>> Gentoo Linux Sicherheitsleitfaden

[Bitte Kapitel auswählen]

1. Einleitung

Dieser Guide ist für Leute gedacht, die Gentoo Linux in einer Server-Basierten Umgebung einsetzen, oder das Gefühl haben mehr Sicherheit zu brauchen.

Notiz
Wenn Sie nach dem Lesen dieses Guides an noch mehr Sicherheit interessiert sind, sollten Sie einen Blick auf das Hardened Gentoo Projekt werfen

2. Was vor der Installation beachtet werden sollte

2.1 Physische Sicherheit

Egal wieviele Sicherheitsmassnahmen Sie integrieren, sie können leicht umgangen werden, wenn der Hacker direkten Zugriff auf Ihre Maschine hat. Stellen Sie sicher, dass Ihre Hardware nicht einfach so zugänglich ist. Zum Beispiel könnten Sie Ihre Maschine in einem speziellen Serverraum einschliessen. Die Gehäuse zu versiegeln ist auch eine gute Idee. Um das höchste Niveau an Sicherheit zu erreichen, können Sie Ihr BIOS so einstellen, dass es nur von der Festplatte bootet. Deaktivieren Sie auch das Booten von Diskette und von CD-ROM. Für den Übervorsichtigen kann es auch sinnvoll sein, das BIOS-Passwort zu aktivieren. BIOS-Passwörter sind auch eine gute Idee für Notebook-Benutzer.

2.2 Dämon/Dienst Planung

Dokumentieren Sie, welche Dienste die Maschine anbieten soll oder anbieten darf. Dies wird Ihnen helfen ein besseres Partitionsschema für das System zu erstellen. Es kann auch das Aufspüren von Eindringlingen erheblich erleichtern. Natürlich brauchen Sie das nicht zu dokumentieren, wenn Sie nur einen oder ein paar Computer benutzen und Sie der einzige Nutzer sind. Zum Beispiel: Wenn die Maschine als Firewall agieren soll sollten auf der Maschine **keine** Dienste ausser vielleicht sshd laufen.

Dokumentieren Sie dies und die aktuelle Version von SSH - es wird Ihnen helfen, das zu aktualisierende System wiederzufinden - für den Fall, dass jemand ein Sicherheitsloch in sshd findet. Dies wird Ihnen auch dabei helfen festzulegen, wer Zugriff auf das System haben sollte.

2.3 Partitions-Schemata

Goldene Regeln:

- Jedes Verzeichnis auf das ein Benutzer Schreibrechte haben muss (/home und /tmp, /var) sollte auf einer seperaten Partition liegen und Disk-Quotas benutzen. Portage benutzt /var/tmp zum kompilieren, folglich muss diese Partition gross sein. Dies reduziert das Risiko, daß ein Benutzer "/" komplett füllen kann.
- Jedes Verzeichnis, in das nicht in der Distribution enthaltene Pakete installiert werden sollen, sollten auf einer seperaten Partition liegen. Nach dem Filesystem Hierarchy Standard ist dies /opt oder /usr/local. Wenn diese seperate Partitionen sind, bleiben Sie bei einer eventuellen Neuinstallation des Systems bestehen.
- Versuchen sie, statische Daten in eine eigene Partition verschieben und diese Partition nur lesbar einhängen. Wenn sie wirklich übervorsichtig sind, dann könnten Sie statische Daten auch auf einem nur lesbaren Medium speichern - zum Beispiel einer CD-ROM.

2.4 Der Benutzer root

Der Benutzer 'root' ist der mächtigste Benutzer im System und sollte nie für irgendetwas Anderes als administrative Aufgaben eingesetzt werden. Wenn ein Angreifer root-Zugang erreicht, dann können Sie Ihrem System nicht mehr länger trauen - Sie haben dann keine andere Wahl, als neu zu installieren.

Goldene Regeln bezüglich 'root'

- Erstellen Sie immer einen Benutzer für die tägliche Arbeit. Wenn dieser Benutzer root-Zugang benötigt, dann fügen Sie diesen Benutzer zur Gruppe wheel hinzu. Dies erlaubt einem normalen benutzer "nach root zu su'en".
- Lassen Sie X oder irgendeine andere Benutzeranwendung niemals als root laufen.

- Benutzen Sie immer absolute Pfadangaben, wenn Sie als root angemeldet sind. Sonst ist es möglich, root eine andere Anwendung ausführen zu lassen, als er denkt (wenn zum Beispiel jemand an PATH manipuliert hat und root ohne *su* - su'ed). Dann wird root den Pfad des Nutzers benutzen.
- Wenn ein Benutzer nur ein paar Kommandos, anstatt von allen, die root benutzen kann, dann überlegen Sie, vielleicht auf sudo zurückzugreifen - aber seien Sie vorsichtig damit.
- Verlassen Sie nie den Terminal, wenn root angemeldet ist!

Gentoo hat einen allgemeinen Schutz gegen normale Benutzer, die versuchen su einzusetzen. Die Standardeinstellung von PAM besagt, dass ein Benutzer in der Gruppe wheel sein muss, um su benutzen zu dürfen.

2.5 Sicherheitsrichtlinien

Es gibt verschiedene Gründe, weshalb Sicherheitsrichtlinien benötigt werden.

- Sie können nicht behaupten ein sicheres Netzwerk zu haben, ohne jemals definiert zu haben, was sicher ist.
- Es ist fast unmöglich potientielle Angreifer zu erwischen, Netzwerkprobleme zu lösen, oder Prüfungen zu dirigieren ohne den Netzwerkverkehr abzuhören oder in private Home-Verzeichnisse zu sehen. Aber dieses Reinhören ohne Erlaubnis des Nutzers ist in den meisten Ländern illegal. Und da 60% aller Angriffe von innerhalb eines Unternehmens kommen ist es wichtig, dass Sie die Augen offen halten.
- Sie können von Ihren Anwendern nicht erwarten, dass sie sich Gedanken über Sicherheit machen, wenn Sie niemals erklärt haben, warum es wichtig ist oder wie sie sich selbst und ihre Kollegen schützen sollten.
- Gute Richtlinien und Netzwerkdokumentation zahlen sich immer aus - egal wie.
- Die Polizei oder die Staatliche Kriminalbehörde kann Ihnen nicht helfen den Angreifer dingfest zu machen, wenn diese nicht wissen wie Ihre Netzwerkkonfiguration aussieht oder welche Dienste Sie anbieten.
- Was werden Sie tun, wenn es einen Angriff gab? Sie müssen definieren, was Sie tun würden und wen Sie informieren würden. Würden Sie bei jeder Gelegenheit die Polizei oder ein CERT-Team anrufen? Die würden Sie nicht ernst nehmen.

Dies sollte eigentlich darlegen warum es wichtig ist Richtlinien auf Systemen mit mehr als einem Benutzer festzulegen und warum es wichtig ist die Anwender zu erziehen.

Eine Richtlinie ist ein Dokument (oder mehrere Dokumente) mit Antworten auf die Fragen "wer", "wo", "warum" und "was". Jeder Anwender in Ihrem System/Netzwerk sollte es lesen, verstehen und unterschreiben. Es ist wichtig, daß Sie sich die Zeit nehmen den Anwendern beim Verstehen der Richtlinie zu helfen: dem Grund, weshalb diese Richtlinie unterschrieben werden muß und was passiert, wenn sie direkt gegen die Richtlinie verstossen (dies sollte in der Richtlinie aufgeführt sein). Dies sollte mindestens einmal im Jahr wiederholt werden, da sich die Richtlinie ändern kann, aber auch um die Anwender immer wieder aufs Neue daran zu erinnern.

Notiz
Erstellen Sie Richtlinien, die einfach zu lesen und in jedem Zusammenhang sehr präzise sind.

Eine Sicherheitsrichtlinie sollte mindestens die folgenden Punkte beinhalten:

Akzeptable Anwendung
- Bildschirmschoner
- Behandlung von Kennwörtern
- Herunterladen von Programmen
- Wissen darüber, ob diese überwacht werden
- Benutzung von Antiviren-Software
- etc.

Behandlung von sensitiven Daten (jegliche Schriftliche Form, Papier oder Digital
- Sauberer Schreibtisch und verschlossene, vertrauliche Informationen
- PC herunterfahren vorm Verlassen
- Benutzung von Verschlüsselung
- Behandlung von Schlüsseln für vertraute Mitarbeiter
- Behandlung von vertraulichem Material auf Reisen

Behandlung der Computerausstattung auf Reisen
- Behandlung des Laptops auf Reisen und bei Hotelaufenthalten

Die Richtlinie für die IT-Abteilung kann sich von der für die normalen Angestellten leicht unterscheiden.

Die Sicherheitsrichtlinie kann riesig werden und wichtige Informationen können leicht vergessen gehen. Die Richtline für die IT-Abteilung kann Informationen enthalten, die gegenüber den normalen Benutzern als vertraulich gelten. Somit ist es sinnvoll, sich in kleinen Richtlinien fortzubewegen: Richtlinie für akzeptable Bedienung, Richtlinie für Passwörter, für E-Mail und für Fernzugriff.

Beispiele für Richtlinien können beim The SANS Security Policy Project gefunden werden. Wenn Sie ein kleines Netzwerk haben und diese Richtlinie für zu gross halten, dann sollten sie einen Blick auf das RFC2196 werfen, dass ein Sicherheitshandbuch darstellt

3. Die Sicherheit nach/während der Installation anziehen

3.1 USE Flags

Die /etc/make.conf-Datei enthält die benutzerdefinierten und die /etc/make.profile/make.defaults-Datei die Standard USE Flags. Die wichtigen Flags für diesen Guide sind *PAM* (Pluggable Authentication Module), *tcpd* (TCP Wrapper) und *SSL* (Secure Socket Layer). Diese sind in den Standard USE Flags enthalten.

3.2 GRUB Passwort

Grub unterstützt 2 verschiedene Wege für Passwortkontrolle in seiner Konfigurationsdatei(/boot/grub/grub.conf): Zum einen normalen Text und zum anderen md5+salt Verschlüsselung.

Fügen Sie ein Passwort in /boot/grub/grub.conf ein.

Befehlsauflistung 1: /boot/grub/grub.conf

```
timeout 5
password changeme
```

Dies wird das Passwort **changeme** hinzufügen; wenn kein Passwort eingegeben ist, wird die Standard-Boot-Einstellung genommen.

Sollte ein md5-Passwort genommen werden, dann müssen Sie das Passwort ins Crypt-Format konvertieren (*man crypt*), daß das selbe Format wie die Shadow-Datei hat. Zum Beispiel könnte das verschlüsselte Passwort **changeme** so aussehen: **1T7/dgdIJ$dJM.n2wZ8RG.oEiIOwJUs.**

Oder das Passwort direkt in der GRUB Shell konvertieren:

Befehlsauflistung 2: md5crypt in der GRUB Shell

```
#/sbin/grub

    GRUB  version 0.92  (640K lower / 3072K upper memory)

   [ Minimal BASH-like line editing is supported.  For the first word, TAB
     lists possible command completions.  Anywhere else TAB lists the possible
     completions of a device/filename. ]

grub> md5crypt

Password: ********
// Typed changeme
Encrypted: $1$T7/dgdIJ$dJM.n2wZ8RG.oEiIOwJUs.

grub> quit
```

Dann kopieren und fügen Sie das Passwort in /boot/grub/grub.conf ein.

Befehlsauflistung 3: /boot/grub/grub.conf

```
timeout 5
password --md5 $1$T7/dgdIJ$dJM.n2wZ8RG.oEiIOwJUs.
```

Der Zeitablauf von 5 Sekunden wird sinnvoll, wenn das System fernbedient wird und bei einem Neustart ohne Tastatureingaben auskommen muss. Mehr Informationen über Grub-Passwörter können Sie bekommen, wenn Sie *info grub* ausführen.

3.3 LILO Passwort

LILO unterstützt auch zwei Arten des Behandelns von Passwörtern : Global und Imagerelativ -- beide in Klartext.

Das globale wird am Anfang der Konfigurationsdatei gesetzt:

Befehlsauflistung 4: /etc/lilo.conf

```
password=changeme
restricted
delay=3
```

Im anderen Fall fügen Sie es einfach beim entsprechenden Image hinzu.

Befehlsauflistung 5: /etc/lilo.conf

```
image=/boot/bzImage
      read-only
      password=changeme
      restricted
```

Wenn die *restricted*-Option nicht angegeben wurde, dann wird jedes Mal nach einem Passwort gefragt.

Um die Änderungen an lilo.conf zu übernehmen, müssen Sie */sbin/lilo* ausführen.

3.4 Einschränkung der Konsolenbenutzung.

/etc/securetty enthält Terminaltypen die es Ihnen ermöglichen/erlauben festzulegen, von welchen *TTY* Geräten aus root sich einloggen darf.

Wir empfehlen, dass sie alle Zeilen bis auf *vc/1* auskommentieren. Dies stellt sicher, dass sich root nur einmal und nur an einem Terminal einloggen kann.

Notiz
Benutzer in der wheel Gruppe können weiterhin auf anderen Konsolen per *su* - root werden.
Befehlsauflistung 6: /etc/securetty

```
vc/1
```

4. Mehr Protokolle (Logs)

Zusätzliche Protokolle sollten hinzugefügt werden um Warnungen oder Fehler aufzuspüren, die vor einem momentanen oder bereits durchgeführten Angriff warnen könnten. Angreifer beobachten ein Netzwerk oder durchsuchen dies oft, bevor sie angreifen.

Es ist auch unersetzlich, dass die Protokolldateien einfach zu lesen und zu verwalten sind. Gentoo Linux gibt ihnen die Möglichkeit bei der Installation zwischen drei verschiedenen Protokollierungsprogrammen zu wählen.

4.2 Loggen: Syslogd

Syslogd ist das gängigste Protokollierungsprogramm für Linux und Unix. Es beinhaltet keine Protokollrotation. Diese Eigenschaft wird durch das Verwenden von /usr/sbin/logrotate in einem Cron Job und korrekt konfigurierten Einstellungen in /etc/logrotate.conf übernommen. Wie oft die Protokollrotation stattfinden sollte hängt von der Systembelastung ab.

Hierunter sehen Sie die Standard Konfiguration /syslog.conf mit einigen zuästzlichen Features. Wir haben die *cron* und *tty* Zeilen unkommentiert und eine Remote Logging Server hinzugefügt. Um die Sicherheit weiter zu erhöhen, können Sie Logs an zwei Orten schreiben lassen.

Befehlsauflistung 7: /etc/syslog.conf

```
#  /etc/syslog.conf     Configuration file for syslogd.
#
#                       For more information see syslog.conf(5)
#                       manpage.
#                       This is from Debian, we are using it for now
#                       Daniel Robbins, 5/15/99

#
```

```
# First some standard logfiles.  Log by facility.
#

auth,authpriv.*                 /var/log/auth.log
*.*;auth,authpriv.none          -/var/log/syslog
cron.*                          /var/log/cron.log
daemon.*                        -/var/log/daemon.log
kern.*                          -/var/log/kern.log
lpr.*                           -/var/log/lpr.log
mail.*                          /var/log/mail.log
user.*                          -/var/log/user.log
uucp.*                          -/var/log/uucp.log
local6.debug                    /var/log/imapd.log

#
# Logging for the mail system. Split it up so that
# it is easy to write scripts to parse these files.
#
mail.info                       -/var/log/mail.info
mail.warn                       -/var/log/mail.warn
mail.err                        /var/log/mail.err

# Logging for INN news system
#
news.crit                       /var/log/news/news.crit
news.err                        /var/log/news/news.err
news.notice                     -/var/log/news/news.notice

#
# Some `catch-all' logfiles.
#
*.=debug;\
        auth,authpriv.none;\
        news.none;mail.none     -/var/log/debug
*.=info;*.=notice;*.=warn;\
        auth,authpriv.none;\
        cron,daemon.none;\
        mail,news.none          -/var/log/messages

#
# Emergencies and alerts are sent to everybody logged in.
#
*.emerg                         *
*.=alert                        *

#
# I like to have messages displayed on the console, but only on a virtual
# console I usually leave idle.
#
daemon,mail.*;\
       news.=crit;news.=err;news.=notice;\
       *.=debug;*.=info;\
       *.=notice;*.=warn       /dev/tty8

#Setup a remote logging server
*.*                         @logserver

# The named pipe /dev/xconsole is for the `xconsole' utility.  To use it,
# you must invoke `xconsole' with the `-file' option:
#
#    $ xconsole -file /dev/xconsole [...]
#
# NOTE: adjust the list below, or you'll go crazy if you have a reasonably
#      busy site..
#
#daemon.*,mail.*;\
#       news.crit;news.err;news.notice;\
#       *.=debug;*.=info;\
#       *.=notice;*.=warn       |/dev/xconsole

local2.*                -/var/log/ppp.log
```

Der Angreifer wird höchstwahrscheinlich versuchen seine Spuren zu verwischen, indem er die Protokolldateien bearbeitet oder löscht. Sie können es für den Angreifer schwerer machen indem sie

kann nach Programmnamen protokollieren (wie syslogd) und beinhaltet reguläre Ausdrucksübereinstimmung und die Möglichkeit Kommandos auszuführen. Sehr gut um Handeln zu können, wenn nötig.

Die Standard Konfiguration ist zunächst ausreichend. Wenn Sie benachrichtigt werden wollen, wenn z.B. ein Anmeldevorgang fehlschlägt benutzen Sie eines der folgenden Skripte.

Für Postfix.

Befehlsauflistung 8: /usr/local/sbin/mail_pwd_failures.sh für Postfix

```
#! /bin/sh
echo "$3" | mail -s "Warning (program : $2)" root
```

Für Qmail:

Befehlsauflistung 9: /usr/local/sbin/mail_pwd_failures.sh für Qmail

```
#!/bin/sh
echo "To: root
Subject:Failure (Warning: $2)
$3
" | /var/qmail/bin/qmail-inject -f root
```

Denken Sie daran das Skript mit */bin/chmod +x /usr/local/sbin/mail_pwd_failures.sh* ausführbar zu machen.

Unkommentieren Sie dann die Zeile unter "Password failures" in /etc/metalog/metalog.conf wie folgt:

Befehlsauflistung 10: /etc/metalog/metalog.conf

```
command = "/usr/local/sbin/mail_pwd_failures.sh"
```

4.4 Loggen: Syslog-ng

Syslog-ng enthält einige derselben Funktionen wie Syslog und Metalog mit einem kleinen Unterschied. Es ermöglicht die Filterung von Nachrichten basierend auf Level und Inhalt (wie Metalog), bietet entferntes Protokollieren (wie syslog) und kann Protokolle von syslogd verarbeiten. Sogar Streams von Solaris, schreiben an ein TTY, Ausführen von Programmen und die Nutzung als Protokollierungsserver sind möglich. Grundlegend ist dies das Beste aus beiden Protokollierern kombiniert mit einer erweiterten Konfiguration.

Eine klassische, leicht modifizierte Konfigurationsdatei.

Befehlsauflistung 11: /etc/syslog-ng/syslog-ng.conf

```
options { long_hostnames(off); sync(0); };

#Quelle von der das Log gelesen werden soll
source src { unix-stream("/dev/log"); internal(); };
source kernsrc { file("/proc/kmsg"); };

#Ziele festlegen
destination authlog { file("/var/log/auth.log"); };
destination syslog { file("/var/log/syslog"); };
destination cron { file("/var/log/cron.log"); };
destination daemon { file("/var/log/daemon.log"); };
destination kern { file("/var/log/kern.log"); };
destination lpr { file("/var/log/lpr.log"); };
destination user { file("/var/log/user.log"); };
destination mail { file("/var/log/mail.log"); };

destination mailinfo { file("/var/log/mail.info"); };
destination mailwarn { file("/var/log/mail.warn"); };
destination mailerr { file("/var/log/mail.err"); };

destination newscrit { file("/var/log/news/news.crit"); };
destination newserr { file("/var/log/news/news.err"); };
destination newsnotice { file("/var/log/news/news.notice"); };

destination debug { file("/var/log/debug"); };
destination messages { file("/var/log/messages"); };
```

```
filter f_daemon { facility(daemon); };
filter f_kern { facility(kern); };
filter f_lpr { facility(lpr); };
filter f_mail { facility(mail); };
filter f_user { facility(user); };
filter f_debug { not facility(auth, authpriv, news, mail); };
filter f_messages { level(info..warn)
        and not facility(auth, authpriv, mail, news); };
filter f_emergency { level(emerg); };

filter f_info { level(info); };
filter f_notice { level(notice); };
filter f_warn { level(warn); };
filter f_crit { level(crit); };
filter f_err { level(err); };
filter f_failed { match("failed"); };
filter f_denied { match("denied"); };

#Filter und Ziele verbinden
log { source(src); filter(f_authpriv); destination(authlog); };
log { source(src); filter(f_syslog); destination(syslog); };
log { source(src); filter(f_cron); destination(cron); };
log { source(src); filter(f_daemon); destination(daemon); };
log { source(kernsrc); filter(f_kern); destination(kern); };
log { source(src); filter(f_lpr); destination(lpr); };
log { source(src); filter(f_mail); destination(mail); };
log { source(src); filter(f_user); destination(user); };
log { source(src); filter(f_mail); filter(f_info); destination(mailinfo); };
log { source(src); filter(f_mail); filter(f_warn); destination(mailwarn); };
log { source(src); filter(f_mail); filter(f_err); destination(mailerr); };

log { source(src); filter(f_debug); destination(debug); };
log { source(src); filter(f_messages); destination(messages); };
log { source(src); filter(f_emergency); destination(console); };

#Standard-Log
log { source(src); destination(console_all); };
```

Sehr einfach zu konfigurieren, aber es ist auch sehr einfach etwas zu übersehen, da die Konfigurationsdatei riesig ist. Der Autor verspricht zudem noch einige zusätzliche Funktionen wie Verschlüsselung, Authentifizierung, Komprimierung und MAC (Mandatory Access Control) Kontrolle. Mit diesen Optionen wird es perfekt sein für Netzwerkprotokollierung, da der Angreifer die Protokolle nicht ausspionieren kann.

Syslog-ng hat auch noch andere Vorteile - es muss nicht als root laufen!

5. Partitionen mounten

Mountet man eine ext2, ext3 oder eine reiserfs Partition, so gibt es mehrere Optionen die man in /etc/fstab einfügen kann. Diese Optionen sind:

- *nosuid* - Ignoriert das SUID bit und behandelt es einfach wie eine normale Datei.
- *noexec* - Verhindert das Ausführen von Dateien von dieser Partition.
- *nodev* - Ignoriert Geräte.

Leider können diese Einstellungen leicht umgangen werden, indem man einen nicht-direkten Pfad ausführt. Jedoch wenn man /tmp auf noexec setzt, stoppt das etwa 99% aller Script-Kiddies, da deren Exploits so gestaltet sind dass sie direkt von /tmp ausgeführt werden.

Befehlsauflistung 12: /etc/fstab

```
/dev/sda1 /boot ext2 noauto,noatime 1 1
/dev/sda2 none swap sw 0 0
/dev/sda3 / reiserfs notail,noatime 0 0
/dev/sda4 /tmp reiserfs notail,noatime,nodev,nosuid,noexec 0 0
/dev/sda5 /var reiserfs notail,noatime,nodev 0 0
/dev/sda6 /home reiserfs notail,noatime,nodev,nosuid 0 0
/dev/sda7 /usr reiserfs notail,noatime,nodev,ro 0 0
/dev/cdroms /cdrom0 /mnt/cdrom iso9660 noauto,ro 0 0
proc /proc proc defaults 0 0
```

Warnung
Setzt man /tmp in noexec Modus, kann dies dazu führen, dass einige Scripts nicht richtig ausgeführt werden.
Notiz
Plattenquotas werden im Kapitel Quotas behandelt
Notiz
Beachten Sie dass ich /var weder in *noexec* noch in *nosuid* Modus setze, obwohl Dateien von diesem Mountpunkt normalerweise niemals ausgeführt werden. Der Grund dafür ist, dass qmail in /var/qmail installiert ist und berechtigt sein muss eine suid-Datei auszuführen und auf sie zuzugreifen. Ich setze /usr in read-only Modus, da ich hier nichts verändere solange ich Gentoo nicht aktualisiere. Dann mounte ich das Dateisystem erneut in read-write Modus, aktualisiere und mounte dann erneut in read-only.
Notiz
Selbst wenn sie qmail nicht benutzen, braucht Gentoo trotzdem noch die Ausführberechtigung in /var/tmp, da dort ebuilds hergestellt werden. Jedoch kann hierfür ein alternativer Pfad eingerichtet werden, wenn Sie darauf bestehen /var im *noexec* Modus zu betreiben.

6. Einschränkungen für Benutzer/Gruppen

6.1 /etc/security/limits.conf

Die Kontrolle von Ressourcenbegrenzungen kann sehr effektiv sein, wenn es darum geht eine lokale DoS Attacke zu verhindern oder die maximal erlaubten Logins für eine Gruppe oder einen Benutzer zu handhaben.

Befehlsauflistung 13: /etc/security/limits.conf

```
*    soft core      0
*    hard core      0
*    hard nproc     15
*    hard rss       10000
*    -    maxlogins 2
@dev hard core      100000
@dev soft nproc     20
@dev hard nproc     35
@dev -    maxlogins 10
```

Wenn Sie dabei sind den Wert von *nproc* oder *maxlogins* gleich 0 zu setzen, sollten sie diesen Benutzer vielleicht lieber löschen. Das Beispiel oben setzt die Einstellungen für die Gruppe *dev* für Prozesse, Kerndateien und *maxlogins*. Der Rest erhält einen Standardwert.

Notiz
/etc/security/limits.conf ist Teil des PAM Paketes und wird nur auf Pakete angewendet, die PAM benutzen.

6.2 /etc/limits

/etc/limits ist recht ähnlich zur Limit-Datei /etc/security/limits.conf. Der einzige Unterschied ist das Format und daß diese nur auf Benutzern oder Wild-Cards (aber keinen Gruppen) beruht. Werfen wir einen Blick auf die Konfiguration:

Befehlsauflistung 14: /etc/limits

```
*   L2 C0 U15 R10000
kn L10 C100000 U35
```

Hier setzen wir die Standardeinstellungen und eine spezielle Einstellung für den Anwender **kn**. Limits sind ein Teil des Shadow-Paketes und betreffen nur das Shadow-Login-Programm. Es ist nicht notwendig irgendwelche Beschränkungen in dieser Datei zu setzen, wenn Sie die PAM-Einstellung in /etc/make.conf vorgenommen haben und PAM vollständig konfiguriert haben.

6.3 Quotas

Warnung
Stellen Sie sicher, dass ihr Dateisystem Quotas unterstützt. ReiserFS zum Beispiel tut es nicht!

Die Anwendung von Quotas auf einem Dateisystem verhindert, daß Anwender den Datenträger mit Ihren Daten überfüllen oder überhaupt schreiben können. Die Kernel-Option wird bei der Kernelkonfiguration unter *File systems->Quota support* aktiviert. Nehmen Sie die Einstellung vor, kompilieren Sie den Kernel neu und starten Sie mit diesem Ihren Computer neu.

Starten Sie die Installation mit *emerge quota*. Passen Sie Ihre /etc/fstab an, indem Sie usrquota und grpquota bei den Partitionen hinzufügen, für die Sie die Nutzungsbeschränkung festlegen wollen.

Befehlsauflistung 15: /etc/fstab

```
/dev/sda1 /boot ext2 noauto,noatime 1 1
/dev/sda2 none swap sw 0 0
/dev/sda3 / reiserfs notail,noatime 0 0
/dev/sda4 /tmp ext3 notail,noatime,nodev,nosuid,noexec,usrquota,grpquota 0 0
/dev/sda5 /var ext3 notail,noatime,nodev,usrquota,grpquota 0 0
/dev/sda6 /home ext3 notail,noatime,nodev,nosuid,usrquota,grpquota 0 0
/dev/sda7 /usr reiserfs notail,noatime,nodev,ro  0 0
/dev/cdroms/cdrom0 /mnt/cdrom iso9660 noauto,ro 0 0
proc /proc proc defaults 0 0
```

Auf jeder Partition auf der Sie Quotas aktiviert haben, erstellen Sie nun die Quota-Dateien (quota.user und quota.group) und setzen diese in die Wurzel der Partition.

Befehlsauflistung 16: Erstellen der Quota-Dateien

```
# touch /tmp/quota.user
# touch /tmp/quota.group
# chmod 600 /tmp/quota.user
# chmod 600 /tmp/quota.group
```

Dieser Schritt muss auf jeder Partition durchgeführt werden, auf der Quotas aktiviert wurden. Nachdem Sie die Quota-Dateien erstellt und konfiguriert haben, müssen Sie das *quota* Initskript dem Default Runlevel hinzufügen.

Befehlsauflistung 17: Quota zum Default Runlevel hinzufügen

```
# rc-update add quota default
```

Wir werden das System nun so konfigurieren, dass die Quotas einmal wöchentlich gecheckt werden. Dazu fügen Sie folgende Zeile in die /etc/crontab ein.

Befehlsauflistung 18: Quota Check in der crontab

```
0 3 * * 0 /sbin/quotacheck -avug
```

Nachdem Sie den Rechner neu gestartet haben, ist es an der Zeit, die Quotas für die Benutzer und Gruppen festzulegen. *edquota -u kn* wird den in $EDITOR festgelegten Editor starten (Standard ist nano), damit Sie die Quotas des Benutzers kn bearbeiten können. -g wird genau dasselbe, allerdings für Gruppen machen.

Befehlsauflistung 19: Bearbeiten der Quotas für den Benutzer kn

```
Quotas for user kn:
/dev/sda4: blocks in use: 2594, limits (soft = 5000, hard = 6500)
        inodes in use: 356, limits (soft = 1000, hard = 1500)
```

Für weitere Informationen lesen Sie bitte **man edquota** oder Das Quota Mini-Howto

6.4 /etc/login.defs

Wenn die Richtlinie besagt, dass die Anwender jede Woche ihr Passwort ändern müssen, dann setzen Sie die Variable *PASS_MAX_DAYS* auf 14 und *PASS_WARN_AGE* auf 7. Es wird ausserdem empfohlen, dass Sie alternde Passwörter benutzen, da Brute-Force Angriffe jedes Passwort finden können - alles nur eine Frage der Zeit. Wir empfehlen ausserdem, dass Sie *LOG_OK_LOGINS* auf yes setzen.

6.5 /etc/login.access

Die login.access ist auch ein Teil des Shadow-Paketes, dass eine Login Zugangs-Kontrolltabelle anbietet. Die Tabelle wird benutzt um zu kontrollieren, wer und wer nicht einloggen darf, basierend auf dem Benutzernamen, dem Gruppennamen oder dem Hostnamen von dem der Versuch gestartet wird. Normalerweise sind alle Anwender des Systems berechtigt sich anzumelden; aus diesem Grunde ist die Datei nur mit Kommentaren und Beispielen gefüllt. Je nachdem wie Sie Ihren Server oder Ihren Arbeitsplatzrechner schützen empfehlen wir die Datei so anzupassen, das niemand anderes als Sie selbst (also der Administrator) Zugang zur Konsole bekommt.

Notiz
Diese Einstellungen sind nicht für root anwendbar.
Befehlsauflistung 20: /etc/login.access

```
-:ALL EXCEPT wheel sync:console
-:wheel:ALL EXCEPT LOCAL .gentoo.org
```

Wichtig
Seien Sie vorsichtig bei der Bearbeitung der Datei. Wenn Sie nicht aufpassen, dann können Sie sich aussperren wenn Sie nicht über root-Rechte verfügen.
Notiz
Diese Einstellungen wirken sich nicht auf SSH aus, da SSH /bin/login normalerweise nicht ausführt. Dies kann durch die Benutzung von "UseLogin yes" in /etc/ssh/sshd_config ermöglicht werden. Das führt dazu, dass SSH login benutzt und die Einstellungen benutzt werden.

Dies erstellt Loginzugriff so, dass Mitglieder von wheel sich an der Konsole einloggen können oder wenn ihre Quelle die gentoo.org Domäne ist. Vielleicht ein wenig zu paranoid, aber sicher ist sicher.

7. Dateiberechtigungen.

7.1 Von allen lesbar

Normale Benutzer sollten zu Konfigurationsdateien oder Passwörtern keinen Zugang haben. Ein Angreifer kann Passwörter aus einer Datenbank oder von einer Webseite stehlen und verunstalten oder noch schlimmer: Daten löschen. Deswegen ist es notwendig, dass die Berechtigungen korrekt gesetzt sind. Wenn Sie sicher sind, dass eine Datei nur von root benutzt wird, geben Sie ihr die Berechtigung *0600* und ordnen Sie diese mit *chown* dem richtigen Benutzer zu.

7.2 Welt/Gruppen-Schreibbar.

Befehlsauflistung 21: Auffinden von Dateien und Verzeichnissen, die von allen schreibar sind

```
# /usr/bin/find / -type f \( -perm -2 -o -perm -20 \) \
   -exec ls -lg {} \; 2>/dev/null >writable.txt
# /usr/bin/find / -type d \( -perm -2 -o -perm -20 \) \
   -exec ls -ldg {} \; 2>/dev/null >>writable.txt
```

Dies schafft eine riesige Datei mit Berechtigungen von allen Dateien, die entweder Schreibberechtigungen für alle oder eine Gruppe haben. Überprüfen Sie die Berechtigungen und eliminieren Sie die für alle schreibbaren Dateien durch das Ausführen von /bin/chmod o-w für die Dateien.

7.3 SUID/SGID Dateien

SUID/SGID Dateien (Dateien bei denen das superuser bit gestzt wurde) ist ein Weg für normale Benutzer Dinge zu tun die normalerweise nur root darf. Diese Dateien können zu lokalen root-Brüchen führen (wenn sie Sicherheitslöcher enthalten), da so eine Datei mit root Berechtigungen ausgeführt wird. Diese Dateien sind gefährlich und sollten unter allen Umständen vermieden werden. Wenn Sie die Dateien nicht benutzen. Führen sie chmod 0 aus oder entfernen sie das Paket (unmerge) aus dem sie stammen (überprüfen sie das Paket mit *qpkg -f*. Wenn sie es nicht längst installiert haben, tun Sie dies mit *emerge gentoolkit*). Ansonsten schalten Sie das suid bit einfach mit chmod-s ab.

Befehlsauflistung 22: Auffinden von setuid Dateien

```
# /usr/bin/find / -type f \( -perm -004000 -o -perm -002000 \) \
  -exec ls -lg {} \; 2>/dev/null >suidfiles.txt
```

Dies erzeugt eine Datei mit einer Liste aller SUID/SGID Dateien.

Befehlsauflistung 23: Liste der setuid binären Dateien

```
/bin/su
/bin/ping
/bin/mount
/bin/umount
/var/qmail/bin/qmail-queue
/usr/bin/chfn
```

```
/usr/bin/chsh
/usr/bin/crontab
/usr/bin/chage
/usr/bin/expiry
/usr/bin/sperl5.6.1
/usr/bin/newgrp
/usr/bin/passwd
/usr/bin/gpasswd
/usr/bin/procmail
/usr/bin/suidperl
/usr/lib/misc/pt_chown
/usr/sbin/unix_chkpwd
/usr/sbin/traceroute
/usr/sbin/pwdb_chkpwd
```

Standardmäßig hat Gentoo Linux nicht viele SUID Dateien (es hängt davon, was Sie installiert haben), aber Sie könnten eine Liste wie die obige erhalten. Viele dieser Befehle sollten nicht von normalen Benutzern benutzt werden, sondern nur von root. Schalten sie das suid bit bei *ping*, *mount*, *umount*, *chfn*, *chsh*, *newgrp*, *suidperl*, *pt_chown* und *traceroute* aus. Sie tun dies mit dem Befehl *chmod -s* bei jeder einzelnen Datei. Entfernen Sie das bit nicht von *su*, *qmail-queue* oder *unix_chkpwd*. Dies würde dazu führen, dass Sie nicht mehr su benutzen und mail empfangen könnten. Durch entfernen des bits entfernen Sie die Möglichkeit, dass ein normaler User (oder Angreifer) root Zugriff durch eine dieser Dateien erlangen kann.

Die einzigen SUID Dateien die ich auf meinem System habe sind *su*, *passwd*, *gpasswd*, *qmail-queue*, *unix_chkpwd* und *pwdb_chkpwd*. Aber wenn sie X benutzen, könnten sie einige mehr haben, denn X benötigt diesen Zugriff.

8. PAM (Pluggable Authentication Modules)

PAM ist eine Sammlung von shared libraries, die eine Alternative für Authentifizierungen in Programmen darstellen. Die PAM Einstellungen von Gentoo Linux sind relativ vernünftig, aber es gibt immer Platz für Verbesserungen. Zunächst installieren wir cracklib.

Befehlsauflistung 24: Installieren von cracklib

```
# emerge cracklib
```

Befehlsauflistung 25: /etc/pam.d/passwd

```
auth     required pam_pwdb.so shadow nullok
account  required pam_pwdb.so
password required pam_cracklib.so difok=3 retry=3 minlen=8 dcredit=2 ocredit=2
password required pam_pwdb.so md5 use_authtok
session  required pam_pwdb.so
```

Dies fügt die cracklib hinzu, welche sicherstellt, dass der Benutzer eine minimale Passwortlänge von 8 Zeichen benutzt; Bestehend aus mindestens 2 Zahlen, 2 Unterschiedlichen und es müssen mindestens 3 Zeichen anders sein als beim letzten Passwort. Dies zwingt den Benutzer ein gutes Passwort zu wählen (Passwortrichtlinien). In der Dokumentation von PAM finden Sie weitere Optionen.

Befehlsauflistung 26: /etc/pam.d/sshd

```
auth     required pam_pwdb.so nullok
auth     required pam_shells.so
auth     required pam_nologin.so
auth     required pam_env.so
account  required pam_pwdb.so
password required pam_cracklib.so difok=3 retry=3 minlen=8 dcredit=2 ocredit=2 use_authtok
password required pam_pwdb.so shadow md5
session  required pam_pwdb.so
session  required pam_limits.so
```

Jeder andere Dienst der nicht mit einer PAM Datei in /etc/pam.d konfiguriert ist wird die "andere" Regel benutzen. Die Standardeinstellung sind auf *deny* gesetzt, so wie es sein sollte. Jedoch habe ich gerne viele Protokolle und deswegen habe ich *pam_warn.so* hinzugefügt. Die letzte Konfiguration ist *pam_limits* welche von /etc/security/limits.conf kontrolliert wird. Siehe das passende Kapitel hierzu.

Befehlsauflistung 27: /etc/pam.d/other

```
auth     required pam_deny.so
auth     required pam_warn.so
account  required pam_deny.so
account  required pam_warn.so
password required pam_deny.so
password required pam_warn.so
session  required pam_deny.so
session  required pam_warn.so
```

9. TCP Wrappers

Ist ein Weg um Zugang zu kontrollieren für Dienste die normalerweise von inetd ausgeführt werden (welches Gentoo nicht hat) aber es kann auch von xinetd und anderen Diensten benutzt werden.

Notiz
Der Dienst sollte tcpd in seinem Serverargument (in xinetd) asugeführt werden. Schauen Sie für mehr Informationen in das xinetd Kapitel.

Befehlsauflistung 28: /etc/hosts.deny

```
ALL:PARANOID
```

Befehlsauflistung 29: /etc/hosts.allow

```
ALL: LOCAL @wheel
time: LOCAL, .gentoo.org
```

Wie Sie sehen können ist das Format sehr ähnlich dem in /etc/login.access. Tcpd unterstützt einen spezifischen Dienst und sie arbeiten nicht im selben Gebiet von Sicherheit. Diese Einstellungen gelten nur für Dienste die TCP Wrapper benutzen.

Es ist auch möglich Befehle auszuführen wenn auf einen Dienst zugegriffen wird (kann benutzt werden wenn Weiterleiten für Benutzer die sich einwählen aktiviert wird) aber es nicht empfohlen, da Menschen dazu neigen mehr Probleme zu schaffen als sie versuchen zu beheben. Ein Beispiel könnte sein, dass sie ein Script konfigurieren um email zu senden jedes mal wenn jemand die deny-Regel trifft, aber ein Angreifer könnte so eine DoS Attacke ausführen indem er darauf weiter zugreift. Dies schafft viel I/O und viele mails, deswegen tun Sie es nicht! Lesen Sie *man 5 hosts_access* für weitere Informationen.

10. Kernelsicherheit

10.1 Funktionsentfernung

Eine grundlegende Regel ist die Entfernung von allem was sie nicht brauchen. Das schafft einen kleinen Kernel und entfernt auch die Verwundbarkeiten die in Treibern oder anderen Eigenschaften liegen können.

Ziehen Sie auch in betracht loadable module support(=ladbare-Modulunterstützung) auszuschalten. Auch wenn es möglich ist Module ohne diese Eigenschaft hinzuzufügen (root kits), wird es doch schwerer für den normalen Angreifer root kits über Kernelmodule zu installieren.

10.2 /proc (kernel flags)

Viele Kernel Parameter können durch das /proc Dateisystem verändert werden, oder durch die Benutzung von *sysct1*.

Um dynamisch Kernelparameter und -variablen sofort zu ändern benötigen Sie *CONFIG_SYSCTL* in Ihrem Kernel. Die ist voreingestellt im Standard 2.4 Kernel.

Befehlsauflistung 30: Entfernen von ping-Paketen

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Dies sperrt icmp Typ 0 (auch bekannt als Ping) Pakete. Der Grund hierfür ist, dass icmp Nutzlast mit anderen Informationen beinhalten kann als Sie denken. Administratoren benutzen Ping als Diagnoseprogramm und beschweren sich oft wenn sie Ping nicht benutzen können. Es gibt keinen Grund für einen Außenstehenden die Möglichkeit zu haben Ping zu benutzen, aber ab und zu kann es hilfreich für Eingeweihte sein, diese Möglichkeit zu haben. Das Problem kann dadurch gelöst werden, indem man icmp type 0 in der Firewall deaktiviert.

Befehlsauflistung 31: Ignorieren von broadcast-Pings

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

Dies sperrt Antworten auf Anfragen, Sie wollen schliesslich nicht ein Schlumpfverstärker werden. Schlumpfverstärker oder X-mass trees ist eine Methode die es einem Angreifer erlaubt einen moderaten Teil von Traffic zu senden und geradezu eine Explosion von Traffic zu verursachen am beabsichtigten Ziel.

Befehlsauflistung 32: Sperren von source routed Paketen

```
# /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_source_route
```

Akzeptieren sie keine source routed Pakete. Angreifer können source routing benutzen um Traffic zu erzeugen der vorgibt aus dem Netzwerk zu kommen, jedoch weitergeleitet wurde den Pfad von dem es ursprünglich kam. Sperren sie Source Routing denn es wird selten für legitime Zwecke genutzt.

Befehlsauflistung 33: Sperren von Umleitungsakzeptanz

```
# /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects
```

Sperren Sie ICMP Umleitungsakzeptanz. ICMP Umleitungen können benutzt werden um Ihre routing tables zu verändern, möglicherweise zu einem schlimmen Ende.

Befehlsauflistung 34: Schutz gegen bad error messages

```
# /bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
```

Schalten sie den Schutz gegen bad error messages ein.

Befehlsauflistung 35: Ermöglichen von rückwärtiger Pfadfilterung

```
# for i in /proc/sys/net/ipv4/conf/*; do
        /bin/echo "1" > $i/rp_filter
done
```

Notiz
Wenn Sie IP forwarding aktivieren, werden Sie auch dieses Resultat erhalten.

Stellen Sie reverse path filtering an. Dies hilft durch automatisches Ablehnen von Quelladressen, die nicht mit dem Netzwerkinterface übereinstimmen, dabei sicherzustellen, dass Pakete legitime Quelladressen benutzen. Dies hat Sicherheitsvorteile, da es IP Spoofing verhindert.

Warnung
Die Nutzung von reverse path filtering kann auch ein Problem darstellen, wenn sie asymetrisches Routing benutzen (Pakete von Ihnen zu einem Host nehmen einen anderen Weg als Pakete vom host zu Ihnen) oder wenn Sie einen Non-Routing Host betreiben, der verschiedene IP-Adressen an verschiedenen Interfaces hat.

Befehlsauflistung 36: Protokollieren aller spoofed, source routed und umgeleiteten Pakete

```
# /bin/echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
```

Protokollieren von spoofed, source routed und umgeleiteten Pakete.

Befehlsauflistung 37: Aktivieren von IP forwarding

```
# /bin/echo "0" > /proc/sys/net/ipv4/ip_forward
```

Stellen Sie sicher, dass IP forwarding ausgeschaltet ist. Wir wollen es nur für einen multi-homed Host.

Alle diese Einstellungen werden zurückgesetzt, wenn die Maschine neu gestartet wird. Daher schlage ich vor, dass Sie folgendes Script zum *rc-update add procparam default* Default Runlevel hinzufügen und ausführbar *chmod +x /etc/init.d/procparam* machen.

Befehlsauflistung 38: /etc/init.d/procparam

```
#!/sbin/runscript

depend() {
 use checkroot
}
```

```
start() {
 ebegin "Setting /proc options."
 /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_all
 /bin/echo "1" > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
 /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_source_route
 /bin/echo "0" > /proc/sys/net/ipv4/conf/all/accept_redirects
 /bin/echo "1" > /proc/sys/net/ipv4/icmp_ignore_bogus_error_responses
 for i in /proc/sys/net/ipv4/conf/*; do
   /bin/echo "1" > $i/rp_filter
 done
 /bin/echo "1" > /proc/sys/net/ipv4/conf/all/log_martians
 /bin/echo "0" > /proc/sys/net/ipv4/ip_forward
 eend 0
}
```

10.3 Grsecurity

Der Patch von Grsecurity ist Standard im Gentoo Kernel aber per Voreinstellung deaktiviert. So aktivieren Sie ihn:

Konfigurieren Sie Ihren Kernel wie Sie es normalerweise auch würden und konfigurieren Sie dann die Grsecurity Option: (wählen sie customized) und stellen sie folgende Optionen ein:

- Buffer Overflow Protection
  - Openwall non-executable stack
  - GCC trampoline support
- Filesystem Protections
  - Proc restrictions
  - Linking restrictions
  - Secure file descriptors
  - Chroot jail restrictions (aktivieren sie alle Optionen unterhalb dieser)
- Kernel Auditing
  - Log execs within chroot
  - (Un)Mount logging
  - Signal logging
  - Fork failure logging
  - Log set*ids to root
  - Time change logging
- Executable Protections
  - Dmesg restriction
  - Randomized PIDs
  - Altered default IPC permissions (kann verhindern, dass einige Programme korrekt ausgeführt werden)
  - Restricted ptrace
- Network Protections
  - Randomized IP IDs
  - Randomized TCP source ports
  - Altered Ping IDs
  - Randomized TTL
- Miscellaneous Features
  - BSD-style coredumps (erzeugt coredumps wie core.named)

Jetzt kompilieren und installieren sie ihren Kernel mit verbesserter Sicherheit.

10.4 Kerneli

Kerneli ist ein Patch der Verschlüsselung zum existierenden Kernel hinzufügt. Durch patchen des Kernel erhalten Sie neue Optionen wie: Kryptrographische Chiffrierung, Zusammenfassungsalgorithmen und Kryptrographische-Schleifenfilter.

Warnung
Der Kerneli Patch ist momentan nicht in einer stabilen Version für den neuesten Kernel verfügbar, also Vorsicht beim Gebrauch.

10.5 Andere Kernel Patches

- The OpenWall Project (nicht für 2.4 Kernel)
- Linux Intrusion Detection System

- Rule Set Based Access Control
- NSA's security enhanced kernel
- Wolk

Und es gibt wahrscheinlich vieles mehr.

11. Sichern von Diensten

11.1 Benutzung von xinetd

xinetd ist ein Ersatz für inetd (welchen Gentoo nicht hat), den Internet-Dienst-Daemon. Er unterstützt Zugriffskontrolle basierend auf den Adressen der entfernten Hosts und der Zugriffszeit. Es beinhaltet auch ausführliche Protokollfähigkeiten, inklusive Serverstartzeit, Adresse des entfernten Hosts, entfernter Benutzername, Serverlaufzeit und geforderte Abläufe.

Wie bei allen anderen Diensten ist es wichtig eine gute Standardkonfiguration zu haben. Da aber *xinetd* von root benutzt wird und Protokolle unterstützt, von denen Sie möglicherweise die Funktionsweise nicht verstehen, raten wir Ihnen es nicht zu benutzen. Wenn Sie es aber doch benutzen wollen, fügen Sie so mehr Sicherheit hinzu:

Befehlsauflistung 39: Installieren von xinetd

```
# emerge xinetd tcp-wrappers
```

Ergänzen Sie die Konfigurationsdatei um:

Befehlsauflistung 40: /etc/xinetd.conf

```
defaults
{
 only_from      = localhost
 instances      = 10
 log_type       = SYSLOG authpriv info
 log_on_success = HOST PID
 log_on_failure = HOST
 cps            = 25 30
}

# Dies konfiguriert pserver (cvs) durch xinetd mit den folgenden Einstellungen:
# maximal 10 Instanzen (10 Verbindungen gleichzeitig)
# Begrenzung von pserver auf tcp
# benutzen des Benutzer-cvs um diesen Dienst laufen zu lassen
# Anbinden der Schnittstelle an nur 1 IP
# Zulassen von Zugriff von 10.0.0.*
# Begrenzung der Zeit in der Entwickler auf das cvs
# zugreifen können von 08Uhr bis 17Uhr
# Benutzung von tcpd wrappers (Zugriffskontrolle kontrolliert durch
# /etc/hosts.allow und /etc/hosts.deny)
# max_load ist an der Maschine auf 1.0 gesetzt
# die disable flag (sperren) steht auf nein, aber ich bevorzuge sie zu
# haben, für den Fall das es gesperrt werden sollte
service cvspserver
{
 socket_type    = stream
 protocol       = tcp
 instances      = 10
 protocol       = tcp
 wait           = no
 user           = cvs
 bind           = 10.0.0.2
 only_from      = 10.0.0.0
 access_times   = 8:00-17:00
 server         = /usr/sbin/tcpd
 server_args    = /usr/bin/cvs --allow-root=/mnt/cvsdisk/cvsroot pserver
 max_load       = 1.0
 log_on_failure += RECORD
 disable        = no
}
```

Für mehr Informationen lesen Sie *man 5 xinetd.conf*.

11.2 SSH

Die einzige Sicherheitsverstärkung die OpenSSH benötigt, ist eine stärkere Verschlüsselung basierend auf Public Key Verschlüsselung. Zu viele Seiten (wie http://www.sourceforge.net, http://www.php.net und http://www.apache.org) haben wegen Passwortlecks oder schlechten Paswörtern unter unauthorisiertem Eindringen in ihre Systeme gelitten.

Befehlsauflistung 41: /etc/ssh/sshd_config

```
#Aktivieren Sie nur Version 2
Protocol 2

#Kein direkter root Zugriff
PermitRootLogin no

#Benutzung von RSA Schlüsselauthentifizierung
RSAAuthentication yes
PubkeyAuthentication yes
AuthorizedKeysFile      .ssh/authorized_keys

# Sperren von .rhost Dateien und normaler Passwordauthentifizierung
RhostsAuthentication no
PasswordAuthentication no
PermitEmptyPasswords no

AllowHosts *.gentoo.org

#Nur Mitglieder der wheel oder admin erhalten Zugriff
AllowGroups wheel admin

#Von allen Leuten in diesen beide Gruppen erhalten nur kn und ns Zugriff
AllowUsers kn bs

#Hinzufügen des Protokollierungslevels
SyslogFacility AUTH
LogLevel INFO

#bind
ListenAddress 127.0.0.1
```

Jetzt ist das einzige was Ihre Benutzer machen müssen einen Schlüssel zu erstellen (auf der Maschine von der sie sich einloggen wollen) mit folgendem Befehl:

Befehlsauflistung 42: Erstellen eines RSA Schlüsselpaares

```
# /usr/bin/ssh-keygen -t rsa
```

Und tippen Sie einen Passsatz (auch Mantra genannt) ein:

Befehlsauflistung 43: Ausgabe von ssh-keygen

```
Generierung des öffentlichen/privaten rsa Schlüsselpaares.
Geben sie den Dateinamen ein unter dem der Schlüssel gespeichert wird (/home/kn/.ssh/id_rsa):[En
Verzeichnis erstellt '/home/kn/.ssh'.
Passsatz eingeben (leer für keinen Passsatz): [Passsatz eingeben]
Denselben Passsatz erneut eingeben: [Erneut Passsatz eingeben]
Ihre Identifikation wurde in /home/kn/.ssh/id_rsa gespeichert.
Ihr öffentlicher Schlüssel wurde in /home/kn/.ssh/id_rsa.pub gespeichert.
Der Fingerabdruck des Schlüssels ist:
07:24:a9:12:7f:83:7e:af:b8:1f:89:a3:48:29:e2:a4 kn@knielsen
```

Dies fügt zwei Dateien zu Ihrem ~/.ssh/ Verzeichnis mit den Namen id_rsa und id_rsa.pub hinzu. Die Datei id_rsa ist Ihr privater Schlüssel und sollte von anderen Leuten außer Ihnen Ferngehalten werden. Die andere Datei id_rsa.pub soll an jeden Server verteilt werden zu dem Sie Zugriff haben. Fügen Sie den Schlüssel in das home Verzeichnis des Benutzers in ~/.ssh/authorized_keys ein, so sollte der Benutzer die Möglichkeit haben sich einzuloggen.

Ihre Benutzer sollten diesen privaten Schlüssel gut verwahren. Packen Sie es auf ein Medium, dass sie immer mit sich tragen oder lassen Sie es auf ihrer Workstation (fügen Sie dies in die Passwortrichtlinien ein).

Mehr über OpenSSH finden Sie auf der Webseite.

11.3 X

XFree ist von Haus aus als Xserver konfiguriert. Dies kann gefährlich sein, denn X benutzt unverschlüsselte TCP-Verbindungen und wartet auf xclients.

Wichtig
Wenn Sie diesen Dienst nicht brauchen, dann deaktivieren Sie ihn!

Aber wenn Sie Ihren Arbeitsplatz als Xserver betreiben möchten, dann benutzen Sie das Kommando /usr/X11R6/bin/xhost nur mit äußerster Vorsicht. Dieses Kommando erlaubt Clients von anderen Rechnern sich mit Ihrer Anzeige zu verbinden und diese auch zu nutzen. Dies kann sinnvoll sein, wenn Sie eine X-Anwendung von einem anderen Rechner brauchen und die einzige Verbindung zwischen den Rechnern ein Netzwerk ist. Die Syntax lautet /usr/X11R6/bin/xhost +hostname.

Warnung
Benutzen Sie nie das xhost + feature! Dies wird jedem Client erlauben sich mit Ihrem X zu Verbinden und dieses unter Kontrolle bringen. Wenn ein Angreifer Zugang zu Ihrem X erlangt, dann kann er Ihre Tastatureingaben überwachen und Kontrolle über Ihren Desktop erlangen.

Eine sichere Lösung ist, dieses Feature vollständig zu deaktivieren indem Sie X mit *startx -- -nolisten tcp* starten oder dies auf Dauer über eine entsprechende Einstellung in der Konfigurationsdatei verhindern.

Befehlsauflistung 44: /usr/X11R6/bin/startx

```
defaultserverargs="-nolisten tcp"
```

Um sicherzustellen, dass startx bei einem emergen einer neuen XFree Version überschrieben werden kann, müssen Sie die Datei beschützen. Fügen Sie die folgende Zeile in die /etc/make.conf ein:

Befehlsauflistung 45: /etc/make.conf

```
CONFIG_PROTECT_MASK="/usr/X11/bin/startx"
```

Wenn sie einen grafischen Login Manager benutzen, benötigen Sie einen anderen Ansatz.

Für *gdm* (Gnome Display Manager)

Befehlsauflistung 46: /etc/X11/gdm/gdm.conf

```
[server-Standard]
command=/usr/X11R6/bin/X -nolisten tcp
```

Für *xdm* (X Display Manager) und *kdm* (KDE Display Manager)

Befehlsauflistung 47: /etc/X11/xdm/Xservers

```
:0 local /usr/bin/X11/X -nolisten tcp
```

11.4 FTP

Das Benutzen von FTP (File Transfer Protocol) ist im Allgemeinen eine schlechte Idee. Es benutzt unverschlüsselte Daten, lauscht auf zwei Ports (normalerweise 20 und 21), und anonyme Logins sind das, wonach Angreifer gerne suchen (um Warez zu verteilen). Da das ftp-Protokoll einige Sicherheitslücken enthält, benutzen Sie bitte alternativ *sftpd* oder HTTP. Wenn dies nicht möglich sein sollte, dann sichern Sie Ihre Dienste so gut wie nur möglich ab und bereiten Sie sich vor.

11.5 Pure-ftpd

Pure-ftpd ist ein Abkömmling des originalen trollftpd. Für mehr Sicherheit und Funktionalität wurde es von Frank Dennis modifiziert.

Benutzen Sie virtuelle Benutzer (niemals Systemkonten) indem Sie die AUTH-Option aktivieren. Setzen Sie diese auf *-lpuredb:/etc/pureftpd.pdb* und erstellen Sie Ihre Benutzer mittels */usr/bin/pure-pw*.

Befehlsauflistung 48: /etc/conf.d/pure-ftpd

```
## Anzahl der gleichzeitigen Verbindungen - insgesamt und je IP ##
MAX_CONN="-c 30"
MAX_CONN_IP="-C 10"

## Keine Uploads erlauben, wenn die Partition voller als dieser Wert hier ist##
DISK_FULL="-k 90%"

AUTH="-lpuredb:/etc/pureftpd.pdb"

## Diverse andere ##
MISC_OTHER="-A -E -X -U 177:077 -d -4 -L100:5 -I 15"
```

Desweiteren konfigurieren Sie die *MISC_OTHER* Einstellung so, dass folgendes verboten ist: keine Anonyma (*-E*) und chroot auf jeden (*-A*). Benutzer können keine Dateien lesen oder schreiben, die mit einem . (Punkt) beginnen (-X), maximale Leerlaufzeit (*-I*), Rekursion begrenzen (*-L*) und einen sinnvollen *umask*.

Warnung
Benutzen Sie **nicht** die Option *-w* oder *-W* ! Wenn Sie eine Warez Seite möchten, dann hören Sie nun bitte auf dieses Dokument zu lesen!

Mehr dazu gibts auf http://www.pureftpd.org

11.6 Proftpd

Proftpd hat einige Sicherheitsprobleme, aber es hat den Anschein als seien die meisten repariert worden. Weitere Verbesserungen wären:

Befehlsauflistung 49: /etc/proftpd/proftpd.conf

```
ServerName "Mein ftp Daemon"
#Zeigen Sie nicht den Ident des Servers
ServerIdent on "Hau ab!"

#Vereinfacht es virtuelle Benutzer anzulegen
RequireValidShell off

#Benutzen Sie eine alternative Passwort- und Gruppendatei (passwd benutzt das Crypt-Format)
AuthUserFile "/etc/proftpd/passwd"
AuthGroupFile "/etc/proftpd/group"

# Berechtigungen
Umask 077

# Timeouts und Beschränkungen
MaxInstances 30
MaxClients 10 "Nur 10 Verbindungen erlaubt"
MaxClientsPerHost 1 "Sie sind schon eingeloggt"
MaxClientsPerUser 1 "Sie sind schon eingeloggt"
TimeoutStalled 10
TimeoutNoTransfer 20
TimeoutLogin 20

#jeden "chroot"-en
DefaultRoot ~

#nicht als root laufen lassen
User  nobody
Group nogroup

#Jeden Transfer aufzeichnen
TransferLog /var/log/transferlog

#Probleme mit Zeichenersetzung
DenyFilter \*.*/
```

Alles Andere hängt von Ihnen und Ihren Lesekenntnissen ab (http://www.proftpd.org).

11.7 Vsftpd

Vsftpd (dies steht für "Very Secure ftp") ist ein kleiner FTP-Dämon mit einer einfachen Standardkonfiguration. Er ist einfach und hat nicht so viele Funktionen (wie z.B. virtuelle Benutzer) wie sie PureFTP und ProFTP anbieten.

Befehlsauflistung 50: /etc/vsftpd

```
anonymous_enable=NO
local_enable=YES

#nur lesbar
write_enable=NO

#Aufzeichnen von Übertragungen aktivieren
xferlog_std_format=YES

idle_session_timeout=20
data_connection_timeout=20
nopriv_user=nobody

chroot_list_enable=YES
chroot_list_file=/etc/vsftpd/chrootlist

ls_recurse_enable=NO
```

Wie Sie sehen können, gibt es keine Möglichkeit für diesen Dienst individuelle Rechte und eine standardmässige chroot-Aktion zu konfigurieren. Aber wenn es um die anonymen Einstellungen geht, dann entwickelt sich dies zum Vorteil. Manchmal kann es gut sein einen anonymen FTP-Server z.B. zum Verteilen von Open-Source zu haben und dieser Server passt hierfür perfekt.

11.8 Apache

Apache (1.3.26) kommt mit einer recht gut eingestellten Konfigurationsdatei, aber auch hier müssen wir einige Dinge verbessern wie z.B. verbinden mit einer Adresse und Verhindern des Datenverlustes bei der Übertragung. Folgende Optionen sollten sie in der Konfigurationsdatei anpassen:

Wenn Sie *ssl* in Ihrer /etc/make.conf vor der Installation von Apache gesetzt haben, dann sollten Sie Zugang zu einem SSL-fähigen Server haben. Fügen Sie folgende Zeile ein um dieses Feature zu aktivieren.

Befehlsauflistung 51: /etc/conf.d/apache

```
HTTPD_OPTS="-D SSL"
```

Befehlsauflistung 52: /etc/apache/conf/apache.conf

```
#Lassen Sie ihn auf die richtige IP hören
Listen 127.0.0.1
BindAddress 127.0.0.1
#Es ist keine gute Idee nobody oder nogroup für
#jeden Prozess der nicht als root läuft zu benutzen -
#(erstellen Sie den Benutzer apache mit der Gruppe apache)
User apache
Group apache
#Wir halten Apache davon ab, die Serverversion auszuplaudern
ServerSignature Off
ServerTokens min
```

Apache wird mit *--enable-shared=max* und *--enable-module=all* kompiliert. Dies wird von Vorneherein alle Module aktivieren, sodass Sie alle Module in der *LoadModule*-Sektion (also *LoadModule* und *AddModule*) auskommentieren müssen, die Sie nicht benötigen. Starten Sie den Dienst neu, indem Sie */etc/init.d/apache restart* ausführen.

Die Dokumentation gibt es auf http://www.apache.org

11.9 Qmail

Qmail wird als der sicherste Mail-Server angesehen. Er wurde mit Sicherheit (und Paranoia) im Hinterkopf geschrieben. Es erlaubt von Haus aus kein Relaying und hatte seit 1996 kein Sicherheitsloch. Starten Sie einfach ein *emerge qmail* und konfigurieren Sie es danach.

11.10 BIND

Wichtig
Bind is bekannt für seine lausige Sicherheits Vergangenheit und sollte nicht leicht genommen werden. Wie jeder andere Dienst sollte auch Bind **niemals** als root laufen, ändern Sie die Standard Konfiguration für diesen Dienst also bitte nicht.

Sie finden Dokumentation zu Bind beim Internet Software Konsortium. Das "BIND 9 Administrator Reference Manual" ist auch in doc/arm verfügbar.

11.11 Djbdns

Über dkbdns gibt es nicht viel zu sagen - ausser das der Autor bereit ist Geld darauf zu verwetten dass es sicher ist. Also versuchen Sie es : http://www.djbdns.org/. Es arbeitet anders als Bind v.9 aber Sie werden einen Einstieg finden.

11.12 Samba

Samba ist ein Protokoll für Dateiaustausch mit Microsoft/Novell-Netzwerken und es sollte **nicht** über das Internet benutzt werden. Dennoch benötigt es Absicherung.

Befehlsauflistung 53: /etc/samba/smb.conf

```
[global]
  #An ein Interface binden
  interfaces = eth0 10.0.0.1/32

  #Sicherstellen, dass die Passwörter verschlüsselt werden
  encrypt passwords = yes
  directory security mask = 0700

  #Kommunikation von 10.0.0.* erlauben
  hosts allow = 10.0.0.

  #Benutzerauthentifizierung aktivieren
  #(Also nicht den "Share-Mode" benutzen)
  security = user

  #Priviligierte Accounts verbieten
  invalid users = root @wheel

  #Maximale Größe die SMB für einen Share anzeigt (kein Limit)
  max disk size = 102400

  #Richtlinie für Passwörter
  min password length = 8
  null passwords = no

  #Wenn möglich PAM benutzen
  obey pam restrictions = yes
  pam password change = yes
```

Stellen Sie sicher, dass die Berechtigungen für jedes Share richtig eingestellt sind und denken Sie daran die Dokumentation zu lesen.

Starten Sie nun den Server neu und fügen Sie die Benutzer hinzu, die Zugriff auf diesen Server haben sollen. Dies wird über das Aufrufen von /usr/bin/smbpasswd mit dem Parameter -a ermöglicht.

11.13 Chroot oder Virtuelle Server.

Einen Dienst zu chrooten stellt eine Möglichkeit dar einen Dienst (oder Benutzer) auf für ihn vorgesehene Ressourcen zu beschränken und zu verhindern, daß er Zugang zu Bereichen (oder Informationen) erlangt, die zu einem unberechtigten Besitz von root-Rechten führen könnte. Indem man einen Dienst als ein anderer Benutzer als root laufen lässt (nobody, apache, named) kann ein Angreifer nur Zugriff auf die Ressourcen des entsprechenden Accounts bekommen. Dies bedeutet, dass ein Angreifer nie root-Rechte erlangen kann, selbst wenn der entsprechende Dienst eine Sicherheitslücke hätte.

Einige Dienste wie zum Beispiel pure-ftpd und bind haben eingebaute Fähigkeiten für "chrooting" und andere Dienste bieten dies nicht. Wenn der Dienst es anbietet, dann benutzen Sie es, andernfalls müssen Sie in die Materie einsteigen und einen eigenen Benutzer erstellen. Lassen Sie es uns nun versuchen und eine eigene chroot-Umgebung aufbauen. Um einen Einstieg zu finden und zu sehen wie chroot arbeitet versuchen wir es zuerst mit bash (als einfachen Einstieg ins Lernen).

Erstellen Sie das /chroot Verzeichnis mitttels *mkdir /chroot*. Nun müssen wir herausfinden, mit welche dynamischen Bibliotheken *bash* benötigt (wenn sie mit *-static* kompiliert wurde, dann ist dieser Schritt nicht nötig).

Das folgende Kommando wird eine Liste der von bash benutzten Bibliotheken ausgeben.

Befehlsauflistung 54: Benutzte Bibliotheken auflisten

```
# ldd /bin/bash
  libncurses.so.5 => /lib/libncurses.so.5 (0x4001b000)
  libdl.so.2 => /lib/libdl.so.2 (0x40060000)
  libc.so.6 => /lib/libc.so.6 (0x40063000)
  /lib/ld-linux.so.2 => /lib/ld-linux.so.2 (0x40000000)
```

Nun erstellen wir die Umgebung für bash

Befehlsauflistung 55: chroot-Umgebung für bash erstellen

```
# mkdir /chroot/bash
# mkdir /chroot/bash/bin
# mkdir /chroot/bash/lib
```

Nun kopieren wir die von bash benutzten Bibliotheken (/lib) in das chroot und kopieren das Kommando bash in das Verzeichnis /bin im chroot. Dies wird die originale Umgebung herstellen - nur eben mit weniger Funktionalität. Nachdem das Kopieren abgeschlossen wurde versuchen Sie ein *chroot /chroot/bash*. Wenn Sie ein Prompt bekommen, dass als aktuelles Verzeichnis / angibt, dann hat alles funktioniert. Wenn nicht, dann werden Sie unter Umständen eine Fehlermeldung bekommen, die die fehlende Datei angibt. Manche dynamische Bibliotheken bauen aufeinander auf.

Sie werden feststellen, dass innerhalb der chroot-Umgebung nichts anderes als *echo* funktioniert. Dies ist deshalb so, weil wir keine anderen Kommandos in unserer chroot-Umgebung haben und *echo* ein in bash eingebauter Befehl ist.

Dies ist in etwa der Weg den Sie gehen würden, um einen "ge-chrooteten" Dienst zu erstellen. Der einzige Unterschied ist, dass Dienste manchmal auf Geräten und Konfigurationsdateien in /etc basieren. Kopieren Sie diese einfach in die chroot-Umgebung (Geräte können mit cp -a kopiert werden) und editieren Sie das Init-Script sodaß es die chroot-Umgebung vor der Ausführung verwendet. Es kann schwierig sein herauszufinden, welche Konfigurationsdateien und Geräte ein Dienst benutzt. Dies ist der Punkt, an dem *strace* nützlich wird. Starten Sie den Service mit /usr/bin/strace bash und suchen Sie nach open, read, stat und vielleicht noch connect. Dies wird Ihnen eine Idee darüber geben, welche Dateien Sie kopieren müssen. Aber in den meisten Fällen kopieren Sie einfach die passwd-datei (vorher editieren und die Benutzer entfernen, die mit dem Dienst nichts zu tun haben), /dev/zero, /dev/log und /dev/random in die neue Umgebung.

Ein weiterer Weg eine sichere Umgebung zu erstellen besteht darin, eine virtuelle Serverumgebung zu erstellen. Dies wird eine Kopie der exisitierenden Linuxinstallation erstellen und es virtuell booten. Dies bedeutet, dass wenn der Server angegriffen wird in Wirklichkeit nur der virtuelle Server angegriffen wird, nicht die echte Installation.

Beispiele von virtuellen Servern:

- Usermode Linux und ein HowTo über Usermode Linux.

12. Firewalls

12.1 Eine Firewall

Oftmals wird eine Firewall als die ultimative Sicherheitsmassnahme bezeichnet - was aber nur bedingt stimmt. In den meisten Fällen kann eine falsch konfigurierte Firewall ein System sogar noch mehr verunsichern. Eine Firewall ist auch eine Software und sollte genau so wie jeder andere Dienst behandelt werden, denn auch hier können Bugs vorhanden sein (die hier Sicherheitslöcher sind).

Also denken Sie nach, bevor Sie eine Firewall in Betrieb nehmen! Brauchen Sie wirklich eine? Wenn Sie der Meinung sind, daß sie eine brauchen, dann verfassen Sie eine Richtlinie wie sie funktionieren sollte, welcher Art sie sein soll und wer sie betreiben sollte.

Firewalls werden für folgende beiden Zwecke verwendet:

- Um Benutzer (Würmer/Angreifer) draussen zu halten
- Um Benutzer (Angestellte/Kinder) drinnen zu halten

Es gibt im Allgemeinen drei Arten von Firewalls:

- Paket-Filter
- >>>> S.o please translate >>>> Circuit relay
- Applikationsgateway

Eine Firewall sollte auf einer dedizierten Maschine ohne weitere Dienste laufen (und wenn, dann höchstens noch ssh) und so abgesichert werden wie dieser Leitfaden es vorschlägt.

12.2 Paket-Filter

Jeglicher Netzwerkverkehr basiert auf Paketen. Viel Verkehr wird in kleinere Pakete transformiert (da diese einfacher zu handhaben sind) und bei der Ankunft am Ziel wieder in der richtigen Reihenfolge wieder zusammengesetzt. Jedes Paket enthält Informationen darüber wie es wohin transportiert werden soll. Und genau diese Informationen macht sich eine Firewall mit Paketfilter zu nutze. Filtern basiert auf:

- Erlauben oder verbieten von Paketen entsprechend der Quell-/Ziel-IP-Adresse
- Erlauben oder verbieten von Paketen entsprechend des Quell-/Ziel-Ports
- Erlauben oder verbieten von Paketen entsprechend dem verwendeten Protokoll
- Erlauben oder verbieten von Paketen entsprechend von bestimmten Einstellungen im Protokoll

Normalerweise wird nur anhand der Daten im Kopf eines Paketes und nicht im eigentlichen Inhalt vollzogen.

Schwächen:

- Adressinformationen in einem Paket könnten gefälscht (**gespoofed**) an den Sender übermittelt werden
- Daten oder Anfragen im erlaubten Paket könnten ungewollte Daten enthalten die ein Angreifer zu seinen Zwecken benutzen könnte um z.B. Schwächen in den Diensten oder hinter der Firewall zu benutzen.
- Normalerweise kann ein Fehler die Firewall unbrauchbar machen

Vorteile:

- Einfach und schnell zu implementieren
- Kann Warnungen vor Angriffen verursachen, bevor diese stattfinden (erkennen von Portscans)
- Geeignet um SYN-Attacken zu beenden

Beispiele für freie Paketfilter für Linux:

- Iptables
- Ipchains
- SmoothWall

12.3 Circuit Relay

Oder auch Circuit Level Gateways sind Firewalls, die Verbindungen validieren bevor die Erlaubnis für den Datenaustausch erteilt wird. Dies bedeutet, dass Pakete entsprechend dem Inhalt des Paketkopfes erlaubt oder verboten werden; dies aber hängt davon ab, ob die Verbindung an beiden Enden gültig entsprechend konfigurierbaren Regeln ist, bevor sie geöffnet oder Daten ausgetauscht werden. Filtern basiert auf:

- Quell-/Zieladresse
- Quell-/Zielport
- Zeitraum
- Protokoll
- Nutzer

- Passwort

Jeglicher Verkehr wird validiert, überwacht und Verkehr ohne diese Informationen wird verboten.

Schwächen:

- Operiert auf der Transportebene und kann u.U. grundlegende Veränderungen in der Programmierung die normalerweise die Transportfunktionen regelt erfordern.

12.4 Applikationsgateway

Der Gateway auf der Applikationsebene ist ein Proxy für eine Applikation, die Daten mit dem Remotesystem unter Verwendung seiner Clients austauscht. Er wird vor der Öffentlichkeit hinter einer DMZ oder einer Firewall ohne Verbindung zur Aussenwelt gesichert. Der Filter basiert auf:

- Erlauben oder verbieten basierend auf Herkunft/Ziel
- Entsprechend dem Paketinhalt
- Dateizugriff abhängig von Dateityp oder -Erweiterung beschränken

Vorteile:

- Dateien können zwischengespeichert werden - das erhöht die Netzwerkleitung
- Detailliertes aufzeichnen von Verbindungen
- Skaliert perfekt (manche Proxy-Server können die zwischengespeicherten Daten teilen)
- Kein direkter Zugriff von Aussen
- Kann Inhalte "on the fly" modifizieren

Schwächen:

- Die Konfiguration ist Komplex

Applikationsgateways werden als die sicherste Lösung angesehen, da sie nicht als root laufen müssen und Richtung Internet nicht öffentlich sind.

Beispiel eines freien Applikationsgateways:

- Squid

12.5 Iptables

Um iptables ans Laufen zu kriegen, muss es im Kernel aktiviert werden. Ich habe sie als Module eingefügt (das Kommando iptables wird diese wenn benötigt laden) und den Kernel neu kompiliert. Für mehr Informationen zur Konfiguration von iptables lesen Sie Iptables Tutorial Chapter 2: Preparations. Nachdem Sie den Kernel neu kompiliert haben (oder noch während der Kernel kompiliert wird) müssen Sie die iptables-Kommandos hinzufügen. Führen Sie einfach nur *emerge iptables* aus und alles sollte funktionieren.

Nun probieren Sie bitte ob alles funktioniert, indem Sie *iptables -L* ausführen. Wenn irgendetwas nicht funktioniert, dann sollten Sie die Konfiguration nochmals überprüfen.

Iptables ist der neue und extrem verbesserte Paketefilter in Linux 2.4.x. Es ist der Nachfolger von ipchains aus dem Linux 2.2.x Kernel. Eine der großen Verbesserungen ist, dass iptables nun in der Lage ist "stateful" Packet Filtering zu performen. Mit "stateful" Packet filtering is es möglich die Spur jeder errichteten TCP Verbindung zu verfolgen.

Eine TCP Verbindung besteht aus einer Serie von Paketen die Informationen über die Quelladresse, die Zieladresse und einem Zähler der das richtige Zusammensetzen der Daten ermöglicht. TCP ist im Gegensatz zu UDP ein verbindungsorientiertes Protokol, UDP ist verbindungslos.

Bei der Prüfung der Header der TCP Pakete kann ein "stateful" Paketfilter bestimmen, ob ein empfangenes TCP Paket zur einer bestehenden Verbindung gehört oder nicht und das Paket entweder akzeptieren oder wegwerfen.

Mit einem "stateless" Paketfilter is es möglich, dem Paket Filter Pakete durch das manipulieren des TCP Paket Header unterzuschieben die eigentlicht gedroppt werden sollten. Dies kann durch das manipulieren des SYN Flag oder anderer Flags im TCP Header erreicht werden. Mit "stateful" Packet Filtering ist es möglich solche Pakete zu droppen, da Sie keiner bestehenden Verbindung zuzuordnen

sind. Damit wird auch die Möglichkeit von "stealth scans" verhindertn, da solche Pakete ebenfalls keiner bestehenden Verbindung zuzuordnen sind.

Iptables bietet einige weitere Möglichkeiten wie zum Beispiel Wiederholungsbegrenzung (rate limiting). Diese Fähigkeit ist extrem nützlich, wenn man einen sicheren DoS (Denial of Service)-Angriff wie auch einen SYN-Angriff verhindern will.

Eine TCP-Verbindung wird durch einen sogenannten Drei-Wege-Handschlag aufgebaut. Wenn eine TCP Verbindung sendet der Client ein Paket mit einem SYN Flag an den Server. Wenn die Server-Seite das SYN Paket empfängt reagiert sie, in dem sie ein SYN+ACK Paket an den client schickt. Wenn der SYN+ACK vom Client empfangen wird, erkennt dieser wiederum mit einem dritten ACL Paket die Verbindung an.

Ein SYN-Angriff geschieht, wenn nur ein SYN-Paket gesendet wird, aber das Senden des SYN+ACK Paket fehlschlägt. Der Client kann ein Paket mit einer gefälschten IP-Adresse senden, da es keine Antwort benötigt. Der Server wird beim Empfang eines SYN Paket einen Eintrag in die Liste halb-geöffneter Verbindungen machen und dann auf den finalen ACK warten, bis der Eintrag gelöscht wird. Der Queue hat eine begrenzte Anzahl von Slots, wenn alle Slots belegt sind können keine neuen Verbindungen aufgebaut werden. Wenn der ACK nicht innerhalb eines begrenzten Zeitraums beim Server ankommt timed die Verbindung aus, der Eintrag wird aus der Queue gelöscht. Die Timeout Einstellungen variieren, liegen aber typischerweise im Bereich von 30 bis 60 Sekunden oder mehr. Die Clientseite initiert die Attacke durch das Aussenden einer größtmöglichen Zahl von SYN Paketen mit verschiedenen Source IP Adressen. Dadurch wird die Liste halb-geöffneter Verbindungen schnell gefüllt, so dass andere Clients davon abgehalten werden eine Verbindung zu diesem Server aufzubauen.

Hier wird das Ratenlimit besonders hilfreich. Es ist möglich die Anzahl von SYN-Pakten von einer bestimmten Quelle zu begrenzen, aber durch Gebrauch von *-m limit --limit 1/s* begrenzt dies das Limit der SYN-Pakete für eins pro Quelle und daher begrenzt die SYN-Flut auf unsere Ressourcen.

Jetzt einiger praktischer Kram!

Wenn iptables in den Kernel geladen wird, hat es 5 Aufhänger an die Sie ihre Regeln hängen können. Sie heissen *INPUT*, *OUTPUT*, *FORWARD*, *PREROUTING* und *POSTROUTING*. Diese Listen nennt man Ketten, da sie per zugefügter Regel funktionieren und überprüfen die Regeln eine nach der anderen in der Reihenfolge wie sie hinzugefügt wurden. Wenn eine Regel auf ein Paket nicht zutrifft wird es an die nächste Regel weitergeleitet.

Sie können Regeln direkt in die 5 Hauptketten setzen oder Ketten erstellen und diese als Regel zu einer existierenden Kette hinzufügen. Iptables unterstützt die folgenden Optionen.

Option:	**Beschreibung:**
-A	Anhängen
-D	Löschen
-I	Einfügen
-R	Ersetzen
-L	Auflisten
-F	Löscht alle Regeln in der Kette oder in allen Ketten
-Z	Keine Counter in der Kette oder in allen Ketten
-C	Teste dieses Paket an der Kette
-N	Erstellen einer neuen benutzerdefinierten Kette
-X	Löschen einer benutzerdefinierten Kette
-P	Richtlinie der Kette bezüglich des Ziels ändern
-E	Ändern des Kettennamens
-p	Protokoll
-s	Quelladresse/maske
-d	Zieladresse/maske
-i	Eingabename (Ethernetname)
-o	Ausgabename (Ethernetname)
-j	Jump (Ziel für Regel)
-m	Erweiterter Treffer (Kann erweiterung benutzen)
-n	Numerische Ausgabe von Adressen und Ports
-t	Zu ändernde Tabelle
-v	Ausführliche Ausgabe
-x	Zahlen Erweitern (exakte Werte anzeigen)
-f	Nur auf die zweiten oder weitere Pakete achten
-V	Paketversion
--line-numbers	Zeilennummern mit ausgeben

Zuerst werden wir versuchen alle ICMP-Pakete an unsere Maschine zu blocken - nur um uns mit iptables vertraut zu machen.

Befehlsauflistung 56: Alle ICMP-Pakete blocken

```
# iptables -A INPUT -p icmp -j DROP
```

Zuerst legen wir die Kette fest, an die es angehängt werden soll, dann das Protokoll und dann das Ziel. Das Ziel kann eine Benutzer spezifierte Regel oder eines der speziellen Ziele *ACCEPT*, *DROP*, *REJECT*, *LOG*, *QUEUE*, *MASQUERADE* sein. In diesem Fall benutzen wir *DROP* daß das Paket ohne irgendeine Antwort an den Client fallen lässt.

Nun versuchen Sie ein *ping localhost*. Es wird nicht möglich sein, eine Antwort zu bekommen, da das komplette ICMP-Protokoll eingehend geblockt wird. Es wird auch nicht möglich sein, andere Maschinen azupingen, da die Pakete nicht mehr von den anderen Rechnern in unseren Rechner kommen können. Nun leeren Sie die Kette um ICMP wieder zum Laufen zu bekommen.

Befehlsauflistung 57: Alle Regeln leeren (Flush)

```
# iptables -F
```

Nun sehen wir uns die Zustandsmaschinerie in iptables an. Wenn wir eine Prüfung bezüglich des Verbindungszustandes an eth0 haben wollen, könnten wir dies folgendermassen aktivieren:

Befehlsauflistung 58: Pakete die zu einer bereits bestehenden Verbindung gehören akzeptieren

```
# iptables -A INPUT -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
```

Dies wird in der INPUT-Kette alle Pakete die zu einer bereits bestehenden oder einer verwandten Verbindung gehören akzeptieren. Man könnte auch jedes Paket, dass nicht in der Zustandstabelle abgedeckt wurde fallen lassen, indem man *iptables -A INPUT -i eth0 -m state --state INVALID -j DROP* direkt davor aufruft. Das aktiviert die Zustandssteuerung in iptables da es die Erweiterung state lädt. Wenn Sie nun von Aussen mit Ihrer Maschine, dann könnten Sie *--state NEW* benutzen. Iptables enthält einige unterschiedliche Module für unterschiedliche Anwendungszwecke. Einige dieser Module sind:

Modul/ Treffer	Beschreibung	Erweiterte Optionen
mac	Prüfung auf die Quell-MAC-Adressen der eingehenden Pakete.	--mac-source
state	Prüfung auf Zustand	--state (passende Werte sind ESTABLISHED,RELATED, INVALID, NEW)
limit	Trefferrate begrenzen	--limit, --limit-burst
owner	Prüfung auf diverse Fähigkeiten des Paketgernators	--uid-owner userid --gid-owner groupid --pid-owner processid --sid-owner sessionid
unclean	Diverse Gültigkeitsprüfungen auf den Paketen	

Lassen Sie uns nun eine benutzerdefinierte Kette erstellen und in einer der existierenden Ketten einbetten:

Befehlsauflistung 59: Benutzerdefinierte Kette erstellen

```
// Neue Kette mit einer Regel erstellen
# iptables -X mychain
# iptables -N mychain
# iptables -A mychain -i eth0 -m state --state ESTABLISHED,RELATED -j ACCEPT
// Die Standardrichtlinie sagt, dass jeglicher ausgehender Verkehr erlaubt ist, aber eingehender
# iptables -P OUTPUT ACCEPT
# iptables -P INPUT DROP
// And add it to the INPUT chain
# iptables -A INPUT -j mychain
```

Indem man die Regel in die INPUT-Kette einpasst bekommt man die Richtlinie: Alles darf raus, aber alles reinkommende wird verworfen ("gedroppt").

Wenn Sie mehr Dokumentation haben wollen, dann werfen Sie einen Blick auf die Netfilter/iptables Dokumentation

Schauen wir uns nun ein komplettes Beispiel an. In diesem Falle sagt meine Firewall/Gateway Richtline:

- Verbindungen zur Firewall wird nur über SSH erlaubt (Port 22)
- Das lokale Netz soll Zugriff auf HTTP, HTTPS und SSH haben (DNS sollte auch erlaubt sein)
- ICMP-Verkehr könnte kritische Daten enthalten und sollte deswegen nicht erlaubt sein. Natürlich gibt es einige Ausnahmen
- Portscans sollten erkannt und aufgezeichnet werden
- SYN-Angriffe sollten abgewehrt werden
- Jeglicher anderer Verkehr sollte blockiert und aufgezeichnet werden

Befehlsauflistung 60: /etc/init.d/firewall

```
#!/sbin/runscript
IPTABLES=/sbin/iptables
IPTABLESSAVE=/sbin/iptables-save
IPTABLESRESTORE=/sbin/iptables-restore
FIREWALL=/etc/firewall.rules
DNS1=212.242.40.3
DNS2=212.242.40.51
#Innen
IIP=10.0.0.2
IINTERFACE=eth0
LOCAL_NETWORK=10.0.0.0/24
#Aussen
OIP=217.157.156.144
OINTERFACE=eth1

opts="${opts} showstatus panic save restore showoptions rules"

depend() {
  need net procparam
}

rules() {
  stop
  ebegin "Setze interne Regeln"

  einfo "Setze Standardregel auf Fallenlassen"
  $IPTABLES -P FORWARD DROP
  $IPTABLES -P INPUT   DROP
  $IPTABLES -P OUTPUT  DROP

  #Standardregel
  einfo "Erstelle Zustands-Kette"
  $IPTABLES -N allowed-connection
  $IPTABLES -F allowed-connection
  $IPTABLES -A allowed-connection -m state --state ESTABLISHED,RELATED -j ACCEPT
  $IPTABLES -A allowed-connection -i $IINTERFACE -m limit -j LOG --log-prefix \
      "Böses Paket von ${IINTERFACE}:"
  $IPTABLES -A allowed-connection -j DROP

  #ICMP Verkehr
  einfo "Erstelle ICMP-Kette"
  $IPTABLES -N icmp_allowed
  $IPTABLES -F icmp_allowed
  $IPTABLES -A icmp_allowed -m state --state NEW -p icmp --icmp-type \
      time-exceeded -j ACCEPT
  $IPTABLES -A icmp_allowed -m state --state NEW -p icmp --icmp-type \
      destination-unreachable -j ACCEPT
  $IPTABLES -A icmp_allowed -p icmp -j LOG --log-prefix "Bad ICMP traffic:"
  $IPTABLES -A icmp_allowed -p icmp -j DROP

  #Eingehender Verkehr
  einfo "Erstelle Kette für eingehenden SSH-Verkehr"
  $IPTABLES -N allow-ssh-traffic-in
  $IPTABLES -F allow-ssh-traffic-in
  #Flood-Schutz
  $IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
      ALL RST --dport ssh -j ACCEPT
  $IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
      ALL FIN --dport ssh -j ACCEPT
```

```
$IPTABLES -A allow-ssh-traffic-in -m limit --limit 1/second -p tcp --tcp-flags \
    ALL SYN --dport ssh -j ACCEPT
$IPTABLES -A allow-ssh-traffic-in -m state -state RELATED,ESTABLISHED -p tcp -dport ssh -j ACC

#Ausgehender Verkehr
einfo "Erstelle Kette für ausgehenden SSH-Verkehr"
$IPTABLES -N allow-ssh-traffic-out
$IPTABLES -F allow-ssh-traffic-out
$IPTABLES -A allow-ssh-traffic-out -p tcp --dport ssh -j ACCEPT

einfo "Erstelle Kette für ausgehenden DNS-Verkehr"
$IPTABLES -N allow-dns-traffic-out
$IPTABLES -F allow-dns-traffic-out
$IPTABLES -A allow-dns-traffic-out -p udp -d $DNS1 --dport domain \
    -j ACCEPT
$IPTABLES -A allow-dns-traffic-out -p udp -d $DNS2 --dport domain \
   -j ACCEPT

einfo "Creating outgoing http/https traffic chain"
$IPTABLES -N allow-www-traffic-out
$IPTABLES -F allow-www-traffic-out
$IPTABLES -A allow-www-traffic-out -p tcp --dport www -j ACCEPT
$IPTABLES -A allow-www-traffic-out -p tcp --dport https -j ACCEPT

#Portscanner fangen
einfo "Erstelle Portscan-Erkennungs-Kette"
$IPTABLES -N check-flags
$IPTABLES -F check-flags
$IPTABLES -A check-flags -p tcp --tcp-flags ALL FIN,URG,PSH -m limit \
    --limit 5/minute -j LOG --log-level alert --log-prefix "NMAP-XMAS:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL FIN,URG,PSH -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL ALL -m limit --limit \
    5/minute -j LOG --log-level 1 --log-prefix "XMAS:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL ALL -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG \
    -m limit --limit 5/minute -j LOG --log-level 1 --log-prefix "XMAS-PSH:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL SYN,RST,ACK,FIN,URG -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags ALL NONE -m limit \
    --limit 5/minute -j LOG --log-level 1 --log-prefix "NULL_SCAN:"
$IPTABLES -A check-flags -p tcp --tcp-flags ALL NONE -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,RST SYN,RST -m limit \
    --limit 5/minute -j LOG --log-level 5 --log-prefix "SYN/RST:"
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,RST SYN,RST -j DROP
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,FIN SYN,FIN -m limit \
    --limit 5/minute -j LOG --log-level 5 --log-prefix "SYN/FIN:"
$IPTABLES -A check-flags -p tcp --tcp-flags SYN,FIN SYN,FIN -j DROP

# Ungültige Zustände in den Ketten einpassen
einfo "Passe Ketten in INPUT an"
$IPTABLES -A INPUT -m state --state INVALID -j DROP
$IPTABLES -A INPUT -j icmp_allowed
$IPTABLES -A INPUT -j check-flags
$IPTABLES -A INPUT -i lo -j ACCEPT
$IPTABLES -A INPUT -j allow-ssh-traffic-in
$IPTABLES -A INPUT -j allowed-connection

einfo "Passe Ketten in FORWARD an"
$IPTABLES -A FORWARD -m state --state INVALID -j DROP
$IPTABLES -A FORWARD -j icmp_allowed
$IPTABLES -A FORWARD -j check-flags
$IPTABLES -A FORWARD -o lo -j ACCEPT
$IPTABLES -A FORWARD -j allow-ssh-traffic-in
$IPTABLES -A FORWARD -j allow-www-traffic-out
$IPTABLES -A FORWARD -j allowed-connection

einfo "Passe Ketten in OUTPUT an"
$IPTABLES -A OUTPUT -m state --state INVALID -j DROP
$IPTABLES -A OUTPUT -j icmp_allowed
$IPTABLES -A OUTPUT -j check-flags
$IPTABLES -A OUTPUT -o lo -j ACCEPT
$IPTABLES -A OUTPUT -j allow-ssh-traffic-out
$IPTABLES -A OUTPUT -j allow-dns-traffic-out
```

```
  $IPTABLES -A OUTPUT -j allow-www-traffic-out
  $IPTABLES -A OUTPUT -j allowed-connection

  #erlaube den Clients über NAT (Network Address Translation) zu routen
  $IPTABLES -t nat -A POSTROUTING -o $IINTERFACE -j MASQUERADE
  eend $?
}

start() {
  ebegin "Starte Firewall"
  if [ -e "${FIREWALL}" ]; then
    restore
  else
    einfo "${FIREWALL} existiert nicht. Benutze Standardregeln."
    rules
  fi
  eend $?
}

stop() {
  ebegin "Halte Firewall an"
  $IPTABLES -F
  $IPTABLES -t nat -F
  $IPTABLES -X
  $IPTABLES -P FORWARD ACCEPT
  $IPTABLES -P INPUT   ACCEPT
  $IPTABLES -P OUTPUT  ACCEPT
  eend $?
}

showstatus() {
  ebegin "Status"
  $IPTABLES -L -n -v --line-numbers
  einfo "NAT status"
  $IPTABLES -L -n -v --line-numbers -t nat
  eend $?
}

panic() {
  ebegin "Setze Panikregeln"
  $IPTABLES -F
  $IPTABLES -X
  $IPTABLES -t nat -F
  $IPTABLES -P FORWARD DROP
  $IPTABLES -P INPUT   DROP
  $IPTABLES -P OUTPUT  DROP
  $IPTABLES -A INPUT -i lo -j ACCEPT
  $IPTABLES -A OUTPUT -o lo -j ACCEPT
  eend $?
}

save() {
  ebegin "Sichere Firewallregeln"
  $IPTABLESSAVE > $FIREWALL
  eend $?
}

restore() {
  ebegin "Stelle Firewallregeln wieder her"
  $IPTABLESRESTORE < $FIREWALL
  eend $?
}

restart() {
  svc_stop; svc_start
}

showoptions() {
  echo "Benutzung: $0 {start|save|restore|panic|stop|restart|showstatus}"
  echo "start)      Wird die Standardeinstellung wieder herstellen oder anderndfalls zu Regeln z
  echo "stop)       Alle Regeln löschen und alles akzeptieren"
  echo "rules)      zu Einstellungen der neuen regeln zwingen"
```

```
  echo "save)       speichert die Regeln in ${FIREWALL}"
  echo "restore)    stellt die Regeln von ${FIREWALL} wieder her"
  echo "showstatus) Status anzeigen"
}
```

Kostenloser Ratschlag für das Erstellen einer Firewall:

- Erstellen Sie vor der Implementierung eine Richtlinie für die Firewall
- Halten Sie sie einfach
- Erlangen Sie Wissen über die Protokolle (lesen Sie das RFC (Request For Comments))
- Denken Sie daran, dass eine Firewall ein weiteres Paket ist, dass als root läuft
- Testen Sie die Firewall

Wenn Sie denken, dass iptables schwer zu verstehen sind oder es zu lange dauert eine sinnvolle Firewall zu erstellen, dann könnten Sie auch Shorewall benutzen. Es benutzt im Grunde genommen iptables um eine Firewall zu erstellen, aber es konzentriert sich auf Regeln und nicht auf spezielle Protokolle.

12.6 Squid

Squid ist ein sehr leistungsstarker Proxy Server, er hat eingebaute Filter, lehnt Traffic auf Grund folgender Merkmale ab: Zeit, regelmäßiger Ausdruckspfad/URI, Quell- und Zieladresse (IP), Domäne, Browser, der authentifizierte Benutzername, Mime-Typ und Port (Protokoll). Wahrscheinlich habe ich einige Funktionen vergessen, aber es ist schwer jede Funktion der gesammten Funktionsliste abzudecken.

Im folgenden Beispiel habe ich einen Banner Filter hinzugefügt anstatt eines Filters basiernd auf pornographischen Seiten. Der Grund dafür ist, dass Gentoo.org **nicht** als eine pornographische Seite aufgelistet werden sollte. Ausserdem will ich meine Zeit nicht damit verbringen einige gute Seiten für Sie zu finden.

In diesen Fall diktiert meine Richtlinie:

- Surfen (HTTP/HTTPS) ist während der Arbeitszeiten erlaubt (MO-FR 8-16 und SA 8-13), wenn sie länger da sind, sollten sie arbeiten und nicht surfen.
- Downloaden ist nicht erlaubt (.exe, .com, .arj, .zip, .asf, .avi, .mpg, .mpeg etc.)
- Banner sind unerwünscht, daher werden sie herausgefiltert und mit einem transparenten GIF ersetzt (hier können Sie kreativ werden!)
- Jede andere kommende oder gehende Verbidung mit dem Internet ist nicht erlaubt.

Dies wird in 4 **einfachen** Schritten implementiert .

Befehlsauflistung 61: /etc/squid/squid.conf

```
# Anbinden an eine IP und einen Port
http_port 10.0.2.1:3128

# Standardkonfiguration
hierarchy_stoplist cgi-bin ?
acl QUERY urlpath_regex cgi-bin \?
no_cache deny QUERY

# Hinzufügen von grundlegenden Listen der Zugriffskontrolle
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

# Hinzufügen wer auf diesen Proxy Server zugreifen kann
acl localnet src 10.0.0.0/255.255.0.0

# Und welche Ports
acl SSL_ports port 443
acl Safe_ports port 80
acl Safe_ports port 443
acl purge method PURGE

# Hinzufügen von Listen zur Zugriffskontrolle basierend
# auf regelmäßigen Ausdrücken innerhalb von URLs
acl archives urlpath_regex "/etc/squid/files.acl"
acl url_ads url_regex "/etc/squid/banner-ads.acl"
```

```
# Hinzufügen von Listen zur Zugriffskontrolle basierend
# auf Datum und Uhrzeit
acl restricted_weekdays time MTWHF 8:00-17:00
acl restricted_weekends time A 8:00-13:00

acl CONNECT method CONNECT

# Erlauben von Managmentzugriff von Localhost
http_access allow manager localhost
http_access deny manager

# Nur Purge Anfragen von Localhost erlauben
http_access allow purge localhost
http_access deny purge

# Verweigern von Anfragen an unbekannte Ports
http_access deny !Safe_ports

# Verweigern von CONNECT an alle ausser SSL Ports
http_access deny CONNECT !SSL_ports

# Meine eigenen Regeln


# Hinzufügen einer Seite zur Darstellung,
# wenn ein Banner entfernt wurde
deny_info NOTE_ADS_FILTERED url_ads

# Dann diese verweigern
http_access deny url_ads

# Verweigern aller Archive
http_access deny archives

# Begrenzung des Zugriffs auf Arbeitszeiten
http_access allow localnet restricted_weekdays
http_access allow localnet restricted_weekends

# Verweigern von allem anderen
http_access deny all
```

Als nächstes fügen Sie alle Dateitypen ein, von denen sie wollen, dass ihre Benutzer sie nicht herunterladen können. Ich habe zip, viv, exe, mp3, rar, ace, avi, mov, mpg, mpeg, au, ra, arj, tar, gz und z Dateien gewählt.

Befehlsauflistung 62: /etc/squid/files.acl

```
\.[Zz][Ii][pP]$
\.[Vv][Ii][Vv].*
\.[Ee][Xx][Ee]$
\.[Mm][Pp]3$
\.[Rr][Aa][Rr]$
\.[Aa][Cc][Ee]$
\.[Aa][Ss][Ff]$
\.[Aa][Vv][Ii]$
\.[Mm][Oo][Vv]$
\.[Mm][Pp][Gg]$
\.[Mm][Pp][Ee][Gg]$
\.[Aa][Uu]$
\.[Rr][Aa]$
\.[Aa][Rr][Jj]$
\.[Tt][Aa][Rr]$
\.[GgZz]$
\.[Zz]$
```

Notiz
Beachten Sie bitte die [] mit Gross- und Kleinbuchstaben fuer jeden Buchstaben. Dies dient dazu, dass niemand es umgehen kann indem er eine Datei mit AvI abruft anstatt avi.

Als nächstes fügen wir die regelmäßigen Ausdrücke ein um Banner zu identifizieren. Sie werden wahrscheinlich viel kreativer sein als ich:

Befehlsauflistung 63: /etc/squid/banner-ads.acl

```
/adv/.*\.gif$
/[Aa]ds/.*\.gif$
/[Aa]d[Pp]ix/
/[Aa]d[Ss]erver
/[Aa][Dd]/.*\.[GgJj][IiPp][FfGg]$
/[Bb]annerads/
/adbanner.*\.[GgJj][IiPp][FfGg]$
/images/ad/
/reklame/
/RealMedia/ads/.*
^http://www\.submit-it.*
^http://www\.eads.*
^http://ads\.
^http://ad\.
^http://ads02\.
^http://adaver.*\.
^http://adforce\.
adbot\.com
/ads/.*\.gif.*
_ad\..*cgi
/Banners/
/SmartBanner/
/Ads/Media/Images/
^http://static\.wired\.com/advertising/
^http://*\.dejanews\.com/ads/
^http://adfu\.blockstackers\.com/
^http://ads2\.zdnet\.com/adverts
^http://www2\.burstnet\.com/gifs/
^http://www.\.valueclick\.com/cgi-bin/cycle
^http://www\.altavista\.com/av/gifs/ie_horiz\.gif
```

Nun der letzte Teil: Wir wollen diese Datei anzeigen, wenn das Banner entfernt wird. Es ist grundlegend eine halbe HTML Datei mit einem 4x4 transparenten GIF Bild.

Befehlsauflistung 64: /etc/squid/errors/NOTE_ADS_FILTERED

```
<HTML>
<HEAD>
<META HTTP-EQUIV="REFRESH" CONTENT="0; URL=http://www.insecurity.dk/images/4x4.gif">
<TITLE>FEHLER: Die angeforderte URL konnte nicht angezeigt werden</TITLE>
</HEAD>
<BODY>
<H1>Anzeige gefiltert!</H1>
```

Notiz
Schliessen sie die <HTML> <BODY> Tags nicht. Dies wird von Squid erledigt.

Wie Sie sehen können hat Squid eine Vielzahl von Möglichkeiten und ist sehr effektiv zum Filtern und als Proxy. Es kann sogar alternative Squid Proxys benutzen um an sehr grosse Netzwerke angepasst zu werden. Die Konfiguration, die ich hier aufgelistet habe ist hauptsächlich fuer kleine Netzwerke mit 1-20 Benutzern geeignet.

Jedoch die Kombinierung von Paketfilterung (iptables) und dem Anwendungsgateway (squid) ist wahrscheinlich die beste Lösung, selbst wenn Squid selber an einem sicheren Ort stationiert ist und niemand von ausserhalb darauf zugreifen kann. Wir müssen uns weiterhin Gedanken machen um Angriffe von Innen.

Nun müssen Sie den Proxy Server in die Einstellungen des Browsers Ihrer Benutzers einbinden. Das Gateway verhindert, dass die Benutzer jeglichen Kontakt mit der Aussenwelt haben, solange sie nicht den Proxy benutzen.

Notiz
In Mozilla geschieht dies in Bearbeiten->Einstellungen->Erweitert->Proxies (bzw. Edit->Preferences->Advanced->Proxies).

Es kann auch transparent geschehen, indem man iptables benutzt um den gesamten Traffic an einen Squid Proxy weiterzuleiten. Dies kann erreicht werden indem man eine Weiterleitungs/Prerouting Regel fürs Gateway hinzufügt:

Befehlsauflistung 65: Ermöglichen von Portweiterleitung an unseren Proxy Server

```
# iptables -t nat -A PREROUTING -p tcp --dport 80 -j DNAT --to proxyhost:3128
# iptables -t nat -A PREROUTING -p tcp --dport 443 -j DNAT --to proxyhost:3128
```

12.7 Also was haben wir gelernt?

Wir lernten, dass:

- Eine Firewall selbst ein Risiko sein kann. Eine schlecht konfigurierte Firewall ist schlechter als überhaupt keine.
- Wie man ein grundlegendes Gateway und einen transparenten Proxy erstellt
- Der Schlüssel zu einer guten Firewall ist das Protokoll zu kennen, dass Sie zulassen wollen
- Dass IP Traffic nicht immer legitime Daten beinhaltet, z.B. ein ICMP Paket mit zusätzlicher Nutzlast
- Wie man SYN Angriffe vereitelt
- Filtern von HTTP Traffic indem man anstössige Bilder und Downloaden von Viren entfernt.
- Kombinieren von Paketfiltern und Anwendungsgateways geben eine bessere Kontrolle.

Nun, wenn sie **wirklich** müssen, schaffen Sie sich eine Firewall an, die ihre Bedürfnisse deckt.

13. Auffinden von Eindringlingen

13.1 Aide (Advanced Intrusion Detection Environment)

Aide ist ein Host basierendes Eindringlingserkennungssystem (eine kostenlose Alternative zu Tripwire). Und wenn Sie bereits mit Tripwire vertraut sind, sollten sie keine Schwierigkeiten haben die Konfigurationsdateien für Aide zu erlernen.

Die Konfigurationsdatei basiert auf regelmäßigen Ausdrücken, Makros und Regeln für Dateien und Verzeichnisse. Wir haben die folgenden Makros:

Makro	Beschreibung	Syntax
ifdef	wenn definiert	@@ifdef "name"
ifndef	wenn nicht definiert	@@ifndef "name"
define	definiert eine Variable	@@define "name" "value"
undef	undefiniert eine Variable	@@undef "name"
ifhost	wenn "hostname"	@@ifhost "hostname"
ifnhost	wenn "hostname" nicht	@@ifnhost "hostname"
endif	Endif muss benutzt werden nach jedem der obrigen Makros ausser define und undef	@@endif

Diese Makros sind sehr praktisch, wenn sie mehr als ein Gentoo System haben und auf allen Aide benutzen wollen. Aber nicht alle Maschinen oder vielleicht sogar Benutzer benuten denselben Dienst.

Als nächstes haben wir Gruppen von Flags um überprüfungen an Dateien und Ordnern durchzuführen. Diese sind eine Kombination aus Berechtigungen, Dateieigenschaften und kryptographischen Hashes/ Checksummen.

Flag	Beschreibung
p	Berechtigungen
i	Inode
n	Anzahl der Links
u	Benutzer
g	Gruppe
s	Grösse
b	Blockzahl
m	mtime
a	atime
c	ctime
S	überprüfung ob die Grösse wächst
md5	MD5 Checksumme
sha1	SHA1 Checksumme
rmd160	RMD160 Checksumme
tiger	Tiger Checksumme
R	p+i+n+u+g+s+m+c+md5
L	p+i+n+u+g
E	Leere Gruppe
>	Wachsende Protokolldatei p+u+g+i+n+S

Und wenn Aide mit mhash Unterstützung kompiliert ist, hat es noch einige weitere Funktionen:

Flag Beschreibung
haval HAVAL Checksumme
gost GOST Checksumme
crc32 CRC32 Checksumme

Nun können sie ihre eigenen auf den oben genannten Flags basierenden Regeln definieren, indem Sie diese folgendermassen kombinieren:

Befehlsauflistung 66: Erstellen eines Regelsatzes für AIDE

```
All=R+a+sha1+rmd160
Norm=s+n+b+md5+sha1+rmd160
```

Das letzte was wir tun müssen um unsere eigene Konfigurationsdatei zu erstellen ist zu schauen wie man diese Regeln einer Datei oder einem Verzeichnis hinzufügt. Grundlegend tippen sie einfach den Datei- oder Verzeichnisnamen und die Regel ein. Aide wird alle Dateien rekursiv hinzufügen, solange Sie nicht etwas anderes angeben.

FlagBeschreibung
! Diese Datei oder dieses Veriechnis nicht hinzufügen.
= Dieses Verzeichnis hinzufügen, aber nicht rekursiv.

Lassen sie uns also ein vollständiges Beispiel betrachten

Befehlsauflistung 67: /etc/aide/aide.conf

```
@@ifndef TOP DIR
@@define TOPDIR /
@@endif

@@ifndef AIDEDIR
@@define AIDEDIR /etc/aide
@@endif

@@ifhost smbserv
@@define smbactive
@@endif

# Der Ort der Datenbank die gelesen werden soll.
database=file:@@{AIDEDIR}/aide.db

# Der Ort der Datenbank, die erstellt werden soll.
database_out=file:aide.db.new

verbose=20
report_url=stdout

# Regeldefinition
All=R+a+sha1+rmd160
Norm=s+n+b+md5+sha1+rmd160

@@{TOPDIR} Norm
!@@{TOPDIR}etc/aide
!@@{TOPDIR}dev
!@@{TOPDIR}proc
!@@{TOPDIR}root
!@@{TOPDIR}tmp
!@@{TOPDIR}var/log
!@@{TOPDIR}var/run
!@@{TOPDIR}usr/portage
@@ifdef smbactive
!@@{TOPDIR}etc/smb/private/secrets.tdb
@@endif
=@@{TOPDIR}home Norm
```

Im obrigen Beispiel definieren wir einige Makros, die angeben wo das topdir startet und wo das Aide Verzeichnis ist. Aide überprüft die /etc/aide/aide.db Datei wenn die Intigrität einer Datei überprüft wird. Jedoch wenn ein Update vorgenommen wird oder eine neue Datei erstellt wird, speichert es die Informationen in /etc/aide/aide.db.new. Dies geschieht, damit die ursprünglich Datenbankdatei nicht automatisch überschrieben wird. Die Option report_URL ist eine "noch kommende" Funktion die

wirklich noch keine Bedeutung hat. Die Absicht der Autoren war es, dass es möglich wäre eine Email zu senden oder vielleicht sogar ein Script auszuführen.

Nach der Konfiguration sollten Sie Ihre Datenbankdatei erstellen indem Sie aide -i ausführen und dann die Datei /etc/aide/aide.db.new nach /etc/aide/aide.db kopieren und den Check zu cron hinzufügen durch crontab -e als root.

Notiz
Abhängig von ihren CPU, Festplattenzugriff und den benutzten Flags für Dateien, kann dies einige Zeit in Anspruch nehmen

Befehlsauflistung 68: Aide als cronjob einrichten

```
0 3   * * * /usr/bin/aide -u
```

Notiz
Denken sie daran es so einzustellen, dass sie die Post für root bekommen. Ansonsten werden Sie niemals wissen was Aide berichtet.

In diesem Fall läuft es einmal um 03 Uhr. Dies geschieht dann, denn ich will die Benutzer beim Arbeiten nicht stören. Beachten Sie, dass ich die *-u* (Update) Option benutzen statt *-C* (überprüfen). Denn *-u* überprüft auch die Dateien und überschreibt nicht die Originaldatenbankdatei, was Zeit spart, denn so müssen sie nur eine Datei kopieren, wenn Veränderungen festgestellt werden. überprüfen sie einfach die Veränderungen um zu sehen, ob Sie es selbst waren oder ein Angreifer der die Veränderungen gemacht hat bevor sie diese kopieren!

Nun gibt es einige Probleme damit die Datenbankdateien lokal zu speichern, denn der Angreifer wird (Wenn er weiss, dass Aide installiert ist) höchstwahrscheinlich versuchen die Datenbankdatei zu verändern, ein Update bei der Datenbankdatei durchzuführen oder /usr/bin/aide zu verändern. Deswegen sollten sie eine CD erstellen oder anderes Medium auf das Sie eine Kopie der Datenbankdatei und der Aide Binärdateien ablegen.

Weitere Informationen gibt es auf der AIDE Projektseite.

13.2 Snort

Snort ist ein "Network Intrusion Detection System (NIDS)". Zur Installation und Konfiguration benutzen Sie die folgenden Beispiele.

Befehlsauflistung 69: Dem System einen Anwender hinzufügen

```
# user add snort -d /var/log/snort -s /dev/null
# chown -R snort /var/log/snort
```

Befehlsauflistung 70: /etc/conf.d/snort

```
PID FILE=/var/run/snort_eth0.pid
MODE="full"
NETWORK="10.0.0.0/24"
LOG DIR="/var/log/snort"
CONF=/etc/snort/snort.conf
SNORT_OPTS="-D -s -u snort -dev -l $LOGDIR -h $NETWORK -c $CONF"
```

Befehlsauflistung 71: /etc/snort/snort.conf

```
// Schritt 1
var HOME_NET 10.0.0.0/24
var EXTERNAL_NET any
var SMTP $HOME_NET
var HTTP_SERVERS $HOME_NET
var SQL_SERVERS $HOME_NET
var DNS_SERVERS [10.0.0.2/32,212.242.40.51/32]
var RULE_PATH ./

// Schritt 2
preprocessor frag2
preprocessor stream4: detect_scans detect_state_problems detect_scans disable_evasion_alerts
preprocessor stream4_reassemble: ports all
preprocessor http_decode: 80 8080 unicode iis_alt_unicode double_encode iis_flip_slash full_whit
preprocessor rpc_decode: 111 32771
preprocessor bo: -nobrute
preprocessor telnet_decode
```

```
// Schritt 3
include classification.config

// Schritt 4
include $RULE_PATH/bad-traffic.rules
include $RULE_PATH/exploit.rules
include $RULE_PATH/scan.rules
include $RULE_PATH/finger.rules
include $RULE_PATH/ftp.rules
include $RULE_PATH/telnet.rules
include $RULE_PATH/smtp.rules
include $RULE_PATH/rpc.rules
include $RULE_PATH/rservices.rules
include $RULE_PATH/dos.rules
include $RULE_PATH/ddos.rules
include $RULE_PATH/dns.rules
include $RULE_PATH/tftp.rules
include $RULE_PATH/web-cgi.rules
include $RULE_PATH/web-coldfusion.rules
include $RULE_PATH/web-iis.rules
include $RULE_PATH/web-frontpage.rules
include $RULE_PATH/web-misc.rules
include $RULE_PATH/web-attacks.rules
include $RULE_PATH/sql.rules
include $RULE_PATH/x11.rules
include $RULE_PATH/icmp.rules
include $RULE_PATH/netbios.rules
include $RULE_PATH/misc.rules
include $RULE_PATH/attack-responses.rules
include $RULE_PATH/backdoor.rules
include $RULE_PATH/shellcode.rules
include $RULE_PATH/policy.rules
include $RULE_PATH/porn.rules
include $RULE_PATH/info.rules
include $RULE_PATH/icmp-info.rules
include $RULE_PATH/virus.rules
# include $RULE_PATH/experimental.rules
include $RULE_PATH/local.rules
```

Befehlsauflistung 72: /etc/snort/classification.config

```
config classification: not-suspicious,Not Suspicious Traffic,3
config classification: unknown,Unknown Traffic,3
config classification: bad-unknown,Potentially Bad Traffic, 2
config classification: attempted-recon,Attempted Information Leak,2
config classification: successful-recon-limited,Information Leak,2
config classification: successful-recon-largescale,Large Scale Information Leak,2
config classification: attempted-dos,Attempted Denial of Service,2
config classification: successful-dos,Denial of Service,2
config classification: attempted-user,Attempted User Privilege Gain,1
config classification: unsuccessful-user,Unsuccessful User Privilege Gain,1
config classification: successful-user,Successful User Privilege Gain,1
config classification: attempted-admin,Attempted Administrator Privilege Gain,1
config classification: successful-admin,Successful Administrator Privilege Gain,1

# NEW CLASSIFICATIONS
config classification: rpc-portmap-decode,Decode of an RPC Query,2
config classification: shellcode-detect,Executable code was detected,1
config classification: string-detect,A suspicious string was detected,3
config classification: suspicious-filename-detect,A suspicious filename was detected,2
config classification: suspicious-login,An attempted login using a suspicious username was detec
config classification: system-call-detect,A system call was detected,2
config classification: tcp-connection,A TCP connection was detected,4
config classification: trojan-activity,A Network Trojan was detected, 1
config classification: unusual-client-port-connection,A client was using an unusual port,2
config classification: network-scan,Detection of a Network Scan,3
config classification: denial-of-service,Detection of a Denial of Service Attack,2
config classification: non-standard-protocol,Detection of a non-standard protocol or event,2
config classification: protocol-command-decode,Generic Protocol Command Decode,3
config classification: web-application-activity,access to a potentially vulnerable web applicati
config classification: web-application-attack,Web Application Attack,1
```

```
config classification: misc-activity,Misc activity,3
config classification: misc-attack,Misc Attack,2
config classification: icmp-event,Generic ICMP event,3
config classification: kickass-porn,SCORE! Get the lotion!,1
```

Weitere Informationen gibt es auf der Snorts Webseite.