```

Achtung: Das obige Beispiel kann sehr leicht zu DoS (Denial of Service, Verbindungsaufbau abgelehnt) führen, indem man versucht, sehr viele Verbindungen in kurzer Zeit aufzubauen. Viele E-Mails bedeuten viel Dateiaktivität, die lediglich durch das Senden von ein paar Paketen erreicht wird.

Die Wichtigkeit von Protokollen und Alarmen

Es ist leicht einzusehen, dass die Behandlung von Protokollen und Alarmen eine wichtige Angelegenheit in einem sicheren System ist. Stellen Sie sich vor, ein System ist perfekt konfiguriert und zu 99% sicher. Wenn ein Angriff unter dieses 1% fällt, und es keine Sicherheitsmaßnahmen gibt, dies erstens zu erkennen und zweitens einen Alarm auszulösen, so ist das System überhaupt nicht sicher.

Debian GNU/Linux stellt Werkzeuge zur Verfügung, die die Analyse von Protokolldateien übernehmen. Am beachtenswertesten sind swatch [26], logcheck oder loganalysis (alle Pakete werden ein wenig Anpassung benötigen, um unnötige Dinge aus den Berichten zu entfernen). Wenn sich das System in Ihrer Nähe befindet, könnte es nützlich sein, das System-Protokoll auf einer virtuellen Konsole auszugeben. Die ist nützlich, da Sie so (auch von weiter weg oder im Vorbeigehen) sehen können, ob sich das System richtig

[24] Bei älteren Veröffentlichungen von Debian sollte Sie Folgendes ausführen:

```
$ apt-cache showpkg libwrap0 | egrep '^[[:space:]]' | sort -u | \ sed 's/,libwrap0$//;s/^[[:space:]]\+//'
```

[25] Beachten Sie hier die Schreibweise, da *spawn* nicht funktionieren wird.

[26] Es gibt darüber einen ziemlich guten Artikel von http://www.spitzner.net/swatch.html.

verhält. Debians `/etc/syslog.conf` wird mit einer auskommentierten Standardkonfiguration ausgeliefert. Um diese Ausgabe einzuschalten, entfernen Sie die Kommentarzeichen vor den entsprechenden Zeilen und starten **syslog** neu (**`/etc/init.d/syslogd restart`**):

```
daemon,mail.*;\
      news.=crit;news.=err;news.=notice;\
      *.=debug;*.=info;\
      *.=notice;*.=warn       /dev/tty8
```

Um die Protokolle farbig zu gestalten, sollten Sie einen Blick auf colorize, ccze oder glark werfen. Es gibt noch eine Menge über die Analyse von Protokollen zu sagen, das hier nicht behandelt werden kann. Eine gute Quelle für weitere Informationen sind Bücher wie http://books.google.com/books?id=UyktqN6Gn-WEC. In jedem Fall sind selbst automatische Werkzeuge dem besten Analysewerkzeug nicht gewachsen: Ihrem Gehirn.

Nutzung und Anpassung von logcheck

Das Paket **logcheck** ist in Debian auf drei Pakete verteilt: logcheck (das Hauptprogramm), logcheck-database (eine Datenbank regulärer Ausdrücke für das Programm) und logtail (gibt Protokollzeilen aus, die noch nicht gelesen wurden). Der Standard unter Debian (in `/etc/cron.d/logcheck`) ist, dass **logcheck** jede Stunde und nach jedem Neustart ausgeführt wird.

Wenn dieses Werkzeug in geeigneter Weise angepasst wurde, kann es sehr nützlich sein, um den Administrator zu alarmieren, wenn etwas ungewöhnliches auf dem System passiert. **Logcheck** kann vollständig angepasst werden, so dass es Mails über Ereignisse aus den Protokollen sendet, die Ihrer Aufmerksamkeit bedürfen. Die Standard-Installation umfasst Profile zum Ignorieren von Ereignissen und Verstößen gegen die Sicherheitsrichtlinie für drei unterschiedliche Einsatzbereiche (Workstation, Server und paranoid). Das Debian-Paket umfasst die Konfigurationsdatei `/etc/logcheck/logcheck.conf`, die vom Programm eingelesen wird, und die definiert, an welchen Benutzer die Testergebnisse geschickt werden sollen. Es stellt außerdem einen Weg für Pakete zur Verfügung, um neue Regeln in folgenden Verzeichnisses zu erstellen: `/etc/logcheck/cracking.d/_packagename_` `/etc/logcheck/violations.d/_packagename_`, `/etc/logcheck/violations.ignore.d/_packagename_`, `/etc/logcheck/ignore.d.paranoid/_packagename_`, `/etc/logcheck/ignore.d.server/_packagename_`, und `/etc/logcheck/ignore.d.workstation/_packagename_`. Leider benutzen das noch nicht viele Pakete. Wenn Sie ein Regelwerk entwickelt haben, dass für andere Benutzer nützlich sein könnte, schicken Sie bitte einen Fehlerbericht für das entsprechende Paket (als ein *wishlist*-Fehler). Mehr Informationen finden Sie unter `/usr/share/doc/logcheck/README.Debian`.

logcheck konfiguriert man am besten, indem man nach der Installation die Hauptkonfigurationsdatei `/etc/logcheck/logcheck.conf` bearbeitet. Verändern Sie den Benutzer, an den die Berichte geschickt werden (standardmäßig ist das Root). Außerdem sollten Sie auch den Schwellenwert für Berichte festlegen. logcheck-database hat drei Schwellenwerte mit steigender Ausführlichkeit: Workstation (Arbeitsplatz), Server und paranoid. »server« ist der Standardwert, »paranoid« wird nur für Hochsicherheitsmaschinen empfohlen, auf denen so wenig Dienste wie möglich laufen. »workstation« eignet sich für relativ geschützte, nicht kritische Maschinen. Wenn Sie neue Protokoll-Dateien hinzufügen wollen, müssen Sie diese nur zu `/etc/logcheck/logcheck.logfiles` hinzufügen. Es ist für die standardmäßige Syslog-Installation eingerichtet.

Wenn Sie dies geschafft haben, sollten Sie die nächsten Tage/Wochen/Monate die verschickten Mails überprüfen. Falls Sie Nachrichten finden, die Sie nicht erhalten wollen, fügen Sie die regulären Ausdrücke (regular expressions, vergleiche regex(7) und egrep(1)), die zu diesen Nachrichten passen, in `/etc/logcheck/ignore.d.reportlevel/local` ein. Versuchen Sie, dass der reguläre Ausdruck mit der gesamten Protokollzeile übereinstimmt. Details, wie man Regeln schreibt, finden Sie in `/usr/`

`share/doc/logcheck-database/README.logcheck-database.gz`. Das ist ein andauernder Prozess der Abstimmung. Wenn nur noch relevante Meldungen verschickt werden, können Sie davon ausgehen, dass dieser Prozess beendet ist. Beachten Sie, dass **logcheck**, selbst wenn er läuft, Ihnen keine Mail schickt, wenn er nichts Relevantes auf Ihrem System findet (so bekommen Sie höchstens eine Mail pro Woche, wenn Sie Glück haben).

Konfiguration, wohin Alarmmeldungen geschickt werden

Debian wird mit einer Standardkonfiguration für `Syslog` (in `/etc/syslog.conf`) ausgeliefert, so dass Meldungen je nach System in die passenden Dateien geschrieben werden. Das sollte Ihnen bereits bekannt sein. Falls nicht, werfen Sie einen Blick auf die Datei `syslog.conf` und deren Dokumentation. Wenn Sie ein sicheres System betreuen wollen, sollte Ihnen bekannt sein, wohin Protokoll-Meldungen geschickt werden, so dass sie nicht unbeachtet bleiben.

Zum Beispiel ist es für viele Produktiv-Systeme sinnvoll, Meldungen auch auf der Konsole auszugeben. Aber bei vielen solcher Systeme ist es wichtig, eine neue Maschine zu haben, die für die anderen als ein Loghost fungiert (d.h. sie empfängt die Protokolle aller anderen Systeme).

Sie sollten auch an Mails für Root denken, da viele Programme zur Sicherheitskontrolle (wie snort) ihre Alarme an die Mailbox von Root senden. Diese Mailbox zeigt normalerweise auf den ersten Benutzer, der auf dem System erstellt wurde (prüfen Sie dazu `/etc/aliases`). Sorgen Sie dafür, dass Roots Mails irgendwo hin geschickt werden, wo sie auch gelesen werden (lokal oder in der Ferne).

Es gibt noch andere Konten mit besonderen Funktionen und andere Aliase auf Ihrem System. Auf einem kleinen System ist es wohl am einfachsten, sicherzustellen, dass alle Aliase auf das Root-Konto verweisen, und dass Mails an Root in das persönliche Postfach des Systemadministrators weiter geleitet werden.

FIXME: It would be interesting to tell how a Debian system can send/receive SNMP traps related to security problems (jfs). Check: snmptrapfmt, snmp and snmpd.

Nutzen eines Loghosts

Ein Loghost ist ein Server, der die syslog-Daten über ein Netzwerk sammelt. Wenn eine Ihrer Maschinen geknackt wird, kann der Eindringling seine Spuren nicht verwischen, solange er den Loghost nicht ebenfalls geknackt hat. Demzufolge muss der Loghost besonders sicher sein. Aus einer Maschine einen Loghost zu machen, ist relativ einfach: Starten Sie den syslogd einfach mit

```
syslogd -r
```

und ein neuer Loghost ist geboren. Um dies unter Debian dauerhaft zu machen, editieren Sie `/etc/default/syslogd` und ändern Sie die Zeile

```
SYSLOGD=""
```

in

```
SYSLOGD="-r"
```

Als nächstes konfigurieren Sie die anderen Maschinen, so dass sie ihre Daten an den Loghost zu senden. Fügen Sie einen Eintrag, ähnlich dem Folgenden, zu der `/etc/syslog.conf` hinzu:

```
facility.level              @Ihr_Loghost
```

Schauen Sie in die Dokumentation, um zu erfahren, wodurch Sie *facility* und *level* ersetzen können; sie sollten nicht wörtlich übernommen werden. Wenn Sie alles in der Ferne mitprotokollieren wollen, schreiben Sie einfach:

```
*.*                         @Ihr_Loghost
```

in Ihre `syslog.conf`. Sowohl lokal als auch aus der Ferne mitzuprotokollieren, ist die beste Lösung (ein Angreifer könnte davon ausgehen, dass er seine Spuren verwischt hat, nachdem er die lokale Log-Datei gelöscht hat). Für weitere Informationen sehen Sie sich die Handbuchseiten syslog(3), syslogd(8) and syslog.conf(5) an.

Zugriffsrechte auf Protokolldateien

Es ist nicht nur wichtig zu entscheiden, wie Warnungen genutzt werden, sondern auch, wer hierauf Zugriff hat, d.h. wer Protokolldateien (falls Sie nicht einen Loghost verwenden) lesen oder verändern kann. Sicherheitsalarme, die ein Angreifer verändern oder abschalten kann, sind im Falle eines Eindringens nicht viel wert. Außerdem sollten Sie berücksichtigen, dass Protokolldateien einem Eindringling ziemlich viel Informationen über Ihr System verraten, wenn er auf sie Zugriff hat.

Einige Zugriffsrechte auf Protokolldateien sind nach der Installation nicht gerade perfekt (aber das hängt natürlich von Ihrer lokalen Sicherheitsrichtlinie ab). Zuerst einmal müssen `/var/log/lastlog` und `/var/log/faillog` nicht für normale Benutzer lesbar sein. In der Datei `lastlog` können Sie sehen, wer sich zuletzt angemeldet hat. In `faillog` befindet sich eine Zusammenfassung fehlgeschlagener Anmeldeversuche. Der Autor empfiehlt, die Rechte von beiden auf 660 zu setzen (mit **chmod 660**). Werfen Sie einen kurzen Blick auf Ihre Protokolldateien und entscheiden Sie sehr vorsichtig, welche Protokolldateien Sie les- oder schreibbar für einen Benutzer mit einer anderen UID als 0 und einer anderen Gruppe als »adm« oder »root« machen. Sie können dies sehr leicht auf Ihrem System überprüfen:

```
#  find /var/log -type f -exec ls -l {} \; | cut -c 17-35 |sort -u
(überprüfen, welchen Benutzern die Dateien unter /var/log gehören)
#  find /var/log -type f -exec ls -l {} \; | cut -c 26-34 |sort -u
(überprüfen, welchen Gruppen die Dateien unter /var/log gehören)
# find /var/log -perm +004
(Dateien, die von jedem Benutzer gelesen werden können)
#  find /var/log \! -group root \! -group adm -exec ls -ld {} \;
(Dateien, die nicht der Gruppe root oder adm gehören)
```

Um anzupassen, wie neue Protokolldateien erstellt werden, müssen Sie wahrscheinlich das Programm anpassen, das sie erstellt. Wenn die Protokolldateien ausgewechselt werden, können Sie das Verhalten der Erstellung und Auswechslung anpassen.

Den Kernel patchen

Debian GNU/Linux stellt verschiedene Patches für den Linux-Kernel zur Verfügung, welche die Sicherheit erhöhen:

- Erkennung von Eindringlingen für Linux (http://www.lids.org, enthalten im Paket lids-2.2.19). Dieser Kernelpatch erleichtert Ihnen, Ihr Linuxsystem abzuhärten, indem er Ihnen ermöglicht, Prozesse einzuschränken, zu verstecken und zu schützten, sogar vor Root. Er führt Fähigkeiten für eine zwingende Zugangskontrolle ein.

- http://trustees.sourceforge.net/ (im Paket trustees). Dieser Patch fügt ein ordentliches, fortgeschrittenes Rechtemanagement Ihrem Linux-Kernel hinzu. Besondere Objekte, die »trustees« (Treuhänder) genannt werden, sind mit jeder Datei oder Verzeichnis verbunden. Sie werden im Speicher des Kernels abgelegt und erlauben so eine schnelle Abfrage aller Rechte.

- NSA Enhanced Linux (in package selinux). Backports of the SElinux-enabled packages are available at https://salsa.debian.org/selinux-team. More information available at SElinux in Debian Wiki page [http://wiki.debian.org/SELinux], at Manoj Srivastava's [http://www.golden-gryphon.com/software/security/selinux.xhtml] and Russell Cookers's [http://www.coker.com.au/selinux/] SElinux websites.

- Der http://people.redhat.com/mingo/exec-shield/ aus dem Paket kernel-patch-exec-shield. Dieser Patch schützt vor einigen Pufferüberläufen (stack smashing attacks).

- The Grsecurity patch [http://www.grsecurity.net/], provided by the kernel-patch-2.4-grsecurity and kernel-patch-grsecurity2 packages [27] implements Mandatory Access Control through RBAC, provides buffer overflow protection through PaX, ACLs, network randomness (to make OS fingerprinting more difficult) and many more features [http://www.grsecurity.net/features.php].

- kernel-patch-adamantix bietet die Patches an, die für die Debian-Distribution http://www.adamantix.org/ entwickelt wurden. Dieser Patch für den Kernel 2.4.x führt einige Sicherheitsfähigkeiten wie nichtausführbaren Speicher durch den Einsatz von http://pageexec.virtualave.net/ und Mandatory Access Control auf Grundlage von http://www.rsbac.org/ ein. Andere Features sind http://www.vanheusden.com/Linux/sp/, ein mit AES verschlüsseltes Loop-Gerät, Unterstützung von MPPE und eine Zurückportierung von IPSEC v2.6.

- cryptoloop-source: Dieser Patch erlaubt Ihnen, die Fähigkeiten der Crypto-API des Kernels zu verwenden, um verschlüsselte Dateisysteme mit dem Loopback-Gerät zu erstellen.

- Kernel-Unterstützung von IPSEC (im Paket kernel-patch-openswan). Wenn Sie das IPsec-Protokoll mit Linux verwenden wollen, benötigen Sie diesen Patch. Damit können Sie ziemlich leicht VPNs erstellen, sogar mit Windows-Rechnern, da IPsec ein verbreiteter Standard ist. IPsec-Fähigkeiten wurden in den Entwicklungskernel 2.5 eingefügt, so dass dieses Feature standardmäßig im zukünftigen Kernel 2.6 enthalten sein wird. Homepage: http://www.openswan.org. *FIXME*: Der neuste Kernel 2.4 in Debian enthält eine Rückeinbindung des IPSEC-Codes aus 2.5. Kommentar dazu.

Die folgenden Sicherheitspatches für den Kernel sind nur noch für alte Kernelversionen in Woody verfügbar und werden nicht mehr weiterentwickelt:

- http://acl.bestbits.at/ (ACLs, Listen zur Zugangskontrolle) für Linux im Paket kernel-patch-acl. Dieser Kernelpatch stellt Listen zur Zugangskontrolle zur Verfügung. Das ist eine fortgeschrittene Methode, um den Zugang zu Dateien einzuschränken. Es ermöglicht Ihnen, den Zugang zu Dateien und Verzeichnisses fein abzustimmen.

- Der Patch für den Linux-Kernel http://www.openwall.com/linux/ von Solar Designer, der im Paket kernel-patch-2.2.18-openwall enthalten ist. Er enthält eine nützliche Anzahl von Beschränkungen des

[27] Notice that this patch conflicts with patches already included in Debian's 2.4 kernel source package. You will need to use the stock vanilla kernel. You can do this with the following steps:

```
# apt-get install kernel-source-2.4.22 kernel-patch-debian-2.4.22
# tar xjf /usr/src/kernel-source-2.4.22.tar.bz2
# cd kernel-source-2.4.22
# /usr/src/kernel-patches/all/2.4.22/unpatch/debian
```

For more information see http://bugs.debian.org/194225, http://bugs.debian.org/199519, http://bugs.debian.org/206458, http://bugs.debian.org/203759, http://bugs.debian.org/204424, http://bugs.debian.org/210762, http://bugs.debian.org/211213, and the http://lists.debian.org/debian-devel/2003/09/msg01133.html

Kernels wie eingeschränkte Verweise, FIFOs in `/tmp`, ein begrenztes `/proc`-Dateisystem, besondere Handhabung von Dateideskriptoren, einen nichtausführbaren Bereich des Stapelspeichers des Benutzers und andere Fähigkeiten. Hinweis: Dieser Patch ist nur auf die Kernelversion 2.2 anwendbar, für 2.4 werden von Solar keine Pakete angeboten.

- kernel-patch-int. Auch dieser Patch fügt kryptografische Fähigkeiten zum Linux-Kernel hinzu. Er war bis zu den Debian-Releases bis Potato nützlich. Er funktioniert nicht mehr mit Woody. Falls Sie Sarge oder eine neuere Version verwenden, sollten Sie einen aktuelleren Kernel einsetzen, in dem diese Features bereits enthalten sind.

Wie auch immer, einige Patches werden von Debian noch nicht zur Verfügung gestellt. Wenn Sie denken, dass manche von ihnen hinzugefügt werden sollten, fragen Sie danach auf https://www.debian.org/devel/wnpp/index.de.html.

Schutz vor Pufferüberläufen

Pufferüberlauf (buffer overflow) wird eine verbreitete Art von Angriffen auf Software [28] genannt, welche die unzureichende Überprüfung von Eingabegrenzen ausnutzen (ein Programmierfehler, der häufig bei der Programmiersprache C auftritt), um durch Programmeingaben Befehle auf der Maschine auszuführen. Diese Attacken über Server, die auf Verbindungen warten, oder über lokal installierte Software, die einem Benutzer größere Privilegien gewährt (`setuid` oder `setgid`) kann zu einem kompromittierten System führen.

Es gibt hauptsächlich vier Methoden, um sich gegen Pufferüberläufe zu schützen:

- Patchen Sie den Kernel, um das Ausführen des Stapelspeichers zu verhindern. Sie können entweder Exec-Shield, OpenWall oder PaX (ist in den Grsecurity- und Adamantixpatches enthalten) verwenden.

- Verbessern Sie den Quellcode, indem Sie Werkzeuge einsetzen, die Teile finden, die zu dieser Verwundbarkeit führen könnten.

- Übersetzen Sie den Quellcode neu, um vernünftige Prüfungen einzuführen, um Überläufe zu verhindern. Benutzen Sie dazu den http://www.research.ibm.com/trl/projects/security/ssp/ Patch für GCC (der von http://www.adamantix.org verwendet wird).

Debian GNU/Linux liefert bis einschließlich der Veröffentlichung 3.0 Software, um alle diese Methoden bis auf den Schutz bei der Übersetzung des Quellcodes (das wurde aber schon in http://bugs.debian.org/213994 nachgefragt) zu implementieren.

Beachten Sie, dass selbst wenn Debian einen Compiler zur Verfügung stellen würde, der Schutz vor Stapel- und Pufferüberläufen bieten würde, so doch alle Pakete neu übersetzt werden müssten, um diese Eigenschaft einzuführen. Tatsächlich ist das die Aufgabe der Distribution Adamantix (unter anderen Fähigkeiten). Die Auswirkungen dieses neuen Features auf die Stabilität der Software muss aber noch ermittelt werden (einige Programme und einige Prozessoren werden vielleicht deswegen nicht mehr funktionieren).

Seien Sie auf jeden Fall gewarnt, dass selbst diese Umgehungen des Problems nicht vor Pufferüberläufen schützen können, da es Möglichkeiten gibt, diese zu überlisten, wie in http://packetstorm.linuxsecurity.com/mag/phrack/phrack58.tar.gz des phrack-Magazins oder in COREs Advisory http://online.securityfocus.com/archive/1/269246 beschrieben.

Wenn Sie Ihren Schutz gegen Pufferüberläufe (unabhängig von der gewählten Methode) testen wollen, können Sie paxtest installieren und die angebotenen Tests laufen lassen.

[28] Sie sind in der Tat so verbreitet, dass sie die Grundlage für 20% aller gemeldeten Sicherheitsmängel pro Jahr darstellen, wie von http://icat.nist.gov/icat.cfm?function=statistics herausgefunden wurde.

Kernelpatch zum Schutz vor Pufferüberläufen

Ein Kernelpatch, der Schutz vor Pufferüberläufen bietet, ist der Openwall-Patch, der diese im Linux-Kernel 2.2 verhindern soll. Für 2.4 oder neuere Kernel müssen Sie die Umsetzung von Exec-Shield oder die von PaX (ist im Grsecurity-Patch kernel-patch-2.4-grsecurity und im Adamantix-Patch kernel-patch-adamantix enthalten) benutzen. Für weitere Informationen zum Einsatz dieser Patches lesen Sie „Den Kernel patchen“.

Prüfprogramme für Pufferüberläufe

Zur Nutzung von Werkzeugen zum Aufspüren von Pufferüberläufen benötigen Sie in jedem Fall Programmiererfahrung, um den Quellcode zu reparieren (und neu zu kompilieren). Debian stellt beispielsweise bfbtester (einen Überlauftester, der Programme per Brute-Force (durch Testen aller Möglichkeiten) nach Überläufen der Kommandozeile und von Umgebungsvariablen durchtestet) bereit. Andere interessante Pakete sind auch rats, pscan, flawfinder und splint.

Sichere Übertragung von Dateien

Während der normalen Systemadministration müssen Sie immer mal wieder Dateien auf Ihr System spielen oder von diesem holen. Auf sichere Art und Weise Dateien von einem Host zu einem anderen zu kopieren, wird durch die Benutzung des Paketes ssh erreicht. Eine andere Möglichkeit ist die Nutzung von ftpd-ssl, einem ftp-Server der *Secure Socket Layer* benutzt, um Übertragungen zu verschlüsseln.

Jede dieser Methoden benötigt natürlich einen speziellen Client. Debian stellt Ihnen solche zur Verfügung, zum Beispiel enthält das Paket ssh das Programm **scp**. Es arbeitet wie **rcp**, aber ist komplett verschlüsselt, so dass die *bösen Jungs* noch nicht einmal herausbekommen können, WAS Sie kopieren. Passend zu dem Server gibt es auch ein ftp-ssl Client-Paket. Sie können Clients für diese Software sogar für andere (nicht-UNIXoide) Betriebssysteme finden. **putty** und **winscp** stellen eine secure-copy-Implementierung für jede Version von Microsoft-Betriebssystemen zur Verfügung.

Beachten Sie, dass die Verwendung von **scp** den Benutzern Zugang zum gesamten Dateisystem ermöglicht, es sei denn, dass es in eine **chroot**-Umgebung eingesperrt ist, wie es in „SSH in ein Chroot-Gefngnis einsperren“ beschrieben wird. Wahrscheinlich sogar leichter (abhängig vom verwendeten Daemon) kann auch der FTP in eine **chroot**-Umgebung eingesperrt werden. Das wird in „Absichern von FTP“ beschrieben. Falls Sie sich sorgen, dass Benutzer Ihre lokalen Dateien durchsehen, und Sie verschlüsselte Kommunikation wünschen, können Sie einen FTP-Daemon mit Unterstützung für SSL einrichten oder FTP mit Klartext und VPN verbinden (siehe „Virtual Private Networks (virtuelle private Netzwerke)“.

Einschränkung und Kontrolle des Dateisystems

Benutzung von Quotas

Es ist wichtig, eine gute Quota-Regelung zu haben, da es die Benutzer daran hindert, die Festplatten zu füllen.

Sie können zwei Arten von Quota-Systemen benutzen: Benutzer-Quota und Gruppen-Quota. Wie Sie sich sicher denken können, begrenzt ein User-Quota den Plattenplatz, den ein Benutzer belegen kann, und ein Gruppen-Quota macht dasselbe für Gruppen. Beachten Sie dies, wenn Sie die Größe der Quotas festlegen.

Es gibt ein paar wichtige Punkte, die Sie erwägen sollten, wenn Sie ein Quota-System aufsetzen:

- Halten Sie die Quotas klein genug, so dass die Benutzer Ihren Festplattenplatz nicht aufzehren können.
- Halten Sie die Quotas groß genug, so dass Benutzer sich nicht beschweren oder dass Ihr Mail-Quota Sie daran hindert, nach einer Weile Mails anzunehmen.
- Nutzen Sie Quotas auf allen Bereichen, die Benutzer beschreiben können, auf `/home` ebenso wie auf `/tmp`.

Für jede Partition und jedes Verzeichnis, auf das Benutzer Schreibzugriff haben, sollte ein Quota eingerichtet werden. Berechnen Sie eine sinnvolle Quota-Größe, die Benutzerfreundlichkeit und Sicherheit kombiniert, und weisen Sie diese zu.

Sie wollen also Quotas benutzen. Zuerst müssen Sie prüfen, ob Ihr Kernel Quota unterstützt. Wenn nicht, müssen Sie ihn neu kompilieren. Prüfen Sie anschließend, ob das Paket quota installiert ist. Wenn nicht, installieren Sie es.

Um Quota für die entsprechenden Dateisysteme einzuschalten, müssen Sie nur die Einstellung `defaults` in Ihrer `/etc/fstab` zu `defaults,usrquota` ändern. Wenn Sie Gruppen-Quotas benötigen, ersetzen Sie `usrquota` durch `grpquota`. Sie können auch beides verwenden. Erstellen Sie dann leere `quota.user` und `quota.group` in den Hauptverzeichnissen der Dateisysteme, auf denen Sie Quotas einführen möchten (d.h.

```
touch
/home/quota.user /home/quota.group
```

für das Dateisystem `/home`).

Starten Sie quota neu, indem Sie **/etc/init.d/quota stop;/etc/init.d/quota start** ausführen

```
/etc/init.d/quota stop;/etc/init.d/quota
        start
```

. Nun sollte quota laufen und die Größen können festgelegt werden.

Bearbeiten der Quotas eines bestimmten Benutzer wird mit

edquota -u <user> gemacht

. Gruppen-Quotas können mit

edquota -g <group> geändert werden

. Setzen Sie dann die weiche und die harte Grenze und inode-Quotas, falls Sie es benötigen.

Mehr Informationen über Quotas finden Sie im Handbuch von quot und im Mini-Howto von quota (`/usr/share/doc/HOWTO/de-html/mini/DE-Quota-HOWTO.html`). Sie sollten auch einen Blick auf `pam_limits.so` werfen.

Die für das ext2-Dateisystem spezifischen Attribute (chattr/lsattr)

Zusätzlich zu den normalen Unix-Rechten bieten die ext2- und ext3-Dateisysteme eine Anzahl von besonderen Attributen, die Ihnen mehr Kontrolle über die Dateien auf Ihrem System erlauben. Im Gegensatz zu den gewöhnlichen Rechten werden diese Attribute nicht vom gebräuchlichen Befehl **ls -l** angezeigt und können auch nicht mit **chmod** geändert werden. Um sie zu verwalten, brauchen Sie zwei weitere

Programme, nämlich **lsattr** und **chattr** (im Paket e2fsprogs). Beachten Sie, dass das bedeutet, dass diese Attribute normalerweise bei einem Backup des Systems nicht gespeichert werden. Wenn Sie also eines verändern, könnte es sich lohnen, die aufeinander folgenden **chattr**-Befehle in einem Skript zu speichern, damit Sie sie später wieder zuweisen können, falls Sie ein Backup zurückspielen müssen.

Unter allen Attributen werden die zwei, die für die Erhöhung der Sicherheit am bedeutendsten sind, mit den Buchstaben »i« und »a« bezeichnet. Sie können nur vom Superuser vergeben (oder entfernt) werden:

- Das Attribut »i« (»immutable«, unveränderlich): Eine Datei mit diesem Attribut kann weder verändert noch gelöscht oder umbenannt werden, nicht einmal vom Superuser. Auch ein Link auf sie kann nicht angelegt werden.

- Das Attribut »a« (»append«, anfügen): Dieses Attribut hat den gleichen Effekt wie das Attribut immutable, allerdings mit der Ausnahme, dass Sie immer noch die Datei im Anfügen-Modus öffnen können. Das bedeutet, dass Sie ihr immer noch Inhalt hinzufügen, aber den vorhanden Inhalt nicht verändern können. Dieses Attribut ist besonders für die Protokolldateien nützlich, die unter `/var/log/` gespeichert werden. Beachten Sie aber, dass sie durch Log-Rotations-Skripte manchmal verschoben werden.

Diese Attribute können auch für Verzeichnisse vergeben werden. In diesem Fall ist es jedem unmöglich, den Inhalt des Verzeichnisses zu verändern, also beispielsweise eine Datei umzubenennen oder zu löschen. Wenn das append-Attribut einem Verzeichnis zugewiesen wird, können nur noch Dateien erstellt werden.

Es ist leicht einzusehen, wie das Attribut »a« die Sicherheit verbessert, indem es Programmen, die nicht vom Superuser ausgeführt werden, die Fähigkeit einräumt, Daten hinzuzufügen, aber verhindert, dass älterer Inhalt verändert wird. Dem gegenüber erscheint das Attribut »i« uninteressanter. Schließlich kann der Superuser ja schon die normalen Unix-Rechte verwenden, um den Zugang zu Dateien einzuschränken. Und ein Angreifer, der Zugang zum Konto des Superusers hat, kann immer das Programm **chattr** benutzen, um die Attribute zu entfernen. Ein solcher Eindringlich ist vielleicht zunächst verwirrt, wenn er feststellt, dass er eine Datei nicht löschen kann. Aber Sie sollten nicht davon ausgehen, dass er blind ist – immerhin hat er es geschafft, in Ihr System einzudringen! Einige Handbücher (einschließlich früherer Versionen dieses Dokuments) empfehlen, einfach die Programme **chattr** und **lsattr** vom System zu entfernen, um die Sicherheit zu erhöhen. Aber diese Strategie, die auch als »security by obscurity« (Sicherheit durch Verschleierung) bekannt ist, sollte unter allen Umständen vermieden werden, da sie ein falsches Gefühl von Sicherheit vermittelt.

Dieses Problem lösen Sie auf sichere Art und Weise, indem Sie die Fähigkeiten des Linux-Kernel verwenden, wie es in „Proaktive Verteidigung“ beschrieben wird. Die hier interessante Fähigkeit heißt `CAP_LINUX_IMMUTABLE`: Wenn Sie es vom Satz der Fähigkeiten entfernen (indem Sie zum Beispiel den Befehl `lcap CAP_LINUX_IMMUTABLE` verwenden, ist es nicht mehr möglich, irgendwelche »a« oder »i« Attribute auf Ihrem System zu verändern, auch nicht durch den Superuser! Ein umfassende Strategie könnte also folgendermaßen aussehen:

- Vergeben Sie die Attribute »a« und »i« an von Ihnen gewünschte Dateien.

- Fügen Sie den Befehl **lcap CAP_LINUX_IMMUTABLE** einem der Skripten, die den Start des Systems steuern (startup scripts), hinzu.

- Setzen Sie das Attribut »i« für dieses Skript, andere Startdateien und auch das Programm **lcap** selbst.

- Führen Sie den oben genannten Befehl per Hand aus (oder starten Sie Ihr System neu, um sicherzustellen, dass alles wie gewünscht funktioniert).

Da nun die Fähigkeit von dem System entfernt wurde, kann ein Eindringling keine Attribute der geschützten Dateien ändern und daher diese nicht verändern oder löschen. Wenn er einen Neustart der Maschine erzwingt (was der einzige Weg ist, die Fähigkeiten wieder herzustellen), wird dies leicht zu bemerken

sein. Außerdem werden die Fähigkeiten bei einem Neustart sofort wieder entfernt werden. Die einzige Möglichkeit, eine geschützte Datei zu ändern, ist, das System im Single-User-Modus zu starten oder ein anderes Bootmedium zu verwenden. Beides erfordert physischen Zugang zur Maschine!

Prüfung der Integrität des Dateisystems

Sind Sie sich sicher, dass `/bin/login` auf Ihrer Festplatte immer noch dasselbe Programm ist, das Sie vor ein paar Monaten installiert haben? Was wäre, wenn es sich um eine gehackte Version handelt, die eingegebene Passwörter in einer versteckten Datei ablegt oder sie als Klartext im ganzen Internet herummailt?

Die einzige Methode, um einen gewissen Schutz dafür zu haben, ist es, die Dateien jede(n) Stunde/Tag/Monat (ich ziehe täglich vor) zu prüfen, indem man deren aktuelle und alte MD5-Summe vergleicht. Zwei unterschiedliche Dateien können keine gleichen MD5-Summen haben (die MD5-Summe umfasst 128 Bits, so ist die Wahrscheinlichkeit, dass zwei unterschiedliche Dateien eine gleiche MD5-Summe haben etwa 1 zu 3,4e3803). So sind Sie sicher, solange niemand den Algorithmus gehackt hat, der die MD5-Summen auf Ihrer Maschine erstellt. Dies ist, nun ja, extrem schwer und sehr unwahrscheinlich. Sie sollten diese Überprüfung Ihrer Programme als sehr wichtig ansehen.

Weit verbreitete Werkzeuge hierfür sind sxid, aide (Advanced Intrusion Detection Environment, fortgeschrittene Umgebung zur Erkennung von Eindringlingen), tripwire, integrit und samhain. Das Installieren von **debsums** wird Ihnen helfen, die Integrität des Dateisystems zu überprüfen, indem Sie die MD5-Summen jeder Datei gegen die MD5-Summe aus dem Debian-Archiv-Paket vergleichen. Seien Sie aber gewarnt, dass diese Dateien sehr leicht von einem Angreifer geändert werden können. Außerdem stellen nicht alle Pakete MD5-Summen für die in ihnen enthaltenen Programme zur Verfügung. Weitere Informationen finden Sie unter „Regelmäßiges Überprüfung der Integrität“ und „Einen Schnappschuss des Systems erstellen“.

Sie benutzen vielleicht **locate**, um das gesamte Dateisystem zu indizieren. Wenn das so ist, sollten Sie die Auswirkungen davon berücksichtigen. Das Debianpaket findutils enthält **locate**, das als Benutzer nobody läuft. Daher indiziert es nur Dateien, die von jedermann eingesehen werden können. Wenn Sie dieses Verhalten verändern, werden allerdings alle Orte von Dateien für alle Benutzer sichtbar. Wenn Sie das gesamte Dateisystem indizieren wollen (und nicht nur die Stückchen, die der Benutzer nobody sehen kann), können Sie **locate** durch das Paket slocate ersetzen. slocate wird als eine um Sicherheit erweiterte Version von GNU locate bezeichnet, hat aber tatsächlich weitere Funktionen zum Auffinden von Dateien. Wenn Sie **slocate** benutzen, sieht ein Benutzer nur Dateien, auf die er auch Zugriff hat, während Sie Dateien und Verzeichnisse des gesamten Systems ausschließen können. Das Paket slocate führt seinen Aktualisierungsprozess mit höheren Rechten aus als locate. Außerdem indiziert es jede Datei. Benutzern wird es dadurch ermöglicht, schnell nach jeder Datei zu suchen, die sie sehen können. **slocate** zeigt ihnen keine neuen Dateien an; es filtert die Ausgabe auf Grundlage der UID.

Sie sollten auch bsign oder elfsign einsetzen. elfsign bietet die Möglichkeit, digitale Signaturen an ELF-Binaries anzufügen und diese Signaturen zu überprüfen. Die aktuelle Fassung verwendet PKI, um die Checksummen der Binaries zu signieren. Dies hat den Vorteil, dass festgestellt werden kann, ob das Binary verändert wurde und wer es erstellt hat. bsign verwendet GPG, elfsign benutzt PKI-(X.509)-Zertifikate (OpenSSL).

Aufsetzen einer Überprüfung von setuid

Das Debian-Paket checksecurity enthält einen **Cron**-Job, der täglich in `/etc/cron.daily/checksecurity` ausgeführt wird. [29]. Dieser **Cron**-Job führt das Skript **/usr/sbin/checksecurity** aus, das Informationen über Änderungen sichert.

[29] In älteren Veröffentlichungen war checksecurity in cron integriert und die Datei hieß `/etc/cron.daily/standard`.

Das Standardverhalten sendet diese Informationen nicht an den Superuser. Stattdessen erstellt es eine tägliche Kopie dieser Änderungen unter `/var/log/setuid.changes`. Sie sollten die Variable MAILTO (in `/etc/checksecurity.conf`) auf »root« setzen, damit diese Informationen an ihn gemailt werden. Sehen Sie sich auch checksecurity(8) für weitere Konfigurations-Informationen an.

Absicherung des Netzwerkzugangs

FIXME: mehr (für Debian spezifischer) Inhalt benötigt

Konfiguration der Netzwerkfähigkeiten des Kernels

Many features of the kernel can be modified while running by echoing something into the `/proc` file system or by using **sysctl**. By entering **/sbin/sysctl** `-A` you can see what you can configure and what the options are, and it can be modified running

`/sbin/sysctl -w variable=value`

(vergleiche sysctl(8)) Nur in seltenen Fällen müssen Sie hier etwas bearbeiten. Aber auch hier können Sie die Sicherheit erhöhen. Zum Beispiel:

```
net/ipv4/icmp_echo_ignore_broadcasts = 1
```

Dies ist ein *Windows-Emulator*, weil es sich wie Windows bei Rundrufen (Broadcast-Ping) verhält, wenn es auf 1 gesetzt wird. Das bedeutet, dass ICMP-Echo-Anfragen, die an die Rundrufadresse geschickt werden, ignoriert werden. Anderenfalls macht es gar nichts.

Falls Sie verhindern wollen, dass Ihr System auf ICMP-Echo-Anfragen antwortet, müssen Sie nur diese Konfigurationsoption anschalten:

```
net/ipv4/icmp_echo_ignore_all = 1
```

Verwenden Sie Folgendes, um Pakete mit unmöglichen Adressen (erzeugt durch falsche Routen) in Ihrem Netzwerk zu protokollieren:

```
/proc/sys/net/ipv4/conf/all/log_martians = 1
```

Für weiterführende Informationen, welche Sachen mit `/proc/sys/net/ipv4/*` angestellt werden können, sollten Sie `/usr/src/linux/Documentation/filesystems/proc.txt` lesen. Alle Optionen werden gründlich in `/usr/src/linux/Documentation/networking/ip-sysctl.txt` [30] beschrieben.

Konfiguration von Syncookies

Diese Option ist ein zweischneidiges Schwert. Auf der einen Seite schützt es Ihr System vor dem Überfluten mit syn-Paketen. Auf der anderen Seite verletzt es definierte Standards (RFCs).

```
net/ipv4/tcp_syncookies = 1
```

[30] In Debian kopiert das Paket kernel-source-`version` die Kernelquellen nach `/usr/src/kernel-source-version.tar.bz2`. Ersetzen Sie einfach `version` mit der installierten Kernelversion.

Wenn Sie das dauerhaft für den Kernel festlegen wollen, müssen Sie in `/etc/network/options` `syncookies=yes` festlegen. Jedes Mal, wenn `/etc/init.d/networking` ausgeführt wird (was typischerweise beim Booten geschieht), wird diese Option wirksam. Dagegen wird folgendes nur eine einmalige Wirkung bis zum nächsten Neustart haben:

```
echo 1 > /proc/sys/net/ipv4/tcp_syncookies
```

Diese Option ist nur verfügbar, wenn der Kernel mit `CONFIG_SYNCOOKIES` übersetzt wurde. Alle Kernel von Debian wurden mit dieser Option kompiliert. Sie können das folgendermaßen überprüfen:

```
$ sysctl -A |grep syncookies
net/ipv4/tcp_syncookies = 1
```

Weitere Informationen zu TCP-Syncookies finden Sie unter http://cr.yp.to/syncookies.html.

Absicherung des Netzwerks beim Hochfahren

Wenn Sie die Netzwerkoptionen des Kernels konfigurieren, müssen Sie dafür sorgen, dass sie bei jedem Neustart des Systems geladen werden. Das nachfolgende Beispiel aktiviert neben vielen der oben vorgestellten Optionen auch noch ein paar andere nützliche Optionen.

Tatsächlich gibt es zwei Möglichkeiten, Ihr Netzwerk beim Booten einzurichten. Sie können entweder `/etc/sysctl.conf` konfigurieren (siehe sysctl.conf(5)) oder ein Skript einsetzen, das beim Aktivieren der Netzwerkschnittstellen aufgerufen wird. Die erste Möglichkeit wird auf alle Schnittstellen angewendet, die zweite erlaubt es Ihnen, die Konfiguration für jede Schnittstelle separat zu wählen.

Ein Beispiel einer Konfiguration von `/etc/sysctl.conf`, die einige Netzwerkoptionen auf der Kernelebene absichert, wird unten gezeigt. Beachten Sie darin den Kommentar, dass `/etc/network/options` beim Ausführen von `/etc/init.d/networking` (dies ist in der Startsequenz nach `procps`) einige Werte überschreiben könnte, wenn sich Werte in dieser Datei widersprechen.

```
#
# /etc/sysctl.conf - Configuration file for setting system variables
# See sysctl.conf (5) for information. Also see the files under
# Documentation/sysctl/, Documentation/filesystems/proc.txt, and
# Documentation/networking/ip-sysctl.txt in the kernel sources
# (/usr/src/kernel-$version if you have a kernel-package installed)
# for more information of the values that can be defined here.

#
# Be warned that /etc/init.d/procps is executed to set the following
# variables.  However, after that, /etc/init.d/networking sets some
# network options with builtin values.  These values may be overridden
# using /etc/network/options.
#
#kernel.domainname = example.com

# Additional settings - adapted from the script contributed
# by Dariusz Puchala (see below)
# Ignore ICMP broadcasts
net/ipv4/icmp_echo_ignore_broadcasts = 1
```

```
#
# Ignore bogus ICMP errors
net/ipv4/icmp_ignore_bogus_error_responses = 1
#
# Do not accept ICMP redirects (prevent MITM attacks)
net/ipv4/conf/all/accept_redirects = 0
# _or_
# Accept ICMP redirects only for gateways listed in our default
# gateway list (enabled by default)
# net/ipv4/conf/all/secure_redirects = 1
#
# Do not send ICMP redirects (we are not a router)
net/ipv4/conf/all/send_redirects = 0
#
# Do not forward IP packets (we are not a router)
# Note: Make sure that /etc/network/options has 'ip_forward=no'
net/ipv4/conf/all/forwarding = 0
#
# Enable TCP Syn Cookies
# Note: Make sure that /etc/network/options has 'syncookies=yes'
net/ipv4/tcp_syncookies = 1
#
# Log Martian Packets
net/ipv4/conf/all/log_martians = 1
#
# Turn on Source Address Verification in all interfaces to
# prevent some spoofing attacks
# Note: Make sure that /etc/network/options has 'spoofprotect=yes'
net/ipv4/conf/all/rp_filter = 1
#
# Do not accept IP source route packets (we are not a router)
net/ipv4/conf/all/accept_source_route = 0
```

Um dieses Skript verwenden zu können, müssen Sie es zuerst unter z.B. `/etc/network/interface-secure` (der Name ist nur ein Beispiel) erstellen und es wie folgt aus `/etc/network/interfaces` aufrufen:

```
auto eth0
iface eth0 inet static
        address xxx.xxx.xxx.xxx
        netmask 255.255.255.xxx
        broadcast xxx.xxx.xxx.xxx
        gateway xxx.xxx.xxx.xxx
        pre-up /etc/network/interface-secure
```

In diesem Beispiel wird das Skript aufgerufen, um alle Netzwerkschnittstellen abzusichern, wie unten gezeigt wird, bevor die Schnittstelle eth0 aktiviert wird.

```
#!/bin/sh -e
# Skriptname: /etc/network/interface-secure
#
# Verändert das Standardverhalten für alle Schnittstellen in einigen Bereichen,
# um vor TCP/IP-Spoofing und Angriffen zu schützen.
```

```
#
# Wurde von Dariusz Puchalak beigesteuert
#
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
                                        # Broadcast echo protection enabled.
echo 0 > /proc/sys/net/ipv4/conf/all/forwarding
                                        # IP forwarding disabled.
echo 1 > /proc/sys/net/ipv4/tcp_syncookies # TCP syn cookies protection enabled.
echo 1 >/proc/sys/net/ipv4/conf/all/log_martians # Log strange packets.
# (this includes spoofed packets, source routed packets, redirect packets)
# but be careful with this on heavy loaded web servers.
echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
                                        # Bad error message protection enabled.

# IP spoofing protection.
echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter

# Disable ICMP redirect acceptance.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects

# Disable source routed packets.
echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

exit 0
```

Beachten Sie, dass Sie auch verschiedene Netzwerkoptionen für verschiedene Schnittstellen (falls Sie mehr als eine haben) setzten können, indem Sie die pre-up-Zeile verändern:

```
pre-up /etc/network/interface-secure $IFACE
```

Zusätzlich müssen Sie ein Skript verwenden, das Änderungen nur auf eine bestimmte Schnittstelle anwendet und nicht auf alle Schnittstellen. Beachten Sie aber, dass einige Netzwerkoptionen nur global gesetzt werden können. Dies ist ein Beispielsskript:

```
#!/bin/sh -e
# Skriptname: /etc/network/interface-secure
#
# Verändert das Standardverhalten für alle Schnittstellen in einigen Bereichen,
# um vor TCP/IP-Spoofing und Angriffen zu schützen.
#
# Wurde von Dariusz Puchalak beigesteuert
#

IFACE=$1
if [ -z "$IFACE" ] ; then
   echo "$0: Must give an interface name as argument!"
   echo "Usage: $0 <interface>"
   exit 1
fi

if [ ! -e /proc/sys/net/ipv4/conf/$IFACE/ ]; then
   echo "$0: Interface $IFACE does not exit (cannot find /proc/sys/net/ipv4/conf/)
```

```
   exit 1
fi

echo 0 > /proc/sys/net/ipv4/conf/$IFACE/forwarding  # IP forwarding disabled.
echo 1 >/proc/sys/net/ipv4/conf/$IFACE/log_martians # Log strange packets.
# (this includes spoofed packets, source routed packets, redirect packets)
# but be careful with this on heavy loaded web servers.

# IP spoofing protection.
echo 1 > /proc/sys/net/ipv4/conf/$IFACE/rp_filter

# Disable ICMP redirect acceptance.
echo 0 > /proc/sys/net/ipv4/conf/$IFACE/accept_redirects
echo 0 > /proc/sys/net/ipv4/conf/$IFACE/send_redirects

# Disable source routed packets.
echo 0 > /proc/sys/net/ipv4/conf/$IFACE/accept_source_route

exit 0
```

Eine andere Lösungsmöglichkeit ist es, ein **init.d**-Skript zu erstellen und es beim Booten auszuführen (verwenden Sie update-rc.d, um die passenden **rc.d**-Links herzustellen).

Konfiguration der Firewall

Um die Möglichkeiten einer Firewall zu haben, damit entweder das lokale System oder andere *dahinter* beschützt werden, muss der Kernel mit Firewall-Unterstützung kompiliert worden sein. Der Standardkernel von Debian 2.2 (Linux 2.2) stellt die Paketfilter-Firewall **ipchains** zur Verfügung. Der Standardkernel von Debian 3.0 (Linux 2.4) enthält die *stateful* Paketfilter-Firewall **iptables** (netfilter).

In jedem Fall ist es recht einfach, einen anderen als den mit Debian gelieferten Kernel zu benutzen. Sie finden vorkompilierte Kernel als Pakete vor, die Sie leicht auf Ihrem Debian-System installieren können. Mit Hilfe des Pakets kernel-source-X können Sie auch die Kernelquellen herunterladen und einen maßgeschneiderten Kernel kompilieren, indem Sie **make-kpkg** aus dem Paket kernel-package benutzen.

Auf das Aufsetzen einer Firewall unter Debian wird unter „Hinzufgen von Firewall-Fhigkeiten“ ausführlich eingegangen.

Lösung des Problems der Weak-End-Hosts

Auf Systemen mit mehr als einer Schnittstelle zu verschiedenen Netzwerken können Dienste so eingerichtet werden, dass sie Verbindungen nur zu einer bestimmten IP-Adresse zulassen. Normalerweise verhindert das den Zugang zu diesen Diensten, wenn an sie Anfragen über andere Adressen gestellt werden. Allerdings bedeutet das nicht, dass der Dienst an eine bestimmte *Hardware*-Adresse (Netzwerkkarte) gebunden ist (ein verbreiteter Irrtum). [31]

Das scheint allerdings nicht mit Diensten zu funktionieren, die mit 127.0.0.1 verbunden sind. Sie sollten vielleicht für die Tests raw sockets verwenden.

[31] Um das nachzuvollziehen folgendes Beispiel, das von Felix von Leitner auf der Bugtraq-Mailingliste vorgestellt wurde:

```
host a (eth0 connected to eth0 of host b):      ifconfig eth0 10.0.0.1      ifconfig eth1 23.0.0.1      tcpser
```

Das scheint allerdings nicht mit Diensten zu funktionieren, die mit 127.0.0.1 verbunden sind. Sie sollten vielleicht für die Tests raw sockets verwenden.

Das ist kein Problem von ARP und auch keine Verletzung eines RFCs (es wird in ftp://ftp.isi.edu/in-notes/rfc1122.txt, Abschnitt 3.3.4.2 als *weak end host* bezeichnet). Vergessen Sie nicht, dass IP-Adressen nichts mit dem physischen Schnittstellen zu tun haben.

Im Kernel 2.2 (und davor) konnte dieses Problem so gelöst werden:

```
# echo 1 > /proc/sys/net/ipv4/conf/all/hidden
# echo 1 > /proc/sys/net/ipv4/conf/eth0/hidden
# echo 1 > /proc/sys/net/ipv4/conf/eth1/hidden
.....
```

Bei späteren Kernel kann das folgendermaßen gelöst werden:

- Regeln für iptables
- richtig konfiguriertes Routing [32] oder
- Patchen des Kernels [33]

In diesem Text finden sich viele Fälle, in denen gezeigt wird, wie man einige Dienste (sshd-Server, apache, Druckserver, ...) so konfiguriert, dass sie nur auf einer bestimmten Adresse lauschen. Der Leser sollte in Betracht ziehen, dass das den Zugang aus dem gleichen (lokalen) Netzwerk nicht verhindern kann, wenn nicht die in diesem Abschnitt vorgeschlagenen Schritte ergriffen werden. [34]

FIXME: Comments on Bugtraq indicate there is a Linux specific method to bind to a given interface.

FIXME: Submit a bug against netbase so that the routing fix is standard behavior in Debian?

Schutz vor ARP-Angriffen

Wenn Sie den anderen Kisten in Ihrem LAN nicht trauen (das sollte immer so sein, da es die sicherste Einstellung ist), sollten Sie sich vor den verschiedenen ARP-Angriffen schützen.

Wie Sie wissen, wird das ARP-Protokoll dazu verwendet, IP-Adressen mit MAC-Adressen zu verknüpfen (für alle Details siehe ftp://ftp.isi.edu/in-notes/rfc826.txt). Jedes Mal, wenn Sie ein Paket an eine IP-Adresse schicken, wird eine ARP-Auflösung vorgenommen (zuerst wird in den lokalen ARP-Speicher geschaut, und falls die IP nicht im Speicher ist, wird ein Rundruf (Broadcast) mit der ARP-Anfrage verschickt), um die Hardware-Adresse des Ziels zu finden. Alle ARP-Angriffe zielen darauf ab, Ihrem Rechner vorzugaukeln, dass die IP-Adresse des Rechners B mit der MAC-Adresse des Computers des Angreifers verbunden ist. Dadurch wird jedes Paket, das Sie an den Rechner B, der mit der IP-Adresse verbunden ist, schicken wollen, an den Computer des Eindringlings umgeleitet ...

Diese Angriffe (Verfälschung des ARP-Speichers, ARP-Spoofing, ...) ermöglichen dem Angreifer, auf Netzwerken den Verkehr abzuhören (selbst bei Netzwerken, die über einen Switch laufen). Er kann sich leicht in eine Verbindung einschleusen oder einen Host vom Netzwerk nehmen oder ... ARP-Angriffe sind leistungsfähig und einfach durchzuführen. Es gibt dafür auch einige Werkzeuge wie **arpspoof** aus dem Paket dsniff oder http://arpoison.sourceforge.net/.

[32] Die Tatsache, dass dieses Verhalten durch Routing geändert werden kann, wurde von Matthew G. Marsh in dem Bugtraq-Thread beschrieben:

```
eth0 = 1.1.1.1/24   eth1 = 2.2.2.2/24   ip rule add from 1.1.1.1/32 dev lo table 1 prio 15000   ip rule add fr
```

[33] Wie im Bugtraq-Thread beschrieben, gibt es dafür einige Patches auf http://www.linuxvirtualserver.org/~julian/#hidden und http://www.fe-fe.de/linux-eth-forwarding.diff.

[34] Ein Angreifer, der nicht in der gleichen Broadcast-Domain (also dem gleichen Netzwerk) wie der angegriffene Host ist, kann auf viele Probleme bei Zugang stoßen, nachdem die Anbindung der IP-Adressen konfiguriert wurde. Wenn der Angriff über einen Router läuft, kann es sich als ziemlich schwer herausstellen, die Antworten zurückzubekommen.

Allerdings gibt es immer eine Lösung:

- Verwenden Sie einen statischen ARP-Speicher. So erstellen Sie »statische« Einträge in Ihrem ARP-Speicher:

  ```
  arp -s host_name hdwr_addr
  ```

  Indem Sie statische Einträge für jeden wichtigen Host in Ihrem Netzwerk vergeben, stellen Sie sicher, dass niemand einen (falschen) Eintrag für diese Hosts erstellen oder verändern kann (statische Einträge verfallen nicht und können nicht verändert werden). Auch gefälschte ARP-Antworten werden ignoriert.

- Entdecken Sie verdächtigen ARP-Verkehr. Sie können dazu arpwatch, karpski oder allgemeinere IDS, die auch verdächtigen ARP-Verkehr entdecken können wie snort oder http://www.prelude-ids.org, einsetzen.

- Verwenden Sie einen IP-Filter, der die MAC-Adressen überprüft.

Einen Schnappschuss des Systems erstellen

Bevor Sie das System in produktiven Betrieb nehmen, können Sie einen Schnappschuss des gesamten Systems erstellen. Diesen Schnappschuss können Sie im Falle einer Kompromittierung (siehe Kapitel 11, *Nach einer Kompromittierung (Reaktion auf einem Vorfall)*) benutzen. Sie sollten den Schnappschuss immer dann erneuern, wenn Sie das System aktualisieren, insbesondere wenn Sie auf eine neue Debian-Veröffentlichung upgraden.

Hierfür können Sie beschreibbare, austauschbare Datenträger benutzen, die Sie schreibschützen können. Dies kann eine Diskette (die nach der Benutzung schreibgeschützt wird), eine CD in einem CD-ROM-Laufwerk (Sie können auch wiederbeschreibbare CD-ROMs benutzen, so können Sie sogar alte Sicherheitskopien Ihrer MD5-Summen behalten), eine USB-Platte oder eine MMC-Karte (wenn Ihr System auf diese zugreifen kann und sie schreibgeschützt werden können) sein.

Das folgende Skript erstellt einen solchen Schnappschuss:

```
#!/bin/bash
/bin/mount /dev/fd0 /mnt/floppy
trap "/bin/umount /dev/fd0" 0 1 2 3 9 13 15
if [ ! -f /usr/bin/md5sum ] ; then
 echo "Cannot find md5sum. Aborting."
 exit 1
fi
/bin/cp /usr/bin/md5sum /mnt/floppy
echo "Calculating md5 database"
>/mnt/floppy/md5checksums.txt
for dir in /bin/ /sbin/ /usr/bin/ /usr/sbin/ /lib/ /usr/lib/
do
   find $dir -type f | xargs /usr/bin/md5sum >>/mnt/floppy/md5checksums-lib.txt
done
echo "post installation md5 database calculated"
if [ ! -f /usr/bin/sha1sum ] ; then
 echo "Cannot find sha1sum"
        echo "WARNING: Only md5 database will be stored"
else
 /bin/cp /usr/bin/sha1sum /mnt/floppy
```

```
 echo "Calculating SHA-1 database"
 >/mnt/floppy/sha1checksums.txt
 for dir in /bin/ /sbin/ /usr/bin/ /usr/sbin/ /lib/ /usr/lib/
 do
    find $dir -type f | xargs /usr/bin/sha1sum >>/mnt/floppy/sha1checksums-lib.txt
 done
 echo "post installation sha1 database calculated"
fi
exit 0
```

Beachten Sie, dass das Programm md5sum (und sha1sum, falls verfügbar) auch auf der Diskette gesichert werden muss, so dass Sie es später benutzen können, um die anderen Programme Ihres Systems zu prüfen (für den Fall, dass md5sum oder sha1sum einen Trojaner enthalten). Wenn Sie aber sicher sein wollen, dass Sie eine gültige Kopie von md5sum verwenden, sollten Sie eine statische Kopie von md5sum erstellen und diese verwenden (damit wird verhindert, dass eine manipulierte libc-Bibliothek das Programm beeinträchtigt) oder md5sum nur in einer sauberen Umgebung einsetzen, die Sie etwa mit einer Rettungs-CD-ROM oder einer Live-CD erzeugen können (damit wird verhindert, dass ein manipulierter Kernel das Programm beeinflusst). Ich kann es nicht genug betonen: Wenn Sie ein System haben, in das eingebrochen wurde, können Sie den Ausgaben nicht vertrauen. Sehen Sie sich auch Kapitel 11, *Nach einer Kompromittierung (Reaktion auf einem Vorfall)* an.

Dieser Schnappschuss enthält nicht die Dateien unterhalb von `/var/lib/dpkg/info`, wo MD5-Summen installierter Pakete enthalten sind (die Dateien enden mit `.md5sums`). Sie können diese Informationen zusätzlich kopieren, aber Sie sollten Folgendes beachten:

- Die Dateien mit den MD5-Summen enthalten die MD5-Summen aller Dateien, die ein Debian-Paket enthält, nicht nur die der Systemprogramme. Das hat zur Folge, dass diese Datenbank viel größer ist (5 MB statt 600 KB auf einem Debian GNU/Linux System mit grafischen Subsystem und etwa 2,5 GB Software installiert) und nicht auf ein kleines, transportables Medium wie eine Diskette passt, aber wohl auf einen tragbaren USB-Speicher.

- Nicht alle Debian-Pakete stellen MD5-Summen der installierten Dateien zur Verfügung, da es (derzeit) nicht in der Richtlinie verlangt wird. Sie können allerdings nach der Installation die MD5-Summen aller Pakete mit debsums erstellen:

  ```
  # debsums --generate=missing,keep
  ```

Sobald der Schnappschuss erstellt wurde, sollten Sie sicherstellen, dass das entsprechende Medium schreibgeschützt ist. Sie können es dann als Sicherheitskopie verwenden oder in ein Laufwerk stecken, um jede Nacht mit **cron** die MD5-Summen des Systems mit Ihrem Schnappschuss zu vergleichen.

Wenn Sie keine Überprüfung von Hand einrichten wollen, können Sie immer eines der Integritätssysteme verwenden, die diese Aufgabe und noch vieles mehr für Sie erledigen werden. Weitere Informationen finden Sie unter „Regelmäßiges Überprüfung der Integrität“.

Andere Empfehlungen

Benutzen Sie keine Software, die von svgalib abhängt

SVGAlib ist ganz nett für Konsolen-Liebhaber wie mich, aber in der Vergangenheit wurde mehrfach gezeigt, dass es ziemlich unsicher ist. Exploits durch zgv wurden veröffentlicht und es war einfach, Root zu werden. Versuchen Sie die Nutzung von SVGAlib-Programmen wann immer nur möglich zu vermeiden.

Kapitel 5. Absichern von Diensten, die auf Ihrem System laufen

Dienste knnen auf zwei Arten in einem laufenden System abgesichert werden:

- Sie so einstellen, dass auf sie nur von Zugangspunkten (Interfaces) zugegriffen werden kann, von denen es ntig ist.
- Sie so konfigurieren, dass sie nur von legitimierten Benutzern auf autorisierte Art und Weise benutzt werden knnen.

Dienste knnen durch Zugriffsbeschrnkungen auf Kernel-Ebene (durch eine Firewall) eingeschrnkt werden, so dass auf sie nur von bestimmten Orten aus zugegriffen werden kann. Konfigurieren Sie sie, so dass sie nur auf einer bestimmten Schnittstelle horchen (einige Dienste bieten diese Fhigkeiten nicht). Oder verwenden Sie eine andere Methode, zum Beispiel den Linux-vserver-Patch (fr 2.4.16), mit dem Prozesse an eine bestimmte Schnittstelle gebunden werden knnen.

Was die Dienste angeht, die von **inetd** aufgerufen werden (**telnet**, **ftp**, **finger**, **pop3**, ...), so ist es wert zu erwhnen, dass **inetd** so konfiguriert werden kann, dass er nur auf eine bestimmte Schnittstelle reagiert (unter Verwendung der `service@ip`-Syntax). Dies ist jedoch eine nicht dokumentierte Eigenschaft. Ein Ersatz, der Meta-Daemon **xinetd**, kennt eine `bind`-Option nur fr diesen Zweck. Lesen Sie dazu bitte ixnetd.conf(5).

```
service nntp
{
        socket_type     = stream
        protocol        = tcp
        wait            = no
        user            = news
        group           = news
        server          = /usr/bin/env
        server_args     = POSTING_OK=1 PATH=/usr/sbin/:/usr/bin:/sbin/:/bin
+/usr/sbin/snntpd logger -p news.info
        bind            = 127.0.0.1
}
```

Die folgenden Abschnitte gehen detaillierter darauf ein, wie bestimmte Dienste abhngig von der beabsichtigten Benutzung passend konfiguriert werden.

Absichern von ssh

Wenn Sie immer noch telnet statt ssh benutzen, sollten Sie dieses Handbuch kurz beiseitelegen und dies ndern. Ssh sollte anstelle von telnet fr alle Anmeldungen aus der Ferne benutzt werden. In einer Zeit, in der es leicht ist, Internet-Verkehr mitzuschnffeln und an Klartext-Passwrter heranzukommen, sollten Sie lediglich Protokolle verwenden, die Verschlsselung benutzen. Also fhren Sie sofort ein `apt-get install ssh` auf Ihrem System aus.

Ermuntern Sie alle Benutzer Ihres Systems, ssh anstelle von telnet zu benutzen, oder noch besser: deinstallieren Sie telnet/telnetd. Zustzlich sollten Sie es vermeiden, sich mit ssh als Root einzuloggen, und lieber andere Methoden benutzen, um Root zu werden. Wie zum Beispiel **su** oder **sudo**. Schlielich sollten Sie noch die Datei `/etc/ssh/sshd_config` fr mehr Sicherheit modifizieren:

- Lassen Sie ssh nur auf einer bestimmten Schnittstelle lauschen, falls Sie mehrere haben (und ssh nicht auf allen verfgbar sein soll) oder Sie in Zukunft eine neue Netzwerkkarte einbauen werden (und keine ssh-Verbindungen auf ihr erlauben wollen).

- Versuchen Sie so wenige Anmeldungen als Root wie mglich zu erlauben. Wenn nun jemand Root werden will, bentigt er zwei Anmeldungen. So kann das Root-Passwort nicht so leicht ausgetestet werden.

- **Port 666** oder **ListenAddress 192.168.0.1:666** verändern Sie den Port, auf dem ssh lauscht, so dass ein Eindringling nicht wirklich sicher sein kann, ob ein sshd-Daemon läuft (aber beachten Sie, dass dies lediglich »Sicherheit durch Verschleierung« ist).

- `PermitEmptyPasswords no` Nicht gesetzte Passwörter spotten jeglicher Systemsicherheit.

- Erlauben Sie nur bestimmten Benutzern sich via ssh auf der Maschine anzumelden. `benutzer@host` kann auch verwendet werden, um einen bestimmten Benutzer dazu zu zwingen, nur von einem bestimmten Host aus zuzugreifen.

- Erlauben Sie nur bestimmten Gruppenmitgliedern sich via ssh auf der Maschine einzuloggen. AllowGroups und AllowUsers haben quivalente Verfahrensweisen, um den Zugang zu der Maschine zu verwehren. Es wird nicht berraschen, dass es sich hierbei um DenyUsers und DenyGroups handelt.

- Es ist allein Ihre Wahl, was Sie hier eintragen. Es ist sicherer, Zugriff nur Benutzern zu erlauben, die ssh-Schlssel in der Datei `~/.ssh/authorized_keys` haben. Wenn Sie dies wollen, setzen Sie es auf no.

- Schalten Sie jede Art der Authentifizierung ab, die Sie nicht wirklich bentigen, zum Beispiel `RhostsRSAAuthentication`, `HostbasedAuthentication`, `KerberosAuthentication` oder `RhostsAuthentication`. Sie sollten sie abschalten, auch wenn sie es standardmig bereits sind (siehe dazu die Handbuch-Seite sshd_config(5)).

- Deaktivieren Sie die Protokollversion 1, da diese einige Designschwächen hat, die es einfacher zu machen, Passwörter zu knacken. Für weitere Informationen lesen Sie http://earthops.net/ssh-timing.pdf oder das http://xforce.iss.net/static/6449.php.

- Fgen Sie einen Bannertext (er wird aus der Datei bezogen) fr Benutzer, die sich mit dem ssh-Server verbinden, hinzu. In einigen Lndern sollte das Senden einer Warnung ber unautorisierten Zugriff oder Benutzerberwachung vor dem Zugriff zu einem bestimmten System erfolgen, um sich rechtlich abzusichern.

Sie knnen den Zugriff auf den ssh-Server auch mittels `pam_listfile` oder `pam_wheel` in der PAM-Kontrolldatei beschrnken. Zum Beispiel knnen Sie jeden abhalten, der nicht in der Datei `/etc/loginusers` aufgelistet ist, durch Hinzufgen folgender Zeile zu `/etc/pam.d/ssh`:

```
auth       required     pam_listfile.so sense=allow onerr=fail item=user file=/etc
```

Abschlieend beachten Sie bitte, dass diese Direktiven von einer OpenSSH-Konfigurationsdatei stammen. Derzeit gibt es drei weit verbreitete ssh-Daemonen: ssh1, ssh2 und OpenSSH von den OpenBSD-Leuten. Ssh1 war der erste verfgbare ssh-Daemon und er ist noch der weit verbreitetste (Gerchten zufolge gibt es sogar eine Windows-Version). Ssh2 hat gegenber ssh1 viele Vorteile, abgesehen davon, dass es unter einer unfreien Lizenz verffentlicht wurde. OpenSSH ist ein vllig freier ssh-Daemon, der sowohl ssh1 als auch ssh2 untersttzt. OpenSSH ist die Version, die installiert wird, wenn Sie auf Debian das Paket ssh auswhlen.

You can read more information on how to set up SSH with PAM support in the http://lists.debian.org/debian-security/2001/11/msg00395.html.

SSH in ein Chroot-Gefngnis einsperren

Zurzeit bietet OpenSSH keine Mglichkeit, automatisch Benutzer bei der Verbindung in ein Chroot-Gefngnis einzusperren (die kommerzielle Version bietet diese Funktionalitt). Wie dem auch sei, es gibt auch ein Projekt, das diese Funktionalitt fr OpenSSH anbietet, vergleiche http://chrootssh.sourceforge.net. Es ist aber aktuell noch nicht als Debianpaket verfgbar. Sie sollten stattdessen das `pam_chroot`-Modul, wie in „Den Benutzerzugang einschränken“ beschrieben, verwenden.

In „Chroot-Umgebung für SSH“ knnen Sie verschiedene Optionen finden, um Chroot-Umgebungen fr SSH zu erstellen.

Ssh-Clients

Wenn Sie einen SSH-Client mit einem SSH-Server verwenden, mssen Sie sicherstellen, dass er die selben Protokolle, die vom Server erzwungen werden, untersttzt. Wenn Sie beispielsweise das Paket mindterm verwenden, untersttzt dies nur Protokollversion 1. Jedoch ist der sshd-Server standardmig so konfiguriert, nur Version 2 (aus Sicherheitsgrnden) zu akzeptieren.

Verbieten der bertragung von Dateien

Wenn Sie *nicht* mchten, das Benutzer Dateien zum und vom ssh-Server bertragen, mssen Sie den Zugang zu **sftp-server** *und* zu **scp** einschrnken. Sie knnen dies fr **sftp-server** tun, indem Sie den korrekten `Subsystem`-Wert in `/etc/ssh/sshd_config` eintragen.

Sie knnen auch Benutzer mittels libpam-chroot in eine Chroot-Umgebung einsperren, so dass sie, selbst wenn Dateitransfers erlaubt sind, auf eine bestimmte Umgebung festgelegt sind, die keine Systemdateien enthlt.

Beschrnkung des Zugangs auf Dateientransfers

Sie knnen den Zugang von Benutzern der Gestalt beschrnken, dass sie nur Dateien bertragen knnen, aber keine interaktive Shell erhalten. Dies knnen Sie mit den folgenden Methoden erreichen:

- den Benutzern verbieten, sich auf dem ssh-Server einzuloggen (wie oben beschrieben entweder durch die Konfigurationsdatei oder die PAM-Konfiguration), oder
- den Benutzern nur eine eingeschrnkte Shell wie scponly oder rssh zuweisen. Diese Shells schrnken die Befehle ein, die den Benutzern zur Verfgung stehen, so dass sie auf dem entfernten Rechner keine Befehle ausfhren knnen.

Absichern von Squid

Squid is one of the most popular proxy/cache server, and there are some security issues that should be taken into account. Squid's default configuration file denies all users requests. However the Debian package allows access from 'localhost', you just need to configure your browser properly. You should configure Squid to allow access to trusted users, hosts or networks defining an Access Control List on `/etc/squid/squid.conf`, see the https://web.archive.org/web/20061206052115/http://www.deckle.co.za/squid-users-guide/Main_Page for more information about defining ACLs rules. Notice that Debian provides a minimum configuration for Squid that will prevent anything, except from *localhost* to connect to your proxy server (which will run in the default port 3128). You will need to customize your `/etc/squid/squid.conf` as needed.

Die empfohlene minimale Konfiguration (mit dem Paket vertrieben) sieht wie folgt aus:

```
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl SSL_ports port 443 563
acl Safe_ports port 80          # http
acl Safe_ports port 21          # ftp
acl Safe_ports port 443 563     # https, snews
acl Safe_ports port 70          # gopher
acl Safe_ports port 210         # wais
acl Safe_ports port 1025-65535  # unregistered ports
acl Safe_ports port 280         # http-mgmt
acl Safe_ports port 488         # gss-http
acl Safe_ports port 591         # filemaker
acl Safe_ports port 777         # multiling http
acl Safe_ports port 901         # SWAT
acl purge method PURGE
acl CONNECT method CONNECT
(...)
# Erlaube nur cachemgr Zugriff von localhost
http_access allow manager localhost
http_access deny manager
# Erlaube nur purge Anfragen von localhost
http_access allow purge localhost
http_access deny purge
# Verbiete Anfragen zu unbekannten Ports
http_access deny !Safe_ports
# Verbiete CONNECT zu anderen als SSL-Ports
http_access deny CONNECT !SSL_ports
#
# INSERT YOUR OWN RULE(S) HERE TO ALLOW ACCESS FROM YOUR CLIENTS
#
http_access allow localhost
# And finally deny all other access to this proxy
http_access deny all
#Default:
# icp_access deny all
#
#Allow ICP queries from everyone
icp_access allow all
```

Sie sollten Squid auch entsprechend Ihren System-Ressourcen konfigurieren einschlielich des Cache-Speichers (Option `cache_mem`), der Lage der zwischengespeicherten Dateien und der verwendeten Speichermenge auf der Platte (Option `cache_dir`).

Beachten Sie, dass es bei ungeeigneter Konfiguration vorkommen kann, dass jemand eine Mail ber Squid weiterleitet, da die Protokolle HTTP und SMTP ein hnliches Design haben. Squids Standardkonfiguration verweigert Zugriffe auf Port 25. Wenn Sie Verbindungen an Port 25 erlauben wollen, fgen Sie ihn einfach zu der Safe_ports-Liste hinzu. Dies ist aber *NICHT* empfohlen.

Passendes Aufsetzen und Konfigurieren des Proxy/Cache-Servers ist nur ein Teil der Absicherung Ihrer Site. Eine andere notwendige Aufgabe ist es, Squids Log-Dateien zu analysieren, um sicher zu gehen, dass alles so arbeitet, wie es soll. Es gibt ein paar Pakete in Debian GNU/Linux, die einem Administrator hierbei helfen knnen. Die folgenden Pakete sind in Debian 3.0 (Woody) und Debian 3.1 (Sarge) verfgbar:

- calamaris – Log-Datei-Analysator fr Squid- oder Oops-Proxy-Log-Dateien

- modlogan – ein modularer Log-Datei-Analysator
- sarg – Squid Analysis Report Generator
- squidtaild – Squid-Log-Beobachtungsprogramm

When using Squid in Accelerator Mode it acts as a web server too. Turning on this option increases code complexity, making it less reliable. By default Squid is not configured to act as a web server, so you don't need to worry about this. Note that if you want to use this feature be sure that it is really necessary. To find more information about Accelerator Mode on Squid see the https://web.archive.org/web/20070104164802/http://www.deckle.co.za/squid-users-guide/Accelerator_Mode

Absichern von FTP

Wenn Sie wirklich FTP benutzen mssen (ohne ihn mit sslwrap zu umhllen oder innerhalb eines SSL- oder SSH-Tunnels), sollten Sie ftp in das Home-Verzeichnis der FTP-Benutzer mit chroot einsperren, so dass diese nichts anderes sehen knnen als ihr eigenes Verzeichnis. Andernfalls knnen sie die Wurzel Ihres Dateisystems durchforsten, als htten sie Shell-Zugriff darauf. Sie knnen die folgende Zeile in Ihre `proftpd.conf`-Datei im globalen Abschnitt hinzufgen, um die chroot-Fhigkeiten zu nutzen:

```
DefaultRoot ~
```

Starten Sie ProFTPd neu, indem Sie `/etc/init.d/proftpd restart` eingeben, und prfen Sie, ob Sie noch aus Ihrem Home-Verzeichnis heraus kommen knnen.

Um ProFTPd-DoS-Angriffe durch ../../../ zu verhindern, fgen Sie die folgende Zeile Ihrer `/etc/proftpd.conf` hinzu: `DenyFilter \*.*/`

Vergessen Sie nicht, dass FTP Login- und Authentifizierungs-Passwort als Klartext sendet (dies ist kein Problem, wenn Sie einen anonymen, ffentlichen Dienst anbieten) und es gibt bessere Alternativen in Debian hierzu. Zum Beispiel **sftp** (aus dem Paket ssh). Es gibt auch freie Implementierungen von SSH fr andere Betriebssysteme, zum Beispiel http://www.chiark.greenend.org.uk/~sgtatham/putty/ oder http://www.cygwin.com.

Wenn Sie dennoch einen FTP-Server verwalten, whrend Sie den Benutzern Zugriff via SSH gewhren, knnten Sie auf ein typisches Problem stoen. Benutzer, die innerhalb eines mit SSH abgesicherten Systems auf einen anonymen FTP-Server zugreifen wollen, knnen versuchen, sich auf dem *FTP-Server* einzuloggen. Whrend der Zugriff verweigert werden wird, wird das Passwort trotzdem als Klartext ber das Netz gesendet. Um dies zu verhindern, hat der ProFTPd-Entwickler TJ Saunders einen Patch erstellt, der verhindert, dass Benutzer den anonymen FTP-Server mit gltigen SSH-Zugangsdaten fttern. Mehr Informationen und den Patch finden Sie unter: http://www.castaglia.org/proftpd/#Patches. Dieser Patch wurde auch an Debian gesandt, vergleiche http://bugs.debian.org/145669.

Zugriff auf das X-Window-System absichern

Heutzutage werden X-Terminals in immer mehr Unternehmen benutzt, wo ein Server fr viele Arbeitspltze bentigt wird. Dies kann gefhrlich sein, weil Sie dem Datei-Server erlauben mssen, sich mit den X-Clients zu verbinden (X-Server aus Sicht von X. X vertauscht die Definition von Client und Server). Wenn Sie dem (sehr schlechten) Vorschlag von vielen Dokumentationen folgen, geben Sie auf Ihrer Maschine `xhost +` ein. Dies erlaubt jedem X-Client, sich mit Ihrem System zu verbinden. Fr etwas bessere Sicherheit knnen Sie stattdessen das Kommando `xhost +Rechnername` verwenden, um den Zugriff auf bestimmte Rechner zu begrenzen.

Allerdings ist es eine viel sicherere Lsung, SSH zu benutzen, um X zu tunneln und die gesamte Sitzung zu verschlsseln. Dies geschieht automatisch, wenn Sie sich an einer anderen Maschine via ssh anmelden. Damit dies funktioniert, mssen Sie den ssh-Client und den ssh-Server konfigurieren. Auf dem ssh-Client muss `ForwardX11` in `/etc/ssh/ssh_config` auf `yes` gesetzt sein. Auf dem ssh-Server muss `X11Forwarding` in `/etc/ssh/sshd_config` auf `yes` gesetzt sein und das Paket xbase-clients muss installiert sein. Dies liegt daran, dass der SSH-Server `/usr/X11R6/bin/xauth` (bei Debian-Unstable (`/usr/bin/xauth`) verwendet, wenn er das Pseudo-X-Display aufsetzt. In den Zeiten von SSH sollten Sie die xhost-basierte Zugriffskontrolle komplett ber Bord werfen.

Wenn Sie keinen X-Zugriff von anderen Maschinen bentigen, ist es fr die Sicherheit am besten, die Bindung auf dem TCP-Port 6000 abzuschalten, indem Sie einfach Folgendes eingeben:

```
$ startx -- -nolisten tcp
```

Dies ist das Standard-Verhalten unter XFree 4.1.0 (dem Xserver aus Debian 3.0 und 3.1). Wenn Sie XFree 3.3.6 laufen lassen (d.h. wenn Sie Debian 2.2 benutzen), knnen Sie `/etc/X11/xinit/xserverrc` editieren, damit Sie etwas erhalten wie:

```
#!/bin/sh
exec /usr/bin/X11/X -dpi 100 -nolisten tcp
```

Wenn Sie XDM benutzen, setzen Sie `/etc/X11/xdm/Xservers` auf `:0 local /usr/bin/X11/X vt7 -dpi 100 -nolisten tcp`. Wenn Sie GDM benutzen, stellen Sie sicher, dass die Option `DisallowTCP=true` in `/etc/gdm/gdm.conf` eingetragen ist (was standardmig unter Debian der Fall ist). Dies wird grundstzlich an jede X-Befehlszeile `-nolisten tcp` anhngen [1].

Sie knnen auerdem eine standardmige Zeitgrenze fr die **xscreensaver**-Bildschirmsperre setzen. Auch wenn der Benutzer sie aufheben kann, sollten Sie die Konfigurationsdatei `/etc/X11/app-defaults/XScreenSaver` editieren und die lock-Zeile von

```
*lock:                  False
```

(das ist der Standardwert unter Debian) auf Folgendes ndern:

```
*lock:                  True
```

FIXME: Add information on how to disable the screensavers which show the user desktop (which might have sensitive information).

Lesen Sie mehr zur Sicherheit von X Window in http://www.tldp.org/HOWTO/XWindow-User-HOWTO.html (`/usr/share/doc/HOWTO/en-txt/XWindow-User-HOWTO.txt.gz`).

FIXME: Add info on thread of debian-security on how to change config files of XFree 3.3.6 to do this.

berprfen Ihres Display-Managers

Wenn Sie einen Display-Manager lediglich zur lokalen Nutzung (fr ein schnes graphisches Anmeldefenster) haben wollen, gehen Sie sicher, dass der XDMCP (X Display Manager Control Protocol) Krempel abgeschaltet ist. Unter XDM knnen Sie dies mit der folgenden Zeile in `/etc/X11/xdm/xdm-config` erreichen:

[1] GDM wird `-nolisten tcp` *nicht* anhngen, wenn es `-query` oder `-indirect` in der Befehlszeile findet, da sonst die Anfrage nicht funktionieren wrde.

```
DisplayManager.requestPort:     0
```

Fr GDM mssen Sie in Ihre gdm.conf Folgendes eintragen:

```
[xdmcp]
Enable=false
```

Normalerweise sind unter Debian alle Display-Manager so konfiguriert, dass sie standardmig keine XDMCP-Dienste starten.

Absichern des Druckerzugriffs (die lpd- und lprng-Problematik)

Stellen Sie sich vor, Sie kommen zur Arbeit und der Drucker spuckt endlose Mengen von Papier aus, weil jemand eine DoS-Attacke gegen Ihren Drucker-Daemon durchfhrt. Unangenehm, oder?

In jeder UNIX-Druck-Architektur muss es einen Weg geben, um die Daten des Clients zu dem Druck-Server zu schicken. Traditionell machen dies **lpr** und **lp** so, dass das Client-Kommando die Daten in das Spool-Verzeichnis kopiert oder symbolisch verlinkt (weshalb diese Programme normalerweise SUID oder SGID sind).

Um jede Gefahr zu vermeiden, sollen Sie Ihren Druck-Server besonders sicher halten. Dies heit, dass Sie Ihren Druckdienst so konfigurieren mssen, dass er nur Auftrge von vertrauenswrdigen Rechnern annimmt. Hierzu mssen Sie die Rechner, von denen Sie Druckauftrge entgegennehmen mchten, in die Datei `/etc/hosts.lpd` eintragen.

Allerdings akzeptiert der **lpr**-Daemon auch, wenn Sie dies getan haben, Verbindungen auf Port 515 auf jeder Schnittstelle. Sie sollten sich berlegen, ob Sie Verbindungen von Netzwerken/Rechnern, die nicht drucken drfen, mittels Firewall blocken wollen (der **lpr**-Daemon kann nicht so konfiguriert werden, dass er nur auf eine bestimmte IP-Adresse lauscht).

Sie sollten **Lprng** gegenber **lpr** vorziehen, da er so konfiguriert werden kann, dass er Zugangskontrolle ber IP beherrscht. Und Sie knnen spezifizieren, auf welche Schnittstelle er sich binden soll (wenn auch etwas sonderbar).

Wenn Sie Ihren Drucker nur lokal auf Ihrem System benutzen, werden Sie diesen Dienst nicht ber ein Netzwerk anbieten wollen. Sie sollten dann berlegen, ein anderes Druck-System, wie zum Beispiel das aus dem Paket cups oder http://pdq.sourceforge.net/, das auf den Zugriffsrechten des Gertes `/dev/lp0` beruht, einzusetzen.

Bei cups werden die Druckauftrge mit dem HTTP-Protokoll zum Server bertragen. Dadurch muss der Client nicht ber spezielle Privilegien verfgen, aber dies erfordert, dass der Server auf irgendeinem Port lauscht.

Wenn Sie **cups** jedoch nur lokal benutzen mchten, knnen Sie ihn so konfigurieren, dass er nur auf der Loopback-Schnittstelle lauscht, indem Sie Folgendes in Ihrer `/etc/cups/cupsd.conf` ndern:

```
Listen 127.0.0.1:631
```

Es gibt noch andere Sicherheitsoptionen in dieser Konfigurationsdatei, wie zum Beispiel das Erlauben oder Verweigern von Netzwerken oder Rechnern. Wenn Sie sie allerdings nicht bentigen, belassen Sie es am besten dabei, einfach nur den Port, auf dem gelauscht wird, einzuschrnken. **Cups** liefert auch Doku-

mentation ber den HTTP-Port. Wenn Sie diese potenziell ntzlichen Informationen einem Angreifer von auerhalb nicht enthllen wollen (und der Port offen ist), fgen Sie auerdem Folgendes hinzu:

```
<Location />
 Order Deny,Allow
 Deny From All
 Allow From 127.0.0.1
</Location>
```

Die Konfigurationsdatei kann so angepasst werden, dass zustzliche Fhigkeiten einschlielich SSL- und TLS-Zertifikate oder Verschlsselung mglich werden. Die Handbcher finden Sie unter http://localhost:631/ oder http://cups.org.

FIXME: Add more content (the article on http://www.rootprompt.org provides some very interesting views).

FIXME: Check if PDG is available in Debian, and if so, suggest this as the preferred printing system.

FIXME: Check if Farmer/Wietse has a replacement for printer daemon and if it's available in Debian.

Absichern des Mail-Dienstes

Wenn Ihr Server kein Mail-System ist, mssen Sie nicht wirklich einen Mail-Daemon haben, der auf eingehende Verbindungen reagiert. Aber Sie wollen lokale Mails ausliefern, um beispielsweise Mails an den Root-User von irgendwelchen Alarmsystemen zu erhalten.

Wenn Sie **exim** haben, mssen Sie den Daemon nicht laufen lassen, um dies zu erreichen, da der Standard-**cron**-Job die Mails abarbeitet. Sehen Sie in „Daemons abschalten“, wie man dies erledigt.

Konfiguration eines Nullmailers

Sie brauchen vielleicht einen lokalen Mail-Daemon, damit er die Mails, die vom lokalen Rechner zu einem anderen System geschickt wurden, versenden kann. Dies ist blich, wenn Sie eine Anzahl von Systemen zu administrieren haben und nicht zu jedem von diesen eine Verbindung aufbauen wollen, um die dort lokal verschickten Mails zu lesen. Genau wie all das Protokollieren eines jeden individuellen Systems durch einen zentralen syslog-Server zentralisiert werden kann, so kann auch Mail zu einem zentralen Mail-Server gesandt werden.

Solch ein *nur sendendes* System sollte sorgfltig dafr eingerichtet werden. Der Daemon kann ebenso konfiguriert werden, nur an der Loopback-Adresse zu lauschen.

Die folgenden Konfigurationsschritte mssen nur zur Konfiguration des exim-Pakets in der Debian 3.0 Version vorgenommen werden. Wenn Sie eine neuere Version verwenden (wie z.B. 3.1, das exim4 verwendet), so wurde das Installationssystem verbessert, so dass, wenn der Mail-Transport-Agent konfiguriert wurde, nur lokale Mails zu versenden, es automatisch nur Verbindungen vom lokalen Rechner und keine Verbindungen aus der Ferne zulsst.

In einem Debian 3.0 System mit exim muss man den SMTP-Daemon aus **inetd** wie folgt entfernen:

```
$ update-inetd --disable smtp
```

und den Mail-Daemon so konfigurieren, dass er nur auf der loopback-Schnittstelle lauscht. In **exim** (dem Standard-Mail-Transport-Agent (MTA) unter Debian) tun Sie dies, indem Sie in der Datei `/etc/exim.conf` folgende Zeile hinzufgen:

```
local_interfaces = "127.0.0.1"
```

Starten Sie beide Daemonen neu (inetd und exim) und exim wird lediglich auf den Socket 127.0.0.1:25 lauschen. Seien Sie vorsichtig und deaktivieren Sie erst inetd, oder exim wird nicht neu starten, da der inetd-Daemon bereits eingehende Verbindungen behandelt.

Bei **postfix** editieren Sie `/etc/postfix/main.conf`:

```
inet_interfaces = localhost
```

Wenn Sie lediglich lokale Mails wollen, ist dieses Herangehen besser als den Mailer-Daemon in einen tcp-Wrapper zu hllen oder Firewall-Regeln einzufgen, die den Zugang fr alle limitieren. Wenn Sie jedoch auch auf andere Schnittstellen reagieren mssen, sollten Sie berlegen, ihn vom inetd aufrufen zu lassen und einen tcp-Wrapper einzusetzen, so dass eingehende Verbindungen gegen `/etc/hosts.allow` und `/etc/hosts.deny` geprft werden. Auerdem werden Sie vor unautorisierten Zugriffsversuchen gegen Ihren Mail-Daemon durch angemessenes Protokollieren gewarnt werden wollen.

In jedem Fall knnen Sie Mail-Zustellversuche auf dem SMTP-Level ablehnen, indem Sie die `/etc/exim/exim.conf` abndern, damit Sie Folgendes enthlt:

```
receiver_verify = true
```

Auch wenn Ihr Mail-Server keine Mails zustellen wird, ist diese Konfiguration fr den Relay-Tester auf http://www.abuse.net/relay.html ntig, um festzustellen, dass Ihr Server *nicht* relaisfhig ist.

Wenn Sie Mails nur weiterleiten mchten, knnen Sie in Erwgung ziehen, den Mail-Daemon durch Programme zu ersetzen, die *nur* zum Weiterleiten der Mail zu einem entfernten Mail-Server konfiguriert werden knnen. Debian stellt zurzeit ssmtp und nullmailer fr diese Zwecke zur Verfgung. Auf jeden Fall knnen Sie fr sich selbst alle von Debian angebotenen Mail-Transport-Agents testen [2] und sehen, welcher davon am besten auf Ihr System zugeschnitten ist.

Anbieten eines sicheren Zugangs zu Mailboxen

Wenn Sie entfernten Zugriff auf Mailboxen erlauben wollen, gibt es eine Anzahl von mglichen POP3- und IMAP-Daemonen.[3] Wenn Sie IMAP-Zugriff anbieten, beachten Sie jedoch, dass es ein allgemeines Dateizugriffsprotokoll ist. Es kann das quivalent zu einem Shell-Zugang werden, da Benutzer in der Lage sein knnten, Zugang zu beliebigen Dateien zu erhalten, auf die sie durch ihn zugreifen knnen.

Versuchen Sie beispielsweise, `{server.com}/etc/passwd` als Ihren Eingabepfad zu konfigurieren. Wenn dies gelingt, ist Ihr IMAP-Daemon nicht richtig konfiguriert, um diese Art von Zugriff zu verhindern.

Unter den IMAP-Servern in Debian vermeidet der **cyrus**-Server (im Paket cyrus-imapd) dies, indem er den gesamten Zugriff zu einer Datenbasis in einem beschrnkten Teil des Dateisystems limitiert. Auch **uw-imapd** (installieren Sie entweder das uw-imapd- oder besser, wenn Ihre IMAP-Clients es untersttzen, das uw-imapd-ssl-Paket) kann konfiguriert werden, das Mailverzeichnis der Benutzer in ein Chroot-Gefng-

[2] Die Liste der in Debian verfgbaren Mail-Daemons erhalten Sie wie folgt:

```
$ apt-cache search mail-transport-agent
```

Die Liste wird **qmail** nicht enthalten, da dies nur im Quellcode im Paket qmail-src vertrieben wird.

[3] Eine Liste von Servern/Daemonen die diese Protokolle in Debian anbieten, kann wie folgt erhalten werden:

```
$ apt-cache search pop3-server $ apt-cache search imap-server
```

nis einzusperren, dies ist jedoch nicht standardmig aktiviert. Die angebotene Dokumentation enthlt mehr Informationen, wie man dies konfiguriert.

Es ist ebenso empfehlenswert, einen IMAP-Server laufen zu haben, der keine neuen Benutzer im lokalen System erfordert (dies wrde auch einen Shell-Zugang ermglichen). Sowohl courier-imap (fr IMAP) und courier-pop, teapop (fr POP3) und cyrus-imapd (fr POP3 und IMAP) bieten Server mit Authentifizierungsmethoden neben den lokalen Benutzerkonten. **cyrus** kann alle Authentifizierungsmethoden, die mittels PAM konfiguriert werden knnen, verwenden, whrenddessen **teapop**-Datenbanken (wie postgresql und mysql) fr die Benutzerauthentifizierung nutzen kann.

FIXME: Check: uw-imapd might be configured with user authentication through PAM too.

Sicherer Empfang von Mails

Das Lesen und Empfangen von Mails ist das gebruchlichste Klartext-Protokoll. Wenn Sie POP3 oder IMAP benutzen, um Ihre Mails zu erhalten, senden Sie ein Klartext-Passwort ber das gesamte Netz, so dass ziemlich jeder Ihre Mails von nun an lesen kann. Benutzen Sie stattdessen SSL (Secure Sockets Layer), um Ihre Mails zu empfangen. Wenn Sie einen Shell-Account auf dem Rechner, der als POP oder IMAP-Server agiert, haben, ist die andere Alternative SSH. Hier ist eine beispielhafte `fetchmailrc`, um dies zu zeigen:

```
poll my-imap-mailserver.org via "localhost"
  with proto IMAP port 1236
      user "ref" there with password "hackmich" is alex here warnings 3600
    folders
      .Mail/debian
    preconnect 'ssh -f -P -C -L 1236:my-imap-mailserver.org:143 -l ref
     my-imap-mailserver.org sleep 15 </dev/null > /dev/null'
```

Die wichtige Zeile ist die preconnect-Zeile. Sie startet eine SSH-Verbindung und erstellt den notwendigen Tunnel, durch den automatisch alle Verbindungen zum lokalen Port 1236 verschlsselt an den IMAP-Mail-Server weitergeleitet werden. Eine andere Mglichkeit wre es, **fetchmail** mit SSL-Untersttzung zu benutzen.

Wenn Sie verschlsselte Mail-Dienste wie POP oder IMAP anbieten mchten, `apt-get install stunnel` und starten Sie Ihren Daemon auf diese Weise:

```
stunnel -p /etc/ssl/certs/stunnel.pem -d pop3s -l /usr/sbin/popd
```

Dieses Kommando bindet den angegebenen Daemon (-l) an den Port (-d) und benutzt ein bestimmtes SSL-Zertifikat (-p).

Absichern von BIND

Es gibt verschiedene Dinge, mit denen Sie sich auseinander setzen sollten, um einen Domain-Server-Daemon abzusichern, die hnlich zu den berlegungen sind, wie man einen anderen Dienst absichert:

- Konfigurieren Sie den Daemon selbst, so dass er von auen nicht missbraucht werden kann (siehe auch „Bind-Konfiguration um Missbrauch zu verhindern“). Dies schliet das Einschrnken von Abfragen durch Clients ein: Zonen-Transfers und rekursive Abfragen.

- Einschrnken des Zugriffs des Daemon auf den Server selbst, so dass dem Schaden fr das System im Falle eines Einbruchs Grenzen gesetzt sind. Hierzu gehrt auch, den Daemon als nicht-privilegierten Benutzer

laufen zu lassen (siehe „ndern des BIND-Benutzers“) und ihn in ein Chroot-Gefngnis einzusperren (siehe „Chroot-Gefngnis fr den Name-Server“).

Bind-Konfiguration um Missbrauch zu verhindern

Sie sollten einige Informationen, die von auen ber den DNS-Server abgefragt werden knnen, zurckhalten, so dass man nicht wertvolle Informationen ber Ihre Organisation, die Sie nicht herausgeben wollen, abfragen kann. Dies schliet die folgenden Optionen mit ein: *allow-transfer*, *allow-query*, *allow-recursion* und *version*. Sie knnen dies in dem globalen Abschnitt tun (so wird es auf alle Zonen angewandt) oder jeweils pro Zone. Dies ist im Paket bind-doc dokumentiert. Sobald das Paket installiert ist, knnen Sie hierzu mehr in `/usr/share/doc/bind/html/index.html` lesen.

Stellen Sie sich vor, Ihr Server ist mit dem Internet und Ihrem internen Netzwerk (Ihre interne IP ist 192.168.1.2) verbunden – ein einfacher Server im heimischen Netzwerk. Sie mchten keinen Dienst im Internet anbieten und lediglich DNS-Abfragen von Ihren internen Rechnern erlauben. Sie knnen dies einschrnken, indem Sie Folgendes in Ihre `/etc/bind/named.conf` aufnehmen:

```
options {
            allow-query { 192.168.1/24; } ;
            allow-transfer { none; } ;
            allow-recursion { 192.168.1/24; } ;
            listen-on { 192.168.1.2; } ;
            forward { only; } ;
            forwarders { A.B.C.D; } ;
};
```

Die Option *listen-on* bewirkt, dass sich DNS nur auf die Schnittstelle bindet, die die interne Adresse hat. Aber sogar wenn diese Schnittstelle Verbindung zum Internet hat (zum Beispiel weil Sie NAT benutzen), werden Abfragen nur akzeptiert, wenn sie von internen Hosts kommen. Wenn das System mehrere Schnittstellen hat und Sie kein *listen-on* gesetzt haben, knnten zwar nur interne Benutzer Abfragen starten, aber, da der Port fr Angreifer von auen ansprechbar ist, knnten sie versuchen, den DNS zum Absturz zu bringen (oder durch Speicher-berlauf-Attacken auszunutzen). Sie knnten ihn sogar dazu bringen, lediglich auf 127.0.0.1 zu hren, wenn Sie den DNS-Service nicht fr ein anderes System anbieten wollen.

Der version.bind-Eintrag in der chaos class enthlt die Version des derzeit laufenden Bind-Prozesses. Diese Information wird oft von automatischen Scannern und bsartigen Individuen dazu verwendet herauszufinden, ob ein **bind** fr eine bestimmte Attacke verwundbar ist. Indem Sie falsche oder gar keine Informationen im version.bind-Eintrag zur Verfgung stellen, minimieren Sie die Wahrscheinlichkeit, dass jemand Ihren Server aufgrund der publizierten Version attackieren wird. Um Ihre eigene Version anzugeben, benutzen Sie die *Version*-Direktive auf folgende Art:

```
 options { ... various options here ...
version "Nicht verfgbar."; };
```

Das ndern des version.bind-Eintrags schtzt eigentlich nicht gegen Attacken, aber Sie knnen es als sinnvolle Schutzvorrichtung ansehen.

Eine beispielhafte `named.conf`-Konfigurationsdatei knnte so aussehen:

```
acl internal {
        127.0.0.1/32;           // localhost
        10.0.0.0/8;             // intern
        aa.bb.cc.dd;            // eth0 IP
```

```
};

acl friendly {
        ee.ff.gg.hh;            // slave DNS
        aa.bb.cc.dd;            // eth0 IP
        127.0.0.1/32;           // localhost
        10.0.0.0/8;             // intern
};

options {
        directory "/var/cache/bind";
        allow-query { internal; };
        allow-recursion { internal; };
        allow-transfer { none; };
};
// Ab hier bis zur meineseite.bogus Zone
// ist alles im Grunde die unvernderte Debian-Standardeinstellung.
logging {
        category lame-servers { null; };
        category cname { null; };
};

zone "." {
        type hint;
        file "/etc/bind/db.root";
};

zone "localhost" {
        type master;
        file "/etc/bind/db.local";
};

zone "127.in-addr.arpa" {
        type master;
        file "/etc/bind/db.127";
};

zone "0.in-addr.arpa" {
        type master;
        file "/etc/bind/db.0";
};

zone "255.in-addr.arpa" {
        type master;
        file "/etc/bind/db.255";
};

// Zone, die ich selbst hinzugefgt habe
zone "mysite.bogus" {
        type master;
        file "/etc/bind/named.meineseite";
        allow-query { any; };
        allow-transfer { friendly; };
};
```

Bitte prfen Sie (erneut) die Debian-Fehler-Datenbank (BTS) bezglich Bind, insbesondere http://bugs.debian.org/94760. Fhlen Sie sich ruhig dazu ermutigt, zu diesem Bugreport beizutragen, wenn Sie glauben, ntzliche Informationen hinzufgen zu knnen.

ndern des BIND-Benutzers

Bezglich der Beschrnkung von BINDs Privilegien mssen Sie beachten, dass, wenn Sie BIND als nicht-root Benutzer laufen lassen, BIND neue Netzwerk-Schnittstellen nicht automatisch entdecken kann, zum Beispiel wenn Sie eine PCMCIA-Karte in Ihr Notebook stecken. Lesen Sie die Datei `README.Debian` in Ihrer named-Dokumentation (`/usr/share/doc/bind/README.Debian`) fr mehr Informationen hierzu. Es gab in letzter Zeit viele Sicherheitsprobleme mit BIND, so dass es ntzlich ist, den Benutzer zu wechseln, wenn es mglich ist. Wir werden die Schritte, die dazu ntig sind, detailliert auffhren. Wenn Sie dies automatisch machen lassen wollen, knnen Sie das Skript in „Beispielskript, um die Standard-Installation von Bind zu ändern“ ausprobieren.

Beachten Sie, dass dies nur auf die BIND-Version 8 zutrifft. In den Debian-Paketen fr die BIND-Version 9 wird der Benutzer *bind* erstellt (seit Version 9.2.1-5, ist in *Sarge* enthalten) und mit der Variable OPTIONS in `/etc/default/bind9` verwendet. Wenn Sie BIND 9 einsetzen und Ihr Nameserver nicht als Benutzer *bind* luft, sollten Sie die Einstellungen in dieser Datei berprfen.

Um BIND unter einem anderen Benutzer laufen zu lassen, mssen Sie zunchst einen zustzlichen Benutzer und eine zustzlich Gruppe dafr erstellen (es ist *keine* gute Idee, fr alle Dienste, die Sie nicht als Root laufen lassen, den Benutzer nobody und die Gruppe nogroup zu benutzen). In diesem Beispiel wird der Benutzer und die Gruppe `named` verwendet. Sie knnen diese anlegen, indem Sie die folgenden Kommandos eingeben:

```
addgroup named
adduser --system --home /home/named --no-create-home --ingroup named \
      --disabled-password --disabled-login named
```

Beachten Sie, dass der Benutzer `named` sehr eingeschrnkt ist. Wenn Sie – aus welchen Grnden auch immer – ein weniger eingeschrnktes Setup haben mchten, benutzen Sie:

```
adduser --system --ingroup named named
```

Editieren Sie nun `/etc/init.d/bind` mit Ihrem Lieblingseditor und ndern Sie die Zeile, die mit

```
start-stop-daemon --start
```

anfngt zu[4]:

```
start-stop-daemon --start --quiet --exec /usr/sbin/named -- -g named -u named
```

Alternativ dazu knnen Sie auch die Standardkonfigurationsdatei (bei BIND 8 `/etc/default/bind`) bearbeiten (und erstellen, falls sie nicht vorhanden ist) und Folgendes einfgen:

```
OPTIONS="-u named -g named"
```

ndern Sie die Rechte der Dateien, die von Bind verwendet werden, inklusive `/etc/bind/rndc.key`:

[4] Beachten Sie, dass Sie abhngig von Ihrer Bind-Version die Option `-g` nicht haben, hchstwahrscheinlich wenn Sie bind9 von Sarge (9.2.4) installiert haben.

```
-rw-r-----    1 root     named          77 Jan  4 01:02 rndc.key
```

und den Ort, an dem bind seine PID-Datei erzeugt, z.B. indem Sie `/var/run/named` anstatt von `/var/run` verwenden:

```
$ mkdir /var/run/named
$ chown named.named /var/run/named
$ vi /etc/named.conf
[ ... ndern Sie die Konfigurationsdatei, um diesen neuen Pfad zu verwenden ...]
options { ...
        pid-file "/var/run/named/named.pid";
};
[ ... ]
```

Auerdem mssen Sie, um zu verhindern, dass irgendetwas als Root luft, im init.d-Skript die `reload`-Zeile von:

```
reload)
       /usr/sbin/ndc reload
```

in Folgendes ndern:

```
reload)
        $0 stop
        sleep 1
        $0 start
```

Hinweis: Abhngig von Ihrer Debian-Version mssen Sie auch die `restart`-Zeile ndern. Dies wurde in der Version `1:8.3.1-2` von Debians BIND-Paket repariert.

Alles, was Sie jetzt noch tun mssen, ist, bind mittels `/etc/init.d/bind restart` neu zu starten und dann Ihr Syslog auf zwei Eintrge wie die folgenden zu prfen:

```
Sep  4 15:11:08 nexus named[13439]: group = named
Sep  4 15:11:08 nexus named[13439]: user = named
```

Voil! Ihr named luft *nicht mehr* als Root. Wenn Sie mehr Informationen darber lesen wollen, warum BIND nicht als nicht-root Benutzer auf Debian-Systemen luft, sehen Sie bitte in der Fehlerdatenbank zu Bind nach, insbesondere http://bugs.debian.org/50013 und http://bugs.debian.org/132582, http://bugs.debian.org/53550, http://bugs.debian.org/52745 und http://bugs.debian.org/128129. Fhlen Sie sich ruhig dazu ermuntert, etwas zu den Fehlerbeschreibungen beizutragen, wenn Sie denken, ntzliche Informationen hinzufgen zu knnen.

Chroot-Gefngnis fr den Name-Server

Um die grtmgliche BIND-Sicherheit zu erreichen, mssen Sie nun ein Chroot-Gefngnis (siehe „Allgemeine chroot- und suid-Paranoia“) um Ihren Daemon herum bauen. Es gibt einen einfachen Weg, dies zu erreichen: Die Option `-t` (siehe die Handbuchseite named(8) oder Seite 100 von http://www.nominum.com/content/documents/bind9arm.pdf). Dies wird Bind selbst in ein bestimmtes Verzeichnis chrooten lassen, ohne dass Sie ein eigenes Chroot-Gefngnis aufsetzen und sich Sorgen um dynamische Bibliotheken machen mssen. Die einzigen Dateien, die in diesem Chroot-Gefngnis bentigt werden, sind:

```
dev/null
etc/bind/       – sollte die named.conf und alle Server-Zonen enthalten
sbin/named-xfer – wenn Sie Namen transferieren
var/run/named/  – sollte die PID und den Cache des
              Name-Servers (falls es ihn gibt) enthalten.
                          Dieses Verzeichnis muss fr
                          den named-User schreibbar sein.
var/log/named   – Wenn Sie in eine Datei protokollieren, muss
                          dies fr den named-User schreibbar sein.
dev/log       – syslogd sollte hierauf hren, wenn
                  named so konfiguriert ist, dass er
                  darber protokolliert.
```

Damit Ihr Bind-Daemon vernnftig luft, braucht er bestimmte Zugriffsrechte auf die named-Dateien. Dies ist eine einfache Angelegenheit, da die Konfigurationsdateien immer in `/etc/named/` liegen. Beachten Sie, dass er lediglich Lesezugriff bentigt, es sei denn, es handelt sich um einen sekundren oder zwischenspeichernden (Cache) Name-Server. Wenn dies der Fall ist, mssen Sie ihm Lese- und Schreibzugriff auf die notwendigen Zonen gewhren (damit Zonen-Transfers vom primren Server funktionieren).

Mehr Informationen ber das Chrooten von Bind finden Sie unter http://www.tldp.org/HOWTO/Chroot-BIND-HOWTO.html (betrifft Bind 9) und http://www.tldp.org/HOWTO/Chroot-BIND8-HOWTO.html (betrifft Bind 8). Diese Dokumente sollten auch nach der Installation des Paketes doc-linux-text (Text-Version) oder doc-linux-html (HTML-Version) verfgbar sein. Ein anderes ntzliches Dokument ist http://web.archive.org/web/20011024064030/http://www.psionic.com/papers/dns/dns-linux.

Wenn Sie fr Bind ein komplettes Chroot-Gefngnis aufsetzen (d.h. Sie benutzen nicht nur `-t`), stellen Sie sicher, dass Sie die folgenden Dateien darin haben:[5]

```
dev/log – syslogd sollte hierauf hren
dev/null
etc/bind/named.conf
etc/localtime
etc/group – mit einer einzigen Zeile: "named:x:GID:"
etc/ld.so.cache – mit ldconfig erstellt
lib/ld-2.3.6.so
lib/libc-2.3.6.so
lib/ld-linux.so.2 – symbolischer Link auf ld-2.3.6.so
lib/libc.so.6 – symbolischer Link auf libc-2.3.6.so
sbin/ldconfig – kann gelscht werden, nachdem Chroot aufgesetzt wurde
sbin/named-xfer – wenn Sie Namen transferieren
var/run/
```

Sorgen Sie auch dafr, dass **syslogd** auf `$CHROOT/dev/log` achtet, so dass der Name-Server seine syslog-Eintrge in das lokale System-Protokoll schreiben lassen kann.

Wenn Sie Probleme mit dynamischen Bibliotheken vermeiden wollen, knnen Sie Bind statisch kompilieren. Sie knnen hierzu **apt-get** mit der `source` Option benutzen. Es kann sogar die Pakete herunterladen, die Sie zum Kompilieren bentigen. Sie mssten etwas hnliches wie das Folgende tun:

```
$ apt-get source bind
# apt-get build-dep bind
```

[5] Diese Einstellungen wurden fr die neueren Verffentlichung von Bind noch nicht getestet.

```
$ cd bind-8.2.5-2
  (editieren Sie src/port/linux/Makefile, so dass CFLAGS
  die Option -static beinhaltet
$ dpkg-buildpackage -rfakeroot -uc -us
$ cd ..
# dpkg -i bind-8.2.5-2*deb
```

Nach der Installation werden Sie die Dateien in das Chroot-Gefngnis verschieben mssen.[6] Sie knnen die `init.d`-Skripte in `/etc/init.d` lassen, so dass das System automatisch den Name-Server starten wird, aber editieren Sie sie, indem Sie bei den **start-stop-daemon**-Aufrufen in diesen Skripten `--chroot /location_of_chroot` hinzufgen. Oder verwenden Sie fr BIND die Option *-t*, indem Sie sie in das OPTION-Argument in der Konfigurationsdatei `/etc/default/bind` (fr Version 8) oder `/etc/default/bind9` (fr Version 9) eintragen.

Fr weitere Informationen wie man Chroot-Gefngnisse aufsetzt, siehe „Allgemeine chroot- und suid-Paranoia“.

FIXME: Fge Informationen aus folgenden Quellen ein: http://people.debian.org/~pzn/howto/chroot-bind.sh.txt, http://www.cryptio.net/~ferlatte/config/ (Debian-specific), http://web.archive.org/web/20021216104548/http://www.psionic.com/papers/whitep01.html and http://csrc.nist.gov/fasp/FASPDocs/NISTSecuringDNS.htm.

Absichern von Apache

FIXME: Add content: modules provided with the normal Apache installation (under /usr/lib/apache/X.X/mod_*) and modules that can be installed separately in libapache-mod-XXX packages.

Sie knnen den Zugriff auf Ihren Apache-Server einschrnken, wenn Sie ihn nur intern benutzen wollen (zum Beispiel zu Testzwecken, oder um auf die doc-central-Archive zuzugreifen, etc.) und nicht wollen, dass von auen auf ihn zugegriffen werden kann. Um dies zu tun, benutzen Sie die `Listen` oder `BindAddress` Direktiven in der Datei `/etc/apache/http.conf`.

Benutzen von Listen:

```
Listen 127.0.0.1:80
```

Benutzen von BindAddress:

```
BindAddress 127.0.0.1
```

Starten Sie anschlieend Apache mit `/etc/init.d/apache restart` neu, und Sie werden sehen, dass er nur auf die lokale Schleife achtet.

In jedem Fall sollten Sie, wenn Sie nicht die ganze Funktionalitt, die Apache zur Verfgung stellt, benutzen wollen, mal einen Blick auf die anderen Web-Server aus Debian werfen, zum Beispiel dhttpd.

Die http://httpd.apache.org/docs/misc/security_tips.html stellt viele Informationen zu Sicherheitsmanahmen, die Sie auf einem Apache Web-Server anwenden knnen, bereit (die gleichen Informationen erhalten Sie unter Debian auch durch das Paket apache-doc).

Mehr Informationen zu weiteren Restriktionen von Apache durch Einrichten chroot-Gefngnisses wird in „Chroot-Umgebung für Apache“ bereitgestellt.

[6] Es sei denn, Sie benutzen die `instdir`-Option, wenn Sie **dpkg** aufrufen, aber dann knnte das chroot-Gefngnis etwas komplizierter werden.

Verhindern, dass Benutzer Web-Inhalte verffentlichen

Die Standard-Apache-Installation in Debian erlaubt Benutzern, Inhalt unter `$HOME/public_html` bereitzustellen. Dieser Inhalt kann aus der Ferne mit einer URL wie http://Ihr_Apache_Server/~benutzer abgegriffen werden.

Wenn Sie dies nicht erlauben wollen, mssen Sie in der Konfigurationsdatei `/etc/apache/http.conf` (von Apache 1.3) folgendes Module auskommentieren:

```
LoadModule userdir_module /usr/lib/apache/1.3/mod_userdir.so
```

Wenn Sie Apache 2.0 verwenden, mssen Sie die Datei `/etc/apache2/mods-enabled/user-dir.load` entfernen oder die Standardkonfiguration einschrnken, indem Sie `/etc/apache2/mods-enabled/userdir.conf` bearbeiten.

Falls allerdings das Modul statisch verlinkt wurde (Sie knnen die Module, die einkompiliert wurden, mittels `apache -l` berprfen), mssen Sie das Folgende der Konfigurationsdatei von Apache hinzufgen:

```
Userdir disabled
```

Ein Angreifer kann immer noch die Benutzer herausfinden, da die Antwort des Web-Servers *403 Permission Denied* und nicht *404 Not available* lautet. Mit dem Rewrite-Modul knnen Sie das verhindern.

Rechte der Protokolldateien

Apaches Protokolldateien gehren seit 1.3.22-1 dem Benutzer root und der Gruppe adm mit den Rechten 640. Diese Rechte ndern sich nach einer Rotation. Ein Eindringling, der das System ber den Web-Server erreicht hat, kann so Eintrge in alten Protokolldatei nicht (ohne Rechteerweiterung) entfernen.

Verffentlichte Web-Dateien

Apache-Dateien befinden sich unterhalb von `/var/www`. Direkt nach der Installation bietet die Standardseite einige Informationen zu dem System (hauptschlich dass es ein Debian-System ist, auf welchem Apache luft). Die Standard-Webseiten gehren standardmig dem Benutzer root und der Gruppe root, whrenddessen der Apache-Prozess als Benutzer www-data und Gruppe www-data luft. Dies sollte es Angreifern, die in das System durch den Web-Server eindringen, schwerer machen, die Site zu verunstalten. Sie sollten natrlich die Standard-Webseiten (die Informationen, die Sie der Auenwelt vorenthalten wollen, enthalten knnen) durch Ihre eigenen ersetzen.

Absichern von Finger

Wenn Sie den Finger-Dienst laufen lassen wollen, fragen Sie sich bitte zuerst, ob Sie das auch wirklich tun mssen. Wenn es notwendig sein sollte, werden Sie feststellen, dass Debian viele Finger-Daemonen zur Verfgung stellt (hier die Ausgabe von **apt-cache search fingerd**):

- cfingerd – konfigurierbarer finger-Daemon
- efingerd – ein weiterer Unix-finger-Daemon mit anpassbarer Ausgabe
- ffingerd – ein sicherer finger-Daemon
- fingerd – Remote-User-Informationsserver
- xfingerd – BSD-hnlicher finger-Daemon mit qmail-Untersttzung

ffingerd ist der empfohlene finger-Daemon, wenn Sie vorhaben, einen ffentlichen Dienst anzubieten. In jedem Fall sind Sie zu Folgendem ermutigt, wenn Sie ihn ber inetd, xinetd oder tcpserver aufzusetzen: Schrnken Sie die Anzahl der Prozesse ein, die gleichzeitig laufen drfen, schrnken Sie den Zugriff auf den Finger-Daemon von bestimmten Hosts ein (indem Sie tcp-wrapper benutzen) und lassen Sie ihn nur auf der notwendigen Schnittstelle lauschen.

Allgemeine chroot- und suid-Paranoia

chroot ist eine der mchtigsten Mglichkeiten, einen Daemon oder einen Benutzer oder einen anderen Dienst zu beschrnken. Denken Sie einfach an ein Gefngnis um Ihr Ziel, welches das Ziel nicht verlassen kann (normalerweise, es gibt aber einige Bedingungen, die einem einen Ausbruch aus solch einem Gefngnis ermglichen). Wenn Sie einem Benutzer oder einem Dienst nicht trauen, knnen Sie eine modifizierte root-Umgebung fr ihn erzeugen. Dies kann einiges an Plattenplatz bentigen, da Sie alle bentigten Programme ebenso wie Bibliotheken in das Gefngnis kopieren mssen. Aber danach ist die Wirkung des Schadens, selbst wenn der Benutzer etwas bsartiges macht, auf das Gefngnis beschrnkt.

Viele Dienste, die als Daemonen laufen, knnen von dieser Vorgehensweise profitieren. Die Daemonen, die Sie mit Ihrer Debian-Distribution installieren, laufen jedoch nicht standardmig in einem chroot-Gefngnis.[7]

Dies beinhaltet: Name-Server (wie **bind**), Web-Server (wie **apache**), Mail-Server (wie **sendmail**) und FTP-Server (wie **wu-ftpd**). Wahrscheinlich ist es nur fair zu sagen, dass die Komplexitt von BIND der Grund dafr ist, warum er in den letzten Jahren so oft fr Attacken verwundbar war (vergleiche „Absichern von BIND“).

Jedoch bietet Debian einige Software an, die helfen kann, **chroot**-Umgebungen aufzubauen. Sehen Sie „Automatisches Erstellen von Chroot-Umgebungen“.

Wenn Sie irgendwelche Dienste in Ihrem System laufen lassen, sollten Sie dies so sicher wie nur mglich tun. Dies beinhaltet: Entziehung von root-Privilegien, Starten in beschrnkten Umgebungen (wie ein chroot-Gefngnis) oder Ersetzen durch ein sichereres quivalent.

Seien Sie jedoch gewarnt, dass aus einem **chroot**-Gefngnis ausgebrochen werden kann, wenn der Benutzer, der im Inneren luft, der Superuser ist. Sie mssen also sicherstellen, dass der Dienst als nicht privilegierter Benutzer luft. Durch Einschrnken seiner Umgebung schrnken Sie die welt-lesbaren/ausfhrbaren Dateien, auf die der Dienst zugreifen kann, ein. Auf diese Weise begrenzen Sie die Mglichkeiten einer Rechteerweiterung durch lokale Sicherheitsverwundbarkeiten des Systems. Selbst in dieser Situation knnen Sie nicht vllig sicher sein, dass es fr einen cleveren Angreifer keinen Weg gibt, irgendwie aus dem Gefngnis auszubrechen. Der ausschlieliche Einsatz sicherer Server-Programme, die einen guten Ruf bezglich Sicherheit haben, ist eine zustzliche gute Sicherheitsmanahme. Selbst kleinste Lcher wie offene Datei-Handle knnen von einem versierten Angreifer zum Einbruch in das System verwendet werden. Schlielich war **chroot** nicht als Sicherheits-, sondern als ein Testwerkzeug gedacht.

Automatisches Erstellen von Chroot-Umgebungen

Es gibt verschiedene Programme, um Server und Dienste automatisch in ein Chroot-Gefngnis einzusperren. Debian bietet zurzeit (akzeptiert im Mai 2002) Wietse Venemas **chrootuid** im Paket chrootuid, ebenso wie compartment und makejail an. Diese Programme knnen verwendet werden, um eine eingeschrnkte Umgebung zum Ausfhren beliebiger Programme aufzusetzen (**chrootuid** erlaubt es Ihnen sogar, es unter einem eingeschrnkten Benutzer laufen zu lassen).

Einige dieser Werkzeuge knnen verwendet werden, um das Chroot-Gefngnis leicht aufzusetzen. Zum Beispiel kann das **makejail**-Programm ein chroot-Gefngnis mit kurzen Konfigurationsdateien erzeugen und

[7] Es wird versucht, diese mit *minimalen Rechten* laufen zu lassen, was beinhaltet, Daemonen unter ihren eigenen Benutzern, anstatt unter Root, laufen zu lassen.

aktualisieren. (Es bietet Beispielskonfigurationsdateien fr **bind**, **apache**, **postgresql** und **mysql**.) Es versucht alle Dateien, die vom Daemon bentigt werden, mittels **strace**, **stat** und Debians Paketabhngigkeiten zu bestimmen und in das Gefngnis zu installieren. Weitere Information gibt es unter http://www.floc.net/makejail/. **Jailer** ist ein hnliches Werkzeug und kann von http://www.balabit.hu/downloads/jailer/ heruntergeladen werden und ist auch als Debian-Paket verfgbar.

Allgemeine Klartextpasswort-Paranoia

Sie sollten versuchen, jeden Netzwerk-Dienst, der seine Passwrter als Klartext ber das Netz sendet oder empfngt, wie zum Beispiel FTP/Telnet/NIS/RPC, zu vermeiden. Der Autor empfiehlt jedem, ssh anstelle von telnet und ftp zu verwenden.

Vergessen Sie jedoch nicht, dass die Migration von telnet zu ssh die Sicherheit in keiner Weise erhht, wenn Sie weiterhin Klartext-Protokolle verwenden. Am besten wre es ftp, telnet, pop, imap und http zu entfernen und durch ihre entsprechenden verschlsselten Dienste zu ersetzen. Sie sollten in Erwgung ziehen von diesen Diensten zu deren SSL-Versionen zu wechseln: ftp-ssl, telnet-ssl, pop-ssl, https, ...

Die meisten der oben aufgelisteten Tipps gelten fr jedes Unix-System (Sie werden sie in jedem anderen sicherheitsrelevanten Dokument, das Sie jemals lesen, wiederfinden, wenn es sich auf Linux und andere Unices bezieht).

NIS deaktivieren

Sie sollten, wenn mglich, nicht NIS, den Network Information Service, benutzen, da er das gemeinsame Nutzen von Passwrtern erlaubt. Dies kann sehr unsicher sein, wenn Ihr Setup fehlerhaft ist.

Wenn Sie Passwrter zwischen verschiedenen Maschinen teilen mssen, sollten Sie andere Alternativen in Erwgung ziehen. Zum Beispiel knnen Sie einen LDAP-Server aufsetzen und PAM auf Ihrem System so konfigurieren, dass es den LDAP-Server zur Benutzer-Authentifizierung kontaktiert. Sie finden ein detailliertes Setup in dem http://www.tldp.org/HOWTO/LDAP-HOWTO.html (`/usr/share/doc/HOWTO/en-txt/LDAP-HOWTO.txt.gz`).

Sie knnen mehr ber NIS-Sicherheit in dem http://www.tldp.org/HOWTO/NIS-HOWTO.html (`/usr/share/doc/HOWTO/en-txt/NIS-HOWTO.txt.gz`) lesen.

FIXME (jfs): Add info on how to set this up in Debian.

Absichern von RPC-Diensten

Sie sollten RPC abschalten, wenn Sie es nicht bentigen.

Remote Procedure Call (RPC, etwa entfernter Funktionsaufruf) ist ein Protokoll, das von Programmen verwendet werden kann, um Dienste von anderen Programmen, die auf anderen Computern laufen, anzufordern. Der **portmap**-Dienst kontrolliert RPC-Dienste durch Abbilden von RPC-Programmnummern auf DARPA-Protokoll-Portnummern. Er muss laufen, um RPC-Aufrufe ausfhren zu knnen.

RPC-basierte Dienste hatten eine unrhmliche Geschichte, was Sicherheitslcken betrifft, obwohl dies fr den Portmapper an sich nicht gilt (dieser bietet aber nach wie vor entfernten Angreifern Informationen). Es ist zu beachten, dass einige DDoS-(distributed denial of service, verteilte Dienstverweigerungen)-Angriffe RPC-Lcher benutzen, um in das System einzudringen und als so genannter Agent/Handler zu fungieren.

Sie bentigen RPC nur dann, wenn Sie einen RPC-basierten Dienst verwenden. Die bekanntesten RPC-basierten Dienste sind NFS (Network File System) und NIS (Network Information System). Vergleichen

Sie mit dem vorherigen Abschnitt fr weitere Information ber NIS. Der File Alteration Monitor (FAM), der vom Paket fam bereitgestellt wird, ist ebenso ein RPC-Dienst und hngt deshalb von portmap ab.

NFS-Dienste sind in einigen Netzwerken ziemlich wichtig. Wenn dies fr Sie der Fall ist, mssen Sie einen Ausgleich finden, zwischen Sicherheit und Nutzbarkeit fr Ihr Netzwerk. Sie knnen mehr ber NFS-Sicherheit im http://www.tldp.org/HOWTO/NFS-HOWTO.html (`/usr/share/doc/HOWTO/en-txt/NFS-HOWTO.txt.gz`) finden.

Vollstndiges Deaktivieren von RPC-Diensten

Das Abschalten von portmap ist relativ einfach. Es gibt verschiedene Methoden. Die einfachste in einem Debian 3.0 oder neueren System ist das Paket portmap zu deinstallieren. Wenn Sie eine ltere Version von Debian laufen haben, werden Sie den Dienst, wie in „Daemons abschalten“ beschrieben, abschalten mssen, weil das Programm Teil des Pakets netbase (das nicht deinstalliert werden kann, ohne das System kaputt zu machen) ist.

Beachten Sie, dass einige Desktop-Umgebungen (hauptschlich GNOME) RPC-Dienste verwenden und den Portmapper fr einige der Dateimanager-Eigenschaften bentigen. Wenn dies bei Ihnen der Fall ist, knnen Sie den Zugang zu RPC-Diensten, wie weiter unter beschrieben, beschrnken.

Einschrnken des Zugriffs auf RPC-Dienste

Unglcklicherweise ist es in manchen Fllen nicht mglich, RPC-Dienste vom System zu entfernen. Einige lokale Desktop-Dienste (hauptschlich SGIs fam) sind RPC-basiert und bentigen deshalb einen lokalen Portmapper. Dies bedeutet, dass unter bestimmten Umstnden Benutzer, die eine Desktop-Umgebung (wie GNOME) installieren, den Portmapper auch installieren werden.

Es gibt einige Wege den Zugriff auf den Portmapper und RPC-Dienste zu beschrnken:

- Blockieren des Zugangs zu den Ports, die von diesen Diensten verwendet werden, mit einer lokalen Firewall (vergleiche „Hinzufgen von Firewall-Fhigkeiten“)
- Blockieren des Zugangs zu diesen Diensten mittels TCP-Wrappers, da der Portmapper und einige RPC-Dienste mit `libwrap` (siehe „Die Nutzung von Tcpwrappers STOPP“) kompiliert wurden. Dies bedeutet, dass Sie Zugang zu diesen durch die `hosts.allow` und `hosts.deny` TCP-Wrapper-Konfiguration blockieren knnen.
- Seit Version 5-5 kann das Paket portmap so konfiguriert werden, dass es nur noch auf der Loopback-Schnittstelle lauscht. Um dies zu erreichen, kommentieren Sie die folgende Zeile in der Datei `/etc/default/portmap` aus: `#OPTIONS="-i 127.0.0.1"` und starten Sie den Portmapper neu. Dies ist ausreichend, um lokale RPC-Dienste laufen zu lassen, whrend zur selben Zeit entfernte Systeme am Zugang gehindert werden (lesen Sie dazu auch „Lösung des Problems der Weak-End-Hosts “).

Hinzufgen von Firewall-Fhigkeiten

Das Debian-GNU/Linux-Betriebssystem hat die eingebauten Fhigkeiten des Linux-Kernels. Wenn Sie eine aktuelle Verffentlichung von Debian (mit dem Standardkernel 2.6) installiert haben, steht Ihnen als Firewall **iptables** (netfilter) zur Verfgung[8].

[8] Ist seit Kernel 2.4 verfgbar (was der Standardkernel fr Debian 3.0 war). ltere Kernelversionen (wie 2.2, der in lteren Debian-Verffentlichungen enthalten war) verwendeten **ipchains**. Der Hauptunterschied zwischen **ipchains** und **iptables** ist, dass letzteres auf *stateful packet inspection* (zustandsbehaftete Paketuntersuchung) beruht, so dass Ihnen sicherere (und einfacher zu erstellende) Filterkonfigurationen zur Verfgung stehen. ltere (und nun nicht lnger untersttzte) Debian-Verffentlichungen, die den Kernel 2.0 einsetzen, bentigten einen geeigneten Kernel-Patch.

Firewallen des lokalen Systems

Sie knnen eine Firewall dazu benutzen, den Zugriff auf Ihr lokales System abzusichern und sogar um die Kommunikation von ihm nach Auen zu beschrnken. Firewall-Regeln knnen auch dazu benutzt werden, Prozesse zu sichern, die nicht vernnftig konfiguriert werden knnen, um Dienste *nicht* einigen Netzwerken, IP-Adressen, etc. zur Verfgung zu stellen.

Dieser Schritt ist aber hauptschlich deshalb als letzter in dieser Anleitung, weil es *viel* besser ist, sich nicht alleine auf die Fhigkeiten der Firewall zu verlassen, um ein System zu schtzen. Die Sicherheit eines Systems setzt sich aus mehreren Ebenen zusammen; eine Firewall sollte die letzte sein, wenn bereits alle Dienste abgehrtet worden sind. Sie knnen sich sicherlich leicht eine Konfiguration vorstellen, bei der ein System lediglich von einer eingebauten Firewall geschtzt wird, und der Administrator glckselig die Firewall-Regeln aus irgendwelchen Grnden (Probleme mit dem Setup, Verdruss, Denkfehler, ...) entfernt. Dieses System wre weit geffnet fr Angriffe, wenn es keine anderen Schutzmanahmen auf dem System gibt.

Andererseits knnen Firewall-Regeln auf dem lokalen System dafr sorgen, dass bse Dinge nicht passieren. Sogar wenn die bereitgestellten Dienste sicher konfiguriert sind, kann eine Firewall vor Misskonfigurationen oder frisch installierten Diensten, die noch nicht passend konfiguriert sind, schtzen. Auerdem wird eine strenge Konfiguration *nach Hause telefonierende* Trojaner am Funktionieren hindern, es sei denn, der Firewall-Code wird entfernt. Beachten Sie, dass ein Eindringling *keinen* Superuser-Zugriff bentigt, um ferngesteuerte Trojaner zu installieren (da es erlaubt ist, sich an Ports zu binden, wenn es sich nicht um einen privilegierten Port handelt und die Fhigkeiten (Capabilities) noch vorhanden sind).

Demzufolge wre ein passendes Firewall-Setup, eines mit einer standardmigen Richtlinie, die alles ablehnt, was nicht ausdrcklich erlaubt ist, also:

- Eingehende Verbindungen werden nur zu lokalen Diensten von erlaubten Maschinen gestattet.
- Ausgehende Verbindungen werden nur von Diensten erlaubt, die auf Ihrem System benutzt werden (DNS, Web-Surfen, POP, E-Mail, ...).[9]
- Die Forward-Regel verbietet alles; es sei denn, andere Systeme werden geschtzt (siehe dazu unten).
- Alle anderen eingehenden und ausgehenden Verbindungen werden abgelehnt.

Schtzen anderer Systeme durch eine Firewall

Eine Debian-Firewall kann auch so installiert werden, dass sie mit Firewall-Regeln Systeme *hinter* ihr beschtzt, indem sie die Angriffsflche zum Internet hin einschrnkt. Eine Firewall kann so konfiguriert werden, dass ein Zugriff von Systemen auerhalb des lokalen Netzwerks auf interne Dienste (Ports) unterbunden wird. Zum Beispiel muss auf einem Mail-Server lediglich Port 25 (auf dem der Mail-Dienst aufsetzt) von auen zugnglich sein. Eine Firewall kann so konfiguriert werden, dass sogar, wenn es neben den ffentlich zugnglichen noch andere Netzwerkdienste gibt, direkt an diese gesendete Pakete verworfen werden (dies nennt man *filtern*).

Sie knnen eine Debian GNU/Linux Maschine sogar so konfigurieren, dass sie als Bridge-Firewall (berbrckender Schutzwall) fungiert, d.h. als eine filternde Firewall, die komplett transparent zum gesamten Netzwerk erscheint, ohne IP-Adresse auskommt und daher nicht direkt attackiert werden kann. Abhngig von dem installierten Kernel mssen Sie vielleicht den Bridge-Firewall-Patch installieren und dann *802.1d Ethernet Bridging* in der Kernel-Konfiguration und die neue Option *netfilter (firewalling) Support* aus-

[9] Im Gegensatz zu persnlichen Firewalls fr andere Betriebssysteme, stellt Debian GNU/Linux (noch) keine Firewall-Erstellungs-Schnittstelle zur Verfgung, die Regeln erstellen kann, die einzelne Prozesse oder Benutzer einschrnken. Jedoch kann der Iptables-Code so konfiguriert werden, dass er dies kann (siehe dazu das owner-Modul in der Handbuchseite iptables(8)).

whlen. Sehen Sie dazu „Aufsetzenden einer Bridge-Firewall“, um zu erfahren, wie man dies auf einem Debian GNU/Linux System aufsetzt.

Aufsetzen einer Firewall

Die Debian-Standardinstallation bietet im Gegensatz zu vielen anderen Linux-Distributionen noch keine Methode fr den Administrator, eine Firewall-Konfiguration mit der Standardinstallation einzurichten, aber Sie knnen eine Anzahl von Firewall-Konfigurationspaketen (siehe „Nutzen von Firewall-Paketen“) installieren.

Natrlich ist die Konfiguration einer Firewall immer vom System und dem Netzwerk abhngig. Ein Administrator muss vorher das Netzwerklayout und die Systeme, die er beschtzen will, kennen, die Dienste, auf die zugegriffen werden knnen muss, und ob andere netzwerkspezifischen Erwgungen (wie NAT oder Routing) bercksichtigt werden mssen. Seien Sie vorsichtig, wenn Sie Ihre Firewall konfigurieren. Wie Laurence J. Lane im iptables-Paket sagt:

Die Werkzeuge knnen leicht falsch verwendet werden und eine Menge rger verursachen, indem sie den gesamten Zugang zu einem Computernetzwerk stilllegen. Es ist nicht vllig ungewhnlich, dass sich ein Systemadministrator, der ein System verwaltet, das hunderte oder tausende von Kilometern entfernt ist, irrtmlicherweise selbst davon ausgeschlossen hat. Man kann es sogar schaffen, sich von dem Computer auszusperren, dessen Tastatur unter seinen Fingern liegt. Lassen Sie daher die gebotene Vorsicht walten.

Vergessen Sie nicht: Das bloe Installieren von iptables (oder dem lteren Firewallcode) gibt Ihnen keine Sicherheit, es stellt lediglich die Software zur Verfgung. Um eine Firewall zu haben, mssen Sie sie konfigurieren!

Wenn Sie keine Ahnung haben, wie Sie Ihre Firewall-Regeln manuell aufsetzen sollen, sehen Sie in dem *Packet Filtering HOWTO* und *NAT HOWTO* aus dem Paket iptables, zu finden unter `/usr/share/doc/iptables/html/` nach.

Wenn Sie nicht viel ber Firewalls wissen, sollten Sie beginnen, indem Sie das http://www.tldp.org/HOWTO/Firewall-HOWTO.html lesen. Installieren Sie das Paket doc-linux-text, wenn Sie es offline lesen wollen. Wenn Sie Fragen stellen wollen oder Hilfe beim Einrichten einer Firewall bentigen, knnen Sie sich an die debian-firewall-Mailingliste wenden, siehe http://lists.debian.org/debian-firewall. Sehen Sie auch „Vorwissen“ fr weitere (allgemeinere) Verweise zu Firewalls. Ein weiterer guter Leitfaden fr Iptables ist http://iptables-tutorial.frozentux.net/iptables-tutorial.html.

Nutzen von Firewall-Paketen

Das manuelle Aufsetzen einer Firewall kann fr neue (und manchmal auch fr erfahrene) Administratoren kompliziert sein. Hierfr hat die Freie-Software-Gemeinschaft eine groe Zahl von Werkzeugen erstellt, die zur einfachen Konfiguration einer lokalen Firewall benutzt werden knnen. Seien Sie gewarnt, dass einige dieser Werkzeuge sich mehr auf lokalen Schutz konzentrieren (auch *personal firewall* genannt), whrend andere vielseitiger sind und dazu benutzt werden knnen, komplexere Regelwerke zum Schutz ganzer Netzwerke zu erstellen.

Einige Programme, die unter Debian zum Aufsetzen von Firewall-Regeln benutzt werden knnen, sind:

- Fr Desktop-Systeme:
  - firestarter, eine GNOME-Anwendung, die sich an Endanwender richtet, die einen Wizard enthlt, der ntzlich ist, um schnell Firewall-Regeln aufzustellen. Die Anwendung enthlt eine graphische Oberflche zum Beobachten, ob eine Firewall-Regel Daten blockiert.
  - guarddog ist ein auf KDE beruhendes Paket zur Erstellung von Firewall-Regeln. Es richtet sich sowohl an Neulinge wie auch an Fortgeschrittene.

- knetfilter ist ein KDE-Programm mit grafischer Oberflche, um Firewall- und NAT-Regeln fr iptables zu verwalten. Es ist eine Alternative zu guarddog, es ist jedoch etwas mehr auf fortgeschrittenere Benutzer ausgelegt.

- fireflier ist ein interaktives Werkzeug, um Firewall-Regeln zu erstellen. Dazu analysiert es den Netzwerkverkehr und Anwendungen. Es basiert auf einem Client-Server-Modell, daher mssen Sie sowohl den Server (fireflier-server) als auch einen der zahlreichen Clients (fireflier-client-gtk (Gtk+-Client), fireflier-client-kde (KDE-Client) oder fireflier-client-qt (QT-Client)) installieren.

- Fr Server-Systeme (textbasiert):

  - fwbuilder, eine objektorientierte graphische Oberflche, die Richtlinien-Compiler fr verschiedene Firewall-Plattformen inklusive Linux' netfilter, BSDs pf (verwendet in OpenBSD, NetBSD, FreeBSD und MacOS X) ebenso wie Zugriffslisten von Routern enthlt. Es ist hnlich zu Enterprise-Firewall-Management-Software. Die vollstndige Funktionalitt von fwbuilder ist auch von der Kommandozeile verfgbar.

  - shorewall, ein Firewall-Konfigurationswerkzeug, das Untersttzung fr IPsec sowie beschrnkte Untersttzung fr Traffic Shaping und die Definition der Firewall-Regeln bietet. Die Konfiguration geschieht durch eine einfache Menge von Dateien, die verwendet werden, um die iptables-Regeln aufzustellen.

  - bastille, diese Hrtungsanwendung ist in Kapitel 6, *Automatisches Abhärten von Debian-Systemen* beschrieben. Einer der Hrtungsschritte, die der Administrator konfigurieren kann, ist eine Definition des erlaubten und verbotenen Netzwerkverkehrs, der verwendet wird, eine Anzahl von Firewall-Regeln, die das System am Start ausfhrt, zu generieren.

Es gibt in Debian auch noch eine Menge anderer Frontends fr Iptables. Eine vollstndige Liste kann auf der http://wiki.debian.org/Firewalls, die auch einen Vergleich der verschiedenen Pakete enthlt, abgerufen werden.

Seien Sie gewarnt, dass manche der zuvor skizzierten Pakete Firewall-Skripte einfhren, die beim Systemstart ausgefhrt werden. Testen Sie diese ausfhrlich, bevor Sie neustarten, oder Sie finden sich selbst ausgesperrt vor Ihrem Rechner wieder. Wenn Sie verschiedene Firewall-Pakete mischen, kann dies zu unerwnschten Nebeneffekten fhren. Gewhnlich wird das Firewall-Skript, das zuletzt ausgefhrt wird, das System konfigurieren (was Sie so vielleicht nicht vorhatten). Sehen Sie hierzu in der Paketdokumentation nach und benutzen Sie nur eines dieser Setups.

Wie bereits zuvor erlutert, sind einige Programme wie firestarter, guarddog und knetfilter graphische Administrations-Schnittstellen, die entweder GNOME oder KDE (die letzte beiden) benutzen. Diese sind viel benutzerorientierter (z.B. fr Heimanwender) als einige der anderen Pakete in der Liste, die sich eher an Administratoren richten. Einige der Programme, die zuvor aufgefhrt wurden (wie **bastille**), fokussieren auf das Erstellen von Firewall-Regeln zum Schutz des Rechners, auf dem sie laufen, sind aber nicht notwendigerweise dafr geschaffen, Firewall-Regeln fr Rechner zu erstellen, die ein Netzwerk schtzen (wie **shorewall** oder **fwbuilder**).

Es gibt einen weiteren Typ von Firewall-Anwendungen: Anwendungs-Proxys. Wenn Sie eine Mglichkeit suchen, eine Unternehmenslsung aufzusetzen, die Pakete filtert und eine Anzahl von transparenten Proxys bietet, die feinabgestimmte Verkehrsanalysen bieten, so sollten Sie zorp genauer betrachten. Dies bietet alles in einem einzelnen Programm. Sie knnen diese Art von Firewall-Rechner auch manuell aufsetzen, indem Sie die Proxys, die in Debian vorhanden sind, fr verschiedene Dienste verwenden. Zum Beispiel fr DNS bind (richtig konfiguriert), dnsmasq, pdnsd oder totd fr FTP frox oder ftp-proxy, fr X11 xfwp, fr IMAP imapproxy, fr Mail smtpd oder fr POP3 p3scan. Fr andere Protokolle knnen Sie entweder einen allgemeinen TCP-Proxy wie simpleproxy oder einen allgemeinen SOCKS-Proxy wie dante-server, tsocks

oder socks4-server verwenden. Typischerweise werden Sie auch ein Web-Cache-System (wie squid) und ein Web-Filtersystem (wie squidguard oder dansguardian) nutzen.

Manuelle init.d-Konfiguration

Eine andere Mglichkeit ist die manuelle Konfiguration Ihrer Firewall-Regeln durch ein init.d-Skript, das die **iptables**-Befehle ausfhrt. Befolgen Sie diese Schritte:

- Sehen Sie das unten aufgefhrte Skript durch und passen Sie es Ihren Anforderungen an.
- Testen Sie das Skript und berprfen Sie die Syslog-Meldungen nach unterdrckten Netzverkehr. Wenn Sie vom Netzwerk aus testen, werden Sie entweder den Beispielshellcode starten wollen, um die Firewall zu entfernen (wenn Sie nichts innerhalb von 20 Sekunden eingeben) oder Sie sollten die *default deny*-Richtliniendefinition auskommentieren (*-P INPUT DROP* und *-P OUTPUT DROP*) und berprfen, dass das System keine gltigen Daten verworfen hat.
- Verschieben Sie das Skript nach `/etc/init.d/meineFirewall`
- The below script takes advantage of Debian's use (since Squeeze) of dependency based boot sequencing. For more information see: https://wiki.debian.org/LSBInitScripts/DependencyBasedBoot and https://wiki.debian.org/LSBInitScripts. With the LSB headers set as they are in the script, insserv will automatically configure the system to start the firewall before any network is brought up, and stop the firewall after any network is brought down.

```
# insserv myfirewall
```

Dies ist das Beispiel-Firewallskript:

```
#!/bin/sh
# Simple example firewall configuration.
#
# Caveats:
# - This configuration applies to all network interfaces
#   if you want to restrict this to only a given interface use
#   '-i INTERFACE' in the iptables calls.
# - Remote access for TCP/UDP services is granted to any host,
#   you probably will want to restrict this using '--source'.
#
# chkconfig: 2345 9 91
# description: Activates/Deactivates the firewall at boot time
#
# You can test this script before applying with the following shell
# snippet, if you do not type anything in 10 seconds the firewall
# rules will be cleared.
#------------------------------------------------------------------
#  while true; do test=""; read  -t 20 -p "OK? " test ; \
#  [ -z "$test" ] && /etc/init.d/myfirewall clear ; done
#------------------------------------------------------------------

PATH=/bin:/sbin:/usr/bin:/usr/sbin

# Services that the system will offer to the network
TCP_SERVICES="22" # SSH only
```

```
UDP_SERVICES=""
# Services the system will use from the network
REMOTE_TCP_SERVICES="80" # web browsing
REMOTE_UDP_SERVICES="53" # DNS
# Network that will be used for remote mgmt
# (if undefined, no rules will be setup)
# NETWORK_MGMT=192.168.0.0/24
# Port used for the SSH service, define this is you have setup a
# management network but remove it from TCP_SERVICES
# SSH_PORT="22"

if ! [ -x /sbin/iptables ]; then
    exit 0
fi

fw_start () {

  # Input traffic:
  /sbin/iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  # Services
  if [ -n "$TCP_SERVICES" ] ; then
  for PORT in $TCP_SERVICES; do
    /sbin/iptables -A INPUT -p tcp --dport ${PORT} -j ACCEPT
  done
  fi
  if [ -n "$UDP_SERVICES" ] ; then
  for PORT in $UDP_SERVICES; do
    /sbin/iptables -A INPUT -p udp --dport ${PORT} -j ACCEPT
  done
  fi
  # Remote management
  if [ -n "$NETWORK_MGMT" ] ; then
    /sbin/iptables -A INPUT -p tcp --src ${NETWORK_MGMT} --dport ${SSH_PORT} -j AC
  else
    /sbin/iptables -A INPUT -p tcp --dport ${SSH_PORT}  -j ACCEPT
  fi
  # Remote testing
  /sbin/iptables -A INPUT -p icmp -j ACCEPT
  /sbin/iptables -A INPUT -i lo -j ACCEPT
  /sbin/iptables -P INPUT DROP
  /sbin/iptables -A INPUT -j LOG

  # Output:
  /sbin/iptables -A OUTPUT -j ACCEPT -o lo
  /sbin/iptables -A OUTPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
  # ICMP is permitted:
  /sbin/iptables -A OUTPUT -p icmp -j ACCEPT
  # So are security package updates:
  # Note: You can hardcode the IP address here to prevent DNS spoofing
  # and to setup the rules even if DNS does not work but then you
  # will not "see" IP changes for this service:
  /sbin/iptables -A OUTPUT -p tcp -d security.debian.org --dport 80 -j ACCEPT
  # As well as the services we have defined:
  if [ -n "$REMOTE_TCP_SERVICES" ] ; then
```

```
  for PORT in $REMOTE_TCP_SERVICES; do
    /sbin/iptables -A OUTPUT -p tcp --dport ${PORT} -j ACCEPT
  done
  fi
  if [ -n "$REMOTE_UDP_SERVICES" ] ; then
  for PORT in $REMOTE_UDP_SERVICES; do
    /sbin/iptables -A OUTPUT -p udp --dport ${PORT} -j ACCEPT
  done
  fi
  # All other connections are registered in syslog
  /sbin/iptables -A OUTPUT -j LOG
  /sbin/iptables -A OUTPUT -j REJECT
  /sbin/iptables -P OUTPUT DROP
  # Other network protections
  # (some will only work with some kernel versions)
  echo 1 > /proc/sys/net/ipv4/tcp_syncookies
  echo 0 > /proc/sys/net/ipv4/ip_forward
  echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
  echo 1 > /proc/sys/net/ipv4/conf/all/log_martians
  echo 1 > /proc/sys/net/ipv4/ip_always_defrag
  echo 1 > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
  echo 1 > /proc/sys/net/ipv4/conf/all/rp_filter
  echo 0 > /proc/sys/net/ipv4/conf/all/send_redirects
  echo 0 > /proc/sys/net/ipv4/conf/all/accept_source_route

}

fw_stop () {
  /sbin/iptables -F
  /sbin/iptables -t nat -F
  /sbin/iptables -t mangle -F
  /sbin/iptables -P INPUT DROP
  /sbin/iptables -P FORWARD DROP
  /sbin/iptables -P OUTPUT ACCEPT
}

fw_clear () {
  /sbin/iptables -F
  /sbin/iptables -t nat -F
  /sbin/iptables -t mangle -F
  /sbin/iptables -P INPUT ACCEPT
  /sbin/iptables -P FORWARD ACCEPT
  /sbin/iptables -P OUTPUT ACCEPT
}


case "$1" in
  start|restart)
    echo -n "Starting firewall.."
    fw_stop
    fw_start
    echo "done."
    ;;
  stop)
```

```
    echo -n "Stopping firewall.."
    fw_stop
    echo "done."
    ;;
  clear)
    echo -n "Clearing firewall rules.."
    fw_clear
    echo "done."
    ;;
  *)
    echo "Usage: $0 {start|stop|restart|clear}"
    exit 1
    ;;
  esac
exit 0
```

Um nicht alle Iptables-Regeln in das Init.d-Skript einfgen zu mssen, knnen Sie auch das Programm **iptables-restore** verwenden, um die Regeln zu laden, die zuvor mit **iptables-save** gespeichert wurden. Um dies zu tun, mssen Sie Ihre Regeln erstellen und das Regelwerk statisch speichern (z.B. in `/etc/default/firewall`).

Konfiguration von Firewall-Regeln mittels ifup

Sie knnen auch die Netzwerkkonfiguration in `/etc/network/interfaces` verwenden, um Ihre Firewall-Regeln einzurichten. Dafr mssen Sie Folgendes tun:

- Erstellen Sie Ihre Firewall-Regeln fr die aktivierte Schnittstelle.
- Sichern Sie Ihre Regeln mit **iptables-save** in eine Datei in `/etc`, zum Beispiel `/etc/iptables.up.rules`.
- Konfigurieren Sie `/etc/network/interfaces`, diese Regeln zu verwenden:

```
iface eth0 inet static
        address x.x.x.x
        [.. interface configuration ..]
        pre-up iptables-restore < /etc/iptables.up.rules
```

Wahlweise knnen Sie auch Regeln erstellen, die beim Herunterfahren der Netzwerkschnittstelle ausgefhrt werden. Dazu erzeugen Sie diese, speichern sie in `/etc/iptables.down.rules` und fgen diese Anweisung zur Schnittstellenkonfiguration hinzu:

```
    post-down iptables-restore < /etc/iptables.down.rules
```

Fr weitergehende Firewall-Konfigurationsskripte durch ifupdown knnen Sie die zu jeder Schnittstelle verfgbaren Hooks (Einspringpunkte) wie in den `*.d/`-Verzeichnissen verwenden, die mit **run-parts** aufgerufen werden (vergleiche run-parts(8)).

Testen Ihrer Firewall-Konfiguration

Testen Ihrer Firewall-Konfiguration ist so einfach und so schwierig, wie das Starten Ihres Firewall-Skripts (oder die Aktivierung der Konfiguration, die Sie in Ihrer Firewall-Konfigurationsanwendung definierten). Wenn Sie jedoch nicht sorgfltig genug sind und Sie Ihre Firewall aus der Ferne konfigurieren (z.B. durch eine SSH-Verbindung), knnten Sie sich selbst aussperren.

Es gibt mehrere Mglichkeiten, dies zu verhindern. Eine ist das Starten eines Skriptes in einem separaten Terminal, das Ihre Firewall-Konfiguration entfernt, wenn es keine Eingabe von Ihnen erhlt. Ein Beispiel dafr ist:

```
$  while true; do test=""; read  -t 20 -p "OK? " test ; \
  [ -z "$test" ] && /etc/init.d/firewall clear ; done
```

Eine andere Mglichkeit ist das Einfhren einer Hintertr in Ihr System durch einen alternativen Mechanismus, der es Ihnen erlaubt, das Firewall-System entweder zurckzusetzen oder ein Loch in es schlgt, wenn irgendetwas krumm luft. Dafr knnen Sie knockd verwenden und es so konfigurieren, dass eine spezielle Portverbindungsversuchssequenz die Firewall zurcksetzt (oder eine temporre Regel hinzufgt). Selbst wenn die Pakete von der Firewall zurckgewiesen werden, werden Sie Ihr Problem lsen knnen, da **knockd** auf der Schnittstelle lauscht und Sie *sieht*.

Das Testen einer Firewall, die ein internes Netz schützt, ist eine andere Aufgabe. Schauen Sie sich dafür einige Werkzeuge an, die es für entfernte Ausnutzbarkeitsbewertungen gibt (siehe „Programme zur Fernprüfung der Verwundbarkeit“), um das Netzwerk von außerhalb nach innen (oder aus einer beliebig anderen Richtung) bezüglich der Effektivität der Firewall-Konfiguration zu testen.

Kapitel 6. Automatisches Abhärten von Debian-Systemen

Nachdem Sie nun all die Informationen aus den vorherigen Kapiteln gelesen haben, fragen Sie sich vielleicht: »Ich habe sehr viele Dinge zu erledigen, um mein System abzusichern. Könnte man das nicht automatisieren?« Die Antwort lautet: »Ja, aber seien Sie vorsichtig mit automatischen Werkzeugen.« Manche Leute denken, dass ein Absicherungswerkzeug nicht die Notwendigkeit einer guten Systemadministration abschafft. Täuschen Sie sich also nicht selbst, indem Sie denken, dass Sie all die Prozesse automatisieren könnten und sich alle betreffenden Angelegenheiten von selbst erledigen würden. Sicherheit ist ein andauernder Prozess, an dem der Administrator teilnehmen muss. Er kann nicht einfach wegbleiben und irgendwelche Werkzeuge die Arbeit erledigen lassen, weil kein einzelnes Werkzeug die Umsetzung aller Sicherheitsrichtlinien, aller Angriffe oder aller Umgebungen bewältigen kann.

Seit Woody (Debian 3.0) gibt es zwei unterschiedliche Pakete, die zur Erhöhung der Sicherheit nützlich sind. Das Paket harden versucht, auf Basis der Paket-Abhängigkeiten schnell wertvolle Sicherheitspakete zu installieren und Pakete mit Mängeln zu entfernen. Die Konfiguration der Pakete muss der Administrator erledigen. Das Paket bastille implementiert gegebene Sicherheitsregeln für das lokale System, die auf einer vorhergehenden Konfiguration durch den Administrator basieren (Sie können auch mit einfachen Ja/Nein-Fragen durch die Konfiguration geführt werden).

Harden

Das Paket harden versucht es einfacher zu machen, Rechner, die gute Sicherheit benötigen, zu installieren und zu administrieren. Dieses Paket sollte von Leuten benutzt werden, die eine schnelle Hilfe bei der Erhöhung der Systemsicherheit haben wollen. Es installiert automatisch einige Werkzeuge, die die Sicherheit auf unterschiedliche Art und Weise erhöhen: Werkzeuge zur Eindringlingserkennung, Werkzeuge zur Sicherheitsanalyse und mehr. harden installiert die folgenden *virtuellen* Pakete (d.h. sie enthalten nichts, hängen aber von anderen Paketen ab oder empfehlen diese):

- harden-tools: Werkzeuge, welche die Sicherheit des Systems erhöhen (Integritätsprüfung, Eindringlingserkennung, Kernel-Patches, ...)
- harden-environment: hilft eine abgesicherte Umgebung zu konfigurieren (derzeit leer)
- harden-servers: entfernt Server, die aus irgendeinem Grund als unsicher gelten
- harden-clients: entfernt Clients, die aus irgendeinem Grund als unsicher gelten
- harden-remoteaudit: Werkzeuge, um Systeme aus der Ferne zu überprüfen
- harden-nids: hilft bei der Installation eines Systems zur Entdeckung von Netzwerkeindringlingen
- harden-surveillance: hilft bei der Installation von Werkzeugen zum Überwachen von Netzwerken und Diensten

Nützliche Pakete, für die keine Abhängigkeit besteht:

- harden-doc: stellt dieses und andere sicherheitsrelevante Dokumente zur Verfügung
- harden-development: Entwicklungswerkzeuge, um sicherere Programme zu erstellen

Seien Sie vorsichtig, wenn Sie Software installiert haben, die Sie brauchen (und aus bestimmten Gründen nicht deinstallieren wollen) und die aufgrund eines Konflikts mit einem der oben aufgeführten Pakete nicht

installiert werden kann. In diesem Fall können Sie harden nicht vollständig nutzen. Die harden-Pakete machen eigentlich gar nichts. Zumindest nicht unmittelbar. Sie haben jedoch absichtliche Paketkonflikte mit bekannten, unsicheren Paketen. Auf diese Art wird die Paketverwaltung von Debian die Installation dieser Paketen nicht erlauben. Wenn Sie zum Beispiel bei installiertem harden-servers-Paket versuchen, einen telnet-Daemon zu installieren, wird Ihnen apt Folgendes sagen:

```
# apt-get install telnetd
Die folgenden Pakete werden ENTFERNT:
  harden-servers
Die folgenden NEUEN Pakete werden installiert:
  telnetd
Möchten Sie fortfahren? [J/n]
```

Dies sollte im Kopf des Administrators eine Alarmglocke auslösen, der sein Vorgehen überdenken sollte.

Bastille Linux

http://www.bastille-unix.org ist ein Werkzeug zur automatischen Abhärtung, das ursprünglich für die Linux-Distributionen Red Hat und Mandrake gedacht war. Wie auch immer: Das Paket bastille aus Debian (seit Woody) ist durch Patches angepasst, um dieselbe Funktionalität unter Debian GNU/Linux Systemen zur Verfügung zu stellen.

Bastille kann mit verschiedenen Oberflächen bedient werden (alle sind in ihrem eigenen Handbuch dokumentiert), die dem Administrator erlauben:

- Schritt für Schritt Fragen zur erwünschten Sicherheit Ihres Systems zu beantworten (siehe Interactuve-Bastille(8)),
- Standardeinstellungen zur Sicherheit (zwischen locker, moderat und paranoid) für eine bestimmte Einrichtung (Server oder Arbeitsplatz-Rechner) zu benutzen, und Bastille entscheiden zu lassen, welche Sicherheitsregelungen eingeführt werden sollen (siehe BastilleChooser(8)),
- eine vorgefertigte Konfigurationsdatei (von Bastille oder vom Administrator) zu nehmen und eine vorgegebene Sicherheitsregelung zu benutzen (siehe AutomatedBastille(8)).

Kapitel 7. Die Infrastruktur für Sicherheit in Debian

Das Sicherheitsteam von Debian

Debian hat ein Sicherheitsteam, das aus fünf Mitgliedern und zwei Sekretären besteht. Es ist für die Sicherheit in der *Stable*-Veröffentlichung verantwortlich. Das bedeutet, dass es Sicherheitslücken nachgeht, die in Software auftauchen (indem es Foren wie Bugtraq oder vuln-dev beobachtet), und ermittelt, ob davon die *Stable*-Veröffentlichung betroffen ist.

Das Sicherheitsteam von Debian ist auch der Ansprechpartner für Probleme, die von den Programmautoren oder Organisationen wie http://www.cert.org behandelt werden und die mehrere Linux-Anbieter betreffen können. Das gilt für alle Probleme, die nicht debianspezifisch sind. Die Kontaktadresse des Sicherheitsteams ist hmailto:team@security.debian.org, die nur die Mitglieder des Sicherheitsteams lesen.

Heikle Informationen sollten an die erste Adresse geschickt werden und unter Umständen mit dem Schlüssel von Debian Security Contact (der sich in Debians Schlüsselbund befindet) verschlüsselt werden.

Wenn das Sicherheitsteam ein mögliches Problem erhält, wird es untersuchen, ob die *Stable*-Veröffentlichung davon betroffen ist. Wenn dies der Fall ist, wird eine Ausbesserungen des Quellcodes vorgenommen. Diese Ausbesserung schließt manchmal ein, dass Patches der Programmautoren zurückportiert werden (da das Originalprogramm gewöhnlich einige Versionen weiter ist als das in Debian). Nachdem die Ausbesserung getestet wurde, werden neue Pakete vorbereitet und auf der Seite hsecurity-master.debian.org veröffentlicht, damit sie mit **apt** abgerufen werden können (siehe „Ausführen von Sicherheitsaktualisierungen“). Zur gleichen Zeit wird eine *Debian-Sicherheits-Ankündigung* (DSA) auf der Webseite veröffentlicht und an öffentliche Mailinglisten einschließlich http://lists.debian.org/debian-security-announce und Bugtraq geschickt.

Einige andere häufige Fragen zum Sicherheitsteam von Debian können unter „Fragen zu Debians Sicherheitsteam“ gefunden werden.

Debian-Sicherheits-Ankündigungen

Debian-Sicherheits-Ankündigungen (DSA) werden erstellt, sobald eine Sicherheitslücke entdeckt wird, die ein Debian-Paket betrifft. Diese Ankündigungen, die von einem Mitglied des Sicherheitsteams signiert sind, enthalten Informationen zu den betroffenen Versionen und den Orten der Aktualisierungen und ihrer MD5-Summen. Die Informationen sind:

- Versionsnummer der Ausbesserung
- Art des Problems
- Ob es aus der Ferne oder lokal ausnutzbar ist
- Kurze Beschreibung des Pakets
- Beschreibung des Problems
- Beschreibung des Exploits
- Beschreibung der Ausbesserung

DSAs werden sowohl auf der http://www.de.debian.org/ als auch auf den http://www.debian.org/security/ veröffentlicht. Das passiert normalerweise nicht, bis die Website neu erstellt wurde (alle vier Stunden). Daher könnten sie nicht sofort vorhanden sein. Somit ist die vorzugswürdige Informationsquelle die Mailingliste debian-security-announce.

Interessierte Benutzer können auch den RDF-Kanal verwenden, um die DSAs automatisch auf ihren Desktop herunterzuladen (dies wird auf einigen Portalen über Debian gemacht). Einige Anwendungen, wie etwa **Evolution** (ein E-Mail-Client und Hilfsprogramm für persönliche Informationen) und **Multiticker** (ein GNOME-Applet) können verwendet werden, um die Ankündigungen automatisch herunterzuladen. Der RDF-Kanal befindet sich unter http://www.debian.org/security/dsa.rdf.

DSAs, die auf der Webseite veröffentlicht wurden, können aktualisiert werden, nachdem sie an öffentliche Mailinglisten verschickt wurden. Eine typische Aktualisierung ist, einen Querverweis auf Datenbanken mit Sicherheitslücken hinzuzufügen. Auch Übersetzungen der DSAs [1] werden nicht an die Sicherheitsmailinglisten geschickt, sondern sind direkt auf der Webseite enthalten.

Querverweise der Verwundbarkeiten

Debian stellt eine vollständige http://www.debian.org/security/crossreferences zur Verfügung, die alle verfügbaren Verweise für die Ankündigungen seit 1998 enthält. Diese Tabelle soll die http://cve.mitre.org/cve/refs/refmap/source-DEBIAN.html ergänzen.

Sie werden bemerken, dass die Tabelle Verweise auf Sicherheitsdatenbanken wie http://www.securityfocus.com/bid, http://www.cert.org/advisories/ und http://www.kb.cert.org/vuls und auf die CVE-Bezeichnungen (siehe unten) enthält. Diese Verweise werden zur Nutzerfreundlichkeit angeboten, aber nur der CVE-Verweise werden regelmäßig überprüft und eingefügt. Dieses Feature wurde im Juni 2002 der Webseite hinzugefügt.

Das Hinzufügen von Querverweisen auf diese Sicherheitsdatenbanken hat folgende Vorteile:

- Es erleichtert Benutzern von Debian zu erkennen und nachzuvollziehen, welche allgemeinen (veröffentlichten) Ankündigungen schon von Debian abgedeckt sind.
- Systemadministratoren können mehr über die Verwundbarkeit und ihre Auswirkungen lernen, wenn sie den Querverweisen folgen.
- Diese Informationen können benutzt werden, um Ausgaben von Verwundbarkeitsscannern, die Verweise auf CVE enthalten, zu überprüfen, um falsche Positivmeldungen auszusortieren (vergleichen Sie „Der Scanner X zur Einschätzung der Verwundbarkeit sagt, dass mein Debian-System verwundbar wäre?“).

CVE-Kompatibilität

Debians Sicherheitsankündigungen wurden am 24. Februar 2004 http://www.de.debian.org/security/CVE-certificate.jpg [2].

Debian developers understand the need to provide accurate and up to date information of the security status of the Debian distribution, allowing users to manage the risk associated with new security vulnerabilities. CVE enables us to provide standardized references that allow users to develop a https://cve.mitre.org/compatible/enterprise.html.

Das Projekt http://cve.mitre.org wird von der MITRE Corporation betreut und stellt eine Liste von standardisierten Bezeichnungen für Verwundbarkeiten und Sicherheitslücken zur Verfügung.

[1] Übersetzungen sind in bis zu zehn verschiedenen Sprachen verfügbar.
[2] Der vollständige http://cve.mitre.org/compatible/phase2/SPI_Debian.html ist bei CVE erhältlich.

Debian ist überzeugt, dass es außerordentlich wichtig ist, die Benutzer mit zusätzlichen Informationen im Zusammenhang mit Sicherheitsproblemen, welche die Debian-Distribution betreffen, zu versorgen. Indem CVE-Bezeichnungen in den Ankündigungen enthalten sind, können Benutzer leichter allgemeine Verwundbarkeiten mit bestimmten Aktualisierungen von Debian in Verbindung bringen. Dies verringert die Zeit, die benötigt wird, um Verwundbarkeiten, die unsere Benutzer betreffen, abzuarbeiten. Außerdem vereinfacht es die Organisation der Sicherheit in einer Umgebung, in der schon Sicherheitswerkzeuge, die CVE verwenden, wie Erkennungssysteme von Eindringlingen in Netzwerk oder Host oder Werkzeuge zur Bewertung der Sicherheit eingesetzt werden, unabhängig davon, ob sie auf der Debian-Distribution beruhen.

Debian stellt CVE-Bezeichnungen sind in allen DSAs seit September 1998 zur Verfügung. Alle Ankündigungen können auf der Webseite von Debian abgerufen werden. Auch Ankündigungen von neuen Verwundbarkeiten enthalten CVE-Bezeichnungen, wenn sie zum Zeitpunkt ihrer Veröffentlichung verfügbar waren. Ankündigungen, die mit einer bestimmten CVE-Bezeichnung verbunden sind, können direkt über Debians Sicherheitsdatenbank (Debian Security Tracker) gesucht werden (siehe unten).

In einige Fällen finden Sie eine bestimmte CVE-Bezeichnung in veröffentlichten Ankündigungen nicht. Beispiele dafür sind:

- Keine Produkte von Debian sind von der Verwundbarkeit betroffen.
- Es gibt noch keine Ankündigung, welche die Verwundbarkeit abdeckt; das Sicherheitsproblem wurde vielleicht als http://bugs.debian.org/cgi-bin/pkgreport.cgi?tag=security gemeldet, aber eine Ausbesserung wurde noch nicht getestet und hochgeladen.
- Eine Ankündigung wurde veröffentlicht, bevor eine CVE-Bezeichnung einer bestimmten Verwundbarkeit zugewiesen wurde (sehen Sie auf der Webseite nach einer Aktualisierung).

Sicherheitsdatenbank

The central database of what the Debian security teams know about vulnerabilities is the http://security-tracker.debian.org. It cross references packages, vulnerable and fixed versions for different suites, CVE names, Debian bug numbers, DSA's and miscellaneous notes. It can be searched, e.g. by CVE name to see which Debian packages are affected or fixed, or by package to show unresolved security issues. The only information missing from the tracker is confidential information that the security team received under embargo.

Das Paket **debsecan** verwendet die Informationen in der Datenbank, zum den Administrator eines Systems darüber zu informieren, welche der installierten Pakete verwundbar sind und für welche Pakete Sicherheitsaktualisierungen verfügbar sind.

Die Infrastruktur des Sicherheitsprozesses in Debian

Da Debian im Moment eine große Anzahl von Architekturen unterstützt, fragen Administratoren manchmal, ob es bei einer bestimmten Architektur bis zu einer Sicherheitsaktualisierung länger dauert als bei einer anderen. Tatsächlich sind Aktualisierungen auf allen Architekturen zur selben Zeit verfügbar, abgesehen von seltenen Umständen.

Pakete im Sicherheitsarchiv werden automatisch erstellt, genauso wie im normalen Archiv. Allerdings werden Sicherheitsaktualisierungen etwas anderes behandelt als normale Aktualisierungen, die von den Paketbetreuern vorgenommen werden, da in manchen Fällen vor einer Veröffentlichung die Aktualisierungen nochmals getestet werden müssen, eine Ankündigung geschrieben werden muss oder eine Woche

oder mehr gewartet werden muss, um zu verhindern, dass der Fehler veröffentlicht wird, bevor nicht alle Linux-Anbieter eine vernünftige Chance hatten, ihn zu beheben.

Folglich arbeitet das Archiv der Sicherheitsuploads nach dem folgenden Ablauf :

- Jemand findet ein Sicherheitsproblem.
- Jemand löst das Problem und lädt die Lösung in den Eingang von security-master.debian.org hoch (dieser *jemand* ist normalerweise ein Mitglied des Sicherheitsteams, kann aber auch ein Paketbetreuer mit einer passenden Verbesserung sein, der sich zuvor mit dem Sicherheitsteam in Verbindung gesetzt hat). Die Änderungsübersicht (changelog) beinhaltet ein *testing-security* oder *stable-security* als Zieldistribution.
- Die hochgeladenen Dateien werden von einem Debian-System überprüft, verarbeitet und in die Warteschleife der angenommenen Dateien weitergeleitet. Danach werden die Buildds benachrichtigt. Auf die Dateien in der Warteschleife kann das Sicherheitsteam und (auf indirektem Wege) die Buildds zugreifen.
- Buildds, die Sicherheit unterstützen, holen sich das Quellpaket (mit einer höheren Priorität als normale Paketerstellungen), erstellen Pakete und schicken die Protokolle ans Sicherheitsteam.
- Das Sicherheitsteam antwortet auf die Protokolle und die neu erstellten Pakete werden in die Warteschleife der ungeprüften Dateien hochgeladen, wo sie von einem Debian-System verarbeitet und in die Warteschleife der angenommenen Dateien verschoben werden.
- Wenn das Sicherheitsteam ein Quellpaket akzeptiert (d.h. dass es für alle Architekturen korrekt Pakete erstellt, und dass es die Sicherheitslücke schließt und keine neuen Probleme hervorruft), führt es ein Skript aus, das:
  - das Paket im Sicherheitsarchiv installiert,
  - die `Paket-`, `Quell-` und `Veröffentlichungsdateien` von security.debian.org auf dem gewöhnlichen Weg aktualisiert (**dpkg-scanpackages**, **dpkg-scansources**, ...),
  - eine Vorlage einer Ankündigung erstellt, die das Sicherheitsteam fertig stellen kann und
  - die Pakete zu den vorgeschlagenen Aktualisierungen weiterleitet, so dass sie sobald wie möglich in die echten Archive eingefügt werden können.

Dieser Ablauf, der früher per Hand durchgeführt wurde, wurde während des Freezing-Abschnitts von Debian 3.0 Woody (Juli 2002) getestet und umgesetzt. Dank dieser Infrastruktur war es dem Sicherheitsteam möglich, aktualisierte Pakete für Apache- und OpenSSH-Probleme für alle unterstützen Architekturen (fast 20) in weniger als einem Tag bereitzustellen.

Leitfaden über Sicherheitsaktualisierungen für Entwickler

Debian Entwickler, die mit dem Sicherheitsteam zusammenarbeiten müssen, um in ihren Pakete Probleme zu lösen, sollten in der Entwicklerreferenz im Abschnitt http://www.debian.org/doc/manuals/developers-reference/pkgs.html#bug-security nachsehen.

Paketsignierung in Debian

Dieser Abschnitt könnte auch mit »Wie man sein Debian GNU/Linux-System sicher upgraded/aktualisiert« überschrieben werden. Es verdient hauptsächlich deshalb einen eigenen Abschnitt, weil es einen

wichtigen Teil der Infrastruktur der Sicherheit darstellt. Die Signierung von Paketen ist ein wichtiges Thema, da es die Manipulation von Paketen in Spiegel und von heruntergeladenen Dateien durch Man-in-the-Middle-Angriffen verhindert. Die automatische Aktualisierung von Software ist eine wichtige Fähigkeit, aber es ist auch wichtig, Gefahren für die Sicherheit zu entfernen, die die Verbreitung von Trojanern und den Einbruch ins System während der Aktualisierung fördern können.[3]

FIXME: probably the Internet Explorer vulnerability handling. certificate chains has an impact on security updates on Microsoft Windows.

Debian stellt keine signierten Pakete zur Verfügung. Es gibt aber seit Debian 4.0 (Codename *Etch*) eine Verfahrensweise, mit der die Integrität von heruntergeladenen Paketen überprüft werden kann.[4] Weiterführende Hinweise können Sie unter „Secure Apt“ finden.

Dieses Problem wird besser im http://www.cryptnet.net/fdp/crypto/strong_distro.html von V. Alex Brennen beschrieben.

Die aktuelle Methode zur Prüfung von Paketsignaturen

Die aktuelle Methode zur Prüfung von Paketsignaturen mit **apt** ist:

- Die `Release`-Datei enthält die MD5-Summe von `Packages.gz` (welche die MD5-Summen der Pakete enthält) und wird signiert. Die Signatur stammt aus einer vertrauenswürdigen Quelle.
- Diese signierte `Release`-Datei wird beim »apt-get update« herunter geladen und zusammen mit `Packages.gz` gespeichert.
- Wenn ein Paket installiert werden soll, wird es zuerst herunter geladen, und dann wird die MD5-Summe erstellt.
- Die signierte `Release`-Datei wird überprüft (ob die Signatur in Ordnung ist) und die MD5-Summe der `Packages.gz`-Datei extrahiert. Die MD5-Summe der `Packages.gz`-Datei wird erstellt und geprüft, und – wenn sie übereinstimmt – wird die MD5-Summe des heruntergeladenen Paketes aus ihr extrahiert.
- Wenn die MD5-Summe des heruntergeladenen Paketes die gleiche ist wie in der `Packages.gz`-Datei, wird das Paket installiert. Andernfalls wird der Administrator alarmiert und das Paket wird im Zwischenspeicher gehalten (so dass der Administrator entscheiden kann, ob es installiert werden soll oder nicht). Wenn das Paket nicht in `Packages.gz` enthalten ist und der Administrator das System so konfiguriert hat, dass nur geprüfte Pakete installiert werden können, wird das Paket ebenfalls nicht installiert.

Durch diese Kette von MD5-Summen ist **apt** in der Lage, zu verifizieren, dass ein Paket aus einer bestimmten Veröffentlichung stammt. Dies ist zwar unflexibler als jedes Paket einzeln zu signieren, kann aber auch mit den unten aufgeführten Plänen kombiniert werden.

This scheme is http://lists.debian.org/debian-devel/2003/12/msg01986.html in apt 0.6 and is available since the Debian 4.0 release. For more information see „Secure Apt“. Packages that provide a front-end to apt need to be modified to adapt to this new feature; this is the case of **aptitude** which was http://lists.debian.org/debian-devel/2005/03/msg02641.html to adapt to this scheme. Front-ends currently known to work properly with this feature include **aptitude** and **synaptic**.

Die Signierung von Paketen wurde innerhalb des Debian-Projekts ausführlich diskutiert. Mehr Informationen hierzu finden Sie unter http://www.debian.org/News/weekly/2001/8/ und http://www.debian.org/News/weekly/2000/11/.

[3] Einige Betriebssysteme wurden schon von Problemen mit automatischen Aktualisierungen heimgesucht, wie z.B. die http://www.cunap.com/~hardingr/projects/osx/exploit.html.
[4] Ältere Veröffentlichungen wie Debian 3.1 (*Sarge*) können mit zurückportierten Versionen des Paketmanagers auf diese Methode zugreifen.

Secure Apt

Die Veröffentlichung von apt 0.6, das seit Debian 4.0 (*Etch*) verfügbar ist, enthält *apt-secure* (auch als *Secure Apt* bekannt), das ein Werkzeug ist, mit dem ein Systemadministrator die Integrität von heruntergeladenen Paketen mit dem oben dargestellten Verfahren überprüfen kann. Diese Veröffentlichung enthält das Werkzeug **apt-key**, um neue Schlüssel zum Schlüsselbund von apt hinzuzufügen, welcher standardmäßig nur den aktuellen Signierungsschlüssel des Debian-Archivs enthält.

Diese Veränderungen basieren auf dem Patch für **apt** (verfügbar in http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=203741), der diese Erweiterung zur Verfügung stellt.

Secure Apt überprüft die Distribution mit der `Release`-Datei. Dies wurde schon unter „Überprüfung der Distribution mit der `Release`-Datei“ dargestellt. Typischerweise erfordert dieser Vorgang kein Mitwirken des Administrators. Aber jedes Jahr müssen Sie eingreifen[5], um den neuen Schlüssel des Archivs hinzuzufügen, wenn dieser ausgewechselt wurde. Weitere Informationen zu den dazu notwendigen Schritten finden Sie unter „Auf sichere Weise einen Schlüssel hinzufügen“.

Diese Fähigkeit befindet sich noch im Entwicklungsstadium. Wenn Sie glauben, dass Sie Fehler gefunden haben, stellen Sie zuerst sicher, dass Sie die neuste Version verwenden (da dieses Paket vor seiner endgültigen Veröffentlichung noch ziemlich verändern werden kann). Falls Sie die aktuelle Version benutzen, schicken Sie einen Fehlerbericht für das Paket apt.

You can find more information at http://wiki.debian.org/SecureApt and the official documentation: http://www.enyo.de/fw/software/apt-secure/ and https://web.archive.org/web/20070206063141/http://www.syntaxpolice.org/apt-secure/.

Überprüfung der Distribution mit der `Release`-Datei

Dieser Abschnitt beschreibt, wie die Überprüfung der Distribution mit Hilfe der `Release`-Datei funktioniert. Dies wurde von Joey Hess geschrieben und ist auch im http://wiki.debian.org/SecureApt abrufbar.

Grundlegende Konzepte

Es gibt ein paar grundlegende Konzepte, die Sie brauchen, um den Rest dieses Abschnitts verstehen zu können.

Eine Prüfsumme ist eine Methode, bei der eine Datei auf eine relativ kurze Zahl heruntergekocht wird, mit welcher der Inhalt der Datei eindeutig identifiziert werden kann. Dies ist wesentlich schwieriger, als es zunächst erscheinen mag. Der am weitesten verbreiteteste Typ von Prüfsummen, MD5, wird gerade unbrauchbar.

Verschlüsselung mit öffentlichen Schlüsseln fußt auf einem Schlüsselpaar: einem öffentlichen Schlüssel und einem privaten Schlüssel. Der öffentliche Schlüssel wird an die Allgemeinheit verteilt. Der private muss geheim bleiben. Jeder, der den öffentlichen Schlüssel hat, kann eine Nachricht verschlüsseln, so dass sie nur noch der Besitzer des privaten Schlüssels lesen kann. Es besteht daneben die Möglichkeit, mit einem privaten Schlüssel eine Datei zu signieren. Wenn eine Datei mit einer digitalen Unterschrift versehen wurde, kann jeder, der den öffentlichen Schlüssel hat, überprüfen, ob die Datei mit diesem Schlüssel unterschrieben wurde. Ohne den privaten Schlüssel lässt sich eine solche Signatur nicht nachmachen.

Diese Schlüssel bestehen aus ziemlich langen Zahlen (1024 oder 2048 Ziffern oder sogar länger). Damit sie leichter zu verwenden sind, haben sie eine kürzere Schlüssel-ID (eine Zahl mit nur acht oder 16 Stellen), mit der sie bezeichnet werden können.

Secure Apt verwendet **gpg**, um Dateien zu unterschreiben und ihre Signaturen zu überprüfen.

[5] Bis ein automatischer Mechanismus entwickelt wurde.

Mit dem Programm **apt-key** wird der Schlüsselbund von GPG für Secure Apt verwaltet. Der Schlüsselbund befindet sich in der Datei `/etc/apt/trusted.gpg` (nicht zu verwechseln mit der verwandten, aber nicht sehr interessanten Datei `/etc/apt/trustdb.gpg`). **apt-key** kann dazu verwendet werden, die Schlüssel im Schlüsselbund anzuzeigen oder um Schlüssel hinzuzufügen oder zu entfernen.

Prüfsummen der `Release`-Datei

Jedes Archiv von Debian enthält eine `Release`-Datei, die jedes Mal aktualisiert wird, wenn ein Paket im Archiv geändert wird. Unter anderem enthält die `Release`-Datei MD5-Summen von anderen Dateien, die sich im Archiv befinden. Ein Auszug einer `Release`-Datei:

```
MD5Sum:
 6b05b392f792ba5a436d590c129de21f            3453 Packages
 1356479a23edda7a69f24eb8d6f4a14b            1131 Packages.gz
 2a5167881adc9ad1a8864f281b1eb959            1715 Sources
 88de3533bf6e054d1799f8e49b6aed8b             658 Sources.gz
```

Die `Release`-Datei enthält auch SHA1-Prüfsummen, was nützlich wird, wenn MD5-Summen vollständig unbrauchbar sind. Allerdings unterstützt apt SHA1 noch nicht.

Werfen wir einen Blick in eine `Packages`-Datei: Wir sehen weitere MD5-Summen, eine für jedes darin aufgeführte Paket. Beispiel:

```
Package: uqm
Priority: optional
...
Filename: unstable/uqm_0.4.0-1_i386.deb
Size: 580558
MD5sum: 864ec6157c1eea88acfef44d0f34d219
```

Mit diesen beiden Prüfsummen kann überprüft werden, ob Sie eine getreue Kopie der `Packages`-Datei heruntergeladen haben, also mit einer MD5-Summe, die mit der in der `Release`-Datei übereinstimmt. Und wenn ein einzelnes Paket heruntergeladen wird, kann auch die MD5-Summe mit dem Inhalt der `Packages`-Datei verglichen werden. Wenn bei einem dieser Schritte ein Fehler auftauchen sollte, bricht Apt den Vorgang ab.

Nichts davon ist neu in Secure Apt, sondern bietet nur die Grundlage für Secure Apt. Beachten Sie, dass es bis jetzt eine Datei gibt, die Apt nicht überprüfen kann: die `Release`-Datei. Bei Secure Apt dreht sich alles darum, dass Apt die `Release`-Datei überprüft, bevor es irgendetwas anderes damit macht. Wenn man das schafft, besteht eine lückenlose Authentifizierungskette von dem Paket, das Sie installieren möchten, bis zum Anbieter des Pakets.

Überprüfung der `Release`-Datei

Damit die `Release`-Datei überprüft werden kann, wird sie mit GPG signiert. Diese Unterschrift kommt in die Datei `Release.gpg`, die mit der `Release`-Datei abgerufen werden kann. Sie sieht in etwa so[6] aus, obwohl sich für gewöhnlich nur GPG ihren Inhalt ansieht:

```
-----BEGIN PGP SIGNATURE-----
Version: GnuPG v1.4.1 (GNU/Linux)
```

[6] Genau genommen handelt es sich um eine ASCII-armored abgetrennte GPG-Signatur.

```
iD8DBQBCqKO1nukh8wJbxY8RAsfHAJ9hu8oGNRAl2MSmP5+z2RZb6FJ8kACfWvEx
UBGPVc7jbHHsg78EhMBlV/U=
=x6og
-----END PGP SIGNATURE-----
```

`Release.gpg` mit Apt überprüfen

Wenn Secure Apt eine `Release`-Datei herunterlädt, lädt es immer auch die `Release.gpg`-Datei herunter. Falls dies misslingen sollte oder die Signatur nicht stimmt, wird es eine Rückmeldung machen und hinweisen, dass die `Packages`-Dateien, auf welche die `Release`-Datei verweist, und alle darin enthaltenen Pakete von einer nicht vertrauenswürdigen Quelle stammen. So würde dies während **apt-get update** aussehen:

```
W: GPG error: http://ftp.us.debian.org testing Release: The following signatures
 couldn't be verified because the public key is not available: NO_PUBKEY 010908312
```

Beachten Sie, dass die zweite Hälfte der langen Nummer die Schlüssel-ID des Schlüssels ist, von dem Apt nichts weiß. Im Beispiel ist sie 2D230C5F.

Falls Sie diese Warnung ignorieren und später versuchen, ein Paket zu installieren, wird Sie Apt nochmals warnen:

```
WARNUNG: Die folgenden Pakete können nicht authentifiziert werden!
  libglib-perl libgtk2-perl
Diese Pakete ohne Überprüfung installieren [j/N]?
```

Wenn Sie nun »J« drücken, haben Sie keine Möglichkeit festzustellen, ob die Datei, die Sie bekommen, wirklich diejenige ist, die Sie auch installieren möchten, oder ob sie eine ganz andere ist, die Ihnen jemand, der die Verbindung mit dem Server abgefangen hat[7], mit einer gemeinen Überraschung unterschieben will.

Hinweis: Sie können diese Abfragen abschalten, indem Sie **apt** mit `--allow-unauthenticated` laufen lassen.

Es lohnt sich auch noch darauf hinzuweisen, dass der Installer von Debian während des Debootstraps des Basissystems, solange Apt noch nicht verfügbar ist, denselben Mechanismus mit signierten `Release`-Dateien verwendet. Der Installer benutzt sogar dieses Verfahren, um Teile von sich selbst zu überprüfen, die er aus dem Netz gezogen hat. Debian signiert im Moment nicht die `Release`-Dateien auf den CDs. Apt kann aber so eingerichtet werden, dass es immer den Paketen von CDs vertraut, so dass dies nicht ein so großes Problem darstellt.

Wie man Apt sagt, wem es vertrauen soll

Die ganze Sicherheit des Verfahrens beruht also darauf, dass es eine `Release.gpg`-Datei gibt, die eine `Release`-Datei signiert, und dass diese Signatur von **apt** mit Hilfe von GPG überprüft wird. Dazu muss es den öffentlichen Schlüssel der Person kennen, welche die Datei unterschrieben hat. Diese Schlüssel werden in Apts eigenem Schlüsselbund (`/etc/apt/trusted.gpg`) gespeichert. Bei der Verwaltung dieser Schlüssel kommt Secure Apt ins Spiel.

Standardmäßig befindet sich bei Debian-Systemen der Schlüssel des Debian-Archivs im Schlüsselbund.

```
# apt-key list
```

[7] Oder Ihren DNS vergiftet hat oder den Server spooft oder die Datei auf einem Spiegel platziert hat, den Sie verwenden, oder ...

```
/etc/apt/trusted.gpg
--------------------
pub   1024D/4F368D5D 2005-01-31 [expires: 2006-01-31]
uid                  Debian Archive Automatic Signing Key (2005) <ftpmaster@debian
```

Im Beispiel ist 4F368D5D die Schlüssel-ID. Beachten Sie, dass dieser Schlüssel nur für ein Jahr gültig ist. Debian tauscht die Schlüssel als letzte Verteidigungslinie gegen Sicherheitsrisiken, die das Knacken eines Schlüssels umfassen, regelmäßig aus.

Mit dem Schlüssel des Archivs wird **apt** dem offiziellen Archiv von Debian vertrauen. Wenn Sie aber weitere Paketdepots zu `/etc/apt/sources.list` hinzufügen wollen, müssen Sie Apt Ihre Schlüssel mitteilen, wenn Sie wollen, dass Apt ihnen vertraut. Sobald Sie den Schlüssel haben und ihn überprüft haben, müssen Sie nur **apt-key add** *`Datei`* laufen lassen, um den Schlüssel hinzuzufügen. Der schwierigste Teil dabei ist, den Schlüssel zu bekommen und ihn zu überprüfen.

Den Schlüssel eines Paketdepots finden

Mit dem Paket debian-archive-keyring werden Schlüssel für **apt** bereitgestellt. Aktualisierungen dieses Pakets führen dazu, dass GPG-Schlüssel für das Debian-Hauptarchiv hinzugefügt (oder gelöscht) werden.

Für andere Archive gibt noch keinen standardisierten Ort, wo sich der Schlüssel für ein Paketdepot befinden soll. Es besteht die grobe Übereinkunft, dass der Schlüssel auf der Webseite des Paketdepots oder im Depot selbst zu finden sein sollte. Wie gesagt ist dies kein echter Standard, so dass Sie den Schlüssel unter Umständen suchen müssen.

The Debian archive signing key is available at https://ftp-master.debian.org/keys.html.[8]

gpg besitzt mit den Schlüsselservern eine standardisierte Möglichkeit, Schlüssel zu verbreiten. Damit kann GPG einen Schlüssel herunterladen und ihn zum Schlüsselbund hinzufügen. Beispiel:

```
$ gpg --keyserver pgpkeys.mit.edu --recv-key 2D230C5F
gpg: requesting key 2D230C5F from hkp server pgpkeys.mit.edu
gpg: key 2D230C5F: public key "Debian Archive Automatic Signing Key (2006) <ftpm
aster@debian.org>" imported
gpg: Anzahl insgesamt bearbeiteter Schlüssel: 1
gpg:               importiert: 1
```

Sie können dann den Schlüssel aus Ihrem Schlüsselbund exportieren und ihn an **apt-key** weiterreichen:

```
$ gpg -a --export 2D230C5F | sudo apt-key add -
gpg: kein uneingeschränkt vertrauenswürdiger Schlüssel 080F67F4 gefunden
OK
```

Die Warnung »gpg: kein uneingeschränkt vertrauenswürdiger Schlüssel 080F67F4 gefunden« bedeutet, dass GPG nicht so konfiguriert wurde, um einem Schlüssel vollständig zu vertrauen. Das Zuweisen von Vertrauensstufen ist Teil des Web-of-Trust von OpenPGP, was hier nicht Gegenstand ist. Daher ist die Warnung unproblematisch. Für gewöhnlich wird dem eignen Schlüssel eines Benutzers vollständig vertraut.

Auf sichere Weise einen Schlüssel hinzufügen

By adding a key to apt's keyring, you're telling apt to trust everything signed by the key, and this lets you know for sure that apt won't install anything not signed by the person who possesses the private key. But

[8] "ziyi" is the name of the tool used for signing on the Debian servers, the name is based on the name of a http://en.wikipedia.org/wiki/Zhang_Ziyi.

if you're sufficiently paranoid, you can see that this just pushes things up a level, now instead of having to worry if a package, or a `Release` file is valid, you can worry about whether you've actually gotten the right key. Is the key file from https://ftp-master.debian.org/keys.html mentioned above really Debian's archive signing key, or has it been modified (or this document lies).

Es ist gut, in Sicherheitsfragen Vorsicht walten zu lassen. Aber ab hier wird es schwieriger, Dinge zu überprüfen. **gpg** arbeitet mit dem Konzept der Kette des Vertrauens (chain of trust), die bei jemandem beginnt, dem Sie vertrauen und der einen anderen Schlüssel unterschreibt usw., bis Sie beim Schlüssel des Archivs sind. Wenn Sie vorsichtig sind, wollen Sie nachprüfen, dass Ihr Archivschlüssel von einem Schlüssel unterschrieben wurde, dem Sie vertrauen können, weil seine Kette des Vertrauens zu jemandem zurückgeht, den Sie persönlich kennen. Dazu sollten Sie eine Debian-Konferenz oder eine lokale LUG zum Unterschreiben der Schlüssel besuchen[9].

Wenn Sie diese Sicherheitsbedenken nicht teilen (können), unternehmen Sie, was auch immer Sie passend finden, wenn Sie eine neue Apt-Quelle oder einen neuen Schlüssel verwenden. Sie könnten demjenigen, der den Schlüssel anbietet, eine Mail schreiben, um den Schlüssel zu überprüfen. Oder Sie vertrauen auf Ihr Glück und gehen davon aus, dass Sie den richten heruntergeladen haben. Das wichtige ist, dass Secure Apt, indem es das Problem darauf reduziert, welchen Archivschlüsseln Sie vertrauen, Sie so vorsichtig und sicher vorgehen lässt, wie es Ihnen passend und notwendig erscheint.

Die Integrität eines Schlüssels überprüfen

You can verify the fingerprint as well as the signatures on the key. Retrieving the fingerprint can be done for multiple sources, you can talk to Debian Developers on IRC, read the mailing list where the key change will be announced or any other additional means to verify the fingerprint. For example you can do this:

```
$ GET http://ftp-master.debian.org/ziyi_key_2006.asc | gpg --import
gpg: key 2D230C5F: public key "Debian Archive Automatic Signing Key (2006)
  <ftpmaster&debian.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1
$ gpg --check-sigs --fingerprint 2D230C5F
pub   1024D/2D230C5F 2006-01-03 [expires: 2007-02-07]
      Key fingerprint = 0847 50FC 01A6 D388 A643  D869 0109 0831 2D23 0C5F
uid   Debian Archive Automatic Signing Key (2006) <ftpmaster@debian.org>
sig!3        2D230C5F 2006-01-03  Debian Archive Automatic Signing Key
                                  (2006) <ftpmaster@debian.org>
sig!         2A4E3EAA 2006-01-03  Anthony Towns <aj@azure.humbug.org.au>
sig!         4F368D5D 2006-01-03  Debian Archive Automatic Signing Key
                                  (2005) <ftpmaster@debian.org>
sig!         29982E5A 2006-01-04  Steve Langasek <vorlon@dodds.net>
sig!         FD6645AB 2006-01-04  Ryan Murray <rmurray@cyberhqz.com>
sig!         AB2A91F5 2006-01-04  James Troup <james@nocrew.org>
```

and then as in „Paketsignierung in Debian“ check the trust path from your key (or a key you trust) to at least one of the keys used to sign the archive key. If you are sufficiently paranoid you will tell apt to trust the key only if you find an acceptable path:

```
$ gpg --export -a 2D230C5F | sudo apt-key add -
```

[9] Nicht alle Schlüssel der Apt-Depots sind überhaupt mit einem anderen Schlüssel unterschrieben. Vielleicht hat derjenige, der das Depot einrichtet, keinen anderen Schlüssel zur Verfügung, oder vielleicht ist es ihm unangenehm, einen Schlüssel mit einer derartig wichtigen Funktion mit seinem Hauptschlüssel zu unterschreiben. Hinweise, wie man einen Schlüssel für ein Depot einrichtet, finden Sie unter „Prüfung der Release-Datei von Debian-fremden Quellen“.

```
Ok
```

Hinweis: Der aktuelle Schlüssel ist mit dem vorhergehenden Archivschlüssel unterschrieben, so dass Sie theoretisch auf Ihrem alten Vertrauen aufbauen können.

Der jährliche Austausch des Archivschlüssels von Debian

Wie schon erwähnt wird der Schlüssel, mit dem das Debian-Archiv signiert wird, jedes Jahr im Januar ausgetauscht. Da Secure Apt noch jung ist, haben wir noch nicht sehr viel Erfahrung damit und es gibt noch ein paar haarige Stellen.

Im Januar 2006 wurde ein neuer Schlüssel für 2006 erstellt und die `Release`-Datei wurde damit unterschrieben. Um aber zu vermeiden, dass Systeme, die noch den alten Schlüssel von 2005 verwenden, nicht mehr korrekt arbeiten, wurde die `Release`-Datei auch mit dem alten Schlüssel unterschrieben. Es war geplant, dass Apt je nach dem verfügbaren Schlüssel eine der beiden Unterschriften akzeptieren würde. Aber es zeigte sich ein Fehler in Apt, da es sich weigerte, der Datei zu vertrauen, wenn es nicht beide Schlüssel hatte und somit beide Unterschriften überprüfen konnte. Dies wurde in der Version 0.6.43.1 ausgebessert. Es gab auch Verwirrung darüber, wie der Schlüssel an Benutzer verteilt wird, die bereits Secure Apt auf ihrem System laufen lassen. Am Anfang wurde er auf die Webseite hochgeladen, ohne Ankündigung und ohne eine echte Möglichkeit, ihn zu überprüfen, und die Benutzer mussten ihn per Hand herunterladen.

Bekannte Probleme bei der Prüfung der Release-Datei

Ein nicht offensichtliches Problem ist, dass Secure Apt nicht funktioniert, wenn Ihre Uhr sehr verstellt ist. Wenn sie auf ein Datum in der Vergangenheit wie 1999 eingestellt ist, wird Apt mit einer nichts sagenden Ausgabe wie dieser abbrechen:

```
W: GPG error: http://archive.progeny.com sid Release: Unknown error executing gpg
```

Dagegen macht **apt-key** das Problem deutlich:

```
gpg: key 2D230C5F was created 192324901 seconds in the future (time warp or clock
gpg: key 2D230C5F was created 192324901 seconds in the future (time warp or clock
pub   1024D/2D230C5F 2006-01-03
uid                  Debian Archive Automatic Signing Key (2006) <ftpmaster@debian
```

Falls die Uhr nicht zu weit vorgeht, behandelt Apt die Schlüssel als abgelaufen.

Wenn Sie Testing oder Unstable verwenden, gibt es ein Problem, wenn Sie in letzter Zeit nicht **apt-get update** ausgeführt haben und mit **apt-get** ein Paket installieren möchten. Apt könnte sich darüber beschweren, dass es nicht authentifiziert werden konnte (Warum passiert das bloß?). **apt-get update** löst das Problem.

Prüfung von Hand

Für den Fall, dass Sie nun zusätzliche Sicherheitsprüfungen einführen wollen, aber nicht die neuste Version von apt einsetzen wollen oder können[10], können Sie das folgende Skript von Anthony Towns benutzen. Dieses Skript führt automatisch neue Sicherheitsüberprüfungen durch, damit ein Benutzer sicher gehen

[10] Entweder weil Sie Stable (*Sarge*) oder eine ältere Veröffentlichung verwenden, oder weil Sie nicht die neuste Version von Apt einsetzen wollen, obwohl wir das Testen wirklich schätzen würden.

kann, dass die Software, die er herunterlädt, die gleiche ist wie die, die von Debian bereitgestellt wird. Das verhindert, dass sich Debian-Entwickler in ein fremdes System einhacken können, ohne dass eine Zurechnung und Rückverfolgung möglich wäre, die durch das Hochladen eines Pakets auf das Hauptarchiv gewährleistet werden. Es kann auch verhindern, dass ein Spiegel etwas fast genau abbildet, das aber eben doch nicht ganz wie in Debian, oder dass veraltete Versionen von instabilen Paketen mit bekannten Sicherheitslücken zur Verfügung gestellt werden.

Dieser Beispielscode, umbenannt nach **apt-check sigs**, sollte auf die folgende Art benutzt werden:

```
# apt-get update
# apt-check-sigs
(... Ergebnisse ...)
# apt-get dist-upgrade
```

Zuerst müssen Sie jedoch Folgendes tun:

- get the keys the archive software uses to sign `Release` files from https://ftp-master.debian.org/keys.html and add them to `~/.gnupg/trustedkeys.gpg` (which is what **gpgv** uses by default).

  ```
  gpg --no-default-keyring --keyring trustedkeys.gpg --import ziyi_key_2006.asc
  ```

- Entfernen Sie alle Zeilen aus `/etc/apt/sources.list`, die nicht die normale »dists«-Struktur benutzen, oder ändern Sie das Skript, so dass es auch mit denen funktioniert.

- Ignorieren Sie die Tatsache, dass Sicherheitsaktualisierungen von Debian keine signierten `Release`-Dateien haben, und das `Sources`-Dateien (noch) keine richtigen Prüfsummen in der `Release`-Datei haben.

- Bereiten Sie sich darauf vor, zu prüfen, dass die richtigen Quellen durch den richtigen Schlüssel signiert wurden.

This is the example code for **apt-check-sigs**, the latest version can be retrieved from http://people.debian.org/~ajt/apt-check-sigs. This code is currently in beta, for more information read http://lists.debian.org/debian-devel/2002/07/msg00421.html.

```
#!/bin/bash

#!/bin/bash

# Copyright (c) 2001 Anthony Towns <ajt@debian.org>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.

rm -rf /tmp/apt-release-check
mkdir /tmp/apt-release-check || exit 1
```

```
cd /tmp/apt-release-check

>OK
>MISSING
>NOCHECK
>BAD

arch=`dpkg --print-installation-architecture`

am_root () {
        [ `id -u` -eq 0 ]
}

get_md5sumsize () {
        cat "$1" | awk '/^MD5Sum:/,/^SHA1:/' |
          MYARG="$2" perl -ne '@f = split /\s+/; if ($f[3] eq $ENV{"MYARG"}) {
print "$f[1] $f[2]\n"; exit(0); }'
}

checkit () {
        local FILE="$1"
        local LOOKUP="$2"

        Y="`get_md5sumsize Release "$LOOKUP"`"
        Y="`echo "$Y" | sed 's/^ *//;s/  */ /g'`"

        if [ ! -e "/var/lib/apt/lists/$FILE" ]; then
                if [ "$Y" = "" ]; then
                        # No file, but not needed anyway
                        echo "OK"
                        return
                fi
                echo "$FILE" >>MISSING
                echo "MISSING $Y"
                return
        fi
        if [ "$Y" = "" ]; then
                echo "$FILE" >>NOCHECK
                echo "NOCHECK"
                return
        fi
        X="`md5sum < /var/lib/apt/lists/$FILE | cut -d\  -f1` `wc -c < /var/lib
/apt/lists/$FILE`"
        X="`echo "$X" | sed 's/^ *//;s/  */ /g'`"
        if [ "$X" != "$Y" ]; then
                echo "$FILE" >>BAD
                echo "BAD"
                return
        fi
        echo "$FILE" >>OK
        echo "OK"
}

echo
```

```
echo "Checking sources in /etc/apt/sources.list:"
echo "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
echo
(echo "You should take care to ensure that the distributions you're downloading
"
echo "are the ones you think you are downloading, and that they are as up to"
echo "date as you would expect (testing and unstable should be no more than"
echo "two or three days out of date, stable-updates no more than a few weeks"
echo "or a month)."
) | fmt
echo

cat /etc/apt/sources.list |
  sed 's/^ *//' | grep '^[^#]' |
  while read ty url dist comps; do
        if [ "${url%%:*}" = "http" -o "${url%%:*}" = "ftp" ]; then
                baseurl="${url#*://}"
        else
                continue
        fi

        echo "Source: ${ty} ${url} ${dist} ${comps}"

        rm -f Release Release.gpg
        lynx -reload -dump "${url}/dists/${dist}/Release" >/dev/null 2>&1
        wget -q -O Release "${url}/dists/${dist}/Release"

        if ! grep -q '^' Release; then
                echo "  * NO TOP-LEVEL Release FILE"
                >Release
        else
                origline=`sed -n 's/^Origin: *//p' Release | head -1`
                lablline=`sed -n 's/^Label: *//p' Release | head -1`
                suitline=`sed -n 's/^Suite: *//p' Release | head -1`
                codeline=`sed -n 's/^Codename: *//p' Release | head -1`
                dateline=`grep "^Date:" Release | head -1`
                dscrline=`grep "^Description:" Release | head -1`
                echo "  o Origin: $origline/$lablline"
                echo "  o Suite: $suitline/$codeline"
                echo "  o $dateline"
                echo "  o $dscrline"

                if [ "${dist%%/*}" != "$suitline" -a "${dist%%/*}" != "$codeline"
                        echo "  * WARNING: asked for $dist, got $suitline/$codelin
                fi

                lynx -reload -dump "${url}/dists/${dist}/Release.gpg" >/dev/null 2
                wget -q -O Release.gpg "${url}/dists/${dist}/Release.gpg"

                gpgv --status-fd 3 Release.gpg Release 3>&1 >/dev/null 2>&1 | sed
                        if [ "$gpgcode" = "GOODSIG" ]; then
                            if [ "$err" != "" ]; then
                                echo "  * Signed by ${err# } key: ${rest#* }"
                            else
```

```
                                    echo "  o Signed by: ${rest#* }"
                                    okay=1
                                fi
                                err=""
                            elif [ "$gpgcode" = "BADSIG" ]; then
                                echo "  * BAD SIGNATURE BY: ${rest#* }"
                                err=""
                            elif [ "$gpgcode" = "ERRSIG" ]; then
                                echo "  * COULDN'T CHECK SIGNATURE BY KEYID: ${rest %%
                                err=""
                            elif [ "$gpgcode" = "SIGREVOKED" ]; then
                                err="$err REVOKED"
                            elif [ "$gpgcode" = "SIGEXPIRED" ]; then
                                err="$err EXPIRED"
                            fi
                        done
                        if [ "$okay" != 1 ]; then
                            echo "  * NO VALID SIGNATURE"
                            >Release
                        fi)
                fi
                okaycomps=""
                for comp in $comps; do
                        if [ "$ty" = "deb" ]; then
                                X=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/binar
                                Y=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/binar
                                if [ "$X $Y" = "OK OK" ]; then
                                        okaycomps="$okaycomps $comp"
                                else
                                        echo "  * PROBLEMS WITH $comp ($X, $Y)"
                                fi
                        elif [ "$ty" = "deb-src" ]; then
                                X=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/sourc
                                Y=$(checkit "`echo "${baseurl}/dists/${dist}/${comp}/sourc
                                if [ "$X $Y" = "OK OK" ]; then
                                        okaycomps="$okaycomps $comp"
                                else
                                        echo "  * PROBLEMS WITH component $comp ($X, $Y)"
                                fi
                        fi
                done
                [ "$okaycomps" = "" ] || echo "  o Okay:$okaycomps"
                echo
        done

echo "Results"
echo "~~~~~~~"
echo

allokay=true

cd /tmp/apt-release-check
diff <(cat BAD MISSING NOCHECK OK | sort) <(cd /var/lib/apt/lists && find . -type
```

```
cd /tmp/apt-release-check
if grep -q ^ UNVALIDATED; then
    allokay=false
    (echo "The following files in /var/lib/apt/lists have not been validated."
    echo "This could turn out to be a harmless indication that this script"
    echo "is buggy or out of date, or it could let trojaned packages get onto"
    echo "your system."
    ) | fmt
    echo
    sed 's/^/    /' < UNVALIDATED
    echo
fi

if grep -q ^ BAD; then
    allokay=false
    (echo "The contents of the following files in /var/lib/apt/lists does not"
    echo "match what was expected. This may mean these sources are out of date,"
    echo "that the archive is having problems, or that someone is actively"
    echo "using your mirror to distribute trojans."
    if am_root; then
        echo "The files have been renamed to have the extension .FAILED and"
        echo "will be ignored by apt."
        cat BAD | while read a; do
            mv /var/lib/apt/lists/$a /var/lib/apt/lists/${a}.FAILED
        done
    fi) | fmt
    echo
    sed 's/^/    /' < BAD
    echo
fi

if grep -q ^ MISSING; then
    allokay=false
    (echo "The following files from /var/lib/apt/lists were missing. This"
    echo "may cause you to miss out on updates to some vulnerable packages."
    ) | fmt
    echo
    sed 's/^/    /' > MISSING
    echo
fi

if grep -q ^ NOCHECK; then
    allokay=false
    (echo "The contents of the following files in /var/lib/apt/lists could not"
    echo "be validated due to the lack of a signed Release file, or the lack"
    echo "of an appropriate entry in a signed Release file. This probably"
    echo "means that the maintainers of these sources are slack, but may mean"
    echo "these sources are being actively used to distribute trojans."
    if am_root; then
        echo "The files have been renamed to have the extension .FAILED and"
        echo "will be ignored by apt."
        cat NOCHECK | while read a; do
            mv /var/lib/apt/lists/$a /var/lib/apt/lists/${a}.FAILED
        done
```

```
    fi) | fmt
    echo
    sed 's/^/    /' > NOCHECK
    echo
fi

if $allokay; then
    echo 'Everything seems okay!'
    echo
fi

rm -rf /tmp/apt-release-check
```

Sie müssen vielleicht bei *Sid* diesen Patch verwenden, da **md5sum** ein »-« an die Summe anfügt, wenn die Ausgabe auf stdin erfolgt:

```
@@ -37,7 +37,7 @@
        local LOOKUP="$2"

        Y="`get_md5sumsize Release "$LOOKUP"`"
-       Y="`echo "$Y" | sed 's/^ *//;s/  */ /g'`"
+       Y="`echo "$Y" | sed 's/-//;s/^ *//;s/  */ /g'`"

        if [ ! -e "/var/lib/apt/lists/$FILE" ]; then
                if [ "$Y" = "" ]; then
@@ -55,7 +55,7 @@
                return
        fi
        X="`md5sum < /var/lib/apt/lists/$FILE` `wc -c < /var/lib/apt/lists/$FILE`"
-       X="`echo "$X" | sed 's/^ *//;s/  */ /g'`"
+       X="`echo "$X" | sed 's/-//;s/^ *//;s/  */ /g'`"
        if [ "$X" != "$Y" ]; then
                echo "$FILE" >>BAD
                echo "BAD"
```

Prüfung der Release-Datei von Debian-fremden Quellen

Beachten Sie, dass, wenn Sie die neuste Version von Apt (mit *Secure Apt*) einsetzen, kein zusätzlicher Aufwand auf Ihrer Seite notwendig sein sollte, wenn Sie keine Debian-fremden Quellen verwenden. Anderenfalls erfordert **apt-get** eine zusätzliche Bestätigung. Dies wird verhindert, wenn `Release`- und `Release.gpg`-Dateien in den Debian-fremden Quellen zur Verfügung stehen. Die `Release`-Datei kann mit **apt-ftparchive** (ist in apt-utils 0.5.0 und später enthalten) erstellt werden, die `Release.gpg` ist nur die abgetrennte Signatur. Beide können mit folgender einfacher Prozedur erstellt werden:

```
$ rm -f dists/unstable/Release
$ apt-ftparchive release dists/unstable > dists/unstable/Release
$ gpg --sign -ba -o dists/unstable/Release.gpg dists/unstable/Release
```

Alternativer Entwurf zur Einzelsignierung von Paketen

Dieser zusätzliche Entwurf, jedes Paket einzeln zu signieren, erlaubt es, Pakete zu prüfen, selbst wenn sie nicht mehr in irgendeiner `Packages`-Datei erwähnt werden. Und so können auch Pakete von Dritten,

für die es nie eine `Packages`-Datei gab, unter Debian installiert werden. Dies wird aber kein Standard werden.

Dieser Entwurf zur Paketsignierung kann mit debsig-verify und debsigs umgesetzt werden. Diese beiden Pakete können in einer .deb-Datei eingebettete Unterschriften erstellen und prüfen. Debian hat bereits jetzt die Möglichkeiten, dies zu tun. Aber es gibt keine Planung, dieses Regelwerk oder ähnliche Werkzeuge umzusetzen, da nunmehr das Schema mit der Signierung des Archivs bevorzugt wird. Die Werkzeuge werden dennoch für Benutzer und Administratoren von Archiven zur Verfügung gestellt, wenn sie diese Vorgehensweise bevorzugen.

Latest **dpkg** versions (since 1.9.21) incorporate a http://lists.debian.org/debian-dpkg/2001/03/msg00024.html that provides this functionality as soon as debsig-verify is installed.

HINWEIS: Derzeit wird `/etc/dpkg/dpkg.cfg` standardmäßig mit der Option »no-debsig« ausgeliefert.

HINWEIS 2: Unterschriften von Entwicklern werden im Moment entfernt, wenn sie in das Paketarchiv gelangen, da die derzeit vorzugswürdige Methode die Überprüfung der Release-Datei ist, wie es oben beschrieben wurde.

Kapitel 8. Sicherheitswerkzeuge in Debian

FIXME: More content needed.

Debian stellt außerdem einige Sicherheitswerkzeuge zur Verfügung, die eine Debian-Maschine zum Zweck der Sicherheit passend einrichten können. Diese Zielsetzung schließt die Sicherung von Systeminformationen durch Firewalls (sowohl auf Paket- als auch auf Anwendungsebene), Eindringlingserkennung (netzwerk- und hostbasiert), Einschätzung der Verwundbarkeit, Antivirus, private Netzwerke und vieles mehr ein.

Seit Debian 3.0 (*woody* ist kryptographische Software in der Hauptdistribution integriert. OpenSSH und GNU Privacy Guard sind in der Standardinstallation enthalten. Außerdem befinden sich jetzt in Web-Browsern und Web-Servern, Datenbanken usw. starke Verschlüsselungsmechanismen. Eine weitergehende Eingliederung von Kryptographie ist für zukünftige Veröffentlichungen geplant. Aufgrund von Exportbeschränkungen in den USA wurde diese Software nicht mit der Hauptdistribution ausgeliefert, sondern war nur auf Seiten außerhalb der USA erhältlich.

Programme zur Fernprüfung der Verwundbarkeit

Die Werkzeuge, um eine Fernprüfung der Verwundbarkeit durchzuführen, sind unter Debian: [1]

- nessus
- raccess
- nikto (Ersatz für **whisker**)

Das weitaus vollständigste und aktuellste Werkzeug ist nessus, welches aus einem Client (nessus) mit graphischer Benutzungsschnittstelle und einem Server (nessusd), der die programmierten Attacken startet, besteht. Nessus kennt verschiedene entfernten Verwundbarkeiten für einige Systeme, einschließlich Netzwerkanwendungen, FTP-Servern, WWW-Servern, usw. Die neusten Sicherheitsplugins sind sogar in der Lage, eine Web-Seite zu analysieren und zu versuchen, interaktive Inhalte zu entdecken, die zu einem Angriff genutzt werden können. Es existieren auch Java- und Win32-Clients, die benutzt werden können, um sich mit dem Nessus-Server zu verbinden. Diese sind jedoch in Debian nicht enthalten.

nikto ist ein Scanner zur Aufdeckung von Schwachstellen bei Webservern und kennt auch einige Anti-IDS-Taktiken (die meisten davon sind keine *Anti-IDS*-Taktiken mehr). Er ist einer der besten verfügbaren CGI-Scanner zur Erkennung von WWW-Servern und kann nur bestimmte Angriffe gegen ihn starten. Die Datenbank, die zum Scannen benutzt wird, kann sehr leicht geändert werden, um neue Informationen einzufügen.

Werkzeuge zum Scannen von Netzwerken

Debian bietet Ihnen einige Werkzeuge zum Scannen von Hosts (aber nicht zur Gefahrenabschätzung). Diese Programme werden in manchen Fällen von Scannern zur Gefahrenabschätzung zu einem ersten »Angriff« gegen entfernte Rechner genutzt, um festzustellen, welche Dienste angeboten werden. Unter Debian sind im Moment verfügbar:

[1] Manche von ihnen sind erhältlich, wenn Sie das Paket harden-remoteaudit installieren.

- nmap
- xprobe
- p0f
- knocker
- isic
- hping2
- icmpush
- nbtscan (für die Prüfung von SMB und NetBIOS)
- fragrouter
- **strobe**(aus dem Paket netdiag)
- irpas

Während xprobe lediglich aus der Ferne das Betriebssystem erkennen kann (indem es TCP/IP-Fingerabdrücke benutzt, machen nmap und knocker beides: das Betriebssystem erkennen und die Ports eines entfernten Rechners scannen. Andererseits können hping2 und icmpush für ICMP-Angriffstechniken benutzt werden.

Nbtscan, das speziell für SMB-Netzwerke entworfen wurde, kann benutzt werden, um IP-Netzwerke zu scannen und diverse Informationen von SMB-Servern zu ermitteln einschließlich der Benutzernamen, Netzwerknamen, MAC-Adressen, ...

Dagegen kann fragrouter dazu verwendet werden, um Systeme zur Eindringlingserkennung zu testen und um zu sehen, ob das NIDS mit fragmentierten Angriffen umgangen werden kann.

FIXME: Check http://bugs.debian.org/153117 (ITP fragrouter) to see if it's included.

FIXME add information based on https://web.archive.org/web/20040725013857/http://www.giac.org/practical/gcux/Stephanie_Thomas_GCUX.pdf which describes how to use Debian and a laptop to scan for wireless (803.1) networks (link not there any more).

Interne Prüfungen

Derzeit kann lediglich das Programm tiger benutzt werden, um interne Prüfungen (auch »white box« genannt) eines Rechners vorzunehmen. Dabei wird festgestellt, ob das Dateisystem richtig aufgesetzt ist, welche Prozesse auf dem Rechner horchen, usw.

Testen des Quellcodes

Debian bietet einige Pakete an, die C/C++-Quellcode prüfen und Programmierfehler finden, die zu möglichen Sicherheitsmängeln führen können:

- flawfinder
- rats
- splint

- pscan

Virtual Private Networks (virtuelle private Netzwerke)

Ein virtuelles privates Netzwerk (VPN) ist eine Gruppe von zwei oder mehreren Computern, die typischerweise zu einem privaten Netzwerk mit begrenztem öffentlichen Netzwerkzugang verbunden sind und sicher über ein öffentliches Netzwerk kommunizieren. VPNs können einen einzelnen Rechner mit einem privaten Netzwerk verbinden (Client-Server) oder ein entferntes LAN mit einem privaten Netzwerk (Server-Server). VPNs verwenden Verschlüsselung, starke Authentifikation von entfernten Benutzern oder Hosts und Methoden, um die Struktur des privaten Netzwerks zu verstecken.

Debian enthält etliche Pakete, die zum Aufsetzen von verschlüsselten virtuellen privaten Netzwerken verwendet werden können:

- vtun
- tunnelv (Abschnitt non-US)
- cipe-source, cipe-common
- tinc
- secvpn
- pptpd
- openvpn
- openswan (http://www.openswan.org/)

FIXME: Update the information here since it was written with FreeSWAN in mind. Check Bug #237764 and Message-Id: <200412101215.04040.rmayr@debian.org>.

Das OpenSWAN-Paket ist wahrscheinlich die beste Wahl, da es nahezu mit allen zusammenarbeiten kann, die das IP-Security-Protokoll IPsec (RFC 2411) benutzen. Aber auch die anderen oben aufgeführten Pakete können Ihnen helfen, möglichst schnell einen sicheren Tunnel aufzusetzen. Das Point-to-Point-Tunneling-Protocol (PPTP) ist ein urheberrechtlich geschütztes Protokoll von Microsoft für VPN. Es wird unter Linux unterstützt, aber es sind einige schwere Sicherheitsprobleme bekannt.

Für weitere Informationen über IPsec und PPTP lesen Sie bitte das http://www.tldp.org/HOWTO/VPN-Masquerade-HOWTO.html, über PPP über SSH das http://www.tldp.org/HOWTO/VPN-HOWTO.html, das http://www.tldp.org/HOWTO/mini/Cipe+Masq.html und das http://www.tldp.org/HOWTO/mini/ppp-ssh/index.html.

Es kann sich auch lohnen, sich http://yavipin.sourceforge.net/ anzusehen. Allerdings scheinen noch keine Pakete für Debian verfügbar zu sein.

Point-to-Point-Tunneling

Wenn Sie einen tunnelnden Server für eine gemischte Umgebung (sowohl Microsofts Betriebssystem als auch Linux-Clients) zur Verfügung stellen wollen und IPsec keine Möglichkeit ist (da es nur in Windows 2000 und Windows XP enthalten ist), können Sie *PoPToP* (Point to Point Tunneling Server) verwenden. Er wird vom Paket pptpd bereitgestellt.

Wenn Sie Microsofts Authentifikation und Verschlüsselung mit dem Server verwenden wollen, die im Paket ppp enthalten sind, sollten Sie Folgendes aus der FAQ beachten:

```
Sie müssen nur dann PPP 2.3.8 einsetzen, wenn Sie zu Microsoft kompatible MSCHAPv2
```

Allerdings müssen Sie auf den Kernel einen Patch anwenden, der im Paket kernel-patch-mppe enthalten ist. Er stellt das Module pp_mppe für den pppd zur Verfügung.

Beachten Sie, dass Verschlüsselung in ppptp erfordert, dass Sie die Nutzerpasswörter in Klartext speichern. Außerdem sind für das MS-CHAPv2-Protokoll http://mopo.informatik.uni-freiburg.de/pptp_mschapv2/.

Infrastruktur für öffentliche Schlüssel (Public Key Infrastructure, PKI)

Mit der Infrastruktur für öffentliche Schlüssel (PKI) wurde eine Sicherheitsarchitektur eingeführt, um den Grad der Vertrauenswürdigkeit von Informationen, die über unsichere Netzwerke ausgetauscht werden, zu erhöhen. Sie beruht auf dem Konzept von öffentlichen und privaten kryptographischen Schlüsseln, um die Identität des Absenders (Signierung) zu überprüfen und die Geheimhaltung zu sichern (Verschlüsselung).

Wenn Sie über die Einrichtung einer PKI nachdenken, sehen Sie sich mit einer breiten Palette von Problemen konfrontiert:

- eine Zertifizierungsstelle (Certification Authority, CA), die Zertifikate ausgeben und bestätigen und unter einer bestimmten Hierarchie arbeiten kann
- ein Verzeichnis, das die öffentlichen Zertifikate der Benutzer enthält
- eine Datenbank (?), um eine List von Widerrufen von Zertifikaten (Certificate Revocation Lists, CRL) zu verwalten
- Geräte, die mit der CA zusammenarbeiten, um Smartcards/USB-Token oder ähnliches zu erzeugen und die Zertifikate sicher zu speichern
- Anwendungen, die die von einer CA ausgestellten Zertifikate benutzen können, um verschlüsselte Kommunikation zu aufzubauen und bestimmte Zertifikate gegen die CRL zu prüfen (zur Authentifizierung und so genannte »full Single Sign On solutions«)
- eine Zeitstempel-Autorität, um Dokumente digital zu signieren
- eine Verwaltungskonsole, von der aus dies alles vernünftig benutzt werden kann (Erstellung von Zertifikaten, Kontrolle der Widerruflisten, usw., ...)

Debian GNU/Linux beinhaltet Softwarepaket, die Ihnen bei einigen dieser PKI-Probleme helfen können. Dazu gehört **OpenSSL** (zur Erstellung von Zertifikaten), **OpenLDAP** (für ein Verzeichnis, um die Zertifikate zu speichern) **gnupg** und **openswan** (mit X.509 Unterstützung). Jedoch stellt Debian zum Zeitpunkt der Veröffentlichung von Woody (Debian 3.0) keine der frei verfügbaren Certificate Authorities wie zum Beispiel pyCA, http://www.openca.org oder die CA-Muster von OpenSSL zur Verfügung. Für weitere Informationen lesen Sie bitte das http://ospkibook.sourceforge.net/.

SSL-Infrastruktur

Debian stellt einige SSL-Zertifikate innerhalb der Distribution zur Verfügung, so dass Sie sie lokal installieren können. Sie befinden sich im Paket ca-certificates. Dieses Paket stellt eine zentrale Sammelstelle für

Zertifikate dar, die an Debian übermittelt und vom Paketverwalter gebilligt (das heißt, verifiziert) wurden. Sie können für alle OpenSSL-Anwendungen, die SSL-Verbindungen verifizieren, nützlich sein.

FIXME: read debian-devel to see if there was something added to this.

Antiviren-Werkzeuge

Es gibt nicht viele Antiviren-Werkzeuge in Debian, wahrscheinlich weil die Benutzer von GNU/Linux nicht von Viren betroffen sind. Das Sicherheitsmodell von Unix trifft eine Unterscheidung zwischen privilegierten Prozessen (Root) und den Prozessen der Benutzer. Daher kann ein »bösartiges« Programm, das ein Benutzer empfängt oder erstellt und dann ausführt, nicht das System »infizieren« oder daran Veränderungen vornehmen. Es existieren dennoch Würmer und Viren für GNU/Linux, auch wenn es (bisher) keinen Virus gab, der sich im Freien weit über eine Debian-Distribution verbreitet hat. Wie dem auch sei, Administratoren sollten vielleicht Antiviren-Gateways aufbauen, um verwundbarere Systeme in ihrem Netzwerk vor Viren zu schützen.

Debian GNU/Linux bietet derzeit die folgenden Werkzeuge zum Erstellen von Antiviren-Umgebungen an:

- http://www.clamav.net, das in Debian seit *Sarge* (der 3.1-Veröffentlichung) enthalten ist. Es sind Pakete sowohl für den Virusscanner (clamav) des Scanner-Daemons (clamav-daemon) als auch für die Daten, die der Scanner benötigt, verfügbar. Da es für die richtige Arbeit eines Antivirus-Programms entscheidend ist, dass seine Daten auf dem neusten Stand sind, gibt es zwei verschiedene Wege, um diese Daten aktuell zu halten: clamav-freshclam eröffnet die Möglichkeit, die Datenbank automatisch über das Internet zu aktualisieren, und clamav-data stellt die Daten unmittelbar zur Verfügung.[2]

- mailscanner ist ein Gateway-Scanner, der in E-Mails Viren und Spam entdeckt. Er arbeitet auf der Grundlage von sendmail oder exim und kann mehr als 17 verschiedene Virensuch-Engines (einschließlich clamav) verwenden.

- libfile-scan-perl, welches File::Scan liefert. Das ist eine Erweiterung von Perl, mit der Dateien nach Viren durchsucht werden können. Mit diesem Modul können plattformunabhängige Virenscanner realisiert werden.

- http://www.sourceforge.net/projects/amavis ist im Paket amavis-ng enthalten und in *Sarge* verfügbar. Es ist ein Virusscanner, der in verschiedene MTAs (Exim, Sendmail, Postfix oder Qmail) integriert werden kann. Er unterstützt mehr als 15 Virensuch-Engines (einschließlich clamav, File::Scan und openantivirus).

- http://packages.debian.org/sanitizer ist ein Werkzeug, welches das Paket procmail verwendet. Es kann den Anhang von E-Mails nach Viren durchsuchen, Anhänge aufgrund ihres Dateinamens abweisen und vieles mehr.

- http://packages.debian.org/amavis-postfix ist ein Skript, das eine Schnittstelle vom Mail-Transport-Agent zu einem oder mehreren kommerziellen Viren-Scannern anbietet (dieses Paket ist lediglich für den MTA postfix bestimmt).

- exiscan ist ein Virusscanner für E-Mails, der in Perl geschrieben wurde. Er arbeitet mit Exim zusammen.

- blackhole-qmail ist ein Spamfilter für Qmail mit eingebauter Unterstützung von Clamav.

[2] Wenn Sie das letztere Paket verwenden und ein offizielles Debian betreiben, wird die Datenbank nicht im Zuge von Sicherheitsaktualisierung auf den neusten Stand gebracht. Sie sollten entweder clamav-freshclam, **clamav-getfiles** verwenden, um neue clamav-data-Pakete zu erstellen, oder die Datenbank über die Seite der Betreuer aktuell halten:

```
 deb http://people.debian.org/~zugschlus/clamav-data/ /
deb-src http://people.debian.org/~zugschlus/clamav-data/ /
```

Einige Gateway-Daemons bieten schon Programmerweiterungen an, um Antiviren-Umgebungen zu erstellen. Dazu gehören exim4-daemon-heavy (die *heavy* Version des Exim MTAs), frox, ein transparenter zwischenspeichernder FTP-Proxyserver), messagewall (ein SMTP-Proxyserver) und pop3vscan (ein transparenter POP3-Proxy).

Zurzeit ist als einziges Programm zum Auffinden von Viren **clamav** in der Hauptdistribution enthalten. Daneben bietet Debian verschiedene Schnittstellen an, mit denen Gateways mit Antivirus-Fähigkeiten für unterschiedliche Protokolle erstellt werden können.

Im Folgenden einige andere freie Antiviren-Projekte, die in der Zukunft in Debian GNU/Linux enthalten sein könnten: http://sourceforge.net/projects/openantivirus/ (siehe http://bugs.debian.org/150698 und http://bugs.debian.org/150695).

FIXME: Is there a package that provides a script to download the latest virus signatures from http://www.openantivirus.org/latest.php?

FIXME: Check if scannerdaemon is the same as the open antivirus scanner daemon (read ITPs).

Allerdings wird Debian *niemals* proprietäre (unfreie und unverbreitbare) Antiviren-Software anbieten. Dazu zählen Panda Antivirus, NAI Netshield, http://www.sophos.com/, http://www.antivirus.com oder http://www.ravantivirus.com. Weitere Hinweise finden Sie im http://www.computer-networking.de/~link/security/av-linux.txt. Das bedeutet nicht, dass diese Software nicht richtig auf einem Debian-System installiert werden kann.[3]

For more information on how to set up a virus detection system read Dave Jones' article https://web.archive.org/web/20120509212938/http://www.linuxjournal.com/article/4882.

GPG-Agent

Es ist heutzutage weit verbreitet, E-Mails digital zu unterschreiben (manchmal auch zu verschlüsseln). Sie können z.B. feststellen, dass viele Menschen auf Mailinglisten ihre E-Mails signieren. Signaturen von öffentlichen Schlüsseln ist im Moment die einzige Möglichkeit festzustellen, ob eine E-Mail vom Absender geschickt wurden und nicht von jemand anderem.

Debian GNU/Linux enthält eine Anzahl von E-Mail-Clients mit der eingebauten Fähigkeit, E-Mails zu signieren. Sie arbeiten entweder mit gnupg oder pgp zusammen:

- evolution.
- mutt.
- kmail.
- icedove (umbenannte Version von Mozillas Thunderbird) mittels des Plugins http://enigmail.mozdev.org/. Dieses Plugin wird durch das Paket enigmail bereitgestellt.
- sylpheed. Abhängig davon wie sich die stabile Version dieses Pakets entwickelt, müssen Sie die *bleeding edge* Version, sylpheed-claws, verwenden.
- gnus ist, wenn mit dem Paket mailcrypt installiert, eine Schnittstelle für **emacs** zu **gnupg**.
- kuvert stellt diese Funktion unabhängig von Ihrem Mail-User-Agent (MUA) zur Verfügung, indem es mit dem Mail-Transport-Agent (MTA) arbeitet.

[3] Tatsächlich gibt es für das Antivirus-Programm *F-prot* das Installationspaket **f-prot-installer**, das zwar nicht frei, aber für Heimanwender *kostenlos* ist. Allerdings lädt dieser Installer nur http://www.f-prot.com/products/home_use/linux/ herunter und installiert sie.

Key-Server ermöglichen es Ihnen, veröffentlichte öffentliche Schlüssel herunterzuladen, damit Sie Signaturen überprüfen können. Einer diese Key-Server ist http://wwwkeys.pgp.net. gnupg kann automatisch öffentliche Schlüssel holen, die sich nicht schon in Ihrem öffentlichen Schlüsselbund befinden. Um beispielsweise **gnupg** so einzurichten, dass es den oben genannten Key-Server verwendet, müssen Sie die Datei `~/.gnupg/options` bearbeiten und folgende Zeile hinzufügen: [4]

```
keyserver wwwkeys.pgp.net
```

Die meisten Key-Server sind miteinander verbunden. Wenn Sie also Ihren öffentlichen Schlüssel einem hinzufügen, wird er an alle anderen Key-Server weitergereicht. Da wäre auch noch das Debian GNU/Linux Paket debian-keyring, das die öffentlichen Schlüssel aller Debian-Entwickler enthält. Der Schlüsselbund von **gnupg** wird in `/usr/share/keyrings/` installiert.

Weitere Informationen:

- http://www.gnupg.org/faq.html.
- http://www.gnupg.org/gph/de/manual.html.
- https://web.archive.org/web/20080201103530/http://www.dewinter.com/gnupg_howto/english/GPGMiniHowto.html.
- https://web.archive.org/web/20080513095235/http://www.uk.pgp.net/pgpnet/pgp-faq/.
- https://web.archive.org/web/20060222110131/http://www.cryptnet.net/fdp/crypto/gpg-party.html.

[4] Weitere Beispiele, wie Sie **gnupg** konfigurieren können, finden Sie in `/usr/share/doc/mutt/examples/gpg.rc`.

Kapitel 9. Der gute Umgang von Entwicklern mit der Sicherheit des OS

Dieses Kapitel handelt von einigen der anerkannten Vorgehensweisen für sicheres Programmieren, wenn Entwickler Pakete für Debian erstellen. Wenn Sie sehr an sicherheitsbewusster Programmierung interessiert sind, sollten Sie David Wheelers http://www.dwheeler.com/secure-programs/ und http://www.securecoding.org von Mark G. Graff und Kenneth R. van Wyk (O'Reilly, 2003) lesen.

Das richtige Vorgehen für die Nachprüfung der Sicherheit und deren Gestaltung

Entwickler, die Software in Pakete packen, sollten größte Anstrengung darauf verwenden sicherzustellen, dass die Installation der Software und ihre Verwendung keine Sicherheitsrisiken für das System oder seine Benutzer eröffnet.

Dazu sollten sie vor der Veröffentlichung der Software oder einer neuen Version den Quellcode des Pakets nachprüfen, um Fehler zu finden, die zu Sicherheitslücken führen können. Bekanntermaßen ist der Aufwand für die Fehlerbehebung in verschiedenen Stadien der Entwicklung unterschiedlich. So ist es leichter (und billiger), Fehler während der Entwicklung auszubessern als später, wenn die Software schon herausgegeben wurde und nur noch gewartet wird (einige Studien behaupten, dass die Kosten in dieser Phase *60* Mal höher sind). Es gibt Hilfsmittel, die versuchen, Fehler automatisch zu entdecken. Entwickler sollten dennoch die verschiedenen Sicherheitsfehler kennen, damit sie sie verstehen und sie so in eigenen (oder fremden) Programmcode entdecken können.

Programmierfehler, die typischerweise zu Sicherheitsproblemen führen, sind insbesondere: http://de.wikipedia.org/wiki/Puffer%C3%BCberlauf, Format-String-Überläufe, Heap-Überläufe und Integer-Überläufe (in C/C++-Programmen), vorübergehende http://de.wikipedia.org/wiki/Symlink-Schwachstelle (in Skripten), http://de.wikipedia.org/wiki/Directory_Traversal, die Einschleusung von Befehlen (auf Servern) und http://de.wikipedia.org/wiki/Cross-Site_Scripting sowie http://de.wikipedia.org/wiki/SQL-Injektion (bei web-orientierten Anwendungen). Eine ausführliche Liste von Sicherheitsfehlern finden Sie in Fortifys http://vulncat.fortifysoftware.com/.

Einige dieser Probleme können Sie nur erkennen, wenn Sie ein Experte in der verwendeten Programmiersprache sind. Aber andere können leicht entdeckt und behoben werden. Zum Beispiel kann eine Symlink-Schwachstelle auf Grund einer falschen Verwendung von temporären Verzeichnissen ohne Weiteres entdeckt werden, indem Sie `grep -r "/tmp/" .` ausführen. Diese Verweise sollten überprüft werden und fest einprogrammierte Dateinamen in temporären Verzeichnissen in Shell-Skripten mit **mktemp** oder **tempfile**, in Perl-Skripten mit File::Temp(3perl) und in C/C++ mit tmpfile(3) ersetzt werden.

Es stehen Ihnen einige Werkzeuge zur Verfügung, die Sie dabei unterstützen, den Quellcode auf Sicherheitsprobleme hin zu überprüfen. Dazu zählen rats, flawfinder und pscan. Weitere Informationen finden Sie in der http://www.de.debian.org/security/audit/tools.

Beim Paketieren von Software sollten Entwickler darauf achten, dass sie allgemein anerkannte Sicherheitsprinzipien einhalten. Dazu gehören:

- Die Software sollte mit so geringen Rechten wie möglich laufen.
    - Falls das Paket Binaries mit setuid oder setgid enthält, wird **Lintian** vor http://lintian.debian.org/reports/Tsetuid-binary.html-, http://lintian.debian.org/reports/Tsetgid-binary.html- und http://lintian.debian.org/reports/Tsetuid-gid-binary.html-Binaries warnen.

- Die Daemons, die in einem Paket enthalten sind, sollten mit den Rechten eines Benutzers laufen, der nur geringe Privilegien besitzt (vergleichen Sie dazu „Benutzer und Gruppen für Software-Daemons erstellen“).

- Automatisierte Aufgaben (also mit **cron**) sollten NICHT mit Root-Rechten laufen. Zumindest sollten mit Root-Rechten keine komplizierten Aufgaben erledigt werden.

Falls Sie diese Prinzipien nicht einhalten können, sollten Sie sichergehen, dass das Programm, das mit umfangreicheren Rechten läuft, auf Sicherheitsprobleme überprüft wurde. Wenn Sie sich nicht sicher sind oder Hilfe benötigen, sollten Sie sich mit dem http://www.de.debian.org/security/audit/ in Verbindung setzen. Wenn Binaries setuid/setgid verwenden, sollten Sie die Richtlinie von Debian zu http://www.debian.org/doc/debian-policy/ch-files#s10.9 beachten.

Für weitere Informationen, insbesondere hinsichtlich Sicherheitsfragen, sollten Sie das http://www.dwheeler.com/secure-programs/ und das https://buildsecurityin.us-cert.gov/portal/ Portal lesen (oder den Programmautor darauf hinweisen).

Benutzer und Gruppen für Software-Daemons erstellen

Wenn Ihre Software als Daemon läuft, der keine Root-Rechte benötigt, müssen Sie für ihn einen Benutzer erstellen. Es gibt zwei Arten von Benutzern in Debian, die für Pakete verwendet werden können: statische UIDs (werden von base-passwd vergeben, eine Liste der statischen Benutzern in Debian finden Sie bei „Benutzer und Gruppen des Betriebssystems“) und dynamisches UIDs, die in einem zugewiesenen Bereich liegen.

Im ersten Fall müssen Sie mit base-passwd eine Benutzer- oder Gruppen-ID erstellen. Wenn der Benutzer verfügbar ist, muss das Paket, das Sie anbieten möchten, eine Abhängigkeit vom Paket base-passwd enthalten.

Im zweiten Fall müssen Sie den Systembenutzer entweder *preinst* oder *postinst* erstellen und dafür sorgen, dass das Paket von `adduser (>= 3.11)` abhängt.

Im folgenden Programmbeispiel soll gezeigt werden, wie der Benutzer oder Gruppe, mit deren Rechten der Daemon laufen wird, bei der Installation oder Aktualisierung des Pakets erstellt wird.

```
[...]
case "$1" in
  install|upgrade)

  # If the package has default file it could be sourced, so that
  # the local admin can overwrite the defaults

  [ -f "/etc/default/packagename" ] && . /etc/default/packagename

  # Sane defaults:

  [ -z "$SERVER_HOME" ] && SERVER_HOME=server_dir
  [ -z "$SERVER_USER" ] && SERVER_USER=server_user
  [ -z "$SERVER_NAME" ] && SERVER_NAME="Server description"
  [ -z "$SERVER_GROUP" ] && SERVER_GROUP=server_group

  # Groups that the user will be added to, if undefined, then none.
```

```
  ADDGROUP=""

  # create user to avoid running server as root
  # 1. create group if not existing
  if ! getent group | grep -q "^$SERVER_GROUP:" ; then
     echo -n "Adding group $SERVER_GROUP.."
     addgroup --quiet --system $SERVER_GROUP 2>/dev/null ||true
     echo "..done"
  fi
  # 2. create homedir if not existing
  test -d $SERVER_HOME || mkdir $SERVER_HOME
  # 3. create user if not existing
  if ! getent passwd | grep -q "^$SERVER_USER:"; then
    echo -n "Adding system user $SERVER_USER.."
    adduser --quiet \
            --system \
            --ingroup $SERVER_GROUP \
            --no-create-home \
            --disabled-password \
            $SERVER_USER 2>/dev/null || true
    echo "..done"
  fi
  # 4. adjust passwd entry
  usermod -c "$SERVER_NAME" \
          -d $SERVER_HOME   \
          -g $SERVER_GROUP  \
             $SERVER_USER
  # 5. adjust file and directory permissions
  if ! dpkg-statoverride --list $SERVER_HOME >/dev/null
  then
      chown -R $SERVER_USER:adm $SERVER_HOME
      chmod u=rwx,g=rxs,o= $SERVER_HOME
  fi
  # 6. Add the user to the ADDGROUP group
  if test -n $ADDGROUP
  then
      if ! groups $SERVER_USER | cut -d: -f2 | \
         grep -qw $ADDGROUP; then
           adduser $SERVER_USER $ADDGROUP
      fi
  fi
  ;;
  configure)

[...]
```

Außerdem müssen Sie für das init.d-Skript sicherstellen,

- dass der Daemon beim Starten seine Rechte ablegt: Wenn die Software nicht selbst den setuid(2) oder seteuid(2) Aufruf absetzt, sollten Sie die Option **--chuid** für start-stop-daemon verwenden.
- dass der Daemon nur angehalten wird, wenn die Benutzer-IDs übereinstimmen. Dafür ist die Option `--user` von **start-stop-daemon** hilfreich.
- dass der Daemon nicht gestartet wird, wenn sein Benutzer oder Gruppe nicht existiert:

```
if ! getent passwd | grep -q "^server_user:"; then
   echo "Server user does not exist. Aborting" >&2
   exit 1
fi
if ! getent group | grep -q "^server_group:" ; then
   echo "Server group does not exist. Aborting" >&2
   exit 1
fi
```

Wenn das Paket einen Systembenutzer erstellt, kann er wieder in *postrm* entfernt werden, wenn das Paket vollständig gelöscht wird (purge). Dabei gibt es allerdings einen Nachteil. Zum Beispiel werden Dateien, die von dem Benutzer des Daemons erstellt wurden, benutzerlos und können später einem neuen Benutzer gehören, dem die gleiche UID zugewiesen wurde [1]. Daher ist nicht zwingend notwendig, dass Benutzer beim vollständigen Löschen eines Pakets entfernt werden. Dies hängt vielmehr vom jeweiligen Paket ab. Im Zweifelsfall sollte der Administrator gefragt werden (mit **debconf**), was passieren soll, wenn ein Paket gelöscht wird.

Maintainers that want to remove users in their *postrm* scripts are referred to the **deluser**/**deluser** `--system` option.

Wenn ein Programm unter einem Benutzer mit beschränkten Rechten läuft, wird sichergestellt, dass Sicherheitsprobleme nicht das gesamte System beschädigen können. Dieses Vorgehen beachtet auch das Prinzip der *geringst möglichen Privilegien*. Denken Sie daran, dass Sie die Rechte eines Programms auch noch durch andere Methoden als beschränkte Benutzerrechte weiter einschränken können [2]. Weitere Informationen finden Sie im Abschnitt http://www.dwheeler.com/secure-programs/Secure-Programs-HOWTO/minimize-privileges.html des Buchs *Secure Programming for Linux and Unix HOWTO*.

[1] Interessante Diskussionen zu diesem Thema finden sich in http://lists.debian.org/debian-mentors/2004/10/msg00338.html und http://lists.debian.org/debian-devel/2004/05/msg01156.html.
[2] Sie können sogar eine SELinux-Richtlinie erstellen.

Kapitel 10. Vor der Kompromittierung

Halten Sie Ihr System sicher

Sie sollten bestrebt sein, Ihr System sicher zu halten, indem Sie seine Verwendung und die es betreffenden Verwundbarkeiten im Auge behalten. Sobald Patches verfügbar sind, sollte Sie diese einspielen. Denn auch wenn Sie zu Beginn ein sehr sicheres System eingerichtet haben, sollten Sie daran denken, dass die Sicherheit eines Systems mit der Zeit nachlässt. Das liegt daran, dass Sicherheitslücken in Systemdiensten entdeckt werden können. Außerdem können Benutzer die Sicherheit untergraben, wenn ihnen das notwendige Verständnis fehlt (z.B. wenn sie aus der Ferne auf ein System mit einem Klartextpasswort oder einem einfach zu erratenden Passwort zugreifen) oder gar weil sie aktiv versuchen, die Sicherheit des Systems auszuschalten (indem sie z.B. zusätzliche Dienste lokal in ihren Konten installieren).

Beobachtung von Sicherheitslücken

Die meisten Administratoren werden sich Sicherheitslücken, die ihr System betreffen, bewusst, wenn sie den dazugehörigen Patch sehen. Sie können aber Angriffen schon im Vorfeld begegnen und vorübergehende Abwehrmaßnahmen einleiten, sobald Sie festgestellt haben, dass Ihr System verwundbar ist. Dies gilt besonders für exponierte Systeme (die also mit dem Internet verbunden sind), die Dienste anbieten. In diesem Fall sollte der Systemadministrator einen Blick auf die bekannten Informationsquellen werfen, um als erster zu wissen, wenn eine Sicherheitslücke für einen kritischen Dienst entdeckt wird.

Typischerweise abonniert man eine Mailingliste für Ankündigungen und beobachtet Webseiten oder Fehlerverfolgungssysteme der Software-Entwickler eines bestimmten Programms. So sollten beispielsweise Apache-Benutzer regelmäßig Apaches http://httpd.apache.org/security_report.html durchsehen und die Mailingliste http://httpd.apache.org/lists.html#http-announce abonnieren.

In order to track known vulnerabilities affecting the Debian distribution, the Debian Testing Security Team provides a https://security-tracker.debian.org/ that lists all the known vulnerabilities which have not been yet fixed in Debian packages. The information in that tracker is obtained through different public channels and includes known vulnerabilities which are available either through security vulnerability databases or http://www.debian.org/Bugs/. Administrators can search for the known security issues being tracked for https://security-tracker.debian.org/tracker/status/release/stable, https://security-tracker.debian.org/tracker/status/release/oldstable, https://security-tracker.debian.org/tracker/status/release/testing, or https://security-tracker.debian.org/tracker/status/release/unstable.

Der Tracker kann mittels einer Benutzerschnittstelle durchsucht werden (nach http://cve.mitre.org/-Namen und dem Paketnamen). Einige Werkzeuge (wie zum Beispiel debsecan, vgl. „Automatisches Überprüfung von Aktualisierungen mit debsecan“) setzen diese Datenbank ein, um auf Verwundbarkeiten des betreffenden Systems hinzuweisen, die noch nicht ausgebessert wurden (d.h. für die eine Ausbesserung bevorsteht).

Sicherheitsbewusste Administratoren können mit diesen Informationen feststellen, welche Sicherheitslücken das System, das sie verwalten, betreffen könnten, wie schwer das Risiko der Lücke wiegt und ob vorübergehend Gegenmaßnahmen zu treffen sind (falls möglich), bis ein Patch verfügbar ist, der das Problem löst.

Sicherheitsprobleme in Veröffentlichungen, die vom Sicherheitsteam von Debian unterstützt werden, sollten irgendwann in Debian-Sicherheits-Ankündigungen (DSA) behandelt werden, die allen Benutzern zur Verfügung gestellt werden (vergleiche „Fortlaufende Aktualisierung des Systems“). Sobald ein Sicherheitsproblem ausgebessert wurde und die Lösung in einer Ankündigung enthalten ist, wird es nicht mehr im Tracker aufgeführt. Sie können es aber immer noch mit einer Suchanfrage (nach dem CVE-Namen) finden, indem Sie http://www.de.debian.org/security/crossreferences verwenden.

Beachten Sie aber, dass die Informationen im Tracker des Debian-Testing-Sicherheitsteams nur bekannte Sicherheitslücken (d.h. solche, die öffentlich sind) beinhalten. In einigen Fällen gibt das Debian-Sicherheitsteam DSA für Pakete heraus, die auf vertraulichen Informationen beruhen, die das Team erhalten hat (z.B. über nicht-öffentliche Mailinglisten der Distributionen oder von Programmautoren). Seien Sie also nicht überrascht, in Sicherheitsankündigungen Sicherheitsprobleme zu entdecken, die nicht im Tracker enthalten sind.

Fortlaufende Aktualisierung des Systems

Sie sollten regelmäßig Sicherheitsaktualisierungen durchführen. Der ganz überwiegende Anteil der Exploits nutzt bekannte Sicherheitslücken aus, die nicht rechtzeitig ausgebessert wurden. Dies wird in der http://www.cs.umd.edu/~waa/vulnerability.html dargestellt, die 2001 auf dem »IEEE Symposium on Security and Privacy« vorgestellt wurde. Das Durchführen einer Aktualisierung wird unter „Ausführen von Sicherheitsaktualisierungen“ beschrieben.

Überprüfung von Hand, welche Sicherheitsaktualisierungen verfügbar sind

Debian besitzt ein Werkzeug, um zu überprüfen, ob ein System aktualisiert werden muss. Viele Benutzer wollen aber einfach von Hand überprüfen, ob Sicherheitsaktualisierungen für ihr System zur Verfügung stehen.

Wenn Sie Ihr System nach der Beschreibung unter „Ausführen von Sicherheitsaktualisierungen“ eingerichtet haben, müssen Sie nur Folgendes tun:

```
# apt-get update
# apt-get upgrade -s
[ ... überprüfen der zu aktualisierenden Pakete ... ]
# apt-get upgrade
# checkrestart
[ ... Neustart der Dienste, die neu gestartet werden müssen ... ]
```

Weiter müssen alle Dienste, deren Bibliotheken aktualisiert wurden, neu gestartet werden. Hinweis: Lesen Sie „Ausführen von Sicherheitsaktualisierungen“ für weitere Informationen zu Bibliotheks- (und Kernel-)Aktualisierungen.

Die erste Zeile wird die Liste der verfügbaren Pakete von den festgelegten Paketquellen herunterladen. Die Option `-s` wird eine Simulation durchführen, d.h. es werden *keine* Pakete heruntergeladen oder installiert. Vielmehr teilt es Ihnen mit, welche heruntergeladen und installiert werden sollen. Durch dieses Ergebnis könnten Sie erfahren, welche Pakete von Debian ausgebessert wurden und als Sicherheitsaktualisierung verfügbar sind. Zum Beispiel:

```
# apt-get upgrade -s
Reading Package Lists... Done
Building Dependency Tree... Done
2 packages upgraded, 0 newly installed, 0 to remove and 0  not upgraded.
Inst cvs (1.11.1p1debian-8.1 Debian-Security:3.0/stable)
Inst libcupsys2 (1.1.14-4.4 Debian-Security:3.0/stable)
Conf cvs (1.11.1p1debian-8.1 Debian-Security:3.0/stable)
Conf libcupsys2 (1.1.14-4.4 Debian-Security:3.0/stable)
```

In this example, you can see that the system needs to be updated with new cvs and cupsys packages which are being retrieved from *woody's* security update archive. If you want to understand why these packages

are needed, you should go to http://security.debian.org and check which recent Debian Security Advisories have been published related to these packages. In this case, the related DSAs are https://lists.debian.org/debian-security-announce/2003/msg00014.html (for cvs) and https://lists.debian.org/debian-security-announce/2003/msg00013.html (for cupsys).

Hinweis: Sie werden Ihr System neustarten müssen, wenn der Kernel aktualisiert wurde.

Überprüfung von Aktualisierungen auf dem Desktop

Seit Debian 4.0 *Lenny* gibt es in Debian update-notifier, das in einer Standardinstallation installiert wird. Es ist eine GNOME-Anwendung, die beim Starten des Desktops mitgestartet wird. Sie kann geprüft, welche Aktualisierungen für Ihr System zur Verfügung stehen, und diese installieren. Dafür verwendet es update-manager.

In dem Stable-Zweig gibt es Aktualisierungen nur zum Entfernen von Sicherheitsproblemen oder dann, wenn eine Zwischenveröffentlichung (point release) angeboten wird. Wenn das System richtig konfiguriert ist, um Sicherheitsaktualisierungen zu erhalten (wie in „Ausführen von Sicherheitsaktualisierungen“ beschrieben), und Sie mit **cron** die Paketinformationen aktualisieren, werden Sie durch ein Desktop-Symbol in dem Benachrichtigungsbereich des Desktops über Aktualisierungen informiert werden.

Diese Benachrichtigung ist nicht aufdringlich und zwingt den Benutzer nicht dazu, die Aktualisierungen zu installieren. Über das Symbol kann der Desktop-Benutzer (mit dem Passwort des Systemadministrators) zu einer einfachen graphischen Benutzeroberfläche gelangen, um sich die verfügbaren Aktualisierungen anzeigen zu lassen und zu installieren.

Diese Anwendung arbeitet damit, dass sie die Paketdatenbank abruft und ihren Inhalt mit dem System vergleicht. Wenn die Datenbank regelmäßig mit **cron** aktualisiert wird, ist ihr Inhalt aktueller als die auf dem System installierten Pakete, worauf die Anwendung Sie hinweisen wird.

Apt richtet eine solche Aufgabe ein (`/etc/cron.d/apt`), die abhängig von der Konfiguration von Apt ausgeführt wird (genauer gesagt je nach *APT::Periodic*). In der GNOME-Umgebung kann dieser Wert über System > Admin > Software origins > Updates oder mit **/usr/bin/software-properties** geändert werden.

Wenn Ihr System täglich die Paketliste herunterladen soll, aber nicht die Pakete selbst, sollte `/etc/apt/apt.conf.d/10periodic` etwa so aussehen:

```
APT::Periodic::Update-Package-Lists "1";
APT::Periodic::Download-Upgradeable-Packages "0";
```

Sie können auch eine andere cron-Aufgabe verwenden, z.B. die von cron-apt installierte (vgl. „Automatisches Überprüfung von Aktualisierungen mit cron-apt“). Damit können Sie auch nur per Hand nach Aktualisierungen suchen.

Benutzer der KDE-Umgebung sollten stattdessen adept und adept-notifier installieren, die vergleichbare Funktionen anbieten, aber nicht in der Standardinstallation enthalten sind.

Automatisches Überprüfung von Aktualisierungen mit cron-apt

Eine andere Methode für automatische Sicherheitsaktualisierungen ist cron-apt. Dieses Paket stellt ein Werkzeug zur Verfügung, mit dem das System in regelmäßigen Abständen (mit einem Cronjob) aktualisiert wird. Es kann so konfiguriert werden, dass es E-Mails mit dem lokalen Mail-Transport-Agent an den Systemadministrator schickt. Standardmäßig wird es nur die Paketliste aktualisieren und neue Pakete herunterladen. Es kann aber so konfiguriert werden, dass es automatisch Aktualisierungen installiert.

Hinweis: Wenn Sie vorhaben, Ihr System automatisch zu aktualisieren (auch wenn Sie sich nur die Pakete herunterladen), sollten Sie sich vielleicht die Distributionsveröffentlichung ansehen, wie in „Überprüfung der Distribution mit der `Release`-Datei“ beschrieben wird. Anderenfalls können Sie sich nicht sicher sein, dass die heruntergeladenen Pakete wirklich aus einer vertrauenswürdigen Quelle stammen.

Weitere Informationen finden Sie auf der http://www.debian-administration.org/articles/162.

Automatisches Überprüfung von Aktualisierungen mit debsecan

Das Programm **debsecan** ermittelt den Sicherheitsstatus, indem es sowohl nicht installierte Sicherheitsaktualisierungen als auch Sicherheitslücken meldet. Im Gegensatz zu cron-apt, das nur Informationen zu verfügbaren Sicherheitsaktualisierungen bereitstellt, bezieht dieses Werkzeug auch Informationen von der Datenbank über Sicherheitslücken, die von Debians Sicherheitsteam verwaltet wird. Darin befinden sich auch Informationen über Lücken, die noch nicht durch eine Sicherheitsaktualisierung geschlossen wurden. Daher kann es Administratoren besser helfen, Sicherheitslücken im Blick zu behalten (wie unter „Beobachtung von Sicherheitslücken“ beschrieben).

Nach der Installation des Debian-Pakets debsecan wird es mit Zustimmung des Administrators eine cron-Aufgabe erstellen, die das Programm aufruft und das Ergebnis an einen bestimmten Benutzer schickt, wenn sie ein verwundbares Paket findet. Sie wird auch Informationen aus dem Internet laden. Bei der Installation wird auch nach dem Ort der Sicherheitsdatenbank gefragt, dieser wird in `/etc/default/debsecan` gespeichert. Er kann leicht so angepasst werden, damit Systeme ohne Internetzugang auf einen lokale Spiegelserver zugreifen können und nur dieser mit der Sicherheitsdatenbank verbunden sein muss.

Beachten Sie jedoch, dass das Sicherheitsteam viele Verwundbarkeiten aufführt, die (wie risikoarme Probleme) nicht mit einer Sicherheitsaktualisierung ausgebessert werden oder bei denen sich später herausstellt, dass sie, anders als zunächst angenommen, Debian nicht betreffen. **Debsecan** wird alle Verwundbarkeiten melden, wodurch diese Meldungen deutlich umfangreicher werden als bei den anderen beschriebenen Werkzeugen.

Weitere Informationen finden Sie auf der http://www.enyo.de/fw/software/debsecan/.

Andere Methoden für Sicherheitsaktualisierungen

Es gibt auch apticron, das ähnlich wie cron-apt nach Aktualisierungen sucht und eine E-Mail an den Administrator schickt. Weitere Informationen über apticron finden Sie auf der http://www.debian-administration.org/articles/491.

Sie können auch einen Blick auf http://clemens.endorphin.org/secpack/ werfen. Es ist ein inoffizielles Programm, um Sicherheitsaktualisierungen von security.debian.org mit Prüfung der Signatur durchzuführen. Es wurde von Fruhwirth Clemens geschrieben. Eine weitere Alternative bietet die Nagios-Erweiterung http://www.unixdaemon.net/nagios_plugins.html#check_debian_packages von Dean Wilson.

Vermeiden Sie den Unstable-Zweig

Falls Sie nicht Zeit darauf verwenden wollen, selbst Pakete zu patchen, wenn Verwundbarkeiten entdeckt werden, sollten Sie auf produktiven Systemen *nicht* Debians Unstable-Zweig einsetzen. Der Hauptgrund dafür ist, dass es für *Unstable* keine Sicherheitsaktualisierungen gibt.

Es ist eine Tatsache, dass manche Sicherheitsprobleme nur in Unstable auftreten und *nicht* in *Stable*. Das rührt daher, dass dort ständig neue Funktionen zu den Anwendungen hinzugefügt werden und auch neue Anwendungen aufgenommen werden, die unter Umständen noch nicht vollständig getestet wurden.

Um im *Unstable*-Zweig Sicherheitsaktualisierungen durchzuführen, müssen Sie unter Umständen eine vollständige Aktualisierung mit einer neuen Version durchführen (was viel mehr als nur das betroffene

Pakete aktualisieren könnte). Sicherheitsaktualisierungen wurden – mit Ausnahmen – nur in den *Stable*-Zweig zurückportiert. Die Grundidee ist, dass mit Sicherheitsaktualisierungen *kein neuer Code* hinzugefügt werden sollte, sondern nur wichtige Probleme beseitigt werden.

Denken Sie daran, dass Sie allerdings den Sicherheitstracker verwenden können (wie unter „Beobachtung von Sicherheitslücken“ beschrieben), um bekannte Sicherheitsprobleme für diesen Zweig nachzuvollziehen.

Sicherheitsunterstützung für den Testing-Zweig

Wenn Sie den *Testing*-Zweig verwenden, müssen Sie einige Problemkreise hinsichtlich der Verfügbarkeit von Sicherheitsaktualisierungen in Betracht ziehen:

- Wenn eine Sicherheitslücke geschlossen wurde, portiert das Sicherheitsteam den Patch nach *Stable* zurück (da Stable normalerweise einige Minor- oder Majorversionen zurückliegt). Die Paketbetreuer sind dafür verantwortlich, Pakete für den *Unstable*-Zweig vorzubereiten. Grundlage dafür ist normalerweise eine neue Veröffentlichung des Originalprogramms. Manchmal ereignen sich die Änderungen fast zur selben Zeit und manchmal enthält eine der Veröffentlichungen eine Ausbesserung einer Sicherheitslücke vor einer anderen. Pakete in *Stable* werden gründlicher getestet als die in *Unstable*, da letztere in den meisten Fällen die neueste Veröffentlichung des Originalprogramms enthält (welches neue, unbekannte Fehler enthalten könnte).

- Gewöhnlich sind Sicherheitsaktualisierungen für den *Unstable*-Zweig verfügbar, wenn der Paketbetreuer ein neues Paket baut, und für den *Stable*-Zweig, wenn das Security Team eine neue Version hochlädt und ein DSA veröffentlicht. Beachten Sie, dass beides nicht des *Testing*-Zweig verändert.

- Wenn keine (neuen) Fehler in der *Unstable*-Version des Pakets entdeckt werden, wandert es nach ein paar Tagen nach *Testing*. Das dauert normalerweise zehn Tage. Es hängt allerdings von der Priorität des Hochladens der Veränderung ab und davon, ob das Paket von *Testing* zurückgehalten wird, da Abhängigkeiten nicht aufgelöst werden können. Beachten Sie, dass wenn das Paket daran gehindert ist, nach Testing zu wandern, auch die Priorität des Hochladens daran nichts ändern kann.

Dieses Verhalten könnte sich je nach dem Status der Veröffentlichung der Distribution verändern. Wenn eine Veröffentlichung unmittelbar bevorsteht, werden auch das Sicherheitsteam oder die Paketbetreuer direkt Aktualisierungen für Testing zur Verfügung stellen.

Zusätzlich kann auch das http://secure-testing-master.debian.net Debian-Testing-Sicherheitsankündigungen (DTSA) für Pakete im Testing-Zweig herausgeben, wenn sofort eine Lücke in diesem Zweig geschlossen werden muss und die normale Vorgehensweise nicht abgewartet werden kann (oder die übliche Vorgehensweise durch andere Pakete blockiert ist).

Benutzer, die von diesem Angebot Gebrauch machen wollen, müssen folgende Zeilen ihrer `/etc/apt/sources.list` (anstatt der Zeilen, die unter „Ausführen von Sicherheitsaktualisierungen“ dargestellt wurden) hinzufügen:

```
    deb http://security.debian.org testing/updates main contrib non-free
# Diese Zeile macht es möglich, auch Quellpakete herunterzuladen
    deb-src  http://security.debian.org testing/updates main contrib non-free
```

Für weitere Informationen zu diesem Angebot können Sie die entsprechende http://lists.debian.org/debian-devel-announce/2006/05/msg00006.html lesen. Dieses Angebot startete offiziell im http://lists.debian.org/debian-devel-announce/2005/09/msg00006.html als zusätzliches Paketdepot und wurde später in das allgemeine Sicherheitsarchiv integriert.

Automatische Aktualisierungen in einem Debian GNU/Linux System

Es sei vorweggeschickt, dass automatische Aktualisierungen nicht vollständig empfohlen werden, da Administratoren die DSAs durchsehen und die Bedeutung einer bestimmten Sicherheitsaktualisierung verstehen sollten.

Wenn Sie Ihr System automatisch aktualisieren wollen, sollten Sie Folgendes durchführen:

- Konfigurieren Sie **apt** so, dass Pakete, die Sie nicht aktualisieren wollen, ihrer momentane Version beibehalten. Das können Sie entweder mit einer Eigenschaft von **apt**, dem *pinning* (festheften), erreichen, oder Sie kennzeichnen sie mit **dpkg** oder **dselect** als *hold* (festgehalten).

  Um Pakete einer bestimmten Veröffentlichung mit pinning festzuheften, müssen Sie `/etc/apt/preferences` bearbeiten (siehe apt_preferences(5)) und Folgendes hinzufügen:

  ```
  Package: *
  Pin: release a=stable
  Pin-Priority: 100
  ```

  FIXME: verify if this configuration is OK.

- Entweder setzen Sie cron-apt ein, wie in „Automatisches Überprüfung von Aktualisierungen mit cron-apt“ beschrieben wird, und erlauben ihm, heruntergeladene Pakete zu installieren. Oder Sie fügen selbst einen Eintrag für **cron** hinzu, damit die Aktualisierung täglich ausgeführt wird. Ein Beispiel:

  ```
  apt-get update && apt-get -y upgrade
  ```

  Die Option `-y` veranlasst **apt**, für alle Fragen, die während der Aktualisierung auftreten können, »yes« anzunehmen. In manchen Fällen sollten Sie die Option `--trivial-only` (nur Bagatellen) der Option `--assume-yes` (ist gleichbedeutend mit `-y`) vorziehen.[1]

- Richten Sie **debconf** so ein, dass während der Aktualisierung keine Eingabe verlangt wird. Auf diese Weise können Aktualisierungen nicht-interaktiv durchgeführt werden.[2]

- Überprüfen Sie die Ergebnisse der Ausführung von **cron**, die an den Superuser gemailt werden (sofern nicht die Umgebungsvariable `MAILTO` im Skript geändert wurde).

Eine sichere Alternative könnte es sein, die Option `-d` (oder `--download-only`) zu verwenden. Das hat zur Folge, dass die benötigten Pakete nur heruntergeladen, aber nicht installiert werden. Und wenn dann die Ausführung von **cron** zeigt, dass das System aktualisiert werden muss, kann das von Hand vorgenommen werden.

Um diese Aufgaben zu erfüllen, muss das System korrekt konfiguriert sein, um Sicherheitsaktualisierungen herunterzuladen. Dies wurde in „Ausführen von Sicherheitsaktualisierungen“ diskutiert.

Allerdings wird dieses Vorgehen ohne eine genaue Analyse nicht für *Unstable* empfohlen, da Sie Ihr System in einen unbrauchbaren Zustand bringen können, wenn sich ein gravierender Fehler in ein wichtiges Paket eingeschlichen hat und auf Ihrem System installiert wird. *Testing* ist vor diesem Problem etwas bes-

[1] Sie können auch die Option `--quiet` (`-q`) verwenden. Sie verringert die Ausgabe von **apt-get** und wird keine Ausgabe produzieren, wenn keine Pakete installiert werden.
[2] Beachten Sie, dass einige Pakete *nicht* **debconf** verwenden könnten. Die Aktualisierung könnte dann hängen bleiben, da Pakete während ihrer Konfiguration Eingaben des Benutzers verlangen.

ser *geschützt*, da gravierende Fehler eine bessere Chance haben entdeckt zu werden, bevor das Paket in den Testing-Zweig wandert (obwohl Ihnen trotzdem *keine* Sicherheitsaktualisierungen zur Verfügung stehen).

Wenn Sie eine gemischte Distribution haben, also eine Installation von *Stable* mit einige Pakete aus *Testing* oder *Unstable*, können Sie mit den Pinning-Eigenschaften oder der Option `--target-release` von **apt-get** herumspielen, um *nur* die Pakete zu aktualisieren, die Sie früher aktualisiert haben.[3]

Regelmäßiges Überprüfung der Integrität

Mit Hilfe der Basisinformationen, die Sie nach der Installation erstellt haben (also mit dem Schnappschuss, der in „Einen Schnappschuss des Systems erstellen“ beschrieben wird), sollte es Ihnen möglich sein, von Zeit zu Zeit die Integrität des Systems zu überprüfen. Eine Integritätsprüfung kann Veränderungen am Dateisystem entdecken, die durch einen Eindringling oder einen Fehler des Systemadministrators entstanden sind.

Überprüfungen der Integrität sollen, wenn möglich, extern durchgeführt werden.[4] Das bedeutet, dass das Betriebssystem des überprüften Systems nicht verwendet wird, um den falschen Eindruck von Sicherheit (also falsche Negative) zu verhindern, der z.B. durch installierte Rootkits entstehen könnte. Die Datenbank, mit der das System verglichen wird, sollte sich daher auf einem nur-lesbaren Medium befinden.

Falls der Einsatz einer externen Prüfung nicht möglich ist, sollten Sie in Betracht ziehen, die Integritätsprüfung mit den verfügbaren Werkzeugen zur Prüfung der Integrität des Dateisystem durchzuführen (wie unter „Prüfung der Integrität des Dateisystems“ beschrieben). Allerdings sollten Vorsichtsmaßnahmen getroffen werden: Die Datenbank für die Integritätsprüfung sollte nur-lesbar sein und Sie sollten auch sicherstellen, dass das Programm, das die Integrität überprüft, (und der Kernel des Betriebssystems) nicht manipuliert wurde.

Einige Werkzeuge, die im Abschnitt über Programme zur Integritätsprüfung beschrieben wurden, wie z.B. **aide**, **integrit** und **samhain**, sind schon so eingerichtet, dass sie regelmäßige Nachprüfungen durchführen (mittels crontab in den ersten beiden Fällen und mittels eines eigenständigen Daemons bei **samhain**). Sie können den Administrator auf verschiedenen Wegen warnen (normalerweise E-Mail, aber **samhain** kann auch Seiten, SNMP-Traps oder einen Alarm an syslog schicken), wenn sich das Dateisystem verändert.

Wenn Sie eine Sicherheitsaktualisierung des System vorgenommen haben, müssen Sie natürlich den Schnappschuss des Systems neu aufzeichnen, um ihn an die Änderungen durch die Sicherheitsaktualisierung anzupassen.

Aufsetzen einer Eindringlingserkennung

Debian GNU/Linux enthält Programme zur Erkennung von Eindringlingen. Das sind Programme, die unpassende oder bösartige Aktivitäten auf Ihrem lokalen System oder auf anderen System in Ihrem lokalen Netzwerk entdecken. Diese Art von Verteidigung ist wichtig, wenn das System sehr entscheidend ist oder Sie wirklich unter Verfolgungswahn leiden. Die gebräuchlichsten Herangehensweisen sind die statistische Entdeckung von Unregelmäßigkeiten und die Entdeckung bestimmter Muster.

Beachten Sie immer, dass Sie einen Alarm-und-Antwort-Mechanismus brauchen, um Ihre Systemsicherheit mit einer dieser Werkzeuge wirklich zu verbessern. Eindringlingserkennung ist Zeitverschwendung, wenn Sie niemanden alarmieren werden.

Wenn ein bestimmter Angriff entdeckt worden ist, werden die meisten Programme zur Eindringlingserkennung entweder den Vorfall mit **syslog** protokollieren oder E-Mails an Root schicken (der Empfänger

[3] Dies ist ein verbreitetes Problem, da viele Benutzer ein stabiles System betreiben wollen, aber einige Pakete aus *Unstable* einsetzen, um die neusten Funktionen zu haben. Das kommt daher, dass sich manche Projekte schneller entwickeln als die Veröffentlichungen von Debians *Stable*.
[4] Ein leichter Weg, das ist tun, ist die Verwendung einer Live-CD wie http://www.knoppix-std.org/, die sowohl die Programme zur Integritätsprüfung als auch die dazugehörige Datenbank enthält.

der E-Mails kann normalerweise eingestellt werden). Ein Administrator muss die Programme passend konfigurieren, so dass falsche Positivmeldungen keinen Alarm auslösen. Alarme können auf einen laufenden Angriff hindeuten und wären später – sagen wir mal am nächsten Tag – nicht mehr nützlich, da der Angriff dann bereits erfolgreich beendet worden sein könnte. Stellen Sie also sicher, dass es eine passende Regelung über die Handhabung von Alarmen gibt, und dass technische Maßnahmen zur Umsetzung dieser Regelung vorhanden sind.

Eine interessante Quelle für Informationen ist http://www.cert.org/tech_tips/intruder_detection_checklist.html.

Netzwerkbasierte Eindringlingserkennung

Programme, die der netzwerkbasierten Eindringlingserkennung dienen, überwachen den Verkehr eines Netzwerkabschnitts und arbeiten auf Grundlage dieser Daten. Genauer ausgedrückt, es werden die Pakete im Netzwerk untersucht, um festzustellen, ob sie mit bestimmten Merkmalen übereinstimmen.

snort ist ein vielseitiger Paketschnüffler und -logger, der Angriffe mit Hilfe einer Bibliothek von Angriffssignaturen erkennt. Es erkennt eine breite Palette von Angriffen und Tests, wie zum Beispiel Pufferüberläufe, verdecktes Abtasten von Ports (stealth port scans), CGI Angriffe, SMB Tests und vieles mehr. **snort** hat auch die Fähigkeit, einen zeitnahen Alarm auszulösen. Dies ist ein Werkzeug, das auf jedem Router installiert werden sollte, um ein Auge auf Ihr Netzwerk zu haben. Installieren Sie es einfach mit `apt-get install snort`, beantworten Sie die Fragen und beobachten Sie die Protokolle. Für einen etwas breiteren Sicherheitsrahmen sollten Sie sich http://www.prelude-ids.org ansehen.

Debians Paket snort hat viele Sicherheitstests standardmäßig eingeschaltet. Jedoch sollten Sie die Konfiguration anpassen, um die Dienste, die auf Ihrem System laufen, zu berücksichtigen. Sie können auch zusätzliche Tests speziell für diese Dienste nutzen.

Es gibt noch andere, einfachere Werkzeuge, die dazu benutzt werden können, Angriffe auf das Netzwerk zu erkennen. portsentry ist ein interessantes Paket, das Sie warnen kann, wenn jemand Ihre Rechner scannt. Auch andere Programme wie ippl oder iplogger erkennen bestimmte IP (TCP und ICMP) Angriffe, auch wenn sie nicht so fortgeschrittene Techniken zur Erkennung von Netzwerkangriffen wie **snort** bieten.

Sie können jedes dieser Werkzeuge mit dem Paket **idswakeup** testen. Das ist ein Shell-Skript, das falsche Alarme verursacht und Signaturen vieler gebräuchlicher Angriffe enthält.

Hostbasierte Eindringlingserkennung

Eine Eindringlingserkennung, die auf einem Host basiert, beruht darauf, Software auf dem zu überwachenden System zu laden, die Protokolldateien und die Überwachungsprogramme des Systems als Datengrundlage verwendet. Sie sucht nach verdächtigen Prozessen, kontrolliert den Zugang zum Host und überwacht u.U. auch Änderungen an kritischen Systemdateien.

tiger ist ein älteres Programm zur Eindringlingserkennung, dass seit der Woody-Distribution auf Debian portiert wurde. **tiger** bietet Tests von verbreiteten Problemen in Zusammenhang mit Einbrüchen, wie der Stärke von Passwörtern, Problemen mit dem Dateisystem, kommunizierenden Prozessen und anderen Möglichkeiten, mit denen Root kompromittiert werden könnte. Dieses Paket umfasst neue, debianspezifische Sicherheitstests, einschließlich der MD5-Summen von installierten Programmen, des Orts von Dateien, die zu keinem Paket gehören und einer Analyse von lokalen lauschenden Prozessen. Die Standardinstallation lässt **tiger** einmal am Tag laufen und einen Bericht erstellen, der an den Superuser geschickt wird und Informationen zu möglichen Kompromittierungen enthält.

Programme zur Protokollanalyse, wie zum Beispiel logcheck, können zusätzliche benutzt werden, um Einbruchsversuche zu erkennen. Siehe „Nutzung und Anpassung von **logcheck**“.

Daneben können Pakete, welche die Integrität des Dateisystems überwachen (siehe „Prüfung der Integrität des Dateisystems“), sehr nützlich sein, um Anomalien in einer abgesicherten Umgebung zu erkennen. Ein erfolgreicher Einbruch wird höchstwahrscheinlich Dateien auf dem lokalen Dateisystem verändern, um die lokalen Sicherheitsrichtlinien zu umgehen, Trojaner zu installieren oder Benutzer zu erstellen. Solche Ereignisse können mit Prüfwerkzeugen der Dateisystemintegrität erkannt werden.

Vermeiden von Root-Kits

Ladbare Kernel-Module (LKM)

Ladbare Kernel-Module sind Dateien, die nachladbare Teile des Kernels enthalten. Sie werden dazu verwendet, die Funktionalität des Kernel zu erweitern. Der Hauptnutzen des Einsatzes von Modulen liegt darin, dass Sie zusätzliche Geräte wie eine Ethernet- oder Soundkarte hinzuzufügen können, ohne dass die Kernelquelle gepatcht und der gesamte Kernel neu übersetzt werden müsste. Allerdings können Cracker LKMs für Root-Kits (knark und adore) benutzen, um auf GNU/Linux Systemen Hintertüren zu öffnen.

LKM-Hintertüren sind ausgeklügelter und schwere zu entdecken als traditionelle Root-Kits. Sie können Prozesse, Dateien, Verzeichnisse und sogar Verbindungen verstecken, ohne den Quellcode der Programme verändern zu müssen. Zum Beispiel kann ein bösartiges LKM den Kernel dazu zwingen, bestimmte Prozesses vor `procfs` zu verstecken, so dass nicht einmal eine unmanipulierte Kopie des Programms **ps** alle Informationen über die aktuellen Prozesse korrekt auflisten.

Erkennen von Root-Kits

Es gibt zwei Herangehensweisen, um Ihr System gegen LKM-Root-Kits zu verteidigen: die aktive Verteidigung und die reaktive Verteidigung. Die Sucharbeit kann einfach und schmerzlos sein oder schwierig und ermüdend, ganz abhängig von der Maßnahme, die Sie ergreifen.

Proaktive Verteidigung

Der Vorteil dieser Art der Verteidigung ist, dass schon verhindert wird, dass das System Schaden nimmt. Eine mögliche Strategie ist, *das Ziel als Erster zu erreichen*, also ein LKM zu laden, das dazu da ist, das System vor anderen böswilligen LKMs zu schützen. Eine andere Maßnahme ist es, dem Kernel Fähigkeiten zu entziehen. Zum Beispiel können Sie aus dem Kernel vollständig die Fähigkeit von ladbaren Kernel-Modulen entfernen. Beachten Sie allerdings, dass es Root-Kits gibt, die selbst in diesen Fällen funktionieren. Es gibt auch welche, die direkt `/dev/kmem` (Kernelspeicher) manipulieren, um sich zu verstecken.

Debian GNU/Linux hat ein paar Pakete, die dazu verwendet werden können, eine aktive Verteidigung aufzusetzen:

lcap - eine benutzerfreundliche Schnittstelle, um dem Kernel *Fähigkeiten* zu entziehen (kernelbasierte Zugriffskontrolle), um das System sicherer zu machen. Beispielsweise wird das Ausführen von `lcap CAP_SYS_MODULE` [5] die Fähigkeit der ladbaren Module entfernen (sogar für Root).[6] Weitere (etwas ältere) Informationen zu Kernelfähigkeiten finden Sie in Jon Corbets Abschnitt http://lwn.net/1999/1202/kernel.php3 auf LWN vom Dezember 1999.

[5] Es gibt über 28 Fähigkeiten einschließlich `CAP_BSET`, `CAP_CHOWN`, `CAP_FOWNER`, `CAP_FSETID`, `CAP_FS_MASK`, `CAP_FULL_SET`, `CAP_INIT_EFF_SET`, `CAP_INIT_INH_SET`, `CAP_IPC_LOCK`, `CAP_IPC_OWNER`, `CAP_KILL`, `CAP_LEASE`, `CAP_LINUX_IMMUTABLE`, `CAP_MKNOD`, `CAP_NET_ADMIN`, `CAP_NET_BIND_SERVICE`, `CAP_NET_RAW`, `CAP_SETGID`, `CAP_SETPCAP`, `CAP_SETUID`, `CAP_SYS_ADMIN`, `CAP_SYS_BOOT`, `CAP_SYS_CHROOT`, `CAP_SYS_MODULE`, `CAP_SYS_NICE`, `CAP_SYS_PACCT`, `CAP_SYS_PTRACE`, `CAP_SYS_RAWIO`, `CAP_SYS_RESOURCE`, `CAP_SYS_TIME` und `CAP_SYS_TTY_CONFIG`. Alle können deaktiviert werden, um Ihren Kernel abzuhärten.

[6] Um dies tun zu können, müssen Sie nicht lcap installieren, aber damit ist es einfacher, als von Hand `/proc/sys/kernel/cap-bound` anzupassen.

Wenn Sie diese vielen Möglichkeiten auf Ihrem GNU/Linux System nicht wirklich brauchen, sollten Sie die Unterstützung für ladbare Module während der Konfiguration des Kernels abschalten. Das erreichen Sie, indem Sie einfach CONFIG_MODULES=n während des Konfiguration Ihres Kernels oder in der Datei `.config` festsetzen. So werden LKM-Root-Kits vermieden, aber Sie verlieren eine leistungsfähige Eigenschaft des Linux-Kernels. Außerdem kann das Abschalten der nachladbaren Module den Kernel überladen, so dass die Unterstützung ladbarer Module notwendig wird.

Reaktive Verteidigung

Der Vorteil reaktiver Verteidigung ist, dass sie die Systemressourcen nicht überlädt. Sie funktioniert durch das Vergleichen von einer Tabelle der Systemaufrufe mit einer bekanntermaßen sauberen Kopie (`System.map`). Eine reaktive Verteidigung kann den Systemadministrator natürlich nur benachrichtigen, wenn das System bereits kompromittiert wurde.

Die Entdeckung von Root-Kits vollbringt unter Debian chkrootkit. Das Programm http://www.chkrootkit.org prüft Anzeichen von bekannten Root-Kits auf dem Zielsystem. Es ist aber kein völlig sicherer Test.

Geniale/paranoide Ideen — was Sie tun können

Dies ist wahrscheinlich der unsicherste und lustigste Abschnitt, da ich hoffe, dass manche der »Wow, das klingt verrückt«-Ideen umgesetzt werden. Im Folgenden werden nur ein paar Ideen vorgestellt, wie Sie Ihre Sicherheit erhöhen können — abhängig von Ihrem Standpunkt aus können Sie sie für genial, paranoid, verrückt oder sicher halten.

- Mit Pluggable Authentication Modules (PAM) herum spielen. Wie in einem phrack 56 Artikel geschrieben wurde, ist das Schöne an PAM, dass »Ihrer Fantasie keine Grenzen gesetzt sind.« Das stimmt. Stellen Sie sich vor, Root kann sich nur mit einen Fingerabdruck oder Abtastung des Auges oder einer Kryptokarte einloggen (warum habe ich hier nur »oder« und nicht »und« gesagt?).

- Faschistisches Protokollieren. Ich würde sagen, dass alles, was wir bisher über Protokollieren besprochen haben, unter »weiches Loggen« fällt. Wenn Sie echtes Protokollieren betreiben wollen, besorgen Sie sich einen Drucker mit Endlos-Papier und schicken ihm alle Protokolle. Hört sich lustig an, ist aber zuverlässig und kann nicht manipuliert oder entfernt werden.

- CD-Distribution. Diese Idee ist sehr leicht zu realisieren und bewirkt ganz gute Sicherheit. Erstellen Sie eine abgesicherte Debian-Distribution mit passenden Firewall-Regeln. Erstellen Sie davon ein bootbares ISO-Image und brennen Sie es auf eine CD-ROM. Jetzt haben Sie eine nur lesbare Distribution mit etwa 600 MB Speicherplatz für Dienste. Stellen Sie lediglich sicher, dass alle Daten, die geschrieben werden sollen, übers Netz geschrieben werden. Für einen Eindringling ist es unmöglich, Schreibzugriff auf diesem System zu erhalten. Alle Änderungen, die ein Eindringling vornimmt, werden mit einem Neustart des Systems rückgängig gemacht.

- Schalten Sie die Modul-Fähigkeiten des Kernels ab. Wenn Sie die Nutzung von Kernel-Modulen während der Kernel-Kompilierung abschalten, werden viele kernelbasierte Hintertüren nicht einsetzbar, da die meisten von ihnen darauf beruhen, modifizierte Kernel-Module zu installieren (siehe oben).

- Protokollieren über ein serielles Kabel (von Gaby Schilders). So lange Server immer noch serielle Schnittstellen haben: Stellen Sie sich ein Protokollsystem für eine Anzahl von Servern vor. Es ist vom Netz abgeschnitten und mit den Servern über einen Multiplexer für serielle Schnittstellen (Cyclades oder ähnliches) verbunden. Jetzt sollen alle Ihre Server die Protokolle an ihre serielle Schnittstelle schicken, einfach nur hinschreiben. Die Protokollmaschine akzeptiert nur einfachen Text als Eingabe auf ihrer seriellen Schnittstelle und schreibt ihn lediglich in eine Protokolldatei. Schließen Sie einen CD- oder DVD-Brenner an. Brennen Sie die Protokolldatei, wenn sie die Größe des Mediums erreicht hat.

Wenn es jetzt nur noch CD-Brenner mit automatischem Medien-Wechsel gäbe ... Nicht so dauerhaft gespeichert wie ein Ausdruck, aber mit dieser Methode kann man größere Mengen handhaben und die CD-ROMs nehmen nicht so viel Platz weg.

- Ändern Sie die Dateiattribute mit **chattr** (dem Tipps-HOWTO von Jim Dennis entnommen). Nachdem Sie Ihr System sauber installiert und konfiguriert haben, verwenden Sie das Programm **chattr** mit dem Attribut `+i`, um Dateien unveränderbar zu machen (die Datei kann nicht gelöscht, umbenannt, verlinkt oder beschrieben werden). Sie sollten dieses Attribut für alle Dateien in `/bin`, `/sbin/`, `/usr/bin`, `/usr/sbin`, `/usr/lib` und den Kerneldateien in root. Sie können auch eine Kopie aller Dateien in `/etc/` mit **tar** oder dergleichen erstellen und das Archiv als unveränderbar kennzeichnen.

  Mit dieser Vorgehensweise können Sie den Schaden zu begrenzen, den Sie anrichten können, wenn Sie als Root eingeloggt sind. Sie können nicht mehr Dateien mit einer fehlgeleiteten Umleitung überschreiben oder Ihr System durch ein fehlplatziertes Leerzeichen im Kommando **rm -fr** unbenutzbar machen (Sie können aber Ihren Daten immer noch einigen Schaden zufügen – aber Ihre Bibliotheken und Programme sind sicherer).

  Dies macht auch verschiedene Sicherheits- und Denial-of-Service (DoS) Exploits entweder unmöglich oder weitaus schwieriger (da viele von ihnen darauf beruhen, Dateien durch Aktionen eines SETUID-Programms zu überschreiben, das *keinen frei wählbaren Shellbefehl zur Verfügung stellt*.

  Eine Unbequemlichkeit dieser Vorgehensweise macht sich bemerkbar, wenn Sie verschiedene Systemprogramme bauen und installieren. Auf der anderen Seite verhindert dies auch, dass **make install** die Dateien überschreibt. Wenn Sie vergessen, das Makefile zu lesen, und die Dateien, die überschrieben werden sollen, mit chattr -i behandelt haben (und die Verzeichnisse, in denen Sie neue Dateien erstellen wollen), schlägt der make-Befehl fehl. Sie müssen nur das Kommando **chattr** ausführen und make neu aufrufen. Sie können diese Gelegenheit gleich dazu benutzen, Ihre alten bin's und libs auszumisten und sie z.B. in ein .old/-Verzeichnis oder Tar-Archiv zu verschieben.

  Beachten Sie, dass dies Sie auch daran hindert, die Pakete Ihres Systems zu aktualisieren, da die Dateien aus den Paketen nicht überschrieben werden können. Also sollten Sie vielleicht ein Skript oder einen anderen Mechanismus haben, der das immutable-Flag auf allen Dateien deaktiviert, bevor Sie ein **apt-get update** ausführen.

- Spielen Sie mit der UTP-Verkabelung herum. Schneiden Sie dazu zwei oder vier Kabel durch und stellen ein Kabel her, das nur Verkehr in eine Richtung zulässt. Verwenden Sie dann UDP-Pakete, um Informationen an die Zielmaschine zu schicken, die ein sicherer Protokollserver oder ein System zur Speicherung von Kreditkartennummern sein kann.

Einrichten eines Honigtopfes (honeypot)

Ein Honigtopf ist ein System, das darauf ausgelegt ist, Systemadministratoren beizubringen, wie Cracker ein System abtasten und darin einbrechen. Es ist eine Systemeinstellung mit der Erwartung und dem Zweck, dass das System abgetastet und angegriffen und möglicherweise darin eingebrochen wird. Wenn Systemadministratoren erfahren, welche Werkzeuge und Methoden Cracker anwenden, können sie daraus lernen, wie sie ihr System und Netzwerk besser schützen.

Debian GNU/Linux-Systeme können leicht als Honigtopf eingerichtet werden, wenn Sie Zeit opfern, sie aufzusetzen und zu überwachen. Sie können leicht den gefälschten Server, die Firewall[7], die den Honigtopf überwacht, und ein Programm, das Eindringling ins Netzwerk entdecken kann, einrichten. Verbinden Sie den Honigtopf mit dem Internet und warten Sie ab. Stellen Sie sicher, dass Sie rechtzeitig alarmiert werden (siehe „Die Wichtigkeit von Protokollen und Alarmen“), wenn in das System eingedrungen wird, damit Sie

[7] Sie sollten typischerweise eine Bridge-Firewall einsetzen, damit die Firewall selbst nicht entdeckt werden kann. Lesen Sie mehr dazu unter „Aufsetzenden einer Bridge-Firewall“.

geeignete Schritte einleiten und den Angriff beenden können, wenn Sie genug gesehen haben. Hier folgen einige Pakete und Probleme, die Sie in Betracht ziehen sollten, wenn Sie einen Honigtopf einrichten:

- die Firewall-Technologie, die Sie verwenden (verfügbar durch den Linux-Kernel)
- syslog-ng: nützlich, um Protokolle des Honigtopfs zu einem entfernen syslog-Server zu schicken
- snort, um allen eingehenden Netzwerkverkehr auf den Honigtopf mitzuschneiden und die Angriffe zu erkennen
- osh, eine eingeschränkte Shell mit Protokollfunktion, die unter SETUID-Root läuft und verbesserte Sicherheit hat (siehe den Artikel von Lance Spitzner weiter unten)
- natürlich alle Daemons, die Sie auf dem falschen Honigtopfserver verwenden wollen: Je nachdem, welche Art von Angreifer Sie analysieren wollen, können Sie den Honigtopf abhärten und die Sicherheitsaktualisierungen einspielen (oder eben *nicht*).
- Integritätsprüfer (siehe „Prüfung der Integrität des Dateisystems“) und das Coroner's Toolkit (tct), um nach dem Angriff eine Analyse durchzuführen
- honeyd und farpd, um einen Honigtopf einzurichten, der auf Verbindungen zu ungenutzten IP-Adressen lauscht und diese an Skripte weiterleitet, die echte Dienste simulieren. Sehen Sie sich auch iisemulator an.
- tinyhoneypot, um einen einfachen Honigtopf-Server mit gefälschten Diensten einzurichten

Falls Sie kein System übrig haben, um die Honigtöpfe und Systeme, die das Netzwerk schützen und kontrollieren, zu bauen, können Sie die Technologie zur Virtualisierung einsetzen, die in **xen** oder **uml** (User-Mode-Linux) enthalten ist. Wenn Sie diesen Weg wählen, müssen Sie Ihren Kernel entweder mit kernel-patch-xen oder kernel-patch-uml patchen.

Sie können mehr über das Aufstellen eines Honigtopfs in Lanze Spitzners exzellentem Artikel http://www.net-security.org/text/articles/spitzner/honeypot.shtml (aus der Know your Enemy Serie). Außerdem stellt das http://project.honeynet.org/ wertvolle Informationen über das Aufstellen von Honigtöpfen und der Analyse von Angriffen auf sie zur Verfügung.

Kapitel 11. Nach einer Kompromittierung (Reaktion auf einem Vorfall)

Allgemeines Verhalten

Wenn Sie während eines Angriffs physisch anwesend sind, sollte Ihre erste Reaktion sein, den Rechner vom Netzwerk zu trennen, indem Sie das Kabel aus der Netzwerkkarte ziehen (wenn das keinen nachteiligen Einfluss auf Ihre Geschäfte hat). Das Netzwerk auf Schicht 1 abzuschalten ist der einzig wirklich erfolgreiche Weg, um den Angreifer aus dem gehackten Rechner herauszuhalten (weiser Ratschlag von Phillip Hofmeister).

Allerdings können einige Werkzeuge, die durch Rootkits, Trojaner oder sogar unehrlichen Benutzern über eine Hintertür installiert wurden, diesen Vorgang erkennen und auf ihn reagieren. Es ist nicht wirklich lustig, wenn Sie sehen, dass `rm -rf /` ausgeführt wird, wenn Sie das Netzwerkkabel ziehen. Wenn Sie nicht bereit sind, dieses Risiko einzugehen, und Sie sich sicher sind, dass in das System eingebrochen wurde, sollten Sie *das Stromkabel herausziehen* (alle, wenn es mehr als eines gibt) und Ihre Daumen drücken. Das hört sich zwar extrem an, verhindert aber tatsächlich eine Logikbombe, die ein Eindringling programmiert haben könnte. Auf jeden Fall sollte ein kompromittiertes System *nicht neugestartet* werden. Entweder sollten die Festplatten in einem anderen System analysiert werden oder Sie sollten ein anderen Medium (eine CD-ROM) benutzen, um das System zu booten und analysieren. Sie sollten *nicht* die Rettungsdisk von Debian verwenden, um das System zu starten. Sie *können* aber die Shell auf der Installationsdisk benutzen (wie Sie wissen, erreichen Sie sie mit Alt+F2), um das System zu analysieren. [1]

Die beste Methode, um ein gehacktes System wiederherzustellen, ist, ein Live-Dateisystem auf einer CD-ROM mit allen Programmen (und Kernel-Modulen) verwenden, die Sie brauchen, um auf das eingebrochene System zugreifen zu können. Sie können das Paket mkinitrd-cd benutzen, um eine solche CD-ROM zu erstellen [2]. Auch die CD-ROM von http://biatchux.dmzs.com/ (früher als Biatchux bekannt) könnte hilfreich sein, da diese Live-CD-ROM forensische Werkzeuge enthält, die in solchen Situationen nützlich sind. Es gibt (noch) kein Programm wie dieses, das auf Debian basiert. Es gibt auch keinen leichten Weg, eine CD-ROM mit Ihrer Auswahl von Debian-Paketen und mkinitrd-cd zu erstellen. Daher werden Sie die Dokumentation lesen müssen, wie Sie Ihre eigenen CD-ROMs machen.

If you really want to fix the compromise quickly, you should remove the compromised host from your network and re-install the operating system from scratch. Of course, this may not be effective because you will not learn how the intruder got root in the first place. For that case, you must check everything: firewall, file integrity, log host, log files and so on. For more information on what to do following a break-in, see http://www.cert.org/tech_tips/root_compromise.html or SANS's https://www.sans.org/white-papers/.

Einige häufige Fragen, wie mit einem gehackten Debian-GNU/Linux-System umzugehen ist, sind unter „Mein System ist angreifbar! (Sind Sie sich sicher?)“ zu finden.

[1] Wenn Sie abenteuerlustig sind, sollten Sie sich am System anmelden und die Informationen aller laufenden Prozesse speichern (Sie bekommen eine Menge aus /proc/nnn/). Es ist möglich, den gesamten ausführbaren Code aus dem Arbeitsspeicher zu ziehen, sogar dann, wenn der Angreifer die ausführbaren Dateien von der Festplatte gelöscht hat. Ziehen Sie danach das Stromkabel.

[2] Das ist auch das Werkzeug, mit dem die CD-ROMs für das Projekt http://www.gibraltar.at/ erstellt werden. Das ist eine Firewall auf einer Live-CD-ROM, die auf der Debian-Distribution beruht.

Anlegen von Sicherheitskopien Ihres Systems

Wenn Sie sich sicher sind, dass das System kompromittiert wurde, vergessen Sie nicht, dass Sie weder der installierten Software noch irgendwelchen Informationen, die sie an Sie liefert, vertrauen können. Anwendungen könnten von einem Trojaner befallen sein, Kernel-Module könnten installiert worden sein, usw.

Am besten ist es, eine komplette Sicherheitskopie Ihres Dateisystems (mittels **dd**) zu erstellen, nachdem Sie von einem sicheren Medium gebootet haben. Debian GNU/Linux CD-ROMs können dazu nützlich sein, da sie auf Konsole 2 eine Shell anbieten, nachdem die Installation gestartet wurde (mit Alt+2 und Enter aktivieren Sie sie). Von dieser Shell aus sollten Sie eine Sicherheitskopie möglichst auf einem anderen Host erstellen (vielleicht auf einen Netzwerk-Datei-Server über NFS/FTP). Dadurch kann eine Analyse des Einbruchs oder eine Neuinstallation durchgeführt werden, während das betroffene System offline ist.

Wenn Sie sich sicher sind, dass es sich lediglich um ein trojanisiertes Kernel-Modul handelt, können Sie versuchen, das Kernel-Image von der Debian-CD-ROM im *rescue*-Modus zu laden. Stellen Sie sicher, dass Sie im *single*-Modus starten, so dass nach dem Kernel keine weiteren Trojaner-Prozesse gestartet werden.

Setzen Sie sich mit dem lokal CERT in Verbindung

Das CERT (Computer and Emergency Response Team) ist eine Organisation, die Ihnen helfen kann, Ihr System nach einem Einbruch wiederherzustellen. Es gibt CERTs weltweit [3]. Sie sollten mit dem lokalen CERT Verbindung aufnehmen, wenn sich ein sicherheitsrelevanter Vorfall ereignet hat, der zu einem Einbruch in Ihr System geführt hat. Die Menschen in der lokalen CERT können Ihnen helfen, Ihr System wiederherzustellen.

Selbst wenn Sie keine Hilfe benötigen, kann es anderen helfen, wenn Sie dem lokalen CERT (oder dem Koordinationszentrum des CERTs) Informationen des Einbruchs zur Verfügung stellen. Die gesammelten Informationen von gemeldeten Vorfällen werden verwendet, um herauszufinden, ob eine bestimmte Verwundbarkeit weit verbreitet ist, ob sich ein neuer Wurm ausbreitet oder welche neuen Angriffswerkzeuge eingesetzt werden. Diese Informationen werden benutzt, um die Internet-Gemeinschaft mit Informationen über die http://www.cert.org/current/ zu versorgen und um http://www.cert.org/incident_notes/ und sogar http://www.cert.org/advisories/ zu veröffentlichen. Ausführliche Informationen, wie (und warum) ein Vorfall gemeldet wird, können Sie auf http://www.cert.org/tech_tips/incident_reporting.html nachlesen.

Sie können auch weniger formale Einrichtungen verwenden, wenn Sie Hilfe brauchen, um Ihr System wiederherzustellen, oder wenn Sie Informationen des Vorfalls diskutieren wollen. Dazu zählen die http://marc.theaimsgroup.com/?l=incidents und die http://marc.theaimsgroup.com/?l=intrusions.

Forensische Analyse

Wenn Sie mehr Informationen sammeln wollen, enthält das Paket tct (The Coroner's Toolkit von Dan Farmer und Wietse Venema) Werkzeuge für eine *post mortem*-Analyse des Systems. tct erlaubt es dem

[3] Dies ist eine Liste einiger CERTs. Ein vollständige Liste erhalten Sie unter http://www.first.org/about/organization/teams/index.html (FIRST ist das Forum von Incident Response and Security Teams): http://www.auscert.org.au (Australien), http://www.unam-cert.unam.mx/ (Mexiko) http://www.cert.funet.fi (Finnland), http://www.dfn-cert.de (Deutschland), http://cert.uni-stuttgart.de/ (Deutschland), http://security.dico.unimi.it/ (Italien), http://www.jpcert.or.jp/ (Japan), http://cert.uninett.no (Norwegen), http://www.cert.hr (Kroatien) http://www.cert.pl (Polen), http://www.cert.ru (Russland), http://www.arnes.si/si-cert/ (Slowenien) http://www.rediris.es/cert/ (Spanien), http://www.switch.ch/cert/ (Schweiz), http://www.cert.org.tw (Taiwan), und http://www.cert.org (USA).

Benutzer, Informationen über gelöschte Dateien, laufende Prozesse und mehr zu sammeln. Sehen Sie für weitere Informationen in die mitgelieferte Dokumentation. Diese und andere Werkzeuge können auch auf http://www.sleuthkit.org/ von Brian Carrier, welches ein Web-Frontend zur forensischen Analyse von Disk-Images zur Verfügung stellt, gefunden werden. In Debian befindet sich sowohl sleuthkit (die Werkzeuge) und autopsy (die grafische Oberfläche).

Forensische Analysen sollten immer auf einer Sicherheitskopie der Daten angewendet werden, *niemals* auf die Daten selbst, da sie durch diese Analyse beeinflusst werden könnten und so Beweismittel zerstört werden würden.

You will find more information on forensic analysis in Dan Farmer's and Wietse Venema's http://www.porcupine.org/forensics/forensic-discovery/ book (available online), as well as in their http://www.porcupine.org/forensics/column.html and their http://www.porcupine.org/forensics/handouts.html. Brian Carrier's newsletter http://www.sleuthkit.org/informer/index.php is also a very good resource on forensic analysis tips. Finally, the http://www.honeynet.org/misc/chall.html are an excellent way to hone your forensic analysis skills as they include real attacks against honeypot systems and provide challenges that vary from forensic analysis of disks to firewall logs and packet captures. For information about available forensics packages in Debian visit https://salsa.debian.org and search for *forensic*.

FIXME: This paragraph will hopefully provide more information about forensics in a Debian system in the coming future.

FIXME: Talk on how to do a debsums on a stable system with the MD5sums on CD and with the recovered file system restored on a separate partition.

FIXME: Add pointers to forensic analysis papers (like the Honeynet's reverse challenge or http://staff.washington.edu/dittrich/).

Analyse von Schadprogrammen

Einige andere Programme aus der Debian-Distribution, die für forensische Analyse verwendet werden können, sind: strace und ltrace

Alle diese Pakete können dazu benutzt werden, um Schurkenprogramme (wie z.B. Hintertüren) zu analysieren, um herauszufinden, wie sie arbeiten und was sie mit dem System anstellen. Einige andere gebräuchliche Werkzeuge sind **ldd** (in libc6), **strings** und **objdump** (beide in binutils).

Wenn Sie eine forensische Analyse von Hintertüren oder verdächtigen Programmen durchführen, die Sie von gehackten Systemen haben, sollten Sie dies in einer sicheren Umgebung durchführen, z.B. in einem bochs-, oder xen-Image oder in einer **chroot**-Umgebung eines Benutzers mit geringen Rechten.[4] Andernfalls könnte auch auf Ihrem eigenen System eine Hintertür eingerichtet oder Root-Rechte erlangt werden.

Falls Sie an der Analyse von Schadprogrammen interessiert sind, sollten Sie das Kapitel http://www.porcupine.org/forensics/forensic-discovery/chapter6.html aus dem Forensik-Buch von Dan Farmer und Wietse Venema lesen.

[4] Seien Sie *äußerst* vorsichtig, wenn sie Chroots einsetzen wollen, da das Binary durch Ausnutzung eines Kernel-Exploits seine Rechte erweitern und es ihm darüber gelingen könnte, Ihr System zu infizieren.

Kapitel 12. Häufig gestellte Fragen / Frequently asked Questions (FAQ)

Dieses Kapitel führt Sie in ein paar der am häufigsten gestellten Fragen in der Security-Mailingliste von Debian ein. Sie sollten sie lesen, bevor Sie dort etwas posten, oder die Leute werden Ihnen »RTFM!« sagen.

Sicherheit im Debian-Betriebssystem

Ist Debian sicherer als X?

Ein System ist so sicher, wie der Administrator fähig ist, es sicher zu machen. Debians Standardinstallation von Diensten zielt darauf ab, *sicher* zu sein. Sie ist aber nicht so paranoid wie andere Betriebssysteme, die Dienste *standardmäßig abgeschaltet*. In jedem Fall muss der Systemadministrator die Sicherheit des System den lokalen Sicherheitsmaßstäben anpassen.

Für eine Übersicht der Sicherheitslücken von vielen Betriebssystemen sollten Sie sich die http://www.cert.org/stats/cert_stats.html ansehen oder sich selber Statistiken mit der http://nvd.nist.gov/statistics.cfm (früher ICAT) erstellen. Sind diese Daten nützlich? Es müssen verschiedene Faktoren berücksichtigt werden, wenn die Daten interpretiert werden sollen. Man sollte beachten, dass diese Daten nicht dazu verwendet werden können, um die Verwundbarkeit eines Betriebssystems mit der eines anderen zu vergleichen. [1] Bedenken Sie außerdem, dass sich einige registrierte Sicherheitslücken im Zusammenhang mit Debian nur auf den *Unstable*-Zweig, also den nicht offiziell veröffentlichten Zweig, beziehen.

Ist Debian sicherer als andere Linux-Distributionen (wie Red Hat, SuSE, ...)?

Der Unterschied zwischen den Linux-Distributionen ist nicht sehr groß mit Ausnahme der Basisinstallation und der Paketverwaltung. Die meisten Distributionen beinhalten zum Großteil die gleichen Anwendungen. Der Hauptunterschied besteht in den Versionen dieser Programme, die mit der stabilen Veröffentlichung der Distribution ausgeliefert werden. Zum Beispiel sind der Kernel, Bind, Apache, OpenSSH, Xorg, gcc, zlib, etc. in allen Linux-Distributionen vorhanden.

Ein Beispiel: Red Hat hatte Pech und wurde veröffentlicht, als foo 1.2.3 aktuell war. Später wurde darin eine Sicherheitslücke entdeckt. Dagegen hatte Debian das Glück, dass es mit foo 1.2.4 ausgeliefert wurde, in dem der Fehler schon behoben war. Das war der Fall beim großen Problem mit http://www.cert.org/advisories/CA-2000-17.html vor ein paar Jahren.

Es besteht eine weitgehende Zusammenarbeit zwischen den jeweiligen Sicherheitsteams der großen Linux-Distributionen. Bekannte Sicherheitsaktualisierungen werden selten (wenn nicht sogar nie) von den Anbietern der Distribution nicht eingespielt. Das Wissen um eine Sicherheitslücke wird niemals vor anderen Anbietern von Distributionen geheim gehalten, da die Ausbesserungen gewöhnlich vom Programmautor oder von http://www.cert.org koordiniert werden. Das hat zur Folge, dass notwendige Sicherheitsaktualisierungen üblicherweise zur selben Zeit veröffentlicht werden. Damit ist die relative Sicherheit der verschiedenen Distributionen ziemlich ähnlich.

Einer großen Vorteile von Debian in Hinblick auf die Sicherheit ist die Leichtigkeit von Systemaktualisierungen mit **apt**. Hier sind ein paar andere Aspekte über die Sicherheit in Debian, die Sie berücksichtigen sollten:

[1] Zum Beispiel könnte es auf Grundlage einiger Daten scheinen, dass Windows NT sicherer ist als Linux. Dies wäre eine fragwürdige Annahme. Das liegt daran, dass Linux-Distributionen normalerweise viel mehr Anwendungen zur Verfügung stellen als Microsofts Windows NT. Dieses Problem des *Abzählens von Sicherheitslücken* wird besser in http://www.dwheeler.com/oss_fs_why.html#security von David A. Wheeler beschrieben.

- Debian bietet mehr Sicherheitswerkzeuge an als andere Distributionen. Vergleichen Sie dazu Kapitel 8, *Sicherheitswerkzeuge in Debian*.

- Debians Standardinstallation ist kleiner (weniger Funktionen) und daher sicherer. Andere Distributionen tendieren im Namen der Benutzerfreundlichkeit dazu, standardmäßig viele Dienst zu installieren, und manchmal sind diese nicht ordentlich konfiguriert (denken Sie an http://www.sophos.com/virusinfo/analyses/linuxlion.html oder http://www.sophos.com/virusinfo/analyses/linuxramen.html). Debians Installation ist nicht so streng wie OpenBSD (dort laufen Daemonen standardmäßig nicht), aber es ist ein guter Kompromiss. [2]

- Debian stellt die besten Verfahren zur Sicherheit in Dokumenten wie diesem vor.

In Bugtraq gibt es viele Debian-Fehler. Heißt das, dass es sehr gefährdet ist?

Die Debian-Distribution enthält eine große und wachsende Zahl von Softwarepaketen, wahrscheinlich sogar mehr als mit vielen proprietären Betriebssystemen geliefert wird. Je mehr Pakete installiert sind, desto größer ist die Möglichkeit von Sicherheitslücken in einem System.

Immer mehr Menschen untersuchen den Quellcode, um Fehler zu entdecken. Es gibt viele Anweisungen im Zusammenhang mit Audits des Quellcodes von großen Softwarekomponenten, die in Debian enthalten sind. Immer wenn ein solcher Audit Sicherheitslücken aufdeckt, werden sie ausgebessert und eine Ankündigung wird an Listen wie Bugtraq geschickt.

Fehler, die in der Debian-Distribution vorhanden sind, betreffen normalerweise auch andere Anbieter und Distributionen. Prüfen Sie einfach den »Debian specific: yes/no«-Abschnitt am Anfang jeder Ankündigung (DSA).

Hat Debian irgendein Zertifikat für Sicherheit?

Die kurze Antwort: Nein.

Die lange Antwort: Zertifikate kosten Geld (besonders ein *seriöses* Sicherheitszertifikat). Niemand hat die Ressourcen aufgebracht, um Debian GNU/Linux beispielsweise mit irgendeinem Level des http://niap.nist.gov/cc-scheme/st/ zertifizieren zu lassen. Wenn Sie daran interessiert sind, eine GNU/Linux-Distribution mit Sicherheitszertifikaten zu haben, stellen Sie uns die Ressourcen zur Verfügung, um dies möglich zu machen.

Es gibt im Moment mindestens zwei Linux-Distributionen, die mit verschiedenen http://en.wikipedia.org/wiki/Evaluation_Assurance_Level-Levels zertifiziert sind. Beachten Sie, dass einige CC-Tests im http://ltp.sourceforge.net vorhanden sind, welche in Debian durch ltp angeboten wird.

Gibt es irgendein Abhärtungsprogramm für Debian?

Ja. http://bastille-linux.sourceforge.net, das sich ursprünglich an anderen Linux-Distributionen (Red Hat und Mandrake) orientierte, es funktioniert derzeit auch mit Debian. Es sind Maßnahmen eingeleitet, um Änderungen am Originalprogramm auch in das Debian-Paket bastille einfließen zu lassen.

Manche Leute glauben jedoch, dass ein Absicherungsprogramm nicht die Notwendigkeit einer guten Administration ersetzen kann.

[2] Ohne die Tatsache in Abrede zu stellen, dass einige Distributionen wie Red Hat oder Mandrake auch die Sicherheit bei ihrer Standardinstallation berücksichtigen, indem der Benutzer *Sicherheitsprofile* auswählen kann, oder Wizards verwendet werden, um beim Einrichten einer *Personal Firewall* zu helfen.

Ich möchte den Dienst XYZ laufen lassen. Welchen sollte ich benutzen?

Einer der größten Stärken von Debian ist die große Vielfalt von Paketen, welche die gleichen Funktionen erfüllen (DNS-Server, Mail-Server, FTP-Server, Web-Server etc.). Das kann einen unerfahrenen Administrator verwirren, wenn er herausfinden will, welches Paket das richtige für ihn ist. Die beste Wahl hängt in der Balance zwischen Ihrem Bedürfnis nach Funktionalität und dem nach Sicherheit in der jeweiligen Situation ab. Im Folgenden einige Fragen, die Sie sich stellen sollten, wenn Sie zwischen ähnlichen Paketen entscheiden müssen:

- Wird die Software noch von ihrem Autor gepflegt? Wann war die letzte Veröffentlichung?
- Ist das Paket ausgereift? Die Versionsnummer sagt *nichts* darüber aus, wie ausgereift es ist. Versuchen Sie seine Geschichte nachzuvollziehen.
- Ist die Software von Fehlern durchsetzt? Gab es Sicherheits-Ankündigungen im Zusammenhang mit ihr?
- Stellt die Software die ganze Funktionalität zur Verfügung, die Sie benötigen? Bietet es mehr, als Sie wirklich brauchen?

Wie mache ich den Dienst XYZ unter Debian sicherer?

Sie werden in diesem Dokument Informationen über das Absichern von einigen Diensten (FTP, Bind) unter Debian GNU/Linux finden. Für Dienste, die hier nicht abgedeckt werden, prüfen Sie die Programm-Dokumentation oder allgemeine Linux-Informationen. Die meisten Sicherheitshinweise für Unix-Systeme sind auch auf Debian anwendbar. So wird Dienst X unter Debian in den meisten Fällen wie in einer anderen Linux-Distribution (oder Un*x, was das betrifft) abgesichert.

Wie kann ich die Banner von Diensten entfernen?

Wenn Sie z.B. nicht wollen, dass Benutzer sich mit Ihrem POP3-Daemon verbinden und dadurch Informationen über Ihr System erlangen, sollten Sie das Banner, das der Dienst den Benutzern zeigt, entfernen (oder verändern). [3] Wie Sie das anstellen können, hängt von der Software ab, mit der Sie einen bestimmten Dienst betreiben. Für **postfix** stellen Sie beispielsweise das SMTP-Banner in `/etc/postfix/main.cf` ein:

```
smtpd_banner = $myhostname ESMTP $mail_name (Debian/GNU)
```

Andere Software kann nicht so leicht verändert werden. ssh muss neu kompiliert werden, um die angezeigte Version zu ändern. Stellen Sie sicher, dass Sie nicht den ersten Teil des Banners (`SSH-2.0`) entfernen, da Clients ihn verwenden, um die von Ihrem Paket unterstützten Protokolle zu identifizieren.

Sind alle Debian-Pakete sicher?

Das Sicherheitsteam von Debian kann nicht alle Pakete aus Debian auf potenzielle Sicherheitslücken hin analysieren, da es einfach nicht genug Ressourcen gibt, um für das gesamte Projekt ein Quellcodeaudit durchzuführen. Allerdings profitiert Debian von den Quellcode-Prüfungen durch die Originalautoren.

Tatsächlich könnte ein Debian-Entwickler in einem Paket einen Trojaner verbreiten und es gibt keine Möglichkeit das nachzuprüfen. Sogar wenn es in einen Zweig von Debian eingeführt werden würde, wäre

[3] Beachten Sie, dass das »security by obscurity« ist und daher auf lange Sicht gesehen wahrscheinlich nicht der Mühe wert ist.

es unmöglich, alle möglichen Situationen abzudecken, in denen der Trojaner ausgeführt werden würde. Das ist der Grund, warum Debian eine *»Keine Gewährleistung«*-Klausel in seiner Lizenz hat.

Allerdings können Debian-Benutzer insofern Vertrauen fassen, als dass der stabile Quellcode eine breite Prüfung hinter sich hat. Die meisten Probleme würden dabei durch Benutzung entdeckt. Es ist nicht zu empfehlen, ungetestete Software auf kritischen Systemen zu installieren, wenn Sie nicht die notwendige Code-Prüfung vornehmen können. In jedem Fall gewährleistet der Aufnahmeprozess in die Distribution (mit digitalen Signaturen), dass im Falle von in die Distribution eingeschleusten Sicherheitsproblemen das Problem letztendlich zum Entwickler zurückgeführt werden kann. Das Debian-Projekt hat diese Angelegenheiten nie auf die leichte Schulter genommen.

Warum sind einige Protokoll- und Konfigurationsdateien für alle lesbar? Ist das nicht unsicher?

Natürlich können Sie die Standardrecht von Debian auf Ihrem System abändern. Der aktuelle Richtlinie in Bezug auf Protokoll- und Konfigurationsdateien besagt, dass sie für alle lesbar sein sollen, *es sei denn*, sie enthalten sensible Informationen.

Seien Sie vorsichtig, wenn Sie Änderungen vornehmen:

- Prozesse könnten nicht mehr in der Lage sein, in Protokolldateien zu schreiben, wenn Sie ihre Rechte einschränken.
- Einige Anwendungen könnten nicht mehr funktionieren, wenn sie ihre Konfigurationsdatei nicht mehr lesen können. Wenn Sie zum Beispiel das Recht, für alle lesbar zu sein, von `/etc/samba/smb.conf` entfernen, kann das Programm **smbclient** nicht funktionieren, wenn es von einem normalen Benutzer ausgeführt wird.

FIXME: Check if this is written in the Policy. Some packages (i.e. ftp daemons) seem to enforce different permissions.

Warum hat /root/ (oder BenutzerX) die Rechte 755?

Tatsächlich kann die gleiche Frage auch für jeden anderen Benutzer gestellt werden. Da Debians Standardinstallation *keine* Dateien unter diesem Verzeichnis abgelegt, sind keine sensiblen Informationen vorhanden, die geschützt werden müssten. Wenn Sie denken, dass diese Rechte für Ihr System zu locker sind, können Sie sie auf 750 einschränken. Für Benutzer sollten Sie „Begrenzung des Zugangs zu Informationen anderer Benutzer“ lesen.

This Debian security mailing list http://lists.debian.org/debian-devel/2000/11/msg00783.html has more on this issue.

Nach der Installation von grsec oder einer Firewall bekomme ich viele Nachrichten auf der Konsole. Wie entferne ich sie?

Wenn Sie Nachrichten auf der Konsole empfangen, aber `/etc/syslog.conf` so eingerichtet haben, dass diese in Dateien oder auf ein spezielles TTY umgeleitet werden, sehen Sie dennoch Nachrichten auf der Konsole, die direkt an sie geschickt werden.

Der Standardloglevel der Konsole ist bei jeden Kernel 7, was bedeutet, dass alle Nachrichten mit einer niedrigeren Priorität auf der Konsole erscheinen werden. Für gewöhnlich haben Firewalls (die LOG-Regel) und einige andere Sicherheitswerkzeuge eine niedrigere Log-Priorität. Daher werden ihre Protokolle direkt an die Konsole geschickt.

Um die Nachrichten, die an die Konsole geschickt werden, zu verringern, können Sie **dmesg** (Option `-n`, vergleichen Sie dmseg(8)) verwenden, das den Ringspeicher des Kernel untersucht und *steuert*. Damit das nach dem nächsten Neustart geändert ist, ändern Sie in `/etc/init.d/klogd` von

```
KLOGD=""
```

in Folgendes ändern:

```
KLOGD="-c 4"
```

Verwenden Sie eine niedrigere Nummer für `-c`, wenn Sie immer noch unerwünschte Nachrichten sehen. Eine Beschreibung der verschiedenen Loglevels befindet sich in `/usr/include/sys/syslog.h`:

```
#define LOG_EMERG       0       /* system is unusable */
#define LOG_ALERT       1       /* action must be taken immediately */
#define LOG_CRIT        2       /* critical conditions */
#define LOG_ERR         3       /* error conditions */
#define LOG_WARNING     4       /* warning conditions */
#define LOG_NOTICE      5       /* normal but significant condition */
#define LOG_INFO        6       /* informational */
#define LOG_DEBUG       7       /* debug-level messages */
```

Benutzer und Gruppen des Betriebssystems

Sind alle Systembenutzer notwendig?

Ja und nein. Debian wird mit einigen vordefinierten Benutzern (mit einer User-ID (UID) < 99 wie in der http://www.de.debian.org/doc/debian-policy/ oder in `/usr/share/doc/base-passwd/README-ME` beschrieben) ausgeliefert. Dadurch wird die Installation einiger Dienste erleichtert, für die es notwendig ist, unter einem passenden Benutzer/UID zu laufen. Wenn Sie nicht vorhaben, neue Dienste zu installieren, können Sie die Benutzer entfernen, denen keine Dateien auf Ihrem System gehören und die keine Dienste laufen lassen. Unabhängig davon ist das Standardverhalten in Debian, dass UIDs von 0 bis 99 reserviert sind und UIDs von 100 bis 999 von Paketen bei der Installation erstellt werden und gelöscht werden, wenn das Pakete vollständig gelöscht wird (purge) wird.

Benutzer, denen keine Dateien gehören, finden Sie leicht mit dem folgenden Kommando[4] (führen Sie es als Root aus, da ein normaler Benutzer nicht genügend Zugriffsrechte haben könnte, um einige sensible Verzeichnisse zu durchsuchen):

```
cut -f 1 -d : /etc/passwd | \
while read i; do find / -user "$i" | grep -q . || echo "$i"; done
```

Diese Benutzer werden von dem Paket base-passwd angelegt. Sie finden Informationen über die Behandlung dieser Benutzer unter Debian in der Dokumentation des Pakets. Es folgt nun eine Liste der Standardbenutzer (mit einer entsprechenden Gruppe):

- root: Root ist (typischerweise) der Superuser.
- daemon: Einige unprivilegierte Daemonen, die Dateien auf die Festplatte schreiben müssen, laufen als daemon.daemon (z.B. **portmap**, **atd**, wahrscheinlich noch andere). Daemonen, die keine eigenen Datei-

[4] Bedenken Sie, dass damit Ihr gesamtes System durchsucht wird. Falls Sie viele Festplatten und Partitionen haben, sollten Sie u.U. den Suchrahmen einschränken.

en besitzen müssen, können stattdessen als nobody.nogroup laufen. Komplexere oder sicherheitsbewusste Daemonen laufen als eigenständige Benutzer. Der Benutzer daemon ist auch praktisch für lokal installierte Daemons.

- bin: aus historischen Gründen beibehalten
- sys: das gleiche wie bei bin. Jedoch gehören /dev/vcs* und `/var/spool/cups` der Gruppe sys.
- sync: Die Shell des Benutzers sync ist `/bin/sync`. Wenn das Passwort auf etwas leicht zu ratendes gesetzt wurde (zum Beispiel »«), kann jeder das System von der Konsole aus synchronisieren lassen, auch wenn er kein Konto hat.
- games: Viele Spiele sind SETGID »games«, damit sie ihre Highscore-Dateien schreiben können. Dies wird in der Richtlinie erklärt.
- man: Das Programm man läuft (manchmal) als Benutzer »man«, damit es Cat-Seiten nach `/var/cache/man` schreiben kann.
- lp: wird von Druck-Daemonen benutzt
- mail: Mailboxen unter `/var/mail` gehören der Gruppe »mail«, wie in der Richtlinie erklärt wird. Der Benutzer und die Gruppe werden auch von verschiedene MTAs zu anderen Zwecken benutzt.
- news: Verschiedene News-Server und ähnliche Programme (zum Beispiel **suck**) benutzen den Benutzer und die Gruppe news auf unterschiedliche Weise. Dateien im news-Spool gehören häufig dem Benutzer und der Gruppe news. Programme wie **inews**, die man benutzen kann, um News zu posten, sind normalerweise SETGID news.
- uucp: Der Benutzer uucp und die Gruppe uucp werden vom UUCP-Subsystem benutzt. Ihnen gehören Spool- und Konfigurationsdateien. Nutzer in der Gruppe uucp können uucico aufrufen.
- proxy: Wie Daemon wird dieser Benutzer und diese Gruppe von manchen Daemonen (insbesondere Proxy-Daemonen) verwendet, die keine spezielle User-ID haben, aber eigene Dateien besitzen müssen. Zum Beispiel wird die Gruppe proxy von **pdnsd** benutzt, und **squid** läuft als Benutzer proxy.
- majordom: **Majordomo** hat auf Debian-Systemen aus historischen Gründen eine statisch zugewiesene UID. Auf neuen Systemen wird sie nicht installiert.
- postgres: **Postgresql**-Datenbanken gehören diesem Benutzer und dieser Gruppe. Alle Dateien in `/var/lib/postgresql` gehören diesem Benutzer, um anständige Sicherheit zu gewährleisten.
- www-data: Einige Web-Server laufen als www-data. Web-Inhalte sollten *nicht* diesem Benutzer gehören, andernfalls wäre ein kompromittierter Web-Server in der Lage, eine Web-Seite zu überschreiben. Daten, die der Web-Server schreibt, einschließlich Protokolldateien, gehören www-data.
- backup: So können Backup-/Wiederherstellungszuständigkeiten lokal an irgendjemanden ohne volle Root-Zugriff delegiert werden.
- operator: operator ist historisch (und praktisch) das einzige »Benutzer«-Konto, in das man sich entfernt einloggen kann, und das nicht von NIS/NFS abhängt.
- list: Mailinglisten-Archive und Daten gehören diesem Benutzer und dieser Gruppe. Manche Mailinglisten-Programme laufen auch unter diesem Benutzer.
- irc: Wird von irc-Daemonen benutzt. Ein statisch zugewiesener Benutzer wird nur wegen eines Fehlers in **ircd** benötigt, das beim Start SETUID() auf sich selbst für eine bestimmte UID ausführt.

- gnats
- nobody, nogroup: Daemonen die keine eigenen Dateien haben laufen als Benutzer nobody und Gruppe nogroup. Demzufolge sollten keine Dateien auf dem gesamten System diesem Benutzer oder dieser Gruppe gehören.

Andere Gruppe, die keinen dazugehörigen Benutzer haben:

- adm: Die Gruppe adm wird zu Zwecken der Überwachung benutzt. Mitglieder dieser Gruppe können viele Dateien in `/var/log` lesen und die xconsole benutzen. `/var/log` war früher einmal `/usr/adm` (und später `/var/adm`), daher der Name dieser Gruppe.
- tty: TTY-Geräte gehören dieser Gruppe. Die Befehle write und wall benutzen dies, um auf die TTYs anderer Leute zu schreiben.
- disk: Roh-Zugriff auf Festplatten. Größtenteils äquivalent zum Root-Zugriff.
- kmem: /dev/kmem und ähnliche Dateien sind von dieser Gruppe lesbar. Dies ist größtenteils ein Relikt aus BSD. Aber jedes Programm, das Lese-Zugriff auf den Systemspeicher braucht, kann so SETGID kmem gemacht werden.
- dialout: Voller und direkter Zugriff auf serielle Schnittstellen. Mitglieder dieser Gruppen können Modems rekonfigurieren, sich irgendwo einwählen, usw.
- dip: Der Name der Gruppe steht für »Dial-up IP«. Mitgliedern der Gruppe dip können Programme wie **ppp**, **dip**, **wvdial** usw. benutzen, um eine Verbindung herzustellen. Die Benutzer in dieser Gruppe können das Modem nicht konfigurieren. Sie können lediglich Programme aufrufen, die es benutzen.
- fax: erlaubt es den Mitgliedern, Fax-Software zu benutzen, um Faxe zu senden und zu empfangen.
- voice: Voicemail, nützlich für Systeme, die Modems als Anrufbeantworter benutzen
- cdrom: Diese Gruppe kann dazu benutzt werden, einer bestimmen Gruppe von Benutzern Zugriff auf CD-ROM-Laufwerke zu geben.
- floppy: Diese Gruppe kann dazu benutzt werden, einer bestimmen Gruppe von Benutzern Zugriff auf Diskettenlaufwerke zu geben.
- tape: Diese Gruppe kann dazu benutzt werden, einer bestimmen Gruppe von Benutzern Zugriff auf Bandlaufwerke zu geben.
- sudo: Mitglieder dieser Gruppe müssen ihr Passwort nicht eingeben, wenn sie **sudo** benutzen. Siehe `/usr/share/doc/sudo/OPTIONS`.
- audio: Diese Gruppe kann dazu benutzt werden, einer bestimmen Gruppe von Benutzern Zugriff auf jedes Audiogerät zu geben.
- src: Dieser Gruppe gehören die Quellcodes, einschließlich der Dateien in `/usr/src`. Sie kann benutzt werden, um einem bestimmten Benutzern die Möglichkeit zu bieten, Quellcode des Systems zu verwalten.
- shadow: `/etc/shadow` ist von dieser Gruppe lesbar. Einige Programme, die auf diese Datei zugreifen müssen, sind SETGID shadow.
- utmp: Diese Gruppe kann nach `/var/run/utmp` und ähnlichen Dateien schreiben. Programme, die darin schreiben können müssen, sind SETGID utmp.

- video: Diese Gruppe kann dazu benutzt werden, einer bestimmen Gruppe von Benutzern Zugriff auf ein Videogerät zu geben.
- staff: Erlaubt Benutzern lokale Modifikationen am System vorzunehmen (`/usr/local`, `/home`), ohne dass sie Root-Privilegien bräuchten. Vergleichen Sie sie mit »adm«, die sich mehr auf Überwachung/Sicherheit bezieht.
- users: Während Debian-Systeme standardmäßig das System einer privaten Benutzergruppe (jeder Benutzer hat seine eigene Gruppe) verwenden, ziehen es manche vor, ein traditionelleres Gruppen-System zu verwenden. In diesem System ist jeder Benutzer Mitglied dieser Gruppe.

Ich entfernte einen Systembenutzer! Wie kann ich dies rückgängig machen?

Wenn Sie einen Systembenutzer entfernt und kein Backup Ihrer `password`- und `group`-Dateien haben, können Sie versuchen, diesen mittels **update-passwd** (vergleichen Sie update-passwd(8)) wiederherzustellen.

Was ist der Unterschied zwischen den Gruppen adm und staff?

Die Gruppe »adm« besteht üblicherweise aus Administratoren. Die Rechte dieser Gruppe erlauben es ihnen, Protokolldateien zu lesen, ohne **su** benutzen zu müssen. Die Gruppe »staff« ist gewöhnlich für Kundendienst- und Junioradministratoren bestimmt und gibt ihnen die Möglichkeit, Dinge in `/usr/local` zu erledigen und Verzeichnisse in `/home` anzulegen.

Warum gibt es eine neue Gruppe, wenn ich einen neuen Benutzer anlege? (Oder warum gibt Debian jedem Benutzer eine eigene Gruppe?)

Das Standardverhalten von Debian ist, dass jeder Benutzer seine eigene, persönliche Gruppe hat. Das traditionelle UN*X-Modell weist alle Benutzer der Gruppe *users* zu. Zusätzliche Gruppe werden erstellt, um den Zugang zu gemeinsam genutzten Dateien, die mit verschiedenen Projektverzeichnissen verbunden sind, einzuschränken. Die Dateiverwaltung wurde schwierig, wenn ein einzelner Benutzer an verschiedenen Projekten arbeitete, da, wenn jemand eine Datei erstellte, diese mit der primären Gruppe des Erstellers (z.B. »users«) verbunden war.

Das Modell von Debian löst dieses Problem, indem es jedem Benutzer seine eigene Gruppe zuweist. So wird mit einer korrekten Umask (0002) und mit dem SETGID-Bit für ein Projektverzeichnis den Dateien, die in diesem Verzeichnis erstellt werden, automatisch die richtige Gruppe zugewiesen. Das erleichtert die Arbeit von Menschen, die an verschiedenen Projekten arbeiten, da sie nicht die Gruppe oder Umasks ändern müssen, wenn sie mit gemeinsam genutzten Dateien arbeiten.

Sie können allerdings dieses Verhalten verändern, indem Sie `/etc/adduser.conf` modifizieren. Ändern Sie die Variable *USERGROUPS* auf »no« ab. Dadurch wird keine neue Gruppe erstellt, wenn ein neuer Benutzer angelegt wird. Sie sollten auch *USERS_GID* die GID der Gruppe zuweisen, der alle Benutzer angehören.

Fragen über Dienste und offene Ports

Warum werden alle Dienste während der Installation aktiviert?

Das ist der Annäherung an das Problem, auf der einen Seite sicherheitsbewusst und auf der anderen Seite benutzerfreundlich zu sein. Anders als OpenBSD, das alle Dienste abschaltet, bis sie vom Administrator aktiviert werden, aktiviert Debian GNU/Linux alle installierten Dienste, bis sie abgeschaltet werden (siehe dazu „Daemons abschalten“). Immerhin haben Sie den Dienst installiert, oder?

Es gab viele Diskussionen auf Debian-Mailinglisten (sowohl auf debian-devel als auch auf debian-security) darüber, welches die bessere Vorgehensweise für eine Standardinstallation ist. Jedoch gab es bisher (10. März 2002) keinen Konsens.

Kann ich inetd entfernen?

Inetd ist nicht leicht zu entfernen, da netbase von dem Paket abhängt, das es enthält (netkit-inetd). Wenn Sie es entfernen wollen, können Sie es entweder abschalten (siehe „Daemons abschalten“) oder das Paket entfernen, indem Sie das Paket equivs benutzen.

Warum ist bei mir Port 111 offen?

Port 111 ist sunrpcs Portmapper und wird standardmäßig bei der Grundinstallationen eines Debian-Systems eingerichtet, da es keine Möglichkeit gibt herauszubekommen, wann ein Programm eines Benutzers RPC gebrauchen könnte, um korrekt zu arbeiten. Jedenfalls wird es meistens von NFS benutzt. Wenn Sie kein NFS benutzen, entfernen Sie es, wie in „Absichern von RPC-Diensten“ erklärt.

In Versionen des Pakets portmap später als 5-5 können Sie sogar den Portmapper installieren, aber ihn nur auf dem Localhost lauschen lassen (dazu müssen Sie `/etc/default/portmap` verändern).

Wozu ist der identd (Port 113) da?

Der Dienst Identd ist ein Authentisierungsdienst, der den Besitzer einer bestimmten TCP/IP-Verbindung zu einem entfernten Server, der die Verbindung annimmt, identifiziert. Wenn ein Benutzer sich mit einem entfernten Host verbindet, schickt **inetd** auf dem entfernten Host üblicherweise eine Anfrage an Port 113 zurück, um Informationen über den Besitzer herauszufinden. Er wird häufig von Mail-, FTP- und IRC-Servern eingesetzt. Er kann auch dazu verwendet werden, um einen Benutzer Ihres lokalen Systems, der ein entferntes System angreift, aufzuspüren.

There has been extensive discussion on the security of **identd** (See http://lists.debian.org/debian-security/2001/08/msg00297.html). In general, **identd** is more helpful on a multi-user system than on a single user workstation. If you don't have a use for it, disable it, so that you are not leaving a service open to the outside world. If you decide to firewall the identd port, *please* use a reject policy and not a deny policy, otherwise a connection to a server utilizing **identd** will hang until a timeout expires (see http://logi.cc/linux/reject_or_deny.php3).

Ich habe Dienste, welche die Ports 1 und 6 verwenden. Welche sind das und wie kann ich sie entfernen?

Sie führen den Befehl `netstat -an` aus und erhalten Folgendes:

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
raw        0      0 0.0.0.0:1               0.0.0.0:*               7
-
raw        0      0 0.0.0.0:6               0.0.0.0:*               7
-
```

Sie sehen *nicht* Prozesse, die auf dem TCP/UDP-Port 1 und 6 lauschen. Tatsächlich sehen Sie einen Prozess, der auf einem *Raw*-Socket für Protokoll 1 (ICMP) und 6 (TCP) lauscht. Ein solches Verhalten ist für Trojaner und einige Systeme zur Eindringlingserkennung wie iplogger und portsentry üblich. Wenn Sie diese Pakete besitzen, löschen Sie sie einfach. Falls nicht, versuchen Sie mit netcats Option `-p` (Prozess) herauszufinden, welcher Prozess diese Lauscher betreibt.

Ich habe festgestellt, dass Port XYZ offen ist. Kann ich ihn schließen?

Ja, natürlich. Die Ports, die Sie offen lassen, hängen von Ihrer individuellen Richtlinie bezüglich öffentlich zugänglicher Dienste ab. Prüfen Sie, ob sie von **inetd** (siehe „Abschalten von **Inetd** oder seinen Diensten“) oder von anderen installierten Paketen geöffnet werden, und leiten Sie passende Maßnahmen ein (d.h. konfigurieren Sie inetd, entfernen Sie das Paket, verhindern Sie, dass der Dienst beim Booten gestartet wird).

Hilft das Löschen von Diensten aus `/etc/services`, um meinen Rechner abzusichern?

Nein, `/etc/services` stellt nur eine Verbindung zwischen virtuellem Namen und Portnummer her. Das Entfernen von Namen aus dieser Datei verhindert (üblicherweise) nicht, dass ein Dienst gestartet wird. Manche Daemonen starten vielleicht nicht, wenn `/etc/services` verändert wurde, aber das ist nicht die Norm. Um einen Dienst richtig abzuschalten, sehen Sie sich „Daemons abschalten“ an.

Allgemeine Sicherheitsprobleme

Ich habe mein Passwort vergessen und kann auf das System nicht mehr zugreifen!

Die nötigen Schritte, um wieder Zugriff erhalten, hängen davon ab, ob Sie die vorgeschlagene Prozedur zum Absichern von **lilo** und BIOS durchgeführt haben oder nicht.

Wenn Sie beides eingeschränkt haben, müssen Sie im BIOS erlauben, von anderen Medien als der Festplatte zu booten, bevor Sie weitermachen können. Wenn Sie auch Ihr BIOS-Passwort vergessen haben, müssen Sie Ihr BIOS zurücksetzen. Dazu öffnen Sie das PC-Gehäuse und entfernen die BIOS-Batterie.

Sobald Sie das Booten von CD-ROM oder Diskette eingeschaltet haben, sollten Sie Folgendes ausprobieren:

- Booten Sie von eine Rettungsdiskette und starten den Kernel.
- Wechseln Sie mit Alt+F2 auf eine virtuelle Konsole.
- Binden Sie die Partition ein, auf der sich Ihr /root befindet.
- Editieren Sie (auf der Rettungsdiskette von Debian 2.2 befindet sich **ae**, Debian 3.0 enthält **nano-tiny**, der **vi** ähnelt) die Datei `/etc/shadow` und ändern Sie die Zeile:

  ```
  root:asdfjl290341274075:XXXX:X:XXXX:X::: (X=irgendeine Ziffer)
  ```

  in Folgendes ändern:

  ```
  root::XXXX:X:XXXX:X:::
  ```

Dies entfernt das vergessene Root-Passwort, das sich im ersten durch Doppelpunkte abgetrennten Feld nach dem Benutzernamen befand. Speichern Sie die Datei ab, starten Sie das System neu und melden Sie sich als Root mit einem leeren Passwort an. Dies wird funktionieren, außer wenn Sie Ihr System etwas sicherer eingestellt haben, d.h. wenn Sie nicht erlauben, dass Benutzer leere Passwörter haben, oder dass Root sich auf einer Konsole einloggen kann.

Falls Sie derartige Maßnahmen getroffen haben, müssen Sie im Single-User-Modus starten. Wenn Sie LILO eingeschränkt haben, müssen **lilo** erneut ausführen, nachdem Sie das Root-Passwort zurückgesetzt

haben. Das ist ziemlich verzwickt, da Ihre `/etc/lilo.conf` verändert werden muss, da das Root-Dateisystem (/) eine RAM-Disk und keine echte Festplatte ist.

Sobald LILO nicht mehr eingeschränkt ist, versuchen Sie Folgendes:

- Drücken Sie Alt, Shift oder Steuerung (Control), kurz bevor das BIOS seine Arbeit beendet hat, und Sie sollten nun einen LILO-Prompt erhalten.
- Geben Sie am Prompt `linux single`, `linux init=/bin/sh` oder `linux 1` ein.
- Sie erhalten einen Shell-Prompt im Single-User-Modus (Sie werden nach dem Passwort gefragt, aber das kennen Sie jetzt ja).
- Binden Sie die Root-Partition (/) im Schreib/Lese-Modus neu ein, indem Sie den Befehl mount verwenden:

```
# mount -o remount,rw /
```

- Ändern Sie das Superuser-Passwort mit **passwd** (da Sie der Superuser sind, werden Sie nicht nach dem alten Passwort gefragt).

Wie muss ich vorgehen, wenn ich meinen Benutzern einen Dienst anbieten möchte, ihnen aber keine Shell-Konten geben will?

Wenn Sie zum Beispiel einen POP-Dienst anbieten wollen, müssen Sie nicht für jeden zugreifenden Benutzer ein Konto anlegen. Am besten setzen Sie hierzu eine Authentifizierung, die auf Verzeichnisses basiert, durch einen externen Dienst (wie Radius, LDAP oder eine SQL-Datenbank) ein. Installieren Sie einfach die gewünschte PAM-Bibliothek (libpam-radius-auth, libpam-ldap, libpam-pgsql oder libpam-mysql), lesen Sie die Dokumentation (Einsteiger sehen bitte unter „Benutzerauthentifizierung: PAM“ nach) und konfigurieren Sie den PAM-nutzenden Dienst, so dass er Ihren Backend benutzt. Bearbeiten Sie dazu die dem Dienst entsprechenden Dateien unter `/etc/pam.d/` und ändern die folgenden Zeile von:

```
auth   required    pam_unix_auth.so shadow nullok use_first_pass
```

beispielsweise für ldap zu:

```
auth   required    pam_ldap.so
```

Im Fall von LDAP-Verzeichnissen liefern manche Dienste LDAP-Schemata mit, die Sie Ihrem Verzeichnis hinzufügen können, um eine LDAP-Authentifizierung zu benutzen. Wenn Sie relationale Datenbanken benutzen, gibt es einen nützlichen Trick: Benutzen Sie die Klausel *where*, wenn Sie die PAM-Module konfigurieren. Wenn Sie beispielsweise eine Datenbank mit der folgenden Tabelle haben:

```
(user_id, user_name, realname, shell, password, UID, GID, homedir, sys, pop, ima
```

Wenn Sie die Attribute der Dienste zu Boolean-Feldern machen, können Sie sie verwenden, um den Zugang zu den verschiedenen Diensten zu erlauben oder zu verbieten. Sie müssen dazu nur die geeigneten Zeilen in folgende Dateien einfügen:

- `/etc/pam.d/imap`:`where=imap=1`.
- `/etc/pam.d/qpopper`:`where=pop=1`.

- `/etc/nss-mysql*.conf:users.where_clause = user.sys = 1;`.
- `/etc/proftpd.conf: SQLWhereClause "ftp=1"`.

Mein System ist angreifbar! (Sind Sie sich sicher?)

Der Scanner X zur Einschätzung der Verwundbarkeit sagt, dass mein Debian-System verwundbar wäre?

Viele Scanner zur Einschätzung der Verwundbarkeit liefern falsche Positivmeldungen, wenn sie auf Debian-Systemen eingesetzt werden. Das liegt daran, dass sie nur die Version eines Softwarepakets überprüfen, um herauszufinden, ob es verwundbar ist. Sie prüfen nicht, ob tatsächlich eine Sicherheitslücke vorhanden ist. Da Debian nicht die Version einer Software ändert, wenn ein Paket repariert wird (häufig werden Ausbesserungen an neueren Veröffentlichungen zurückportiert), neigen einige Werkzeuge dazu zu denken, dass ein aktualisiertes Debian-System verwundbar ist, auch wenn das nicht der Fall ist.

Wenn Sie denken, dass Ihr System auf dem aktuellen Stand der Sicherheitsaktualisierungen ist, sollten Sie die Querverweise zu den Datenbanken mit Sicherheitslücken, in denen die DSAs veröffentlicht sind (vergleichen Sie dazu „Debian-Sicherheits-Ankündigungen“), verwenden, um falsche Positive auszusondern, wenn das Programm, das Sie verwenden, CVE-Referenzen enthält.

Ich habe in meinen Protokolldateien einen Angriff gesehen: Ist mein System kompromittiert?

Ein Hinweis auf einen Angriff heißt nicht notwendigerweise, dass Ihr System gehackt wurde. Leiten Sie die üblichen Schritte ein, um festzustellen, ob das System kompromittiert wurde (siehe Kapitel 11, *Nach einer Kompromittierung (Reaktion auf einem Vorfall)*). Selbst wenn Ihr System hinsichtlich des protokollierten Angriffs nicht verwundbar ist, könnte ein entschlossener Angreifer neben der von Ihnen entdeckten Sicherheitslücke auch eine andere ausgenutzt haben.

Ich habe in meinen Protokollen merkwürdige »MARK«-Einträge gefunden. Wurde ich gehackt?

Sie können die folgenden Zeilen in Ihren Systemprotokollen finden:

```
Dec 30 07:33:36 debian -- MARK --
Dec 30 07:53:36 debian -- MARK --
Dec 30 08:13:36 debian -- MARK --
```

Dies stellt keinen Hinweis auf eine Kompromittierung dar, obwohl Benutzer, die von einer Debian-Release wechseln, es vielleicht merkwürdig finden. Wenn Ihr System keine große Last (oder nicht viele aktive Dienste) hat, können diese Zeilen in alle Protokollen auftauchen. Dies ist ein Hinweis, dass Ihr **syslogd**-Daemon richtig läuft. Aus syslogd(8):

```
-m interval
          Der Syslogd protokolliert regelmäßig einen
          Zeitstempel. Der voreingestellte Abstand zwischen zwei
          – MARK – Zeilen ist 20 Minuten. Er kann mit dieser Option
          geändert werden. Setzen Sie den Abstand auf Null, um
          die Zeitstempel komplett abzuschalten.
```

Ich habe Benutzer gefunden, die laut meinen Protokolldateien »su« benutzen: Bin ich kompromittiert?

Sie könnten in Ihren Protokolldateien Zeilen wie die folgenden finden:

```
Apr  1 09:25:01 server su[30315]: + ??? root-nobody
Apr  1 09:25:01 server PAM_unix[30315]: (su) session opened for user nobody by (
```

Seien Sie nicht zu besorgt. Prüfen Sie, ob dies durch einen **Cron**-Job hervorgerufen wird (normalerweise `/etc/cron.daily/find` oder **logrotate**):

```
$ grep 25 /etc/crontab
25 9    * * *   root    test -e /usr/sbin/anacron || run-parts --report
/etc/cron.daily
$ grep nobody /etc/cron.daily/*
find:cd / && updatedb --localuser=nobody 2>/dev/null
```

Ich habe »possible SYN flooding« in meinen Protokollen entdeckt: Werde ich angegriffen?

Sie sehen Einträge wie diese in Ihren Protokollen:

```
May 1 12:35:25 linux kernel: possible SYN flooding on port X. Sending cookies.
May 1 12:36:25 linux kernel: possible SYN flooding on port X. Sending cookies.
May 1 12:37:25 linux kernel: possible SYN flooding on port X. Sending cookies.
May 1 13:43:11 linux kernel: possible SYN flooding on port X. Sending cookies.
```

Überprüfen Sie mit **netstat**, ob es eine große Anzahl von Verbindungen zum Server gibt. Zum Beispiel:

```
linux:~# netstat -ant | grep SYN_RECV | wc -l
   9000
```

Dies ist ein Anzeichen, dass ein Denial-of-Service-Angriff (DoS) auf den Port X Ihres Systems (am wahrscheinlichsten gegen einen öffentlichen Dienst wie Ihren Web- oder Mailserver). Sie sollten TCP-Syncookies in Ihrem Kernel einschalten, siehe „Konfiguration von Syncookies“. Beachten Sie, dass ein DoS-Angriff Ihr Netzwerk überfluten kann, auch wenn Sie verhindern können, dass er Ihr System zum Absturz bringt. [5] Der einzige effektive Weg, diesen Angriff abzuwehren, ist, mit Ihrem Netzprovider in Verbindung zu treten.

Ich habe seltsame Root-Sessions in meinen Protokollen entdeckt: Wurde ich gehackt?

Sie sehen folgende Art von Einträgen in der Datei `/var/log/auth.log`:

```
May 2 11:55:02 linux PAM_unix[1477]: (cron) session closed for user root
May 2 11:55:02 linux PAM_unix[1476]: (cron) session closed for user root
May 2 12:00:01 linux PAM_unix[1536]: (cron) session opened for user root by
```

[5] Da keine Datei-Deskriptoren mehr vorhanden sind, könnte das System nicht mehr antworten, bis das Zeitlimit der TCP-Verbindungen überschritten wurde.

```
(UID=0)
May 2 12:00:02 linux PAM_unix[1536]: (cron) session closed for user root
```

Sie kommen von einem ausgeführten **Cron**-Job (in unserem Beispiel alle fünf Minuten). Um herauszufinden, welches Programm für diese Jobs verantwortlich ist, überprüfen Sie die Einträge in `/etc/crontab`, `/etc/cron.d`, `/etc/crond.daily` und Roots `crontab` in `/var/spool/cron/crontabs`.

Ich bin Opfer eines Einbruchs, was soll ich jetzt tun?

Es gibt mehrere Schritte, die Sie bei einem Einbruch durchführen sollten:

- Prüfen Sie, ob Ihr System auf dem aktuellen Stand der Sicherheitsaktualisierungen für veröffentlichte Verwundbarkeiten ist. Wenn Ihr System verwundbar ist, erhöht dies die Möglichkeit, dass Ihr System tatsächlich gehackt wurde. Die Wahrscheinlichkeit steigt weiter an, wenn die Sicherheitslücke schon eine Zeit lang bekannt ist, da üblicherweise mehr Angriffsversuche in Bezug auf ältere Verwundbarkeiten bestehen. Hier ist ein Link zu http://www.sans.org/top20/.
- Lesen Sie dieses Dokument, insbesondere den Abschnitt Kapitel 11, *Nach einer Kompromittierung (Reaktion auf einem Vorfall)*.
- Fragen Sie nach Hilfe. Sie können die Mailingliste debian-security benutzen und um Rat fragen, wie Sie Ihr System wiederherstellen oder patchen.
- Benachrichtigen Sie Ihren lokalen http://www.cert.org (wenn einer existiert, ansonsten sollten Sie sich vielleicht direkt mit CERT in Verbindung setzen). Das könnte Ihnen helfen (vielleicht aber auch nicht), aber wenigstens wird CERT über laufende Angriffe informiert. Diese Informationen sind sehr wertvoll, um herauszufinden, welche Werkzeuge und Angriffsarten von der *Blackhat*-Community verwendet werden.

Wie verfolge ich einen Angriff zurück?

Sie können einen Angriff zu seinem Ursprung zurückverfolgen, indem Sie die Protokolle (wenn sie nicht geändert wurden) mit Hilfe eines Systems zur Eindringlingserkennung (siehe „Aufsetzen einer Eindringlingserkennung“), **traceroute**, **whois** oder ähnlicher Werkzeuge (einschließlich forensischer Analyse) durchsehen. Wie Sie auf diese Informationen reagieren und was *Sie* als Angriff betrachten, hängt ausschließlich von Ihren Sicherheitsrichtlinien ab. Ist ein einfacher Scan ein Angriff? Ist die Prüfung auf eine Verwundbarkeit ein Angriff?

Das Programm X in Debian ist angreifbar – was soll ich tun?

Nehmen Sie sich zuerst einen Augenblick Zeit, um zu schauen, ob die Sicherheitslücke in öffentlichen Sicherheitsmailinglisten (wie Bugtraq) oder anderen Foren bekannt gemacht wurde. Das Sicherheitsteam von Debian ist hinsichtlich dieser Listen auf dem Laufenden, daher sollte ihm dieses Problem bereits bekannt sein. Leiten Sie keine weiteren Maßnahmen ein, wenn Sie schon eine Bekanntmachung auf http://security.debian.org sehen.

Wenn anscheinend keine Informationen veröffentlicht wurden, schicken Sie bitte eine E-Mail zu den betroffenen Paketen mit einer detaillierten Beschreibung der Verwundbarkeit (Code, der dies bestätigt, ist auch in Ordnung) an mailto:team@security.debian.org. Dort erreichen Sie das Sicherheitsteam von Debian.

Laut der Versionsnummer eines Paketes läuft bei mir immer noch eine angreifbare Version!

Statt auf eine neue Veröffentlichung zu aktualisieren, portiert Debian sicherheitsrelevante Korrekturen zu der Version zurück, die in der Stable-Veröffentlichung enthalten ist. Der Grund dafür ist, dass sicher

gegangen werden soll, dass die Stable-Veröffentlichung so wenig wie möglich verändert wird. Damit wird verhindert, dass sich Dinge als Folge einer Sicherheitskorrektur unerwartet ändern oder kaputt gehen. Ob Sie eine sichere Version eines Paketes benutzen, stellen Sie fest, indem Sie das Changelog des Paketes durchsehen oder indem Sie die exakte Versionsnummer (ursprüngliche Version -slash- Debian-Release) mit der Nummer aus der Debian-Sicherheits-Ankündigung (DSA) vergleichen.

Bestimmte Software

Proftpd ist für einen Denial-of-Service-Angriff anfällig.

Fügen Sie Ihrer Konfigurationsdatei `DenyFilter \*.*/`hinzu. Mehr Informationen entnehmen Sie http://www.proftpd.org/bugs.html.

Nach der Installation von portsentry sind viele Ports offen.

Dies ist nur die Art und Weise, wie **portsentry** arbeitet. Es öffnet etwas zwanzig ungenutzte Ports und versucht so, Port-Scans zu entdecken.

Fragen zu Debians Sicherheitsteam

The security team keeps its list of Frequently Asked Questions at the http://www.debian.org/security/faq. Please refer to that web page for up to date information.

Anhang A. Versionsgeschichte

Versionsgeschichte

Version 3-19.2	Sun May 19 2024	HolgerWansing<hwansing@mailbox.org>

Translation files synchronised with XML sources 3-19

Version 3-19.1	Mon May 1 2017	MarcosFouces<marcos.fouces@gmail.com>

Translation files synchronised with XML sources 3-19

Version 3-19	April 2017	MarcosFouces<marcos.fouces@gmail.com>

Migrate to Docbook XML.
Build with Publican. No longer use custom Makefile.
Migrate svn repository to git.
Import chinese, italian, spanish, portuguese, japanese, russian, french and german translations to PO format.

Version 3-18	February 2015	ThisKinkhorst<thijs@debian.org>

Clarify FAQ on raw sockets.
Update section 4.5 on GRUB2.
Replace example postrm user removal code with advice to use deluser/delgroup --system

Version 3-17	January 2015	ThisKinkhorst<thijs@debian.org>

Remove mention of MD5 shadow passwords.
Do not recommend dselect for holding packages.
No longer include the Security Team FAQ verbatim, because it duplicates information documented elsewhere and is hence perpetually out of date.
Update section on restart after library upgrades to mention needrestart.
Avoid gender-specific language. Patch by Myriam.
Use LSB headers for firewall script. Patch by Dominic Walden.

Version 3-16	January 2013	JavierFernández-Sanguino Peña.<jfs@debian.org>

Hinweis, dass das Dokument nicht in Hinblick auf die neusten Versionen aktualisiert ist
Verweise auf aktuelle Quellen aktualisiert
Informationen zu Sicherheitsaktualisierungen für neuere Veröffentlichungen aktualisiert
Verweis von Informationen für Entwickler auf Online-Quellen, anstatt die Informationen im Dokument zu belassen, um Dubletten zu vermeiden
Informationen über die Sicherung des Konsolenzugangs erweitert einschließlich der Beschränkung der Magischen S-Abf-Taste
Informationen zu PAM-Modulen aktualisiert einschließlich, wie man Anmeldungen an der Konsole einschränkt, cracklib verwendet und die in /etc/pam.d/login verfügbaren Eigenschaften einsetzt; veraltete Verweise auf Variablen in /etc/login.defs entfernt
Hinweis auf einige PAM-Module, die eine Zweifaktor-Anmeldung durchführen können, für Administratoren, die vollständig auf Passwörter verzichten wollen
Beispielshellskript im Anhang korrigiert
Fehler bei Verweisen korrigiert
Verweis auf das Bastille-Projekt bei Sourceforge anstelle der Site bastille-unix.org, da diese nicht mehr antwortet

Version 3-15	December 2010	JavierFernández-Sanguino Peña<jfs@debian.org>

Verweis auf die Website von Log Analysis geändert, da nicht länger verfügbar

Version 3-14 March 2009 JavierFernández-Sanguino Peña<jfs@debian.org>

Abschnitt über die Auswahl des Dateisystems geändert: Hinweis, dass ext3 jetzt der Standard ist
Name der Pakete, die mit enigmail zusammenhängen, geändert, damit sie den geänderten Namen in Debian entsprechen

Version 3-13 February 2008 JavierFernández-Sanguino Peña<jfs@debian.org>

URLs, die auf Bastille-Linux verweisen, zu www.Bastille-UNIX.org geändert, da die Domain von einem http://bastille-linux.sourceforge.net/press-release-newname.html
Verweise auf Linux-Ramen- und Lion-Wurm ausgebessert
In den Beispielen linux-image anstelle der (alten) Pakete kernel-image verwendet
Von Francesco Poli gemeldete Tippfehler ausgebessert

Version 3-12 August 2007 JavierFernández-Sanguino Peña<jfs@debian.org>

Informationen über Sicherheitsaktualisierungen erneuert; Text über Tiger entfernt und Informationen zu Update-notifier und Expertenwerkzeuge (für Desktops) sowie zu Debsecan eingefügt; auch einige Verweis auf andere verfügbare Werkzeuge eingefügt
Die Firewall-Anwendungen nach Zielgruppen aufgeteilt und Fireflier zur Liste der Firewall-Anwendungen für den Desktop hinzugefügt
Verweis auf Libsafe entfernt, es ist nicht mehr im Archiv (wurde im Januar 2006 entfernt)
Den Ort der Konfiguration von Syslog berichtigt, vielen Dank an John Talbut

Version 3-11 January 2007 JavierFernández-Sanguino Peña<jfs@debian.org>

Änderung von Javier Fernández-Sanguino Peña. Vielen Dank an Francesco Poli für die umfangreiche Durchsicht dieses Dokuments.
Die meisten Verweise auf Woody entfernt, da es nicht länger im Archiv verfügbar ist und es dafür auch keine Unterstützung der Sicherheit mehr gibt
Beschrieben, wie Benutzer eingeschränkt werden, so dass sie nur Dateiübertragungen durchführen können
Einen Hinweise auf die Entscheidung der Änderung der Vertraulichkeit der Mailingliste debian-private hinzugefügt
Den Verweis auf die Anleitung zum Umgang mit Vorfällen aktualisiert
Einen Hinweis darauf eingefügt, dass Entwicklerwerkzeuge (wie Compiler) nicht mehr standardmäßig in Etch installiert werden
Einen Hinweis darauf eingefügt, dass Entwicklerwerkzeuge (wie Compiler) nicht mehr standardmäßig in Etch installiert werden
Den Verweis auf den Master-Security-Server korrigiert
Einen Hinweis auf die Dokumentation zu APT-secure eingefügt
Die Erläuterung der APT-Signaturen verbessert
Einige Stellen auskommentiert, die sich auf noch nicht fertig gestellte Abschnitte der offiziellen öffentlichen Schlüssel von Spiegelservern bezogen
Den Namen des Debian-Testing-Sicherheitsteams korrigiert
In einem Beispiel den Verweis auf Sarge entfernt
Den Abschnitt über Antivirus aktualisiert: ClamAV ist jetzt in der Veröffentlichung enthalten. Erwähnte auch den Installer für F-prot
Alle Verweise auf Freeswan entfernt, da es veraltet ist
Probleme beschrieben, die beim Verändern der Firewall-Regeln aus der Ferne auftreten können, und gab einige Tipps (in Fußnoten)
Informationen zur IDS-Installation aktualisiert, BASE und das Bedürfnis nach einer Protokollierungsdatenbank erwähnt
Den Abschnitt »Bind nicht als Root laufen lassen« neu geschrieben, da dies nicht mehr auf Bind9 zutrifft.
Entfernte auch Verweise auf das init.d-Skript, da die Änderungen in /etc/default vorgenommen werden müssen.
Eine veraltete Möglichkeit, Regeln für die Firewall einzurichten, entfernt, da Woody nicht länger unterstützt wird

Zu dem früheren Hinweis bezüglich LOG_UNKFAIL_ENAB zurückgekehrt, nämlich dass es auf »no« (wie es standardmäßig ist) gesetzt werden sollte
Informationen hinzugefügt, wie das System mit Werkzeugen für den Desktop (einschließlich Update-notifier) aktualisiert wird, und beschrieben, wie man mit Aptitude das System aktualisiert. Angemerkt, dass dselect veraltet ist
Die FAQ aktualisiert und überflüssige Abschnitte entfernt
Den Abschnitt über die forensische Analyse von Schadsoftware überarbeitet und aktualisiert
Einige tote Verweise entfernt oder korrigiert
Viele Tipp- und Grammatikfehler verbessert, die von Francesco Poli mitgeteilt wurden

Version 3-10 November 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Beispiele gegeben, wie rdepends von Apt-cache verwendet wird. Wurde von Ozer Sarilar vorgeschlagen
Den Verweis auf das Benutzerhandbuch von Squid auf Grund seines Umzugs korrigiert. Wurde von Oskar Pearson (dem Betreuer) mitgeteilt
Informationen über umask korrigiert, es kann in logins.defs (nicht limits.conf) für alle Anmelde-Verbindung konfiguriert werden. Auch dargestellt, was Debians Vorgabe ist und was restriktivere Werte für sowohl »user« als auch »root« wären. Vielen Dank an Reinhard Tartler für das Auffinden des Fehlers

Version 3-9 October 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Informationen hinzugefügt, wie man Sicherheitslücken verfolgt. Hinweis auf den Debian-Testing-Sicherheits-Tracker hinzugefügt
Weitere Informationen über die Sicherheitsunterstützung für Testing hinzugefügt
Eine große Anzahl von Tippfehlern mit einem Patch von Simon Brandmair korrigiert
Einen Abschnitt hinzugefügt, wie der Root-Prompt bei Initramfs abgestellt wird. Wurde von Max Attems beigesteuert
Verweise auf Queso entfernt
Hinweis in der Einleitung hinzugefügt, dass nun auch Testing vom Sicherheitsteam unterstützt wird

Version 3-8 July 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Die Hinweise neugeschrieben, wie man SSH in einer Chroot-Umgebung einsperrt, um die verschiedenen Optionen deutlicher herauszustellen. Vielen Dank an Bruce Park, der auf verschiedene Fehler in diesem Anhang hinwies
Den Aufruf von lsof verbessert, wie es von Christophe Sahut vorgeschlagen wurde
Patches von Uwe Hermann zur Verbesserung von Tippfehlern eingefügt
Einen Tippfehler in einer Referenz verbessert, der von Moritz Naumann entdeckt wurde

Version 3-7 April 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Einen Abschnitt über die bewährten Methoden der Debian-Entwickler in Hinblick auf Sicherheitsfragen hinzugefügt
Ein Firewall-Skript mit Kommentaren von WhiteGhost hinzugefügt

Version 3-6 March 2006 JavierFernández-Sanguino Peña<jfs@debian.org>

Einen Patch von Thomas Sjögren eingefügt, der beschreibt, dass `noexec` wie erwartet mit »neuen« Kernel arbeitet, der Informationen über den Umgang mit temporären Dateien und einige Verweise auf externe Dokumentationen hinzufügt
Nach einem Vorschlag von Freek Dijkstra einen Verweis auf Dan Farmers und Wietse Venemas Website über forensische Entdeckungen eingefügt und den Abschnitt über forensische Analyse mit weiteren Verweisen etwas erweitert
Dank Christoph Auer die URL des italienischen CERT korrigiert
Wieder Joey Hess' Informationen aus dem Wiki über Secure Apt verwendet und sie in den Infrastrukturabschnitt eingefügt
Abschnitte in Hinblick auf ältere Versionen (Woody oder Potato) überarbeitet
Einige Darstellungsprobleme mit einem Patch von Simon Brandmair ausgebessert

Patches von Carlo Perassi eingepflegt: ACL-Patches sind überflüssig, ebenso wie die Openwall-Patches, Fixme-Anmerkungen über die Kernelserien 2.2 und 2.4 entfernt, hap ist überflüssig (und nicht in WNPP), Verweise auf Immunix entfernt (StackGuard gehört jetzt Novell) und verbesserte ein FIXME über der Verwendung von bsign oder elfsign
Verweise auf die Webseiten von SELinux aktualisiert, so dass sie auf das Wiki verweisen (derzeit die aktuellste Informationsquelle)
Dateimarkierungen eingefügt und eine einheitlichere Verwendung von »MD5 sum« mit einem Patch von Jens Seidel hergestellt
Patch von Joost van Baal angewendet, mit dem die Informationen im Firewall-Abschnitt verbessert werden (Verweis auf das Wiki anstatt alle verfügbaren Firewall-Paket aufzulisten) (schließt: #339865)
Den FAQ-Abschnitt über die Verwundbarkeitsstatistiken überarbeitet, dank Carlos Galisteo de Cabos Hinweis, dass der Abschnitt veraltet ist
Das Zitat des Social Contracts 1.1 anstatt von 1.0 verwendet, wie von Francesco Poli vorgeschlagen

Version 3-5 November 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Hinweis im SSH-Abschnitt eingefügt, dass Chroot nicht funktioniert, wenn die Option nodev mit der Partition verwendet wird, und auf das neuste ssh-Paket mit dem chroot-Patch verwiesen. Vielen Dank an Lutz Broedel für diese Hinweise
Einen Tippfehler ausgebessert, der von Marcos Roberto Greiner entdeckt wurde (md5sum sollte sha1sum im Code-Schnipsel sein)
Jens Seidels Patch eingefügt, der eine Anzahl von Paketnamen und Tippfehlern verbesserte
Kleine Aktualisierung des Werkzeugabschnitts, Werkzeuge entfernt, die nicht länger verfügbar sind, und einige neue hinzugefügt
Teile des Abschnitts neu geschrieben, in dem es darum geht, wo und in welchen Formaten dieses Dokument erhältlich ist (die Website stellt eine PDF-Version zur Verfügung). Auch angemerkt, dass Kopien auf anderen Sites und Übersetzungen veraltet sein könnten (viele der Treffer auf Google für dieses Handbuch auf anderen Sites sind tatsächlich veraltet).

Version 3-4 August-September 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Die Verbesserung der Sicherheit nach der Installation im Zusammenhang mit der Kernelkonfiguration für den Schutz der Netzwerkebene mit der Datei sysctl.conf verbessert. Wurde von Will Moy zur Verfügung gestellt.
Den Abschnitt über Gdm dank Simon Brandmair verbessert
Ausbesserungen von Tippfehlern, die von Frédéric Bothamy und Simon Brandmair entdeckt wurden
Verbesserungen im Abschnitt »Nach der Installation« im Zusammenhang, wie MD5-Summen (oder SHA-1-Summen) für periodische Überprüfungen erstellt werden
Den Abschnitt »Nach der Installation« in Hinblick auf die Konfiguration von Checksecurity (war veraltet) aktualisiert

Version 3-3 June 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Einen Code-Schnipsel hinzugefügt, um mit Grep-available eine Liste von Paketen zu erstellen, die von Perl abhängen. Wurde so in #302470 erbeten
Den Abschnitt über Netzwerkdienste neu geschrieben (welche installiert sind und wie man sie abschaltet)
Weitere Informationen zum Abschnitt über die Entwicklung eines Honigtopfs hinzugefügt, indem nützliche Debian-Pakete erwähnt werden

Version 3-2 March 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Den Abschnitt über die Konfiguration von Limits mit PAM erweitert
Informationen hinzugefügt, wie pam_chroot für Openssh eingesetzt wird (auf Grundlage der README von pam_chroot)
Einige kleinere Dinge korrigiert, die von Dan Jacobson gemeldet wurden
Die Informationen über Kernelpatches aktualisiert, teilweise auf Grundlage eines Patches von Carlo Perassi sowie durch Anmerkungen zu aufgegebenen Teilen des Kernels und zu neuen Kernelpatches (adamantix)

Einen Patch von Simon Brandmair eingefügt, der einen Satz im Zusammenhang mit Login-Fehlern auf dem Terminal ausbesserte
Mozilla/Thunderbird zu den gültigen GPG-Agenten hinzugefügt, wie von Kapolnai Richard vorgeschlagen wurde
Den Abschnitt über Sicherheitsaktualisierungen, die Aktualisierung von Bibliotheken und des Kernels betreffen, und wie man herausfindet, ob Dienste neu gestartet werden müssen, erweitert
Den Abschnitt über die Firewall neu geschrieben, die Informationen, die Woody betreffen, nach unten verschoben und die übrigen Abschnitte erweitert, einschließlich Hinweisen dazu, wie man von Hand eine Firewall einrichtet (mit einem Beispielsskript) und wie man die Konfiguration der Firewall testen kann
Einige Informationen bezüglich der Veröffentlichung von Debian 3.1 hinzugefügt
Ausführlichere Hinweise zu Kernelupgrades hinzugefügt, die sich besonders an diejenigen richten, die das alte Installationssystem verwenden
Einen kurzen Abschnitt über die experimentelle Veröffentlichung von Apt 0.6 eingefügt, die die Überprüfung von Paketsignaturen enthält. Den alten Inhalt in den Abschnitt verschoben und auch einen Verweis auf die Veränderungen, die in Aptitude vorgenommen wurden, hinzugefügt
Ausbesserungen von Tippfehlern, die von Frédéric Bothamy entdeckt wurden

Version 3-1 January 2005 JavierFernández-Sanguino Peña<jfs@debian.org>

Added clarification to ro /usr with patch from Joost van Baal.
Apply patch from Jens Seidel fixing many typos.
FreeSWAN is dead, long live OpenSWAN.
Added information on restricting access to RPC services (when they cannot be disabled) also included patch provided by Aarre Laakso.
Update aj's apt-check-sigs script.
Apply patch Carlo Perassi fixing URLs.
Apply patch from Davor Ocelic fixing many errors, typos, urls, grammar and FIXMEs. Also adds some additional information to some sections.
Rewrote the section on user auditing, highlight the usage of script which does not have some of the issues associated to shell history.

Version 3-0 December 2004 JavierFernández-Sanguino Peña<jfs@debian.org>

Rewrote the user-auditing information and include examples on how to use script.

Version 2-99 March 2004 JavierFernández-Sanguino Peña<jfs@debian.org>

Added information on references in DSAs and CVE-Compatibility.
Added information on apt 0.6 (apt-secure merge in experimental).
Fixed location of Chroot daemons HOWTO as suggested by Shuying Wang.
Changed APACHECTL line in the Apache chroot example (even if its not used at all) as suggested by Leonard Norrgard.
Added a footnote regarding hardlink attacks if partitions are not setup properly.
Added some missing steps in order to run bind as named as provided by Jeffrey Prosa.
Added notes about Nessus and Snort out-of-dateness in woody and availability of backported packages.
Added a chapter regarding periodic integrity test checks.
Clarified the status of testing regarding security updates (Debian bug 233955).
Added more information regarding expected contents in securetty (since it's kernel specific).
Added pointer to snoopylogger (Debian bug 179409).
Added reference to guarddog (Debian bug 170710).
apt-ftparchive is in apt-utils, not in apt (thanks to Emmanuel Chantreau for pointing this out).
Removed jvirus from AV list.

Version 2-98 JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed URL as suggested by Frank Lichtenheld.
Fixed PermitRootLogin typo as suggested by Stefan Lindenau.

Version 2-97	September 2003	JavierFernández-Sanguino Peña<jfs@debian.org>

Added those that have made the most significant contributions to this manual (please mail me if you think you should be in the list and are not).
Added some blurb about FIXME/TODOs.
Moved the information on security updates to the beginning of the section as suggested by Elliott Mitchell.
Added grsecurity to the list of kernel-patches for security but added a footnote on the current issues with it as suggested by Elliott Mitchell.
Removed loops (echo to 'all') in the kernel's network security script as suggested by Elliott Mitchell.
Added more (up-to-date) information in the antivirus section.
Rewrote the buffer overflow protection section and added more information on patches to the compiler to enable this kind of protection.

Version 2-96	August 2003	JavierFernández-Sanguino Peña<jfs@debian.org>

Removed (and then re-added) appendix on chrooting Apache. The appendix is now dual-licensed.

Version 2-95	June 2003	JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed typos spotted by Leonard Norrgard.
Added a section on how to contact CERT for incident handling (Kapitel 11, *Nach einer Kompromittierung (Reaktion auf einem Vorfall)*).
More information on setting up a Squid proxy.
Added a pointer and removed a FIXME thanks to Helge H. F.
Fixed a typo (save_inactive) spotted by Philippe Faes.
Fixed several typos spotted by Jaime Robles.

Version 2-94	April 2003	JavierFernández-Sanguino Peña<jfs@debian.org>

Following Maciej Stachura's suggestions I've expanded the section on limiting users.
Fixed typo spotted by Wolfgang Nolte.
Fixed links with patch contributed by Ruben Leote Mendes.
Added a link to David Wheeler's excellent document on the footnote about counting security vulnerabilities.

Version 2-93	March 2003	FrédéricSchütz<schutz@math-gen.ch>

rewrote entirely the section of ext2 attributes (lsattr/chattr).

Version 2-92	February 2003	JavierFernández-Sanguino Peña<jfs@debian.org>, FrédéricSchütz<schutz@math-gen.ch>

Merge section 9.3 ("useful kernel patches") into section 4.13 ("Adding kernel patches"), and added some content.
Added a few more TODOs.
Added information on how to manually check for updates and also about cron-apt. That way Tiger is not perceived as the only way to do automatic update checks.
Slightly rewrite of the section on executing a security updates due to Jean-Marc Ranger comments.
Added a note on Debian's installation (which will suggest the user to execute a security update right after installation).

Version 2-91	January/February 2003	JavierFernández-Sanguino Peña<jfs@debian.org>

Added a patch contributed by Frédéric Schütz.
Added a few more references on capabilities thanks to Frédéric.
Slight changes in the bind section adding a reference to BIND's 9 online documentation and proper references in the first area (Hi Pedro!).
Fixed the changelog date - new year :-).
Added a reference to Colin's articles for the TODOs.

Removed reference to old ssh+chroot patches.
More patches from Carlo Perassi.
Typo fixes (recursive in Bind is recursion), pointed out by Maik Holtkamp.

Version 2-9 December 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Reorganized the information on chroot (merged two sections, it didn't make much sense to have them separated).
Added the notes on chrooting Apache provided by Alexandre Ratti.
Applied patches contributed by Guillermo Jover.

Version 2-8 JavierFernández-Sanguino Peña<jfs@debian.org>

Applied patches from Carlo Perassi, fixes include: re-wrapping the lines, URL fixes, and fixed some FIXMEs.
Updated the contents of the Debian security team FAQ.
Added a link to the Debian security team FAQ and the Debian Developer's reference, the duplicated sections might (just might) be removed in the future.
Fixed the hand-made auditing section with comments from Michal Zielinski.
Added links to wordlists (contributed by Carlo Perassi).
Fixed some typos (still many around).
Fixed TDP links as suggested by John Summerfield.

Version 2-7 JavierFernández-Sanguino Peña<jfs@debian.org>

Some typo fixes contributed by Tuyen Dinh, Bartek Golenko and Daniel K. Gebhart.
Note regarding /dev/kmem rootkits contributed by Laurent Bonnaud.
Fixed typos and FIXMEs contributed by Carlo Perassi.

Version 2-6 September 2002 CrisTillman<tillman@voice-trak.com>

Changed around to improve grammar/spelling.
s/host.deny/hosts.deny/ (1 place).
Applied Larry Holish's patch (quite big, fixes a lot of FIXMEs).

Version 2-5.1 September 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed minor typos submitted by Thiemo Nagel.
Added a footnote suggested by Thiemo Nagel.
Fixed an URL link.

Version 2-5.0 August 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Applied a patch contributed by Philipe Gaspar regarding the Squid which also kills a FIXME.
Yet another FAQ item regarding service banners taken from the debian-security mailing list (thread "Telnet information" started 26th July 2002).
Added a note regarding use of CVE cross references in the *How much time does the Debian security team...* FAQ item.
Added a new section regarding ARP attacks contributed by Arnaud "Arhuman" Assad.
New FAQ item regarding dmesg and console login by the kernel.
Small tidbits of information to the signature-checking issues in packages (it seems to not have gotten past beta release).
New FAQ item regarding vulnerability assessment tools false positives.
Added new sections to the chapter that contains information on package signatures and reorganized it as a new *Debian Security Infrastructure* chapter.
New FAQ item regarding Debian vs. other Linux distributions.
New section on mail user agents with GPG/PGP functionality in the security tools chapter.
Clarified how to enable MD5 passwords in woody, added a pointer to PAM as well as a note regarding the max definition in PAM.

Added a new appendix on how to create chroot environments (after fiddling a bit with makejail and fixing, as well, some of its bugs), integrated duplicate information in all the appendix.
Added some more information regarding **SSH** chrooting and its impact on secure file transfers. Some information has been retrieved from the debian-security mailing list (June 2002 thread: *secure file transfers*).
New sections on how to do automatic updates on Debian systems as well as the caveats of using testing or unstable regarding security updates.
New section regarding keeping up to date with security patches in the *Before compromise* section as well as a new section about the debian-security-announce mailing list.
Added information on how to automatically generate strong passwords.
New section regarding login of idle users.
Reorganized the securing mail server section based on the *Secure/hardened/minimal Debian (or "Why is the base system the way it is?")* thread on the debian-security mailing list (May 2002).
Reorganized the section on kernel network parameters, with information provided in the debian-security mailing list (May 2002, *syn flood attacked?* thread) and added a new FAQ item as well.
New section on how to check users passwords and which packages to install for this.
New section on PPTP encryption with Microsoft clients discussed in the debian-security mailing list (April 2002).
Added a new section describing what problems are there when binding any given service to a specific IP address, this information was written based on the Bugtraq mailing list in the thread: *Linux kernel 2.4 "weak end host" issue (previously discussed on debian-security as "arp problem")* (started on May 9th 2002 by Felix von Leitner).
Added information on **ssh** protocol version 2.
Added two subsections related to Apache secure configuration (the things specific to Debian, that is).
Added a new FAQ related to raw sockets, one related to /root, an item related to users' groups and another one related to log and configuration files permissions.
Added a pointer to a bug in libpam-cracklib that might still be open... (need to check).
Added more information regarding forensics analysis (pending more information on packet inspection tools such as **tcpflow**).
Changed the "what should I do regarding compromise" into a bullet list and included some more stuff.
Added some information on how to set up the Xscreensaver to lock the screen automatically after the configured timeout.
Added a note related to the utilities you should not install in the system. Included a note regarding Perl and why it cannot be easily removed in Debian. The idea came after reading Intersect's documents regarding Linux hardening.
Added information on lvm and journalling file systems, ext3 recommended. The information there might be too generic, however.
Added a link to the online text version (check).
Added some more stuff to the information on firewalling the local system, triggered by a comment made by Hubert Chan in the mailing list.
Added more information on PAM limits and pointers to Kurt Seifried's documents (related to a post by him to Bugtraq on April 4th 2002 answering a person that had ``discovered'' a vulnerability in Debian GNU/Linux related to resource starvation).
As suggested by Julián Muñoz, provided more information on the default Debian umask and what a user can access if he has been given a shell in the system (scary, huh?).
Included a note in the BIOS password section due to a comment from Andreas Wohlfeld.
Included patches provided by Alfred E. Heggestad fixing many of the typos still present in the document.
Added a pointer to the changelog in the Credits section since most people who contribute are listed here (and not there).
Added a few more notes to the chattr section and a new section after installation talking about system snapshots. Both ideas were contributed by Kurt Pomeroy.
Added a new section after installation just to remind users to change the boot-up sequence.
Added some more TODO items provided by Korn Andras.
Added a pointer to the NIST's guidelines on how to secure DNS provided by Daniel Quinlan.

Added a small paragraph regarding Debian's SSL certificates infrastructure.
Added Daniel Quinlan's suggestions regarding **ssh** authentication and exim's relay configuration.
Added more information regarding securing bind including changes suggested by Daniel Quinlan and an appendix with a script to make some of the changes commented on in that section.
Added a pointer to another item regarding Bind chrooting (needs to be merged).
Added a one liner contributed by Cristian Ionescu-Idbohrn to retrieve packages with tcpwrappers support.
Added a little bit more info on Debian's default PAM setup.
Included a FAQ question about using PAM to provide services without shell accounts.
Moved two FAQ items to another section and added a new FAQ regarding attack detection (and compromised systems).
Included information on how to set up a bridge firewall (including a sample Appendix). Thanks to Francois Bayart who sent this to me in March.
Added a FAQ regarding the syslogd's *MARK heartbeat* from a question answered by Noah Meyerhans and Alain Tesio in December 2001.
Included information on buffer overflow protection as well as some information on kernel patches.
Added more information (and reorganized) the firewall section. Updated the information regarding the iptables package and the firewall generators available.
Reorganized the information regarding log checking, moved logcheck information from host intrusion detection to that section.
Added some information on how to prepare a static package for bind for chrooting (untested).
Added a FAQ item regarding some specific servers/services (could be expanded with some of the recommendations from the debian-security list).
Added some information on RPC services (and when it's necessary).
Added some more information on capabilities (and what lcap does). Is there any good documentation on this? I haven't found any documentation on my 2.4 kernel.
Fixed some typos.

Version 2-4 | June 2002 | JavierFernández-Sanguino Peña<jfs@debian.org>

Rewritten part of the BIOS section.

Version 2-3.1 | April 2002 | JavierFernández-Sanguino Peña<jfs@debian.org>

Wrapped most file locations with the file tag.
Fixed typo noticed by Edi Stojicevi.
Slightly changed the remote audit tools section.
Added some todo items.
Added more information regarding printers and cups config file (taken from a thread on debian-security).
Added a patch submitted by Jesus Climent regarding access of valid system users to Proftpd when configured as anonymous server.
Small change on partition schemes for the special case of mail servers.
Added Hacking Linux Exposed to the books section.
Fixed directory typo noticed by Eduardo Pérez Ureta.
Fixed /etc/ssh typo in checklist noticed by Edi Stojicevi.

Version 2-3.0 | April 2002 | JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed location of dpkg conffile.
Remove Alexander from contact information.
Added alternate mail address.
Fixed Alexander mail address (even if commented out).
Fixed location of release keys (thanks to Pedro Zorzenon for pointing this out).

Version 2-2 | April 2002 | JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed typos, thanks to Jamin W. Collins.
Added a reference to apt-extracttemplate manpage (documents the APT::ExtractTemplate config).

Added section about restricted SSH. Information based on that posted by Mark Janssen, Christian G. Warden and Emmanuel Lacour on the debian-security mailing list.
Added information on antivirus software.
Added a FAQ: su logs due to the cron running as root.

Version 2-1 April 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Changed FIXME from lshell thanks to Oohara Yuuma.
Added package to sXid and removed comment since it *is* available.
Fixed a number of typos discovered by Oohara Yuuma.
ACID is now available in Debian (in the acidlab package) thanks to Oohara Yuuma for noticing.
Fixed LinuxSecurity links (thanks to Dave Wreski for telling).

Version 2-0 March 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Converted the HOWTO into a Manual (now I can properly say RTFM).
Added more information regarding tcp wrappers and Debian (now many services are compiled with support for them so it's no longer an **inetd** issue).
Clarified the information on disabling services to make it more consistent (rpc info still referred to update-rc.d).
Added small note on lprng.
Added some more info on compromised servers (still very rough).
Fixed typos reported by Mark Bucciarelli.
Added some more steps in password recovery to cover the cases when the admin has set paranoid-mode=on.
Added some information to set paranoid-mode=on when login in console.
New paragraph to introduce service configuration.
Reorganized the *After installation* section so it is more broken up into several issues and it's easier to read.
Wrote information on how to set up firewalls with the standard Debian 3.0 setup (iptables package).
Small paragraph explaining why installing connected to the Internet is not a good idea and how to avoid this using Debian tools.
Small paragraph on timely patching referencing to IEEE paper.
Appendix on how to set up a Debian snort box, based on what Vladimir sent to the debian-security mailing list (September 3rd 2001).
Information on how logcheck is set up in Debian and how it can be used to set up HIDS.
Information on user accounting and profile analysis.
Included apt.conf configuration for read-only /usr copied from Olaf Meeuwissen's post to the debian-security mailing list.
New section on VPN with some pointers and the packages available in Debian (needs content on how to set up the VPNs and Debian-specific issues), based on Jaroslaw Tabor's and Samuli Suonpaa's post to debian-security.
Small note regarding some programs to automatically build chroot jails.
New FAQ item regarding identd based on a discussion in the debian-security mailing list (February 2002, started by Johannes Weiss).
New FAQ item regarding **inetd** based on a discussion in the debian-security mailing list (February 2002).
Introduced note on rcconf in the "disabling services" section.
Varied the approach regarding LKM, thanks to Philipe Gaspar.
Added pointers to CERT documents and Counterpane resources.

Version 1-99 January 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Added a new FAQ item regarding time to fix security vulnerabilities.
Reorganized FAQ sections.
Started writing a section regarding firewalling in Debian GNU/Linux (could be broadened a bit).
Fixed typos sent by Matt Kraai.
Fixed DNS information.
Added information on whisker and nbtscan to the auditing section.

Fixed some wrong URLs.

Version 1-98 January 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Added a new section regarding auditing using Debian GNU/Linux.
Added info regarding finger daemon taken from the security mailing list.

Version 1-97 January 2002 JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed link for Linux Trustees.
Fixed typos (patches from Oohara Yuuma and Pedro Zorzenon).

Version 1-96 December 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Reorganized service installation and removal and added some new notes.
Added some notes regarding using integrity checkers as intrusion detection tools.
Added a chapter regarding package signatures.

Version 1-95 December 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added notes regarding Squid security sent by Philipe Gaspar.
Fixed rootkit links thanks to Philipe Gaspar.

Version 1-94 November 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added some notes regarding Apache and Lpr/lpng.
Added some information regarding noexec and read-only partitions.
Rewrote how users can help in Debian security issues (FAQ item).

Version 1-93 November 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Fixed location of mail program.
Added some new items to the FAQ.

Version 1-92 October 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added a small section on how Debian handles security.
Clarified MD5 passwords (thanks to `rocky').
Added some more information regarding harden-X from Stephen van Egmond.
Added some new items to the FAQ.

Version 1-91 October 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added some forensics information sent by Yotam Rubin.
Added information on how to build a honeynet using Debian GNU/Linux.
Added some more TODOS.
Fixed more typos (thanks Yotam!).

Version 1-9 October 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added patch to fix misspellings and some new information (contributed by Yotam Rubin).
Added references to other online (and offline) documentation both in a section (see „Seien Sie wachsam gegenüber generellen Sicherheitsproblemen!“) by itself and inline in some sections.
Added some information on configuring Bind options to restrict access to the DNS server.
Added information on how to automatically harden a Debian system (regarding the harden package and bastille).
Removed some done TODOs and added some new ones.

Version 1-8 October 2001 JavierFernández-Sanguino Peña<jfs@debian.org>

Added the default user/group list provided by Joey Hess to the debian-security mailing list.
Added information on LKM root-kits („Ladbare Kernel-Module (LKM)“) contributed by Philipe Gaspar.
Added information on Proftp contributed by Emmanuel Lacour.
Recovered the checklist Appendix from Era Eriksson.

Added some new TODO items and removed other fixed ones.
Manually included Era's patches since they were not all included in the previous version.

Version 1-7	September 2001	JavierFernández-Sanguino Peña<jfs@debian.org>, Era-Eriksson<era@iki.fi>

Typo fixes and wording changes.
Minor changes to tags in order to keep on removing the tt tags and substitute prgn/package tags for them.

Version 1-6	August 2001	JavierFernández-Sanguino Peña<jfs@debian.org>

Added pointer to document as published in the DDP (should supersede the original in the near future).
Started a mini-FAQ (should be expanded) with some questions recovered from my mailbox.
Added general information to consider while securing.
Added a paragraph regarding local (incoming) mail delivery.
Added some pointers to more information.
Added information regarding the printing service.
Added a security hardening checklist.
Reorganized NIS and RPC information.
Added some notes taken while reading this document on my new Visor :).
Fixed some badly formatted lines.
Fixed some typos.
Added a Genius/Paranoia idea contributed by Gaby Schilders.

Version 1-5	May 2001	JavierFernández-Sanguino Peña<jfs@debian.org>, JosipRodin<joy@debian.org>

Added paragraphs related to BIND and some FIXMEs.

Version 1-4	May 2001	JavierFernández-Sanguino Peña<jfs@debian.org>

Small setuid check paragraph
Various minor cleanups.
Found out how to use `sgml2txt -f` for the txt version.

Version 1-3	March 2001	JavierFernández-Sanguino Peña<jfs@debian.org>

Added a security update after installation paragraph.
Added a proftpd paragraph.
This time really wrote something about XDM, sorry for last time.

Version 1-2	December 2000	JavierFernández-Sanguino Peña<jfs@debian.org>

Lots of grammar corrections by James Treacy, new XDM paragraph.

Version 1-1	December 2000	JavierFernández-Sanguino Peña<jfs@debian.org>

Typo fixes, miscellaneous additions.

Version 1-0	December 2000	JavierFernández-Sanguino Peña<jfs@debian.org>

Erste Veröffentlichung

Anhang B. Anhang

Der Abhärtungsprozess Schritt für Schritt

Eine Anleitung, die Schritt für Schritt darstellt, wie ein Debian 2.2 GNU/Linux-System nach der Installation abgehärtet wird, ist unten aufgeführt. Das ist nur eine denkbare Herangehensweise für einem solchen Vorgang. Sie ist am Absichern von Netzwerkdiensten orientiert und stellt den gesamten Anlauf der Konfiguration vor. Vergleichen Sie auch „Prüfliste der Konfiguration“.

- Installieren Sie das System. Beachten Sie dabei die Informationen dieses HOWTOs bezüglich der Partitionierung. Nach der Basis-Installation nehmen Sie eine angepasste Installation vor. Wählen Sie keine Task-Pakete aus. Aktivieren Sie shadow-Passwörter.
- Entfernen Sie mit **dselect** alle nicht benötigten, aber ausgewählten Pakete, bevor Sie [I]nstallation wählen. Belassen Sie nur die absolut notwendige Software auf dem System.
- Aktualisieren Sie die ganze Software mit den aktuellen Paketen von security.debian.org, wie bereits unter „Ausführen von Sicherheitsaktualisierungen“ beschrieben.
- Implementieren Sie die in dieser Anleitung vorgeschlagenen Maßnahmen zu Benutzer-Quotas, Ausgestaltung des Anmeldevorgangs und **Lilo**.
- Machen Sie sich eine Liste von allen Diensten, die derzeit auf Ihrem System laufen. Versuchen Sie dazu Folgendes:

  ```
  $ ps aux
  $ netstat -pn -l -A inet
  # /usr/sbin/lsof -i | grep LISTEN
  ```

  Damit das dritte Kommando funktioniert, werden Sie lsof-2.2 installieren müssen (und es als Root laufen lassen). Beachten Sie, dass **lsof** das Wort LISTEN passend zu Ihrer Lokalisation übersetzen kann.
- Um einen unnötigen Dienst zu entfernen, stellen Sie zunächst fest, wie er gestartet wird und welches Paket ihn zur Verfügung stellt. Sie können dies ganz einfach machen, indem Sie das Programm prüfen, das auf dem Socket lauscht. Das nachfolgende Shell-Skript, das die Programme **lsof** und **dpkg** verwendet, macht genau das:

  ```
  #!/bin/sh
  # FIXME: this is quick and dirty; replace with a more robust script snippet
  for i in `sudo lsof -i | grep LISTEN | cut -d " " -f 1 |sort -u` ; do
    pack=`dpkg -S $i |grep bin |cut -f 1 -d : | uniq`
    echo "Service $i is installed by $pack";
    init=`dpkg -L $pack |grep init.d/ `
    if [ ! -z "$init" ]; then
      echo "and is run by $init"
    fi
  done
  ```
- Wenn Sie einen unerwünschten Dienst finden, entfernen Sie das Paket (mit **dpkg --purge**). Oder benutzen Sie **update-rc.d** (siehe „Daemons abschalten“), um ihn aus dem Start-Prozess zu entfernen.
- Überprüfen Sie bei inetd-Diensten (werden durch den Superdaemon gestartet), welche Dienste in `/etc/inetd.conf` aktiviert sind. Verwenden Sie dazu Folgendes:

```
$ grep -v "^#" /etc/inetd.conf | sort -u
```

Deaktivieren Sie dann diejenigen Dienste, die Sie nicht benötigen, indem Sie die Zeile in `/etc/inetd.conf` auskommentieren, das Paket entfernen, oder indem Sie **update-inetd** benutzen.

- Wenn Sie Dienste eingehüllt haben (und **/usr/sbin/tcpd** benutzen), prüfen Sie, ob die Dateien `/etc/hosts.allow` und `/etc/hosts.deny` passend zu Ihrer Richtlinie für die Dienste konfiguriert sind.

- Wenn der Server mehr als eine externe Schnittstelle benutzt, können Sie Dienste darauf beschränken, auf bestimmten Schnittstellen zu lauschen. Ob das möglich ist, hängt aber von den Diensten ab. Wenn Sie zum Beispiel internen FTP-Zugriff erlauben wollen, lassen Sie den FTP-Daemon nur auf der internen Schnittstelle lauschen, nicht auf allen (d.h. 0.0.0.0:21).

- Booten Sie die Maschine neu, oder wechseln Sie in den Single-User-Modus und zurück in den Multi-User-Modus mit:

```
# init 1
(....)
# init 2
```

- Prüfen Sie die nun angebotenen Dienste und wiederholen Sie gegebenenfalls die letzten Schritte.

- Installieren Sie jetzt die benötigten Dienste, falls es noch nicht geschehen ist, und konfigurieren Sie sie passend.

- Prüfen Sie mit folgendem Shell-Befehl, unter welchem Benutzer die verfügbaren Dienste laufen:

```
# for i in `/usr/sbin/lsof -i |grep LISTEN |cut -d " " -f 1 |sort -u`; \
> do user=`ps ef |grep $i |grep -v grep |cut -f 1 -d " "` ; \
> echo "Dienst $i läuft als Benutzer $user"; done
```

Überlegen Sie, ob Sie diese Dienste einem bestimmten Benutzer oder Gruppe zuweisen wollen und sie vielleicht auch in eine chroot-Umgebung einsperren wollen, um die Sicherheit zu erhöhen. Sie können dies tun, indem Sie die `/etc/init.d`-Skripte ändern, die den Dienst starten. Die meisten Dienste benutzen unter Debian **start-stop-daemon**, der die dafür Optionen (`--change-uid` und `--chroot`) zur Verfügung stellt. Ein paar warnende Worte zum Einsperren in eine **chroot**-Umgebung: Sie müssen alle Dateien, die durch das Paket des Dienstes installiert wurden (verwenden Sie dpkg -L), und alle Pakete, von denen es abhängt, in die **Chroot**-Umgebung legen. Informationen, wie das Programm **ssh** in eine **chroot**-Umgebung eingesperrt wird, finden Sie unter „Chroot-Umgebung für SSH“.

- Wiederholen Sie die Schritte oben um zu prüfen, ob nur die gewünschten Dienste laufen und ob sie unter der gewünschten Benutzer/Gruppen-Kombination laufen.

- Testen Sie die installierten Dienste, um festzustellen, ob sie wie erwartet arbeiten.

- Überprüfen Sie das System, indem Sie einen Scanner zur Abschätzung der Verwundbarkeit (zum Beispiel nessus) benutzen, um Angriffsmöglichkeiten (Fehlkonfigurationen, alte oder nicht benötigte Dienste) zu finden.

- Installieren Sie Instrumente zur Entdeckung von Eindringlingen in Netzwerk und Hosts (wie snort und logcheck).

- Wiederholen Sie den Netzwerk-Scan und prüfen Sie, ob das System zur Erkennung von Eindringlingen funktioniert.

Die richtig Paranoiden überlegen sich auch Folgendes:

- Fügen Sie dem System Firewall-Fähigkeiten hinzu, die eingehende Verbindungen nur zu angebotenen Diensten erlauben und ungenehmigte ausgehende Verbindungen verhindern.
- Überprüfen Sie erneut die Installation auf Angriffspunkte mit einem Netzwerk-Scanner.
- Prüfen Sie ausgehende Verbindungen vom System zu Hosts außerhalb mit einem Netzwerk-Scanner, um sicherzustellen, dass ungewollte Verbindungen keinen Weg nach draußen finden.

FIXME: this procedure considers service hardening but not system hardening at the user level, include information regarding checking user permissions, SETUID files and freezing changes in the system using the ext2 file system.

Prüfliste der Konfiguration

Dieser Anhang wiederholt kurz Punkte aus anderen Abschnitten dieser Anleitung in einem verdichteten Prüflisten-Format. Er ist als schnelle Zusammenfassung für Leute gedacht, die bereits diese Anleitung gelesen haben. Es gibt auch andere gute Prüflisten, zum Beispiel Kurt Seifrieds http://seifried.org/security/os/linux/20020324-securing-linux-step-by-step.html und http://www.cert.org/tech_tips/usc20_full.html.

FIXME: This is based on v1.4 of the manual and might need to be updated.

- Beschränkung des physischen Zugriffs und der Boot-Fähigkeiten:
  - Setzen Sie im BIOS ein Passwort.
  - Schalten Sie im BIOS das Booten von Diskette, CD-ROM, ... ab.
  - Setzen Sie ein LILO- bzw. GRUB-Passwort (`/etc/lilo.conf` bzw. `/boot/grub/menu.lst`); stellen Sie sicher, dass die Konfigurationsdatei von LILO oder GRUB nicht lesbar ist.
- Partitionierung:
  - Legen Sie Daten, die von Benutzern geschrieben wurden, Daten, die nicht zum System gehören, und sich ständig ändernde Laufzeitdaten auf eigenen, getrennten Partitionen ab.
  - Setzen Sie die Mount-Optionen `nosuid,noexec,nodev` in `/etc/fstab` bei ext2/3-Partitionen, die keine ausführbaren Programme enthalten sollten, wie zum Beispiel `/home` oder `/tmp`.
- Passworthygiene und Anmeldesicherheit:
  - Wählen Sie ein gutes Root-Passwort.
  - Installieren und benutzen Sie PAM:
    - Fügen Sie die Unterstützung von PAM-MD5 hinzu, und stellen Sie sicher (allgemein gesprochen), dass die Einträge in den `/etc/pam.d/`-Dateien, die Zugriff auf die Maschine gewähren, das zweite Feld in der pam.d-Datei auf `requisite` oder `required` gesetzt haben.
    - Ändern Sie `/etc/pam.d/login`, so dass nur lokale Anmeldungen von Root erlaubt werden.

    - Bezeichnen Sie außerdem autorisierte ttys in `/etc/security/access.conf` und richten Sie diese Datei überhaupt so ein, dass Anmeldungen von Root so weit wie möglich eingeschränkt werden.
    - Fügen Sie pam_limits.so hinzu, wenn Sie Begrenzungen für jeden Benutzer vornehmen wollen.
    - Ändern Sie `/etc/pam.d/passwd`: Erhöhen Sie die minimale Länge von Passwörtern (vielleicht sechs Zeichen) und schalten Sie MD5 ein.
    - Wenn Sie es wünschen, fügen Sie `/etc/group` die Gruppe wheel hinzu; fügen Sie `/etc/pam.d/su` pam_wheel.so group=wheel hinzu.
    - Für angepasste Kontrollen der einzelnen Benutzer nehmen Sie Einträge in pam_listfile.so an den passenden Stellen vor.
    - Erstellen Sie eine Datei `/etc/pam.d/other` und setzen Sie sie mit strenger Sicherheit auf.
  - Setzen Sie in `/etc/security/limits.conf` Begrenzungen (beachten Sie, dass `/etc/limits` nicht benutzt wird, wenn Sie PAM verwenden).
  - Nehmen Sie Einschränkungen in `/etc/login.defs` vor; wenn Sie MD5 oder PAM einschalten, machen Sie auch hier die entsprechenden Änderungen.
  - Nehmen Sie Einschränkungen in `/etc/pam.d/login` vor.
  - Schalten Sie den FTP-Zugriff von Root in `/etc/ftpusers` ab.
  - Schalten Sie Anmeldungen von Root über das Netzwerk ab; benutzen Sie su(1) oder sudo(1) (denken Sie über die Installation von sudo nach).
  - Benutzen Sie PAM, um zusätzliche Auflagen für Anmeldungen zu ermöglichen.
- Andere lokale Sicherheitsangelegenheiten:
  - Kernel-Tweaks (siehe „Konfiguration der Netzwerkfähigkeiten des Kernels“)
  - Kernel-Patches (siehe „Den Kernel patchen“)
  - Schränken Sie die Zugriffsrechte auf Protokolldateien (`/var/log/{last,fail}log`, Protokolle von Apache) ein.
  - Stellen Sie sicher, dass in `/etc/checksecurity.conf` die Prüfung von SETUID eingeschaltet ist.
  - Überlegen Sie sich, an Protokolldateien nur erweiterbar (append-only) und Konfigurationsdateien unveränderbar (immutable) zu machen, indem Sie chattr benutzen (nur ext2/3-Dateisystem).
  - Setzen Sie eine Integritätsprüfung des Dateisystems auf (siehe „Prüfung der Integrität des Dateisystems“). Installieren Sie debsums.
  - Alles auf einem lokalen Drucker mitprotokollieren?
  - Brennen Sie Ihre Konfiguration auf eine bootbare CD und booten Sie hiervon?
  - Abschalten von Kernel-Modulen?
- Einschränkung des Netzwerkzugriffs:

- Installieren und konfigurieren Sie **ssh** (Vorschlag: PermitRootLogin No in `/etc/ssh/sshd_config`, PermitEmptyPasswords No; beachten Sie auch die anderen Vorschläge im Text).
- Schalten Sie **in.telnetd** ab oder entfernen Sie ihn, falls er installiert ist.
- Deaktivieren Sie ganz allgemein alle überflüssigen Dienste in `/etc/inetd.conf`. Benutzen Sie dazu **update-inetd --disable** (oder Sie schalten **inetd** ganz ab oder verwenden einen Ersatz wie **xinetd** oder **rlinetd**).
- Schalten Sie andere überflüssige Netzwerkdienste ab. ftp, DNS, www, usw. sollten nicht laufen, wenn Sie sie nicht brauchen und nicht regelmäßig überwachen. In den meisten Fällen muss ein Mail-Server betrieben werden, sollte aber so konfiguriert sein, dass er nur lokal Mails zustellt.
- Installieren Sie von den Diensten, die Sie brauchen, nicht einfach das verbreiteteste Programm, sondern schauen Sie nach sichereren Versionen, die Debian liefert (oder aus anderen Quellen), um. Was auch immer Sie schließlich benutzen: Stellen Sie sicher, dass Sie die Risiken verstanden haben.
- Setzen Sie **Chroot**-Gefängnisse für auswärtige Benutzer und Daemonen auf.
- Konfigurieren Sie die Firewall und die tcp-Wrapper (d.h. hosts_access(5)); beachten Sie den Trick für `/etc/hosts.deny` im Text.
- Wenn Sie FTP laufen lassen, setzen Sie den ftpd-Server so auf, dass er immer in einer **chroot**-Umgebung im Home-Verzeichnis des Benutzers läuft.
- Wenn Sie X laufen lassen, schalten Sie xhost-Authentifizierung ab und benutzen Sie stattdessen **ssh**. Oder noch besser: Deaktivieren Sie die Weiterleitung von X komplett, falls das möglich ist (fügen Sie -nolisten tcp zu der X-Kommando-Zeile hinzu und schalten Sie XDMCP in `/etc/X11/xdm/xdm-config` ab, indem Sie den requestPort auf 0 setzen).
- Schalten Sie Zugriff von außerhalb auf den Drucker ab.
- Tunneln Sie alle IMAP- oder POP-Sitzungen durch SSL oder **ssh**. Installieren Sie stunnel, wenn Sie diesen Dienst anderen Mail-Benutzern anbieten wollen.
- Setzen Sie einen Log-Host auf, und konfigurieren Sie andere Maschinen, ihre Protokolle an diesen Host zu senden (`/etc/syslog.conf`).
- Sichern Sie BIND, Sendmail und andere komplexe Daemonen ab (starten Sie sie in einer **chroot**-Umgebung und als Pseudobenutzer, der nicht Root ist).
- Installieren Sie tiger oder ein ähnliches Werkzeug zur Erkennung von Eindringlingen in Ihr Netzwerk.
- Installieren Sie snort oder ein ähnliches Werkzeug zur Erkennung von Eindringlingen in Ihr Netzwerk.
- Verzichten Sie, falls möglich, auf NIS und RPC (Abschalten von portmap).

- Angelegenheiten mit Richtlinien:
  - Klären Sie die Benutzer über das Wie und Warum Ihrer Richtlinien auf. Wenn Sie etwas verboten haben, das auf anderen Systemen normalerweise verfügbar ist, stellen Sie Dokumentation bereit, die erklärt, wie man die gleichen Resultate erreicht, indem man andere, sichere Mittel anwendet.
  - Verbieten Sie die Nutzung von Protokollen, die Klartext-Passwörter benutzen (**telnet**, **rsh** und ähnliche, ftp, imap, pop, http, ...).

  - Verbieten Sie Programme, die SVGAlib benutzen.
  - Benutzen Sie Disk-Quotas.
- Bleiben Sie über Sicherheitsangelegenheiten informiert:
  - Abonnieren Sie sicherheitsrelevante Mailinglisten.
  - Richten Sie Sicherheitsaktualisierungen für apt ein – fügen Sie `/etc/apt/sources.list` einen Eintrag (oder Einträge) für http://security.debian.org/ hinzu.
  - Vergessen Sie auch nicht, regelmäßig **apt-get update ; apt-get upgrade** (vielleicht als **Cron**-Job?) laufen zu lassen, wie unter „Ausführen von Sicherheitsaktualisierungen“ beschrieben.

Aufsetzen eines eigenständigen IDS

Sie können sehr leicht eine Debian-Box als eigenständiges Eindringlings-Erkennungs-System (Intrusion Detection System, IDS) aufsetzen, indem Sie snort benutzen und eine webbasierte Schnittstelle zur Überwachung der Alarme über Eindringlinge einrichten:

- Installieren Sie ein Debian-Basis-System ohne zusätzliche Pakete.
- Installieren Sie eine Version von Snort, die Datenbanken unterstützt, und richten Sie Snort so ein, dass die Alarme in der Datenbank protokolliert werden.
- Laden Sie BASE (Basic Analysis and Security Engine) oder ACID (Analysis Console for Intrusion Databases, Konsole zur Analyse für Eindringling-Datenbanken) herunter und installieren Sie es. Konfigurieren Sie es so, dass es die gleiche Datenbank wie Snort verwendet.
- Installieren Sie die notwendigen Pakete.[1]

BASE wird derzeit für Debian im Paket acidbase geliefert, ACID im Paket acidlab.[2] Beide stellen eine graphische WWW-Schnittstelle zur Ausgabe von Snort zur Verfügung.

Neben der Grundinstallationen benötigen Sie auch einen Webserver (wie apache), einen **PHP**-Interpreter und eine relationale Datenbank (wie postgresql oder mysql), wo Snort seine Alarme ablegen kann.

Dieses System sollte mit wenigstens zwei Netzwerk-Schnittstellen ausgestattet sein: Eine verbunden mit einem Verwaltungs-LAN (um die Resultate abzufragen und das System zu verwalten), und eine ohne IP-Adresse, das an mit dem zu beobachtenden Abschnitt des Netzwerks verbunden ist. Sie sollten den Webserver so einrichten, dass er nur auf der Schnittstelle lauscht, die mit dem Verwaltungs-LAN verbunden ist.

Sie sollten beide Schnittstellen in der Standardkonfigurationsdatei von Debian `/etc/network/interfaces` einrichten. Eine Adresse, nämlich die des Verwaltungs-LANs, sollten Sie wie gewöhnlich einrichten. Die andere Schnittstelle muss so konfiguriert werden, dass sie aktiviert wird, wenn das System startet, ihr darf aber keine Interface-Adresse zugewiesen sein. Eine Konfiguration der Schnittstelle könnte folgendermaßen aussehen:

```
auto eth0
iface eth0 inet manual
      up ifconfig $IFACE 0.0.0.0 up
```

[1] Normalerweise werden alle benötigten Pakete installiert, um Abhängigkeiten aufzulösen.
[2] Es kann auch von http://www.cert.org/kb/acid/, http://acidlab.sourceforge.net oder http://www.andrew.cmu.edu/~rdanyliw/snort/ heruntergeladen werden.

```
up ip link set $IFACE promisc on
down ip link set $IFACE promisc off
down ifconfig $IFACE down
```

The above configures an interface to read all the traffic on the network in a *stealth*-type configuration. This prevents the NIDS system to be a direct target in a hostile network since the sensors have no IP address on the network. Notice, however, that there have been known bugs over time in sensors part of NIDS (for example see https://lists.debian.org/debian-security-announce/2003/msg00087.html related to Snort) and remote buffer overflows might even be triggered by network packet processing.

You might also want to read the http://www.faqs.org/docs/Linux-HOWTO/Snort-Statistics-HOWTO.html and the documentation available at the https://www.snort.org/#documents.

Aufsetzenden einer Bridge-Firewall

Diese Informationen trug Francois Bayart bei, um Benutzern zu helfen, eine Linux Bridge/Firewall mit 2.4.x Kernel und iptables aufzusetzen. Ein Kernelpatch wird nicht mehr benötigt, da der Code Standardinhalt der Linux-Kernel-Distribution wurde.

Um die notwendigen Einstellungen im Kernel vorzunehmen, rufen Sie `make menuconfig` oder `make xconfig` auf. Aktivieren Sie im Abschnitt *Networking options* folgende Optionen:

```
[*] Network packet filtering (replaces ipchains)
[ ]   Network packet filtering debugging (NEW)
<*> 802.1d Ethernet Bridging
[*]   netfilter (firewalling) support (NEW)
```

Passen Sie auf, dass Sie Folgendes deaktiviert müssen, wenn Sie Firewall-Regeln anwenden wollen; anderenfalls wird **iptables** nicht funktioniert:

```
[ ]   Network packet filtering debugging (NEW)
```

Anschließend müssen Sie die korrekten Optionen im Abschnitt *IP: Netfilter Configuration* setzen. Dann kompilieren und installieren Sie den Kernel. Wenn Sie dies auf die *Debian-Art* machen wollen, installieren Sie kernel-package und benutzen Sie **make-kpkg**, um ein maßgeschneidertes Debian-Kernelpaket zu erstellen, das Sie mit dpkg auf Ihrem Server installieren können. Sobald der neue Kernel kompiliert und installiert ist, müssen Sie das Paket bridge-utils installieren.

Wenn Sie diesen Schritt abgeschlossen haben, können Sie die Konfiguration Ihrer Bridge fertigstellen. Im nächsten Abschnitt werden Ihnen zwei verschiedene mögliche Konfigurationen einer Bridge vorgestellt. Beide sind mit einer Übersicht eines hypothetischen Netzwerks und den notwendigen Befehlen versehen.

Eine Bridge mit NAT- und Firewall-Fähigkeiten

Die erste Konfigurationsmöglichkeit benutzt die Bridge als Firewall mit Network Address Translation (NAT, Übersetzung der Netzwerkadressen), die einen Server und interne LAN-Clienten schützt. Unten wird eine Darstellung der Anordnung des Netzwerks gezeigt:

```
Internet ---- Router ( 62.3.3.25 ) ---- Bridge (62.3.3.26 gw 62.3.3.25 / 192.168.0
                                                |
                                                |
                                                |---- WWW-Server (62.3.3.27 gw 62.3.3.25
```

```
                                          |
                                          |
                                         LAN --- Zipowz (192.168.0.2 gw 192.168.0.
```

Die folgenden Befehle zeigen, wie diese Bridge konfiguriert werden kann:

```
# Erstellen der Schnittstelle br0
/usr/sbin/brctl addbr br0

# Hinzufügen der Ethernet-Schnittstelle, welche die Bridge benutzen soll
/usr/sbin/brctl addif br0 eth0
/usr/sbin/brctl addif br0 eth1

# Starten der Ethernet-Schnittstelle
/sbin/ifconfig eth0 0.0.0.0
/sbin/ifconfig eth1 0.0.0.0

# Konfigurieren der Ethernet-Bridge
# Die Bridge wird korrekt und unsichtbar (transparente Firewall) sein.
# In einem traceroute ist sie versteckt, und Sie behalten Ihr echtes
# Gateway auf Ihren anderen Computern. Jetzt können Sie ein Gateway
# auf Ihrer Bridge konfigurieren und es auf Ihren anderen Computern als
# neues Gateway einsetzen

/sbin/ifconfig br0 62.3.3.26 netmask 255.255.255.248 broadcast 62.3.3.31

# Ich habe diese internen IPs für mein NAT benutzt
ip addr add 192.168.0.1/24 dev br0
/sbin/route add default gw 62.3.3.25
```

Eine Bridge mit Firewall-Fähigkeiten

Eine zweite denkbare Konfiguration ist ein System, das als transparente Firewall für ein LAN mit einer öffentlichen IP-Adresse aufgesetzt ist.

```
Internet ---- Router (62.3.3.25) ---- Bridge (62.3.3.26)
                                          |
                                          |
                                          |---- WWW-Server (62.3.3.28 gw 62.3.3.25)
                                          |
                                          |
                                          |---- Mail-Server (62.3.3.27 gw 62.3.3.25)
```

Die folgenden Befehle zeigen, wie diese Bridge konfiguriert werden kann:

```
# Erstellen der Schnittstelle br0
/usr/sbin/brctl addbr br0

# Hinzufügen der Ethernet-Schnittstelle, welche die Bridge benutzen soll
/usr/sbin/brctl addif br0 eth0
/usr/sbin/brctl addif br0 eth1
```

```
# Starten der Ethernet-Schnittstelle
/sbin/ifconfig eth0 0.0.0.0
/sbin/ifconfig eth1 0.0.0.0

# Konfigurieren der Ethernet-Bridge
# Die Bridge wird korrekt und unsichtbar (transparente Firewall) sein.
# In einem traceroute ist sie versteckt, und Sie behalten Ihr echtes
# Gateway auf Ihren anderen Computern. Jetzt können Sie ein Gateway
# auf Ihrer Bridge konfigurieren und es auf Ihren anderen Computern als
# neues Gateway einsetzen

/sbin/ifconfig br0 62.3.3.26 netmask 255.255.255.248 broadcast 62.3.3.31
```

Wenn Sie mit **traceroute** die Route des Linux-Mail-Servers verfolgen, sehen Sie die Bridge nicht. Wenn Sie mit **ssh** auf die Bridge zugreifen wollen, müssen Sie ein Gateway haben oder erst auf einen anderen Server wie den »Mail Server« zugreifen, um dann über die interne Netzwerkkarte auf die Bridge zuzugreifen.

Grundlegende Iptables-Regeln

Dies ist ein Beispiel für grundlegende Regeln, die für beide Einstellungen benutzt werden können:

Beispiel B.1. Grundlegende Iptables-Regeln

```
iptables -F FORWARD
iptables -P FORWARD DROP
iptables -A FORWARD -s 0.0.0.0/0.0.0.0 -d 0.0.0.0/0.0.0.0 -m state --state INVALID
iptables -A FORWARD -m state --state ESTABLISHED,RELATED -j ACCEPT

# Zwei lustige Regeln, aber nicht bei klassischen Iptables. Sorry ...
# Limit ICMP
# iptables -A FORWARD -p icmp -m limit --limit 4/s -j ACCEPT
# Übereinstimmende Strings, eine gute, einfache Methode, um Viren sehr
# schnell abzublocken
# iptables -I FORWARD -j DROP -p tcp -s 0.0.0.0/0 -m string --string "cmd.exe"

# Abblocken aller MySQL-Verbindungen, nur um sicher zu gehen
iptables -A FORWARD -p tcp -s 0/0 -d 62.3.3.0/24 --dport 3306 -j DROP

# Regeln für den Linux Mail-Server

# Erlaube FTP-DATA (20), FTP (21), SSH (22)
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.27/32 --dport 20:22 -j ACCEPT

#Dem Mail-Server erlauben, sich mit der Außenwelt zu verbinden
# Beachten Sie: Dies ist *nicht* für die vorherigen Verbindungen
# notwendig (erinnern Sie sich: stateful filtering) und könnte entfernt
# werden:
iptables -A FORWARD -p tcp -s 62.3.3.27/32 -d 0/0 -j ACCEPT

# Regeln für den WWW-Server

# AErlaube HTTP ( 80 ) Verbindungen mit dem WWW-Server
```

```
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.28/32 --dport 80 -j ACCEPT

# Erlaube HTTPS ( 443 ) Verbindungen mit dem WWW-Server
iptables -A FORWARD -p tcp -s 0.0.0.0/0 -d 62.3.3.28/32 --dport 443 -j ACCEPT

# Dem WWW-Server erlauben, sich mit der Außenwelt zu verbinden
# Beachten Sie: Dies ist *nicht* für die vorherigen Verbindungen
# notwendig (erinnern Sie sich: stateful filtering) und könnte entfernt
# werden:
iptables -A FORWARD -p tcp -s 62.3.3.28/32 -d 0/0 -j ACCEPT
```

Beispielskript, um die Standard-Installation von Bind zu ändern

Dieses Skript automatisiert den Vorgang, die Standardinstallation des Name-Servers **bind** in der Version 8 zu ändern, so dass er *nicht* als Root läuft. Hinweis: Bei **bind** in der Version 9 in Debian ist dies standardmäßig so.[3] Diese Version ist demnach der Version 8 von **bind** vorzuziehen.

Dieses Skript ist hier aus historischen Gründen aufgeführt und soll zeigen, wie man diese Art von Veränderungen systemweit automatisieren kann. Das Skript wird den Benutzer und die Gruppe für den Name-Server erstellen und `/etc/default/bind` und `/etc/init.d/bind` so ändern, dass das Programm unter diesem Benutzer läuft. Benutzen Sie es äußerst vorsichtig, da es nicht eingehend getestet wurde.

Sie können die Benutzer auch von Hand erstellen und dann den Patch für das Standard-Init.d-Skript verwenden, der im http://bugs.debian.org/cgi-bin/bugreport.cgi?bug=157245 enthalten ist.

```
#!/bin/sh
# Change the default Debian bind v8 configuration to have it run
# with a non-root user and group.
#
# DO NOT USER this with version 9, use debconf for configure this instead
#
# WARN: This script has not been tested thoroughly, please
# verify the changes made to the INITD script

# (c) 2002 Javier Fernández-Sanguino Peña
#
#    This program is free software; you can redistribute it and/or modify
#    it under the terms of the GNU General Public License as published by
#    the Free Software Foundation; either version 1, or (at your option)
#    any later version.
#
#    This program is distributed in the hope that it will be useful,
#    but WITHOUT ANY WARRANTY; without even the implied warranty of
#    MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
#    GNU General Public License for more details.
#
#     Please see the file `COPYING' for the complete copyright notice.
#
```

[3] Ab der Version 9.2.1-5, also seit der Veröffentlichung von *Sarge*.

```
restore() {
# Just in case, restore the system if the changes fail
  echo "WARN: Restoring to the previous setup since I'm unable to properly chang
  echo "WARN: Please check the $INITDERR script."
  mv $INITD $INITDERR
  cp $INITDBAK $INITD
}


USER=named
GROUP=named
INITD=/etc/init.d/bind
DEFAULT=/etc/default/bind
INITDBAK=$INITD.preuserchange
INITDERR=$INITD.changeerror
AWKS="awk ' /\/usr\/sbin\/ndc reload/ { print \"stop; sleep 2; start;\"; noprint

[ `id -u` -ne 0 ] && {
  echo "This program must be run by the root user"
  exit 1
}

RUNUSER=`ps eo user,fname |grep named |cut -f 1 -d " "`

if [ "$RUNUSER" = "$USER" ]
then
  echo "WARN: The name server running daemon is already running as $USER"
  echo "ERR:  This script will not do any changes to your setup."
  exit 1
fi
if [ ! -f "$INITD" ]
then
  echo "ERR:  This system does not have $INITD (which this script tries to chang
  RUNNING=`ps eo fname |grep named`
  [ -z "$RUNNING" ] && \
    echo "ERR:  In fact the name server daemon is not even running (is it instal
  echo "ERR:  No changes will be made to your system"
  exit 1
fi

# Check if there are options already setup
if [ -e "$DEFAULT" ]
then
  if grep -q ^OPTIONS $DEFAULT; then
    echo "ERR: The $DEFAULT file already has options set."
    echo "ERR:  No changes will be made to your system"
  fi
fi

# Check if named group exists
if [ -z "`grep $GROUP /etc/group`" ]
then
  echo "Creating group $GROUP:"
  addgroup $GROUP
```

```
  else
    echo "WARN: Group $GROUP already exists. Will not create it"
  fi
  # Same for the user
  if [ -z "`grep $USER /etc/passwd`" ]
  then
    echo "Creating user $USER:"
    adduser --system --home /home/$USER \
      --no-create-home --ingroup $GROUP \
      --disabled-password --disabled-login $USER
  else
    echo "WARN: The user $USER already exists. Will not create it"
  fi

  # Change the init.d script

  # First make a backup (check that there is not already
  # one there first)
  if [ ! -f $INITDBAK ]
  then
    cp $INITD $INITDBAK
  fi

  # Then use it to change it
  cat $INITDBAK |
  eval $AWKS > $INITD

  # Now put the options in the /etc/default/bind file:
  cat >>$DEFAULT <<EOF
# Make bind run with the user we defined
OPTIONS="-u $USER -g $GROUP"
EOF

  echo "WARN: The script $INITD has been changed, trying to test the changes."
  echo "Restarting the named daemon (check for errors here)."

  $INITD restart
  if [ $? -ne 0 ]
  then
    echo "ERR:  Failed to restart the daemon."
    restore
    exit 1
  fi

  RUNNING=`ps eo fname |grep named`
  if [ -z "$RUNNING" ]
  then
    echo "ERR:  Named is not running, probably due to a problem with the changes."
    restore
    exit 1
  fi

  # Check if it's running as expected
  RUNUSER=`ps eo user,fname |grep named |cut -f 1 -d " "`
```

```
if [ "$RUNUSER" = "$USER" ]
then
  echo "All has gone well, named seems to be running now as $USER."
else
  echo "ERR:  The script failed to automatically change the system."
  echo "ERR:  Named is currently running as $RUNUSER."
  restore
  exit 1
fi

exit 0
```

Das obige Skript wird, wenn es auf Woodys (Debian 3.0) **bind** (Version 8) angewendet wird, die initd-Datei verändern, nachdem der Benutzer und die Gruppe »named« erstellt wurde.

Schutz der Sicherheitsaktualisierung durch eine Firewall

Nach einer Standard-Installation könnten immer noch Sicherheitslücken auf dem System vorhanden sein. Falls Sie die Aktualisierungen für die verwundbaren Pakete nicht auf einem anderen System herunterladen können (oder security.debian.org zu lokalen Zwecken spiegeln können), müssen Sie sich mit dem Internet verbinden, um die Pakete herunterzuladen.

Wenn Sie sich jedoch mit dem Internet verbinden, setzen Sie Ihr System einer Gefahr aus. Wenn einer Ihrer lokalen Dienste angreifbar ist, könnten Sie kompromittiert sein, noch bevor die Aktualisierung beendet ist! Sie mögen dies paranoid finden, aber eine Analyse vom http://www.honeynet.org zeigt tatsächlich, dass ein System in weniger als drei Tagen kompromittiert werden kann, sogar wenn das System gar nicht der Öffentlichkeit bekannt ist (d.h. nicht in DNS-Einträgen auftaucht).

Wenn Sie eine Aktualisierung Ihres Systems durchführen, das nicht von einem externen System (z.B. einer Firewall) geschützt ist, können Sie trotzdem eine lokale Firewall so konfigurieren, dass Sie nur die Sicherheitsaktualisierung selbst erlaubt. Das Beispiel unten zeigt, wie die lokale Firewall aufgesetzt werden muss, damit nur Verbindungen zu security.debian.org erlaubt werden, während der Rest protokolliert wird.

Im nachfolgenden Beispiel wird ein strenges Regelwerk für eine Firewall dargestellt. Führen Sie diese Befehle auf einer lokalen Konsole aus (und nicht auf einer entfernten), um das Risiko zu verringern, sich aus Ihrem System auszusperren.

```
# iptables -F
# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
# iptables -A OUTPUT -d security.debian.org --dport 80 -j ACCEPT
# iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
```

```
# iptables -A INPUT -p icmp -j ACCEPT
# iptables -A INPUT -j LOG
# iptables -A OUTPUT -j LOG
# iptables -P INPUT DROP
# iptables -P FORWARD DROP
# iptables -P OUTPUT DROP
# iptables -L
Chain INPUT (policy DROP)
target     prot opt source               destination
ACCEPT     all  --  0.0.0.0/0            0.0.0.0/0          state RELATED,ESTABL
ACCEPT     icmp --  0.0.0.0/0            0.0.0.0/0
LOG        all  --  anywhere             anywhere           LOG level warning

Chain FORWARD (policy DROP)
target     prot opt source               destination

Chain OUTPUT (policy DROP)
target     prot opt source               destination
ACCEPT     80   --  anywhere             security.debian.org
LOG        all  --  anywhere             anywhere           LOG level warning
```

Hinweis: Es ist die vorzugswürdige Verfahrensweise, die Policy-Regel *DROP* für die Input-Kette zu verwenden. Seien Sie aber *äußerst* vorsichtig, wenn Sie dies bei einer entfernten Verbindung unternehmen. Wenn Sie das Regelwerk Ihrer Firewall aus der Ferne testen, ist es am besten, wenn Sie ein Skript mit dem Regelwerk laufen lassen (anstatt jede Regel Zeile für Zeile von der Befehlszeile aus einzugeben) und sich vorsorglich eine Hintertür[4]

Selbstverständlich müssen Sie alle Hintertüren abschalten, ehe Sie Ihr System in Betrieb nehmen. Offen halten, so dass Sie wieder Zugriff auf Ihr System bekommen, wenn Sie einen Fehler gemacht haben. Auf diese Weise müssen Sie sich nicht auf den Weg zum entfernten Rechner machen, um die Firewall-Regel, mit der Sie sich ausgeschlossen haben, zu korrigieren.

FIXME: This needs DNS to be working properly since it is required for security.debian.org to work. You can add security.debian.org to /etc/hosts but now it is a CNAME to several hosts (there is more than one security mirror)

FIXME: this will only work with HTTP URLs since ftp might need the ip_conntrack_ftp module, or use passive mode.

[4] Wie z.B. *knockd*. Alternativ dazu können Sie auch eine separate Konsole öffnen und das System nachfragen lassen, ob sich jemand auf der Gegenseite befindet. Wenn keine Eingabe erfolgt, werden die Firewall-Regeln zurückgesetzt. Ein Beispiel dafür ist:

```
#!/bin/bash

while true; do
    read -n 1 -p "Are you there? " -t 30 ayt
    if [ -z "$ayt" ] ; then
        break
    fi
done

# Reset the firewall chain, user is not available
echo
echo "Resetting firewall chain!"
iptables -F
iptables -P INPUT ACCEPT
iptables -P FORWARD ACCEPT
iptables -P OUTPUT ACCEPT
exit 1
```

Chroot-Umgebung für SSH

Es ist eine schwere Aufgabe, eine eingeschränkte Umgebung für **SSH** zu erstellen. Das liegt zum einen an seinen Abhängigkeiten und zum anderen daran, dass **SSH** im Gegensatz zu anderen Servern den Benutzern eine entfernte Shell zur Verfügung stellt. Daher müssen Sie sich überlegen, welche Programme Benutzer in der Umgebung verwenden sollen.

Sie haben zwei Möglichkeiten, eine beschränkte entfernte Shell einzurichten:

- die SSH-Benutzer in ein Chroot-Gefängnis einsperren: Dazu müssen Sie den SSH-Daemon so konfigurieren, dass er Benutzer nach der Authentifizierung in ein Chroot-Gefängnis einsperrt, bevor sie eine Shell bekommen. Jeder Benutzer kann seine eigene Umgebung haben.

- den SSH-Server in ein Chroot-Gefängnis einsperren: Wenn die SSH-Anwendung sich selbst in einer Chroot-Umgebung befindet, sind auch alle Benutzer in diese Umgebung eingesperrt.

Die erste Möglichkeit hat den Vorteil, dass es möglich ist, sowohl unbeschränkte als auch beschränkte Benutzer zu haben. Falls Sie keine Setuid-Anwendungen in der Chroot-Umgebung zur Verfügung stellen, wird es schwieriger, aus dem Gefängnis auszubrechen. Allerdings müssen Sie gegebenenfalls Chroot-Umgebungen für jeden Benutzer einzeln einrichten. Außerdem ist die Konfiguration schwieriger, da es Zusammenarbeit mit dem SSH-Server erfordert. Die zweite Möglichkeit ist leichter zu verwirklichen und schützt vor dem Ausnutzen eines Exploits des SSH-Servers, da auch dieser im Chroot-Gefängnis ist. Jedoch müssen alle Benutzer die gleiche Chroot-Umgebung verwenden. Verschiedene Umgebungen für verschiedene Benutzer sind nicht möglich.

SSH-Benutzer in ein Chroot-Gefängnis einsperren

Sie können den SSH-Server so einrichten, dass er bestimmte Benutzer in eine Chroot-Umgebung einsperrt, so dass sie eine Shell mit nur einer beschränkten Anzahl von Anwendungen zur Verfügung haben.

Einsatz von libpam-chroot

Der wahrscheinlich leichteste Weg ist, das Paket libpam-chroot, das in Debian vorhanden ist, zu verwenden. Wenn Sie es installiert haben, müssen Sie:

- `/etc/pam.d/ssh` verändern, um dieses PAM-Modul zu verwenden. Fügen Sie dazu als letzte Zeile Folgendes ein[5]:

  ```
  session    required   pam_chroot.so
  ```

- eine passende Chroot-Umgebung für die Benutzer einrichten. Sie können versuchen, die Skripte unter `/usr/share/doc/libpam-chroot/examples/` zu verwenden, das Programm makejail benutzen[6] oder eine minimale Debian-Umgebung mit debootstrap aufsetzen. Stellen Sie sicher, dass die Umgebung die notwendigen Geräte enthält.[7]

[5] Sie können die Option *debug* verwenden. Damit wird der Fortschritt des Moduls unter *authpriv.notice* protokolliert.

[6] Mit folgendem Python-Aufruf können Sie eine sehr eingeschränkte Bash-Umgebung für makejail erstellen. Erstellen Sie das Verzeichnis `/var/chroots/users/foo` und eine Datei mit dem Namen `bash.py` und folgendem Inhalt:

```
chroot="/var/chroots/users/foo"
cleanJailFirst=1
testCommandsInsideJail=["bash ls"]
```

Führen Sie dann *makejail bash.py* aus, um eine Benutzer-Umgebung unter `/var/chroots/users/foo` zu erstellen. So testen Sie die Umgebung:

- `/etc/security/chroot.conf` bearbeiten, damit die ausgewählten Benutzer in das Verzeichnis eingesperrt werden, das Sie zuvor eingerichtet haben. Sie sollten getrennte Verzeichnisse für verschiedene Benutzer haben, damit sie weder das ganze System noch sich gegenseitig sehen können.

- SSH konfigurieren: Je nach der eingesetzten OpenSSH-Version funktioniert die Chroot-Umgebung sofort. Seit 3.6.1p2 wird die Funktion *do_pam_session()* aufgerufen, nachdem sshd seine Rechte abgelegt hat. Da chroot() Root-Rechte benötigt, wird es mit Rechtetrennung nicht funktionieren. Allerdings wurde in neueren OpenSSH-Versionen der PAM-Code verändert, so dass do_pam_session vor dem Ablegen der Rechte aufgerufen wird. Daher funktioniert es auch mit aktivierter Rechtetrennung. Falls Sie sie abschalten müssen, müssen Sie `/etc/ssh/sshd_config` so verändern:

  ```
  UsePrivilegeSeparation no
  ```

  Beachten Sie, dass das die Sicherheit Ihres Systems verringern wird, da dann der OpenSSH-Server als *Root* läuft. Das bedeutet, dass wenn eine Angriffsmöglichkeit aus der Ferne gegen OpenSSH entdeckt wird, ein Angreifer *Root*-Rechte anstatt nur *Sshd*-Rechte erlangen wird und somit das gesamte System kompromittiert.[8]

Wenn Sie die *Rechtetrennung* nicht deaktivieren, brauchen Sie im Chroot-Gefängnis `/etc/passwd`, welches die Benutzer-UID enthält, damit die *Rechtetrennung* funktioniert.

Wenn Sie die Option *Rechtetrennung* auf *yes* gesetzt haben und Ihre Version von OpenSSH nicht richtig läuft, müssen Sie sie abschalten. Wenn Sie das unterlassen, werden Benutzer, die sich mit Ihrem Server verbinden wollen und von diesem Modul in eine Chroot-Umgebung eingesperrt werden sollen, Folgendes zu sehen bekommen:

```
$ ssh -l user server
user@server's password:
Connection to server closed by remote host.
Connection to server closed.
```

Dies geschieht, weil der SSH-Daemon, der als »sshd« läuft, nicht den Systemaufruf chroot() ausführen kann. Um die Rechtetrennung abzuschalten, müssen Sie die Konfigurationsdatei `/etc/ssh/sshd_config` wie oben beschrieben verändern.

Beachten Sie, dass, wenn Folgendes fehlt, sich die Benutzer nicht in der Chroot-Umgebung anmelden können:

- Das Dateisystem `/proc` muss in der Chroot-Umgebung des Benutzers gemountet sein.

- Die notwendigen Geräte unter `/dev/pts/` müssen vorliegen. Falls diese Dateien automatisch vom Kernel erstellt werden, müssen Sie sie von Hand unter `/dev/` in der Chroot-Umgebung erstellen.

- Das Home-Verzeichnis des Benutzers muss in der Chroot-Umgebung existieren. Ansonsten wird der SSH-Daemon nicht fortfahren.

```
   # chroot /var/chroots/users/foo/ ls
  bin  dev  etc  lib  proc  sbin  usr
```

[7] Unter Umständen benötigen Sie die Geräte `/dev/ptmx` und `/dev/pty*` und das Unterverzeichnis `/dev/pts/`. Es sollte ausreichen, MAKEDEV im `/dev`-Verzeichnis der Chroot-Umgebung auszuführen, um sie zu erstellen, falls sie nicht existieren. Wenn Sie einen Kernel einsetzen, der die Gerätedateien dynamisch erstellt (Version 2.6), müssen Sie die Dateien /dev/pts/ selbst erstellen und mit den passenden Rechten ausstatten.

[8] Wenn Sie einen Kernel verwenden, der Mandatory-Access-Control (RSBAC/SElinux) unterstützt, müssen Sie die Konfiguration nicht ändern, wenn Sie dem *Sshd*-Benutzer die notwendigen Rechte einräumen, um den Systemaufruf chroot() ausführen zu können.

Sie können diese Probleme mit dem Schlüsselwort *debug* in der PAM-Konfiguration `/etc/pam.d/ssh` debuggen. Falls Sie auf Probleme stoßen, kann es sich als nützlich erweisen, auch den Debugging-Modus des SSH-Clients zu aktivieren.

Hinweis: Diese Informationen sind auch in `/usr/share/doc/libpam-chroot/README.Debian.gz` enthalten (und vielleicht aktueller). Bitte überprüfen Sie, ob dort aktualisierte Informationen vorhanden sind, bevor Sie die oben aufgezeigten Schritte ausführen.

Patchen des ssh-Servers

Debians **sshd** gestattet nicht, die Bewegungen eines Benutzer durch den Server zu beschränken, da er keine **Chroot**-Funktionalität besitzt. Diese ist im Gegensatz dazu Bestandteil des kommerziellen Programms **sshd2** (es verwendet »ChrootGroups« oder »ChrootUsers«, siehe sshd2_config(5)). Allerdings gibt es einen Patch, der **sshd** um diese Funktion erweitert. Den Patch erhalten Sie unter http://chrootssh.sourceforge.net (wurde in http://bugs.debian.org/139047 nachgefragt). Der Patch könnte Bestandteil von zukünftigen Veröffentlichungen des OpenSSH-Pakets werden. Emmanuel Lacour bietet **ssh**-Pakete als Debs mit diesen Fähigkeiten für *Sarge* an. Sie sind unter http://debian.home-dn.net/sarge/ssh/ verfügbar. Beachten Sie aber, dass sie nicht aktuell sein müssen, daher wird empfohlen, den Weg der Kompilierung zu gehen.

Nachdem Sie den Patch angewendet haben, müssen Sie `/etc/passwd` anpassen und darin das Home-Verzeichnis der Benutzer ändern (mit dem speziellen `/./` Kürzel).

```
joebenutzer:x:1099:1099:Joe Zufaelliger Benutzer:/home/joe/./:/bin/bash
```

Dies wird *sowohl* den Fernzugriff auf die Shell *als auch* Fernkopien über den **ssh**-Kanal einschränken.

Gehen Sie sicher, dass Sie alle benötigten Programme und Bibliotheken in den **Chroot**-Pfaden der Benutzer haben. Diese Dateien sollten Root als Eigentümer haben, um Manipulationen durch den Benutzer zu verhindern (zum Beispiel um das **chroot**-Gefängnis zu verlassen). Ein Beispiel könnte so aussehen:

```
./bin:
total 660
drwxr-xr-x    2 root     root         4096 Mar 18 13:36 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-r-xr-xr-x    1 root     root       531160 Feb  6 22:36 bash
-r-xr-xr-x    1 root     root        43916 Nov 29 13:19 ls
-r-xr-xr-x    1 root     root        16684 Nov 29 13:19 mkdir
-rwxr-xr-x    1 root     root        23960 Mar 18 13:36 more
-r-xr-xr-x    1 root     root         9916 Jul 26  2001 pwd
-r-xr-xr-x    1 root     root        24780 Nov 29 13:19 rm
lrwxrwxrwx    1 root     root            4 Mar 30 16:29 sh -> bash


./etc:
total 24
drwxr-xr-x    2 root     root         4096 Mar 15 16:13 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-rw-r--r--    1 root     root           54 Mar 15 13:23 group
-rw-r--r--    1 root     root          428 Mar 15 15:56 hosts
-rw-r--r--    1 root     root           44 Mar 15 15:53 passwd
-rw-r--r--    1 root     root           52 Mar 15 13:23 shells


./lib:
```

```
total 1848
drwxr-xr-x    2 root     root         4096 Mar 18 13:37 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
-rwxr-xr-x    1 root     root        92511 Mar 15 12:49 ld-linux.so.2
-rwxr-xr-x    1 root     root      1170812 Mar 15 12:49 libc.so.6
-rw-r--r--    1 root     root        20900 Mar 15 13:01 libcrypt.so.1
-rw-r--r--    1 root     root         9436 Mar 15 12:49 libdl.so.2
-rw-r--r--    1 root     root       248132 Mar 15 12:48 libncurses.so.5
-rw-r--r--    1 root     root        71332 Mar 15 13:00 libnsl.so.1
-rw-r--r--    1 root     root        34144 Mar 15 16:10
libnss_files.so.2
-rw-r--r--    1 root     root        29420 Mar 15 12:57 libpam.so.0
-rw-r--r--    1 root     root       105498 Mar 15 12:51 libpthread.so.0
-rw-r--r--    1 root     root        25596 Mar 15 12:51 librt.so.1
-rw-r--r--    1 root     root         7760 Mar 15 12:59 libutil.so.1
-rw-r--r--    1 root     root        24328 Mar 15 12:57 libwrap.so.0


./usr:
total 16
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 .
drwxr-xr-x    8 guest    guest        4096 Mar 15 16:53 ..
drwxr-xr-x    2 root     root         4096 Mar 15 15:55 bin
drwxr-xr-x    2 root     root         4096 Mar 15 15:37 lib


./usr/bin:
total 340
drwxr-xr-x    2 root     root         4096 Mar 15 15:55 .
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 ..
-rwxr-xr-x    1 root     root        10332 Mar 15 15:55 env
-rwxr-xr-x    1 root     root        13052 Mar 15 13:13 id
-r-xr-xr-x    1 root     root        25432 Mar 15 12:40 scp
-rwxr-xr-x    1 root     root        43768 Mar 15 15:15 sftp
-r-sr-xr-x    1 root     root       218456 Mar 15 12:40 ssh
-rwxr-xr-x    1 root     root         9692 Mar 15 13:17 tty


./usr/lib:
total 852
drwxr-xr-x    2 root     root         4096 Mar 15 15:37 .
drwxr-xr-x    4 root     root         4096 Mar 15 13:00 ..
-rw-r--r--    1 root     root       771088 Mar 15 13:01
libcrypto.so.0.9.6
-rw-r--r--    1 root     root        54548 Mar 15 13:00 libz.so.1
-rwxr-xr-x    1 root     root        23096 Mar 15 15:37 sftp-server
```

Einsperren des SSH-Servers in einem Chroot-Gefängnis

Wenn Sie eine Chroot-Umgebung erstellen, welche die Dateien des SSH-Servers enthält, z.B. unter `/var/chroot/ssh`, sollten Sie den im **chroot**-Gefängnis eingesperrten **ssh**-Server mit diesem Befehl starten:

```
# chroot /var/chroot/ssh /sbin/sshd -f /etc/sshd_config
```

Das führt dazu, dass der **sshd**-Daemon innerhalb des Chroot-Gefängnisses gestartet wird. Dazu müssen Sie zunächst dafür sorgen, dass das Verzeichnis `/var/chroot/ssh` den SSH-Server und die Werkzeuge enthält, die Benutzer benötigen, die mit dem Server verbunden sind. Wenn Sie das vorhaben, sollten Sie sicherstellen, dass OpenSSH *Rechtetrennung (Privilege Separation)* einsetzt (was standardmäßig so ist). Dazu muss in der Konfigurationsdatei `/etc/ssh/sshd_config` folgende Zeile enthalten sein:

```
UsePrivilegeSeparation yes
```

Dadurch wird der entfernte Daemon so wenig Dinge wie möglich als Root ausführen. Wenn er also einen Fehler enthalten sollte, kann damit nicht aus dem Chroot-Gefängnis ausgebrochen werden. Beachten Sie, dass, anders als wenn Sie eine Chroot-Umgebung für jeden Benutzer einzeln einrichten, in diesem Fall der SSH-Daemon im selben Chroot-Gefängnis wie die Benutzer läuft. Es gibt also mindestens einen Prozess in der Chroot-Umgebung, der als Root läuft. Mit ihm ist es möglich, aus dem Chroot-Gefängnis auszubrechen.

Beachten Sie auch, dass SSH nur funktioniert, wenn die Partition, auf der die Chroot-Umgebung eingerichtet wurde, nicht mit der Option *nodev* gemountet wurde. Wenn Sie diese Option verwenden, bekommen Sie folgende Fehlermeldung: *PRNG is not seeded*, weil `/dev/urandom` nicht in der Chroot-Umgebung funktioniert.

Einrichten eines minimalen Systems (der wirklich leichte Weg)

Sie können mit debootstrap eine minimale Umgebung einrichten, die ausschließlich den SSH-Server enthält. Dafür müssen Sie nur eine Chroot-Umgebung einrichten, wie es im http://www.debian.org/doc/manuals/reference/ch09#_chroot_system beschrieben wird. Diese Vorgehensweise ist idiotensicher (Sie werden alle für die Chroot-Umgebung notwendigen Bestandteile erhalten), aber dies geht auf Kosten von Plattenspeicher. Eine minimale Installation von Debian benötigt einige hundert Megabyte. Dieses minimale System könnte auch Setuid-Dateien enthalten, mit denen ein Benutzer aus dem Chroot-Gefängnis ausbrechen könnte, wenn sie eine Rechteerweiterung zulassen.

Automatisches Erstellen der Umgebung (der leichte Weg)

Mit dem Paket makejail können Sie leicht eine eingeschränkte Umgebung erstellen, da es automatisch den Trace des Server-Daemons verfolgt (mit **strace**) und dafür sorgt, dass er in der eingeschränkten Umgebung läuft.

Der Vorteil von Programmen, die automatisch die **chroot**-Umgebung einrichten, liegt darin, dass sie im Stande sind, Pakete in die **chroot**-Umgebung zu kopieren (und verfolgen sogar die Abhängigkeiten der Pakete, um sicherzustellen, dass sie vollständig sind). Dadurch wird das Bereitstellen von Anwendungen für Benutzer leichter.

Um ein Chroot-Gefängnis aus den von **makejail** zur Verfügung gestellten Beispielen einzurichten, müssen Sie `/var/chroot/sshd` erstellen und folgenden Befehl ausführen:

```
# makejail /usr/share/doc/makejail/examples/sshd.py
```

Dies wird eine Chroot-Umgebung im Verzeichnis `/var/chroot/sshd` erstellen. Beachten Sie, dass diese Chroot-Umgebung nicht voll funktionstüchtig ist, bis Sie:

- Das Dateisystem *procfs* in `/var/chroot/sshd/proc` eingehängt haben. **Makejail** wird es für Sie einhängen. Aber nach einem Neustart werden Sie es erneut einhängen müssen:

```
# mount -t proc proc /var/chroot/sshd/proc
```

Es kann auch automatisch eingebunden werden. Dazu müssen Sie `/etc/fstab` bearbeiten und folgende Zeile eintragen:

```
proc-ssh /var/chroot/sshd/proc  proc none 0 0
```

- Syslog auf das Geräte `/dev/log` in der Chroot-Umgebung horchen lassen. Dazu müssen Sie `/etc/default/syslogd` ändern und *-a /var/chroot/sshd/dev/log* zur Definition der Variablen *SYSLOGD* hinzufügen.

Sehen Sie sich die Beispielsdatei an, um herauszufinden, welche Änderungen an der Umgebung vorgenommen werden müssen. Einige diese Änderungen können nicht automatisch vorgenommen werden, wie z.B. das Kopieren des Home-Verzeichnisses eines Benutzers. Außerdem sollten Sie die Gefährdung von sensiblen Informationen begrenzen, indem Sie nur die Daten bestimmter Benutzer aus den Dateien `/etc/shadow` und `/etc/group` kopieren. Beachten Sie, dass, falls Sie Rechtetrennung verwenden, der Benutzer *sshd* in diesen Dateien vorhanden sein muss.

Die folgende Beispielumgebung wurde (ein wenig) unter Debian 3.0 getestet. Sie basiert auf der Konfigurationsdatei, die mit dem Paket geliefert wird, und beinhaltet das Paket fileutils.

```
.
|-- bin
|   |-- ash
|   |-- bash
|   |-- chgrp
|   |-- chmod
|   |-- chown
|   |-- cp
|   |-- csh -> /etc/alternatives/csh
|   |-- dd
|   |-- df
|   |-- dir
|   |-- fdflush
|   |-- ksh
|   |-- ln
|   |-- ls
|   |-- mkdir
|   |-- mknod
|   |-- mv
|   |-- rbash -> bash
|   |-- rm
|   |-- rmdir
|   |-- sh -> bash
|   |-- sync
|   |-- tcsh
|   |-- touch
|   |-- vdir
|   |-- zsh -> /etc/alternatives/zsh
|   `-- zsh4
|-- dev
|   |-- null
|   |-- ptmx
```

```
|   |-- pts
|   |-- ptya0
(...)
|   |-- tty
|   |-- tty0
(...)
|   `-- urandom
|-- etc
|   |-- alternatives
|   |   |-- csh -> /bin/tcsh
|   |   `-- zsh -> /bin/zsh4
|   |-- environment
|   |-- hosts
|   |-- hosts.allow
|   |-- hosts.deny
|   |-- ld.so.conf
|   |-- localtime -> /usr/share/zoneinfo/Europe/Madrid
|   |-- motd
|   |-- nsswitch.conf
|   |-- pam.conf
|   |-- pam.d
|   |   |-- other
|   |   `-- ssh
|   |-- passwd
|   |-- resolv.conf
|   |-- security
|   |   |-- access.conf
|   |   |-- chroot.conf
|   |   |-- group.conf
|   |   |-- limits.conf
|   |   |-- pam_env.conf
|   |   `-- time.conf
|   |-- shadow
|   |-- shells
|   `-- ssh
|       |-- moduli
|       |-- ssh_host_dsa_key
|       |-- ssh_host_dsa_key.pub
|       |-- ssh_host_rsa_key
|       |-- ssh_host_rsa_key.pub
|       `-- sshd_config
|-- home
|   `-- userX
|-- lib
|   |-- ld-2.2.5.so
|   |-- ld-linux.so.2 -> ld-2.2.5.so
|   |-- libc-2.2.5.so
|   |-- libc.so.6 -> libc-2.2.5.so
|   |-- libcap.so.1 -> libcap.so.1.10
|   |-- libcap.so.1.10
|   |-- libcrypt-2.2.5.so
|   |-- libcrypt.so.1 -> libcrypt-2.2.5.so
|   |-- libdl-2.2.5.so
|   |-- libdl.so.2 -> libdl-2.2.5.so
```

```
|   |-- libm-2.2.5.so
|   |-- libm.so.6 -> libm-2.2.5.so
|   |-- libncurses.so.5 -> libncurses.so.5.2
|   |-- libncurses.so.5.2
|   |-- libnsl-2.2.5.so
|   |-- libnsl.so.1 -> libnsl-2.2.5.so
|   |-- libnss_compat-2.2.5.so
|   |-- libnss_compat.so.2 -> libnss_compat-2.2.5.so
|   |-- libnss_db-2.2.so
|   |-- libnss_db.so.2 -> libnss_db-2.2.so
|   |-- libnss_dns-2.2.5.so
|   |-- libnss_dns.so.2 -> libnss_dns-2.2.5.so
|   |-- libnss_files-2.2.5.so
|   |-- libnss_files.so.2 -> libnss_files-2.2.5.so
|   |-- libnss_hesiod-2.2.5.so
|   |-- libnss_hesiod.so.2 -> libnss_hesiod-2.2.5.so
|   |-- libnss_nis-2.2.5.so
|   |-- libnss_nis.so.2 -> libnss_nis-2.2.5.so
|   |-- libnss_nisplus-2.2.5.so
|   |-- libnss_nisplus.so.2 -> libnss_nisplus-2.2.5.so
|   |-- libpam.so.0 -> libpam.so.0.72
|   |-- libpam.so.0.72
|   |-- libpthread-0.9.so
|   |-- libpthread.so.0 -> libpthread-0.9.so
|   |-- libresolv-2.2.5.so
|   |-- libresolv.so.2 -> libresolv-2.2.5.so
|   |-- librt-2.2.5.so
|   |-- librt.so.1 -> librt-2.2.5.so
|   |-- libutil-2.2.5.so
|   |-- libutil.so.1 -> libutil-2.2.5.so
|   |-- libwrap.so.0 -> libwrap.so.0.7.6
|   |-- libwrap.so.0.7.6
|   `-- security
|       |-- pam_access.so
|       |-- pam_chroot.so
|       |-- pam_deny.so
|       |-- pam_env.so
|       |-- pam_filter.so
|       |-- pam_ftp.so
|       |-- pam_group.so
|       |-- pam_issue.so
|       |-- pam_lastlog.so
|       |-- pam_limits.so
|       |-- pam_listfile.so
|       |-- pam_mail.so
|       |-- pam_mkhomedir.so
|       |-- pam_motd.so
|       |-- pam_nologin.so
|       |-- pam_permit.so
|       |-- pam_rhosts_auth.so
|       |-- pam_rootok.so
|       |-- pam_securetty.so
|       |-- pam_shells.so
|       |-- pam_stress.so
```

```
|       |-- pam_tally.so
|       |-- pam_time.so
|       |-- pam_unix.so
|       |-- pam_unix_acct.so -> pam_unix.so
|       |-- pam_unix_auth.so -> pam_unix.so
|       |-- pam_unix_passwd.so -> pam_unix.so
|       |-- pam_unix_session.so -> pam_unix.so
|       |-- pam_userdb.so
|       |-- pam_warn.so
|       `-- pam_wheel.so
|-- sbin
|   `-- start-stop-daemon
|-- usr
|   |-- bin
|   |   |-- dircolors
|   |   |-- du
|   |   |-- install
|   |   |-- link
|   |   |-- mkfifo
|   |   |-- shred
|   |   |-- touch -> /bin/touch
|   |   `-- unlink
|   |-- lib
|   |   |-- libcrypto.so.0.9.6
|   |   |-- libdb3.so.3 -> libdb3.so.3.0.2
|   |   |-- libdb3.so.3.0.2
|   |   |-- libz.so.1 -> libz.so.1.1.4
|   |   `-- libz.so.1.1.4
|   |-- sbin
|   |   `-- sshd
|   `-- share
|       |-- locale
|       |   `-- es
|       |       |-- LC_MESSAGES
|       |       |   |-- fileutils.mo
|       |       |   |-- libc.mo
|       |       |   `-- sh-utils.mo
|       |       `-- LC_TIME -> LC_MESSAGES
|       `-- zoneinfo
|           `-- Europe
|               `-- Madrid
`-- var
    `-- run
        |-- sshd
        `-- sshd.pid

27 directories, 733 files
```

Bei Debian 3.1 müssen Sie sicherstellen, dass das Gefängnis auch die Dateien für PAM enthält. Falls es nicht schon **makejail** für Sie erledigt hat, müssen Sie folgende Dateien in die Chroot-Umgebung kopiert:

```
$ ls /etc/pam.d/common-*
/etc/pam.d/common-account  /etc/pam.d/common-password
```

```
/etc/pam.d/common-auth     /etc/pam.d/common-session
```

Die Chroot-Umgebung von Hand erstellen (der schwierige Weg)

Es ist möglich, eine Umgebung mit der Trial-and-Error-Methode zu erstellen. Dazu müssen Sie die Traces und die Protokolldateien des **sshd**-Servers überwachen, um die notwendigen Dateien herauszufinden. Die folgende Umgebung, die von José Luis Ledesma zur Verfügung gestellt wurde, ist eine beispielhafte Auflistung der Dateien in einer **chroot**-Umgebung für **ssh** unter Debian 3.0:

```
.:
total 36
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ./
drwxr-xr-x 11 root root 4096 Jun 3 13:43 ../
drwxr-xr-x 2 root root 4096 Jun 4 12:13 bin/
drwxr-xr-x 2 root root 4096 Jun 4 12:16 dev/
drwxr-xr-x 4 root root 4096 Jun 4 12:35 etc/
drwxr-xr-x 3 root root 4096 Jun 4 12:13 lib/
drwxr-xr-x 2 root root 4096 Jun 4 12:35 sbin/
drwxr-xr-x 2 root root 4096 Jun 4 12:32 tmp/
drwxr-xr-x 2 root root 4096 Jun 4 12:16 usr/
./bin:
total 8368
drwxr-xr-x 2 root root 4096 Jun 4 12:13 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rwxr-xr-x 1 root root 109855 Jun 3 13:45 a2p*
-rwxr-xr-x 1 root root 387764 Jun 3 13:45 bash*
-rwxr-xr-x 1 root root 36365 Jun 3 13:45 c2ph*
-rwxr-xr-x 1 root root 20629 Jun 3 13:45 dprofpp*
-rwxr-xr-x 1 root root 6956 Jun 3 13:46 env*
-rwxr-xr-x 1 root root 158116 Jun 3 13:45 fax2ps*
-rwxr-xr-x 1 root root 104008 Jun 3 13:45 faxalter*
-rwxr-xr-x 1 root root 89340 Jun 3 13:45 faxcover*
-rwxr-xr-x 1 root root 441584 Jun 3 13:45 faxmail*
-rwxr-xr-x 1 root root 96036 Jun 3 13:45 faxrm*
-rwxr-xr-x 1 root root 107000 Jun 3 13:45 faxstat*
-rwxr-xr-x 1 root root 77832 Jun 4 11:46 grep*
-rwxr-xr-x 1 root root 19597 Jun 3 13:45 h2ph*
-rwxr-xr-x 1 root root 46979 Jun 3 13:45 h2xs*
-rwxr-xr-x 1 root root 10420 Jun 3 13:46 id*
-rwxr-xr-x 1 root root 4528 Jun 3 13:46 ldd*
-rwxr-xr-x 1 root root 111386 Jun 4 11:46 less*
-r-xr-xr-x 1 root root 26168 Jun 3 13:45 login*
-rwxr-xr-x 1 root root 49164 Jun 3 13:45 ls*
-rwxr-xr-x 1 root root 11600 Jun 3 13:45 mkdir*
-rwxr-xr-x 1 root root 24780 Jun 3 13:45 more*
-rwxr-xr-x 1 root root 154980 Jun 3 13:45 pal2rgb*
-rwxr-xr-x 1 root root 27920 Jun 3 13:46 passwd*
-rwxr-xr-x 1 root root 4241 Jun 3 13:45 pl2pm*
-rwxr-xr-x 1 root root 2350 Jun 3 13:45 pod2html*
-rwxr-xr-x 1 root root 7875 Jun 3 13:45 pod2latex*
-rwxr-xr-x 1 root root 17587 Jun 3 13:45 pod2man*
-rwxr-xr-x 1 root root 6877 Jun 3 13:45 pod2text*
-rwxr-xr-x 1 root root 3300 Jun 3 13:45 pod2usage*
```

```
-rwxr-xr-x 1 root root 3341 Jun 3 13:45 podchecker*
-rwxr-xr-x 1 root root 2483 Jun 3 13:45 podselect*
-r-xr-xr-x 1 root root 82412 Jun 4 11:46 ps*
-rwxr-xr-x 1 root root 36365 Jun 3 13:45 pstruct*
-rwxr-xr-x 1 root root 7120 Jun 3 13:45 pwd*
-rwxr-xr-x 1 root root 179884 Jun 3 13:45 rgb2ycbcr*
-rwxr-xr-x 1 root root 20532 Jun 3 13:45 rm*
-rwxr-xr-x 1 root root 6720 Jun 4 10:15 rmdir*
-rwxr-xr-x 1 root root 14705 Jun 3 13:45 s2p*
-rwxr-xr-x 1 root root 28764 Jun 3 13:46 scp*
-rwxr-xr-x 1 root root 385000 Jun 3 13:45 sendfax*
-rwxr-xr-x 1 root root 67548 Jun 3 13:45 sendpage*
-rwxr-xr-x 1 root root 88632 Jun 3 13:46 sftp*
-rwxr-xr-x 1 root root 387764 Jun 3 13:45 sh*
-rws--x--x 1 root root 744500 Jun 3 13:46 slogin*
-rwxr-xr-x 1 root root 14523 Jun 3 13:46 splain*
-rws--x--x 1 root root 744500 Jun 3 13:46 ssh*
-rwxr-xr-x 1 root root 570960 Jun 3 13:46 ssh-add*
-rwxr-xr-x 1 root root 502952 Jun 3 13:46 ssh-agent*
-rwxr-xr-x 1 root root 575740 Jun 3 13:46 ssh-keygen*
-rwxr-xr-x 1 root root 383480 Jun 3 13:46 ssh-keyscan*
-rwxr-xr-x 1 root root 39 Jun 3 13:46 ssh_europa*
-rwxr-xr-x 1 root root 107252 Jun 4 10:14 strace*
-rwxr-xr-x 1 root root 8323 Jun 4 10:14 strace-graph*
-rwxr-xr-x 1 root root 158088 Jun 3 13:46 thumbnail*
-rwxr-xr-x 1 root root 6312 Jun 3 13:46 tty*
-rwxr-xr-x 1 root root 55904 Jun 4 11:46 useradd*
-rwxr-xr-x 1 root root 585656 Jun 4 11:47 vi*
-rwxr-xr-x 1 root root 6444 Jun 4 11:45 whoami*
./dev:
total 8
drwxr-xr-x 2 root root 4096 Jun 4 12:16 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
crw-r--r-- 1 root root 1, 9 Jun 3 13:43 urandom
./etc:
total 208
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rw------- 1 root root 0 Jun 4 11:46 .pwd.lock
-rw-r--r-- 1 root root 653 Jun 3 13:46 group
-rw-r--r-- 1 root root 242 Jun 4 11:33 host.conf
-rw-r--r-- 1 root root 857 Jun 4 12:04 hosts
-rw-r--r-- 1 root root 1050 Jun 4 11:29 ld.so.cache
-rw-r--r-- 1 root root 304 Jun 4 11:28 ld.so.conf
-rw-r--r-- 1 root root 235 Jun 4 11:27 ld.so.conf~
-rw-r--r-- 1 root root 88039 Jun 3 13:46 moduli
-rw-r--r-- 1 root root 1342 Jun 4 11:34 nsswitch.conf
drwxr-xr-x 2 root root 4096 Jun 4 12:02 pam.d/
-rw-r--r-- 1 root root 28 Jun 4 12:00 pam_smb.conf
-rw-r--r-- 1 root root 2520 Jun 4 11:57 passwd
-rw-r--r-- 1 root root 7228 Jun 3 13:48 profile
-rw-r--r-- 1 root root 1339 Jun 4 11:33 protocols
-rw-r--r-- 1 root root 274 Jun 4 11:44 resolv.conf
drwxr-xr-x 2 root root 4096 Jun 3 13:43 security/
```

```
-rw-r----- 1 root root 1178 Jun 4 11:51 shadow
-rw------- 1 root root 80 Jun 4 11:45 shadow-
-rw-r----- 1 root root 1178 Jun 4 11:48 shadow.old
-rw-r--r-- 1 root root 161 Jun 3 13:46 shells
-rw-r--r-- 1 root root 1144 Jun 3 13:46 ssh_config
-rw------- 1 root root 668 Jun 3 13:46 ssh_host_dsa_key
-rw-r--r-- 1 root root 602 Jun 3 13:46 ssh_host_dsa_key.pub
-rw------- 1 root root 527 Jun 3 13:46 ssh_host_key
-rw-r--r-- 1 root root 331 Jun 3 13:46 ssh_host_key.pub
-rw------- 1 root root 883 Jun 3 13:46 ssh_host_rsa_key
-rw-r--r-- 1 root root 222 Jun 3 13:46 ssh_host_rsa_key.pub
-rw-r--r-- 1 root root 2471 Jun 4 12:15 sshd_config
./etc/pam.d:
total 24
drwxr-xr-x 2 root root 4096 Jun 4 12:02 ./
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ../
lrwxrwxrwx 1 root root 4 Jun 4 12:02 other -> sshd
-rw-r--r-- 1 root root 318 Jun 3 13:46 passwd
-rw-r--r-- 1 root root 546 Jun 4 11:36 ssh
-rw-r--r-- 1 root root 479 Jun 4 12:02 sshd
-rw-r--r-- 1 root root 370 Jun 3 13:46 su
./etc/security:
total 32
drwxr-xr-x 2 root root 4096 Jun 3 13:43 ./
drwxr-xr-x 4 root root 4096 Jun 4 12:35 ../
-rw-r--r-- 1 root root 1971 Jun 3 13:46 access.conf
-rw-r--r-- 1 root root 184 Jun 3 13:46 chroot.conf
-rw-r--r-- 1 root root 2145 Jun 3 13:46 group.conf
-rw-r--r-- 1 root root 1356 Jun 3 13:46 limits.conf
-rw-r--r-- 1 root root 2858 Jun 3 13:46 pam_env.conf
-rw-r--r-- 1 root root 2154 Jun 3 13:46 time.conf
./lib:
total 8316
drwxr-xr-x 3 root root 4096 Jun 4 12:13 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rw-r--r-- 1 root root 1024 Jun 4 11:51 cracklib_dict.hwm
-rw-r--r-- 1 root root 214324 Jun 4 11:51 cracklib_dict.pwd
-rw-r--r-- 1 root root 11360 Jun 4 11:51 cracklib_dict.pwi
-rwxr-xr-x 1 root root 342427 Jun 3 13:46 ld-linux.so.2*
-rwxr-xr-x 1 root root 4061504 Jun 3 13:46 libc.so.6*
lrwxrwxrwx 1 root root 15 Jun 4 12:11 libcrack.so -> libcrack.so.2.7*
lrwxrwxrwx 1 root root 15 Jun 4 12:11 libcrack.so.2 -> libcrack.so.2.7*
-rwxr-xr-x 1 root root 33291 Jun 4 11:39 libcrack.so.2.7*
-rwxr-xr-x 1 root root 60988 Jun 3 13:46 libcrypt.so.1*
-rwxr-xr-x 1 root root 71846 Jun 3 13:46 libdl.so.2*
-rwxr-xr-x 1 root root 27762 Jun 3 13:46 libhistory.so.4.0*
lrwxrwxrwx 1 root root 17 Jun 4 12:12 libncurses.so.4 -> libncurses.so.4.2*
-rwxr-xr-x 1 root root 503903 Jun 3 13:46 libncurses.so.4.2*
lrwxrwxrwx 1 root root 17 Jun 4 12:12 libncurses.so.5 -> libncurses.so.5.0*
-rwxr-xr-x 1 root root 549429 Jun 3 13:46 libncurses.so.5.0*
-rwxr-xr-x 1 root root 369801 Jun 3 13:46 libnsl.so.1*
-rwxr-xr-x 1 root root 142563 Jun 4 11:49 libnss_compat.so.1*
-rwxr-xr-x 1 root root 215569 Jun 4 11:49 libnss_compat.so.2*
-rwxr-xr-x 1 root root 61648 Jun 4 11:34 libnss_dns.so.1*
```

```
-rwxr-xr-x 1 root root 63453 Jun 4 11:34 libnss_dns.so.2*
-rwxr-xr-x 1 root root 63782 Jun 4 11:34 libnss_dns6.so.2*
-rwxr-xr-x 1 root root 205715 Jun 3 13:46 libnss_files.so.1*
-rwxr-xr-x 1 root root 235932 Jun 3 13:49 libnss_files.so.2*
-rwxr-xr-x 1 root root 204383 Jun 4 11:33 libnss_nis.so.1*
-rwxr-xr-x 1 root root 254023 Jun 4 11:33 libnss_nis.so.2*
-rwxr-xr-x 1 root root 256465 Jun 4 11:33 libnss_nisplus.so.2*
lrwxrwxrwx 1 root root 14 Jun 4 12:12 libpam.so.0 -> libpam.so.0.72*
-rwxr-xr-x 1 root root 31449 Jun 3 13:46 libpam.so.0.72*
lrwxrwxrwx 1 root root 19 Jun 4 12:12 libpam_misc.so.0 ->
libpam_misc.so.0.72*
-rwxr-xr-x 1 root root 8125 Jun 3 13:46 libpam_misc.so.0.72*
lrwxrwxrwx 1 root root 15 Jun 4 12:12 libpamc.so.0 -> libpamc.so.0.72*
-rwxr-xr-x 1 root root 10499 Jun 3 13:46 libpamc.so.0.72*
-rwxr-xr-x 1 root root 176427 Jun 3 13:46 libreadline.so.4.0*
-rwxr-xr-x 1 root root 44729 Jun 3 13:46 libutil.so.1*
-rwxr-xr-x 1 root root 70254 Jun 3 13:46 libz.a*
lrwxrwxrwx 1 root root 13 Jun 4 12:13 libz.so -> libz.so.1.1.3*
lrwxrwxrwx 1 root root 13 Jun 4 12:13 libz.so.1 -> libz.so.1.1.3*
-rwxr-xr-x 1 root root 63312 Jun 3 13:46 libz.so.1.1.3*
drwxr-xr-x 2 root root 4096 Jun 4 12:00 security/
./lib/security:
total 668
drwxr-xr-x 2 root root 4096 Jun 4 12:00 ./
drwxr-xr-x 3 root root 4096 Jun 4 12:13 ../
-rwxr-xr-x 1 root root 10067 Jun 3 13:46 pam_access.so*
-rwxr-xr-x 1 root root 8300 Jun 3 13:46 pam_chroot.so*
-rwxr-xr-x 1 root root 14397 Jun 3 13:46 pam_cracklib.so*
-rwxr-xr-x 1 root root 5082 Jun 3 13:46 pam_deny.so*
-rwxr-xr-x 1 root root 13153 Jun 3 13:46 pam_env.so*
-rwxr-xr-x 1 root root 13371 Jun 3 13:46 pam_filter.so*
-rwxr-xr-x 1 root root 7957 Jun 3 13:46 pam_ftp.so*
-rwxr-xr-x 1 root root 12771 Jun 3 13:46 pam_group.so*
-rwxr-xr-x 1 root root 10174 Jun 3 13:46 pam_issue.so*
-rwxr-xr-x 1 root root 9774 Jun 3 13:46 pam_lastlog.so*
-rwxr-xr-x 1 root root 13591 Jun 3 13:46 pam_limits.so*
-rwxr-xr-x 1 root root 11268 Jun 3 13:46 pam_listfile.so*
-rwxr-xr-x 1 root root 11182 Jun 3 13:46 pam_mail.so*
-rwxr-xr-x 1 root root 5923 Jun 3 13:46 pam_nologin.so*
-rwxr-xr-x 1 root root 5460 Jun 3 13:46 pam_permit.so*
-rwxr-xr-x 1 root root 18226 Jun 3 13:46 pam_pwcheck.so*
-rwxr-xr-x 1 root root 12590 Jun 3 13:46 pam_rhosts_auth.so*
-rwxr-xr-x 1 root root 5551 Jun 3 13:46 pam_rootok.so*
-rwxr-xr-x 1 root root 7239 Jun 3 13:46 pam_securetty.so*
-rwxr-xr-x 1 root root 6551 Jun 3 13:46 pam_shells.so*
-rwxr-xr-x 1 root root 55925 Jun 4 12:00 pam_smb_auth.so*
-rwxr-xr-x 1 root root 12678 Jun 3 13:46 pam_stress.so*
-rwxr-xr-x 1 root root 11170 Jun 3 13:46 pam_tally.so*
-rwxr-xr-x 1 root root 11124 Jun 3 13:46 pam_time.so*
-rwxr-xr-x 1 root root 45703 Jun 3 13:46 pam_unix.so*
-rwxr-xr-x 1 root root 45703 Jun 3 13:46 pam_unix2.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_acct.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_auth.so*
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_passwd.so*
```

```
-rwxr-xr-x 1 root root 45386 Jun 3 13:46 pam_unix_session.so*
-rwxr-xr-x 1 root root 9726 Jun 3 13:46 pam_userdb.so*
-rwxr-xr-x 1 root root 6424 Jun 3 13:46 pam_warn.so*
-rwxr-xr-x 1 root root 7460 Jun 3 13:46 pam_wheel.so*
./sbin:
total 3132
drwxr-xr-x 2 root root 4096 Jun 4 12:35 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
-rwxr-xr-x 1 root root 178256 Jun 3 13:46 choptest*
-rwxr-xr-x 1 root root 184032 Jun 3 13:46 cqtest*
-rwxr-xr-x 1 root root 81096 Jun 3 13:46 dialtest*
-rwxr-xr-x 1 root root 1142128 Jun 4 11:28 ldconfig*
-rwxr-xr-x 1 root root 2868 Jun 3 13:46 lockname*
-rwxr-xr-x 1 root root 3340 Jun 3 13:46 ondelay*
-rwxr-xr-x 1 root root 376796 Jun 3 13:46 pagesend*
-rwxr-xr-x 1 root root 13950 Jun 3 13:46 probemodem*
-rwxr-xr-x 1 root root 9234 Jun 3 13:46 recvstats*
-rwxr-xr-x 1 root root 64480 Jun 3 13:46 sftp-server*
-rwxr-xr-x 1 root root 744412 Jun 3 13:46 sshd*
-rwxr-xr-x 1 root root 30750 Jun 4 11:46 su*
-rwxr-xr-x 1 root root 194632 Jun 3 13:46 tagtest*
-rwxr-xr-x 1 root root 69892 Jun 3 13:46 tsitest*
-rwxr-xr-x 1 root root 43792 Jun 3 13:46 typetest*
./tmp:
total 8
drwxr-xr-x 2 root root 4096 Jun 4 12:32 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
./usr:
total 8
drwxr-xr-x 2 root root 4096 Jun 4 12:16 ./
drwxr-xr-x 9 root root 4096 Jun 5 10:05 ../
lrwxrwxrwx 1 root root 7 Jun 4 12:14 bin -> ../bin//
lrwxrwxrwx 1 root root 7 Jun 4 11:33 lib -> ../lib//
lrwxrwxrwx 1 root root 8 Jun 4 12:13 sbin -> ../sbin//
```

Chroot-Umgebung für Apache

Einleitung

Das Programm **chroot** wird häufig dazu benutzt, einen Daemon in einen beschränkten Verzeichnisbaum einzusperren. Sie können es dazu verwenden, um Dienste von anderen abzuschirmen, so dass Sicherheitsprobleme mit einem Softwarepaket nicht den ganzen Server gefährden können. Durch die Verwendung des Skripts **makejail** wird es viel leichter, einen Verzeichnisbaum in einer **chroot**-Umgebung einzurichten und zu aktualisieren.

FIXME: Apache can also be chrooted using http://www.modsecurity.org which is available in libapache-mod-security (for Apache 1.x) and libapache2-mod-security (for Apache 2.x).

Lizenz

Installation des Servers

Diese Vorgehensweise wurde auf Debian GNU/Linux 3.0 (Woody) mit **makejail** 0.0.4-1 (in Debian/Testing) getestet.

- Melden Sie sich als **Root** an und erstellen Sie ein neues Verzeichnis für das Gefängnis:

```
$ mkdir -p /var/chroot/apache
```

- Erstellen Sie einen neuen Benutzer und eine neue Gruppe. Der Apache in der **chroot**-Umgebung wird als diese Benutzer und Gruppe laufen, die für nichts anderes auf dem System verwendet werden. In dem Beispiel heißen sowohl Benutzer als auch Gruppe **chrapach**.

```
$ adduser --home /var/chroot/apache --shell /bin/false \
--no-create-home --system --group chrapach
```

  FIXME: is a new user needed? (Apache already runs as the apache user)

- Installieren Sie ganz normal Apache auf Debian: `apt-get install apache`.

- Richten Sie Apache ein (z.B. definieren Sie Ihrer Subdomains usw.). Weisen Sie in der Konfigurationsdatei `/etc/apache/httpd.conf` den Optionen *Group* und *User* `chrapach` zu. Starten Sie Apache neu und stellen Sie sicher, dass der Server korrekt funktioniert. Danach halten Sie den Server wieder an.

- Installieren Sie **makejail** (ist derzeit in Debian/Testing vorhanden). Sie sollten auch **wget** und **lynx** installieren, da sie von **makejail** benutzt werden, um den Server in der **chroot**-Umgebung zu testen: `apt-get install makejail wget lynx`.

- Kopieren Sie die Beispielkonfigurationsdatei für Apache ins Verzeichnis `/etc/makejail`:

```
# cp /usr/share/doc/makejail/examples/apache.py /etc/makejail/
```

- Bearbeiten Sie `/etc/makejail/apache.py`. Sie müssen die Optionen *chroot*, *users* und *groups* verändern. Um diese Version von **makejail** laufen zu lassen, können Sie auch die Option **packages** hinzufügen. Vergleichen Sie die http://www.floc.net/makejail/current/doc/. Die Konfigurationsdatei könnte beispielsweise so aussehen:

```
chroot="/var/chroot/apache"
testCommandsInsideJail=["/usr/sbin/apachectl start"]
processNames=["apache"]
testCommandsOutsideJail=["wget -r --spider http://localhost/",
                         "lynx --source https://localhost/"]
preserve=["/var/www",
          "/var/log/apache",
          "/dev/log"]
users=["chrapach"]
groups=["chrapach"]
packages=["apache", "apache-common"]
userFiles=["/etc/password",
           "/etc/shadow"]
```

```
groupFiles=["/etc/group",
            "/etc/gshadow"]
forceCopy=["/etc/hosts",
           "/etc/mime.types"]
```

FIXME: some options do not seem to work properly. For instance, `/etc/shadow` and `/etc/gshadow` are not copied, whereas `/etc/password` and `/etc/group` are fully copied instead of being filtered.

- Erstellen Sie den Verzeichnisbaum für chroot: `makejail /etc/makejail/apache.py`.
- Falls `/etc/password` und `/etc/group` vollständig kopiert wurden, geben Sie Folgendes ein:

```
$ grep chrapach /etc/passwd > /var/chroot/apache/etc/passwd
$ grep chrapach /etc/group > /var/chroot/apache/etc/group
```

  Damit werden `/etc/password` und `/etc/group` mit gefilterten Fassungen ersetzt.

- Kopieren Sie die Webseiten und die Protokolle ins Gefängnis. Diese Dateien werden nicht automatisch mitkopiert (sehen Sie sich dazu die Option *preserve* in der Konfigurationsdatei von **makejail** an).

```
# cp -Rp /var/www /var/chroot/apache/var
# cp -Rp /var/log/apache/*.log /var/chroot/apache/var/log/apache
```

- Editieren Sie das Startskript für den Protokoll-Daemon des Systems so, dass er auch den Socket `/var/chroot/apache/dev/log` beobachtet. Ersetzen Sie in `/etc/default/syslogd` `SYSLOGD=""` mit `SYSLOGD=" -a /var/chroot/apache/dev/log"` und starten Sie den Daemon neu (`/etc/init.d/sysklogd restart`).
- Editieren Sie das Startskript von Apache (`/etc/init.d/apache`). Sie müssen ein paar Änderung am Standardstartskript vornehmen, damit es richtig in einem Verzeichnisbaum in einer **chroot**-Umgebung läuft. Da wäre:
  - Legen Sie die Variable *CHRDIR* am Anfang der Datei neu fest.
  - Bearbeiten Sie die Abschnitte *start*, *stop*, *reload* etc.
  - Fügen Sie eine Zeile hinzu, um das `/proc`-Dateisystem innerhalb des Gefängnisses ein- und auszuhängen.

```
#! /bin/bash
#
# apache        Start the apache HTTP server.
#

CHRDIR=/var/chroot/apache

NAME=apache
PATH=/bin:/usr/bin:/sbin:/usr/sbin
DAEMON=/usr/sbin/apache
SUEXEC=/usr/lib/apache/suexec
PIDFILE=/var/run/$NAME.pid
CONF=/etc/apache/httpd.conf
```

```
APACHECTL=/usr/sbin/apachectl

trap "" 1
export LANG=C
export PATH

test -f $DAEMON || exit 0
test -f $APACHECTL || exit 0

# ensure we don't leak environment vars into apachectl
APACHECTL="env -i LANG=${LANG} PATH=${PATH} chroot $CHRDIR $APACHECTL"

if egrep -q -i "^[[:space:]]*ServerType[[:space:]]+inet" $CONF
then
    exit 0
fi

case "$1" in
  start)
    echo -n "Starting web server: $NAME"
    mount -t proc proc /var/chroot/apache/proc
    start-stop-daemon --start --pidfile $PIDFILE --exec $DAEMON \
      --chroot $CHRDIR
    ;;

  stop)
    echo -n "Stopping web server: $NAME"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" --oknodo
    umount /var/chroot/apache/proc
    ;;

  reload)
    echo -n "Reloading $NAME configuration"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" \
      --signal USR1 --startas $DAEMON --chroot $CHRDIR
    ;;

  reload-modules)
    echo -n "Reloading $NAME modules"
    start-stop-daemon --stop --pidfile "$CHRDIR/$PIDFILE" --oknodo \
      --retry 30
    start-stop-daemon --start --pidfile $PIDFILE \
      --exec $DAEMON --chroot $CHRDIR
    ;;

  restart)
    $0 reload-modules
    exit $?
    ;;

  force-reload)
    $0 reload-modules
    exit $?
    ;;
```

```
  *)
    echo "Usage: /etc/init.d/$NAME {start|stop|reload|reload-modules|force-reloa
    exit 1
    ;;
esac

if [ $? == 0 ]; then
  echo .
  exit 0
else
  echo failed
  exit 1
fi
```

FIXME: should the first Apache process be run as another user than root (i.e. add --chuid chrapach:chrapach)? Cons: chrapach will need write access to the logs, which is awkward.

- Ersetzen Sie in `/etc/logrotate.d/apache` `/var/log/apache/*.log` durch `/var/chroot/apache/var/log/apache/*.log`.

- Starten Sie Apache (**/etc/init.d/apache start**) und überprüfen Sie, was im Protokoll des Gefängnisses gemeldet wird (`/var/chroot/apache/var/log/apache/error.log`). Wenn Ihre Konfiguration komplexer sein sollte (z.B. wenn Sie auch PHP und MySQL einsetzen), werden wahrscheinlich Dateien fehlen. Wenn einige Dateien nicht automatisch von **makejail** kopiert werden, können Sie diese in den Optionen *forceCopy* (um Dateien direkt zu kopieren) oder *packages* (um ganze Pakete mit ihren Abhängigkeiten zu kopieren) in der Konfigurationsdatei `/etc/makejail/apache.py` aufführen.

- Geben Sie `ps aux | grep apache` ein, um sicherzustellen, dass Apache läuft. Sie sollten etwas in dieser Art sehen:

```
root 180 0.0 1.1 2936 1436 ? S 04:03 0:00 /usr/sbin/apache
chrapach 189 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 190 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 191 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 192 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
chrapach 193 0.0 1.1 2960 1456 ? S 04:03 0:00 /usr/sbin/apache
```

- Stellen Sie sicher, dass die Apache-Prozesse in einer **chroot**-Umgebung laufen. Betrachten Sie dazu das `/proc`-Dateisystem: `ls -la /proc/process_number/root/.`, wobei `process_number` einer der PID-Nummern ist, die oben aufgeführt wurden (z.B. 189 in der zweiten Reihe). Die Einträge des eingeschränkten Verzeichnisbaums sollten Sie sich auflisten lassen:

```
drwxr-sr-x 10 root staff 240 Dec 2 16:06 .
drwxrwsr-x 4 root staff 72 Dec 2 08:07 ..
drwxr-xr-x 2 root root 144 Dec 2 16:05 bin
drwxr-xr-x 2 root root 120 Dec 3 04:03 dev
drwxr-xr-x 5 root root 408 Dec 3 04:03 etc
drwxr-xr-x 2 root root 800 Dec 2 16:06 lib
dr-xr-xr-x 43 root root 0 Dec 3 05:03 proc
drwxr-xr-x 2 root root 48 Dec 2 16:06 sbin
drwxr-xr-x 6 root root 144 Dec 2 16:04 usr
drwxr-xr-x 7 root root 168 Dec 2 16:06 var
```

Um diesen Test zu automatisieren, geben Sie `ls -la /proc/`cat /var/chroot/apache/var/run/apache.pid`/root/.` ein.

FIXME: Add other tests that can be run to make sure the jail is closed?

Ich mag das, da es so nicht sehr schwierig ist, das Gefängnis einzurichten, und der Server mit nur zwei Zeilen aktualisiert werden kann:

```
apt-get update && apt-get install apache
makejail /etc/makejail/apache.py
```

Weiterführende Informationen

Wenn Sie nach weiteren Informationen suchen, sehen Sie sich die Quellen an, auf denen diese Anleitung beruht: Die http://www.floc.net/makejail/. Diese Programm wurde von Alain Tesio geschrieben.