Global Information Assurance Certification Paper

SANS GCUX Practical Assignment:

Securing Unix Step by Step

Installation of a Secure User-Chrooted SSH Jumphost

Roland Mathis

September 20, 2004

Contents

1 Abstract

This paper describes the setup of a secure SSH jumphost. Besides the usual hardening tasks two additional layers of security are put in place. After a successful authentication the non-privileged users are locked up in a chrooted environment, also called jail, where the users are only allowed to jump via SSH, Telnet or FTP to other servers. They can change their password that required the setup of a daemon (Monit) which monitors the chrooted password files and updates the system files accordingly. In addition the destination servers and ports a user can jump to are restricted with the Owner-socketlookup patch for Iptables.

All user names and sensitive IP addresses have been changed.

2 System specification and risk mitigation plan

2.1 Jumphost definition

The term "jumphost" describes the logical location and the purpose of the server. Its purpose is to let co-workers and contractors log in and "jump" via SSH, Telnet or FTP to another device in the same network. It is used to do maintenance work on servers in a demilitarized zone. Authorized personnel has to have the possibility to maintain servers from anywhere in the world. Without a middle server like a jumphost the maintenance ports had to be opened on each production server. From a security perspective this can not be tolerated. An additional layer of security had to be put into place to mitigate the risk of a break-in on a production server.

Source	**Destination**	Port
Internet	Production Server	TCP/80 (HTTP)
Internet	Jumphost	TCP/22 (SSH)
Jumphost	Production Server	TCP/22 (SSH)

Advantages of a jumphost:

- Dedication to one task only which allows stricter hardening.
- Centralization of user database. Only one place to disable a user.
- Just a few daemons have to run which mitigates privilege escalation.
- In case of a compromise of the jumphost the production servers still work.

2.2 Users

There are two groups of users allowed to log into the jumphost.

The first group are the system administrators. They access the jumphost to do maintenance tasks like adding and removing users and keeping the system up-to-date. More details about these tasks are described in section 8. As there are only a few system administrators and they all need to do the same tasks they are allowed to become root.

The second group are the jailed users. As soon as they were successfully authorized by the jumphost they find themselves in a restricted environment. In this environment only a few commands are available and other restrictions like disk space limits are set.

2.3 Hardware

The hardware used for the server is a Sun Netra Ultrasparc T1 105:

- 440 Mhz CPU
- 500 MB RAM
- 2 hard disks with 4 GB each

This hardware was obviously not chosen for its high performance. We had some servers left from former mail servers that had been replaced with faster machines. To run the SSH daemon these machines are sufficient.

There is probably a little security related advantage of using a Sparc/Linux platform because there are fewer pre-compiled binaries available which could prevent automated worms from causing further damage after a successful break-in. As soon as a worm or attacker is equipped with runnable binaries or shell scripts this does not matter any longer.

In general the installation of Linux on the Sparc hardware was not any different once the installation media had been booted. There are differences in the boot loader configuration which will be discussed later.

To keep the list of installed packages as small as possible an identical Netra T1 has been setup to serve as building host.

2.4 Software

Some of the important software components are briefly introduced in this section. A list of all installed packages will follow later in this paper.

2.4.1 Kernel, Grsecurity and Iptables

At the time of installation the newest Kernel from the 2.4 tree was 2.4.26. In addition to only enable the necessary Kernel features the Kernel was patched with the Owner-socketlookup patch from the extra repository of Netfilter/Iptables. This feature allows filtering on a per-user basis. Also the Kernel has been patched with the Grsecurity to add Kernel level security. In the case of the jumphost the Grsecurity patches suits very well, because it protects from known Chroot breakouts.

2.4.2 Distribution

The choice of the operating system was Debian/Linux. It is the standard Linux distribution used in the enterprise. It has proven to install and run nicely on the Sparcs. The text mode installation can be done via serial connection. The main reason to go for Debian is its package management which makes it really easy to keep the system up-to-date. In general a Debian installation has a very small footprint in comparison with other big distributions. This simplifies the removal of unneeded packages. There could have been other choices, but Debian did the task.

2.4.3 Openssh

Openssh was the natural choice to provide the users with a familiar syntax and to comply with our standards. Its wide spread use has the advantage that security updates are made available quickly. The downside of Openssh's popularity is that vulnerabilities are sought upon and found fairly often. To enhance security Openssh has been patched with the Openssh-chroot patch, which locks the user into a restricted area after he successfully logged in. From within that restricted area only a few commands can be run and permissions are strict.

2.4.4 Quota

The jumphost is required to provide disk space for file transfers. The jailed users' home directories are put on a separate partition and to ensure availability to all users, available disk space for each of them has been restricted. This regulation is done by Quota-tools.

2.4.5 Software summary

Description	Version	URL
Debian Linux	3.0r2	http://www.debian.org
Linux Kernel	2.4.26	http://www.kernel.org/pub/linux/kernel/v2.4
Grsecurity	2.0-2.4.26	http://www.grsecurity.net
Iptables/Netfilter	1.2.11	http://www.netfilter.org
Owner-socketlookup patch		http://www.netfilter.org/patch-o-matic/pom-extra.html
Openssh	3.8.1p1	http://www.openssh.org
Openssh chroot patch	3.8.1p1	http://chrootssh.sourceforge.net/index.php
Quota-tools	3.12	http://sourceforge.net/projects/linuxquota

2.5 Risk mitigation plan

The following table identifies the main risks involved in running the jumphost:

Risk description	Risk mitigation
A remote attacker breaks into the jumphost.	Only SSH is available from outside. Iptables have been installed and configured.
A non-privileged user becomes root.	Administrators have root permission already. They have to be trusted. Other users are jailed as soon as SSH authentications was successful. Unneeded packages have been removed. Jumphost has been hardened.
A successful attacker is not discovered.	Logging has been configured. Logcheck informs administrator about unusual system events. Integrit keeps track of the file checksums and informs accordingly.
A non-privileged user guesses or brute-forces passwords.	Password are MD5 hashed and the obscure option has been used to forbid simple passwords.
A non-privileged user loads malicious modules into the Kernel.	Kernel has been minimized and module support has been disabled.
A jailed user breaks out of the jail.	The Kernel has been patched with Grsecurity which counter-measures well-known break-outs.
A jailed user jumps to a server he is not supposed to.	This concerns especially external users (contractors etc). In a user agreement it is defined which servers a user is allowed to jump to. On the jumphost the destination servers and ports a user is allowed to initiate connections to have been enforced with the help of the Iptables Owner-socketlookup.
A jailed user fills up the hard disk and prevents other jailed users from storing data.	The disks have been partitioned accordingly and quotas restrict available disk space per user.
A jailed user carelessly adds keys to `~/.ssh/authorized\_keys`.	Keys can only be added by the system administrator.
Password changing and synchronization mechanisms could be abused by jailed users.	Restrictive PAM configuration in the jail environment.

3 Base installation

3.1 Preparation and boot

This section describes how the Debian GNU/Linux 3.0r2 distribution has been installed on the Sun Netra Ultrasparc T1 105. More detailed information can be found on http://www.debian.org/releases/stable/sparc/install.

- The Debian GNU/Linux 3.0r2 distribution has been downloaded from http://www.debian.org/CD/http-ftp/ and burned on a CD-ROM.
- The SCSI CD-ROM has been attached to the Netra.

Because there was no screen or keyboard to the Netra attached, a serial connection had to be setup. This was done from a physically close by Linux machine. A roll-over cable connected the serial DB-9 (ttyS1) of this other machine with the Serial A LOM of the jumphost. Once this was setup, a connection from the Linux machine with the Ckermit package (a serial communication software) was established:

```
linux# kermit
C-Kermit 7.0.196, 1 Jan 2000, for Linux
 Copyright (C) 1985, 2000,
  Trustees of Columbia University in the City of New York.
Type ? or HELP for help.
(/root/) C-Kermit>set line /dev/ttyS1
(/root/) C-Kermit>set carrier-watch off
(/root/) C-Kermit>connect
Connecting to /dev/ttyS1, speed 9600.
The escape character is Ctrl-\ (ASCII 28, FS)
Type the escape character followed by C to get back,
or followed by ? to see other options.
----------------------------------------------
ok
```

If the `ok-prompt` (openboot prompt) would not appear a `break` signal had to be sent. With Ckermit this can be accomplished with `[Ctrl]+[\]+[b]`. To see all the available escape characters `[Ctrl]+[\]+[?]` can be issued. If the Netra is not configured to automatically boot up, the `lom>` (Lights Out Management) will appear. Entering `power-on` and sending a `break` will get you to the `ok-prompt`.

Now the the device alias of the connected SCSI-CDROM can be figured out. It usually is `cdrom1`. To boot the Debian installation disk `boot cdrom1` was issued.

```
ok
ok probe-scsi
Primary UltraSCSI bus:
Target 0
  Unit 0   Disk     FUJITSU MAB3045S SUN4.2G1705
Target 1
  Unit 0   Disk     IBM     DDRS34560SUN4.2GS98E
Target 6
  Unit 0   Removable Read Only device    TOSHIBA XM5701TASUN12XCD0997
ok
ok devalias
cdrom1                   /pci@1f,0/pci@1,1/scsi@2/disk@6,0:f
cdrom                    /pci@1f,0/pci@1/pci@1/ide@e/cdrom@2:f
..
ok
ok boot cdrom1
```

3.2 Installation of the Debian base system

Not all the installation steps will be described here. The focus will be on security relevant steps.

3.2.1 Disk partitioning

The table below lists the size and file system types chosen for this installation. Mark that the second hard disk is not partitioned yet. It will be partitioned later (see section 6.1).

The `/boot` partition has to be `ext2` in order to let `openboot` read the Kernel image. For the other partitions the choice was the `ext3` file system because of its journaling capabilities which enables to boot up without manual interaction in case of a system crash. The `sda3` contains traditionally the entire disk (see http://www.debian.org/releases/stable/sparc/ch-partitioning.en.html#s-partition-programs).

The partitioning of the disk has been done with security in mind. Each partition can be mounted with different mount options. For example it will be possible to mount `/tmp` with the options `rw,noexec,nodev` while `/` will be mounted with `rw` only because it contains executables and devices.

Partition	Mount point	Size	File system
sda1	/boot	50 MBytes	ext2
sda2	/	200 MBytes	ext3
sda3	/	Whole disk	Whole disk
sda4	/usr	500 MBytes	ext3
sda5	/var	800 MBytes	ext3
sda6	/tmp	200 MBytes	ext3
sda7	swap	500 MBytes	Linux swap
sda8	/home	Rest sda	ext3

This results are shown in the following partition table. The u-flag marks a swap partition for Solaris, but GNU/Linux chooses to ignore it.

```
Disk /dev/sda (Sun disk label): 16 heads, 135 sectors, 3880 cylinders
Units = cylinders of 2160 * 512 bytes

   Device Flag    Start       End    Blocks   Id  System
/dev/sda1             0        47     50760   83  GNU/Linux native
/dev/sda2  u         47       236    204120   83  GNU/Linux native
/dev/sda3  u          0      3880   4190400    5  Whole disk
/dev/sda4           236       710    511920   83  GNU/Linux native
/dev/sda5           710      1468    818640   83  GNU/Linux native
/dev/sda6          1468      1657    204120   83  GNU/Linux native
/dev/sda7  u       1657      2131    511920   82  GNU/Linux swap
/dev/sda8          2131      3880   1888920   83  GNU/Linux native
```

3.2.2 Finishing base installation and booting the system

- There is no need to install any additional Kernel modules and packages.
- Enable MD5 passwords.
- IP address should be configured statically.
- APT (Debian package handling utiltity) has to be configured to access a close Debian mirror and the Debian security update server.
- At the end the system is made bootable. Silo will take care of the boot process. The configuration of Silo will be explained in section 4.3.3.
- After the system reboot the freshly installed GNU/Linux system starts up.

4 Build Kernel and special packages

The following packages and the Kernel have been built from source for the reasons stated in each section.

In order to keep the jumphost clean from packages not needed for productive environment a build host had been setup. The build host is identical with the jumphost (same hardware, same operating system) but additionally is equipped with development tools in order to compile source code and create Debian packages (e.g. gcc, make, wget, gpg, bunzip2, fakeroot).

The setup of a build host is not explained in this paper. It is important to harden and protect the build host with the same care as the jumphost. If an attacker can get hold of the build host he would be able to fiddle with the build environment and indirectly bring malicious code to the jumphost.

- It is recommended not to build the packages as user `root`. This is a precaution to not mess around with the standard installation of the build host.
- The resulting binaries, man pages etc. are bundled in Debian packages before being transferred to the jumphost. The original package names were complemented with the string `-cc` to avoid accidental updates via `apt-get upgrade`. It also is a good way to get an overview of the packages that had been built from source (`dpkg -l |grep -- -cc`).

4.1 Getting Kernel source

Before building `iptables` the Kernel sources need to be downloaded and unpacked, because the `iptables owner-socketlookup` patch will change the Kernel sources. The steps for configuring, building and installing the new Kernel will be explained in section 4.3.

```
buildhost# wget http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.26.tar.bz2
buildhost# wget http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.26.tar.bz2.sign
buildhost# gpg linux-2.4.26.tar.bz2.sign
```

4.2 Iptables

After a jailed users has successfully logged into the jumphost he will want to connect to another server within the same network to do maintenance work. The connection will be established via SSH, FTP or Telnet.

Every jailed user is allowed to connect to a specific set of servers and ports. This policy is enforced by Iptables patched with the Owner-socketlookup feature which allows to set rules on a connection-per-user basis.

Clearly this strategy could be challenged because it can be bypassed. If for example a user is allowed to connect to host A, but not host B, he could first connect to host A and from there connect to host B. This way he would have circumvented the security policy.

Still it adds a layer of security especially together with the user agreement wherein the relevant parties agreed on what servers will be accessed. It also defines misuse of the jumphost clearer because users have to actively find a way "jumping" around the jumphost regulations.

4.2.1 Getting and verifying source

The sources for Iptables and the extra repository have to be downloaded and verified with GnuPG.

```
buildhost# wget http://www.netfilter.org/files/iptables-1.2.11.tar.bz2
buildhost# wget http://www.netfilter.org/files/iptables-1.2.11.tar.bz2.sig
buildhost# gpg iptables-1.2.11.tar.bz2.sig
buildhost# wget http://www.netfilter.org/files/patch-o-matic-ng-20040621.tar.bz2
buildhost# wget http://www.netfilter.org/files/patch-o-matic-ng-20040621.tar.bz2.sig
buildhost# gpg patch-o-matic-ng-20040621.tar.bz2.sig
```

4.2.2 Unpacking and patching

After unpacking Iptables and patch-o-matic (includes the extra repository) `runme extra` starts an interactive shell program that lets you choose the patches. For the jumphost installation only the Owner-socketlookup patch was applied. The environment variables `KERNEL_DIR` and `IPTABLES_DIR` have to be set according to their location.

```
buildhost# tar xjf iptables-1.2.11.tar.bz2
buildhost# tar xjf patch-o-matic-ng-20040621.tar.bz2
buildhost# cd patch-o-matic-ng-20040621
buildhost# KERNEL_DIR=../linux-2.4.26 IPTABLES_DIR=../iptables-1.2.11/ ./runme extra
..
Testing owner-socketlookup... not applied
..
Do you want to apply this patch [N/y/t/f/a/r/b/w/q/?] y
..
```

4.2.3 Making and packaging

Now that the Iptables source has been patched, it can be built. I chose to statically build it (`NO_SHARED_LIBS=1`) to not minimize the amount of libraries on the system. Because the standard installation on the build host should be left alone the installation directory was set to `tmp/`.

```
buildhost# cd iptables-1.2.11
buildhost# make KERNEL_DIR=../linux-2.4.26 NO_SHARED_LIBS=1
buildhost# mkdir -p tmp/DEBIAN
buildhost# make KERNEL_DIR=../linux-2.4.26 PREFIX=tmp/usr NO_SHARED_LIBS=1 install
```

The minimum requirement to build a Debian package is the `DEBIAN/control` file. `fakeroot` runs `dpkg-deb` in its own environment. It simulates root file privileges. The package is then being transferred to the jumphost where it is installed.

```
buildhost# vi tmp/DEBIAN/control
# -------------------------- DEBIAN/control ---------------------------------
Package: iptables-cc
Priority: standard
Section: net
Architecture: sparc
Maintainer: intern
Version: 1.2.11
Description: IP packet filter administration tools
# ---------------------------------------------------------------------------
buildhost# fakeroot dpkg-deb --build tmp iptables-cc_1.2.11_sparc.deb
```

Strangely enough the dependencies of `net-base` require `ipchains | iptables | ipfwadm` while on other GNU/Linux machines this is not a requirement anymore. This was probably changed in newer versions of `net-base`. With the package named `iptables-cc` this dependency was not fulfilled anymore.

Instead of messing around with `net-base` it seemed easier to just install a dummy `iptables` package with no contents except a `README` and version number high enough to prevent `apt-get upgrade` from installing the full-blown `iptables` package.

```
buildhost# mkdir -p iptables-dummy-package/DEBIAN
buildhost# cd iptables-dummy-package
buildhost# vi DEBIAN/control
# -------------------------- DEBIAN/control ---------------------------------
Package: iptables
Priority: standard
Section: net
```

```
Architecture: sparc
Maintainer: intern
Version: 99.9
Description: This package does not contain any files except this one. I created
this package to meet the netbase dependency. The real iptables package is named
iptables-cc to meet naming conventions (all packages built by cc are to be named
 <package>-cc).
# ------------------------------------------------------------------------------
buildhost# mkdir -p usr/share/doc/iptables
buildhost# vi usr/share/doc/iptables/README
# -------------------------- README ----------------------------------------
This package does not contain any files except this one. I created
this package to meet the netbase dependency. The real iptables package
is named iptables-cc to meet naming conventions (all packages built
by cc are to be named <package>-cc).
# ------------------------------------------------------------------------------
buildhost# fakeroot dpkg-deb --build iptables-dummy-package \
iptables-dummy-package_99.9_sparc.deb
```

4.2.4 Installing and configuring

Both packages had to be installed with `dpkg -i <package-name>` on the jumphost. Because of the dependency issue discussed above the old Iptables package had to be removed with `dpkg --purge --ignore-deps=netbase iptables` command.

The following startup file was put into `/etc/init.d/iptables`. It installs (start) or uninstalls (stop) the rule base. The rule set is enabled during system start-up. This can be done by linking `/etc/init.d/iptables` to the `/etc/rc?.d` run-levels or on the jumphost this was done with the Debian specific command `update-rc.d iptables defaults`.

Running `/etc/init.d/iptables stop` flushes the active rule set and sets the policies of the `INPUT` and `OUTPUT` chain to `ACCEPT`. This has proven to be useful when troubleshooting connection problems.

The entire `/etc/init.d/iptables` is attached in Appendix A.

Installing the rule set with the command `/etc/init.d/iptables start` respectively on system boot will:

1. Set environment variables. It would have been possible to read the values for `IPADDR`, `DNSSRV` and `NTPSRV` from files in `/etc` but this script is meant to be easy to read in order to keep it maintainable.
2. Set the policies of all chains to `DROP` and flush the rule set.

3. Accept everything going to the loopback interface, but drop incoming packets from the network with destination address `127.0.0.1` (loopback).
4. Drop incoming packets coming from illegal sources (IP addresses that are not expected respectively not routed within our network).
5. Drop incoming packets with illegal TCP flags sequences.
6. Drop incoming packets with invalid states.
7. Drop incoming ICMP packets.
8. Accept incoming related and established connection.
9. Accept incoming SSH connections.
10. Accept everything coming from loopback interface.
11. Drop outgoing packets with invalid states.
12. Accept outgoing related and established connection.
13. Accept outgoing echo-requests (ping) packets.
14. Accept outgoing traceroute packets for user root.
15. Accept outgoing NTP time synchronization requests for user root.
16. Accept outgoing DNS requests.
17. Read the file `/etc/iptables.accept` and set outgoing accept rules for specific `<user>:<destinat` triples.

All the traffic in and out of the jumphost is logged via `syslog` facility. Depending on how much traffic will be seen, logging will be reduced. The `/etc/iptables.accept` file looks like this:

```
# ------------------- /etc/iptables.accept ----------------------------------
#
# iptables accept file (per user)
# syntax: <user>:<dest_ip>:<dest_port>
# note:   for tcp connection only!
#

# outgoing ssh, ftp, telnet
root:192.168.254.50:22
user1:192.168.254.51:21
user2:192.168.254.52.23
user2:192.168.254.51.21
```

```
# apt-get update mirror and security.debian.org
root:192.168.254.60:80
root:192.168.254.61:80

# smtp
root:192.168.254.40:25
# ------------------------------------------------------------------------
```

4.3 Kernel

In previous sections the Kernel source has been unpacked (see 4.1) and patched with the Iptables Owner-socketlookup functionality (see 4.2.2. This section describes how to patch the Kernel with Grsecurity, disable modules and unneeded Kernel features.

4.3.1 Patching with Grsecurity

Grsecurity (http://www.grsecurity.ca) adds additional layers of security to the Kernel. It is especially interesting for the jumphost because it hardens the Chroot environment. With simple, on the Internet available and compile-ready code, it is possible to break-out of the Chroot environment. This is threatening security of the jail environment.

There are many other security features Grsecurity is adding to the Kernel. Many of the options have been enabled (see Appendix B). Importantly all of the Chroot options were enabled except "Capability restrictions within chroot". Ping does not work with this option enabled.

```
buildhost# ls -d linux-2.4.26
linux-2.4.26
buildhost# wget http://www.grsecurity.ca/grsecurity-2.0-2.4.26.patch
buildhost# wget http://www.grsecurity.ca/grsecurity-2.0-2.4.26.patch.sign
buildhost# gpg http://www.grsecurity.ca/grsecurity-2.0-2.4.26.patch.sign
buildhost# patch -p0 <./grsecurity-2.0-2.4.26.patch
```

4.3.2 Configuring and making the Kernel

It is not in the scope of this paper to describe the configuration of the Kernel. Important for the security of the system is the fact that module-support and unneeded Kernel functionality had been disabled. See the Kernel configuration in Appendix B for all settings.

`strip vmlinux` minimizes the size of the Kernel image. This is necessary because the Kernel can not be loaded if it is too big.

```
buildhost# make menuconfig
buildhost# make dep && make vmlinux
buildhost# strip vmlinux
```

4.3.3 Configuring Silo

The new Kernel had been transferred to the jumphost and moved to `/boot/vmlinuz-2.4.26`. The `silo.conf` had to be adjusted to load the new Kernel when booting the system. To keep things consistent the link from `/boot/vmlinuz` to `/vmlinux` had to be removed. Then just the two Kernel images remained, both in `/boot`.

The old Kernel should be left in place to leave an opportunity to boot with it, in case the new Kernel cannot boot. This is especially important when booting the new Kernel for the first time. Later on the module utilities will be removed and the old Kernel image will still be bootable but the system will not function fully because of the missing user-level module support.

In the past boot loader passwords had been set. It often caused problems because it is used rarely and system administrators tend to forget it or have changed their employer. The server was put in the data center with restricted access. For these reasons no boot password was set.

The `partition` keyword refers to the disk partition where `/boot` resides. In the case of the jumphost this is a partition on its own (sda1). The `image=` statement points to the relative location of the Kernel image of the named partition. The default Kernel image is labelled `linux`.

```
# ------------------- /etc/silo.conf ----------------------------------------
partition=1
root=/dev/sda2
timeout=100
read-only
image=1/vmlinuz-2.4.26-cc
 label=linux
image=1/vmlinuz-2.4.18-sun4u
 label=linuxorg
# ---------------------------------------------------------------------------
```

4.4 Openssh

The version of Openssh (http://www.openssh.org/) provided by Debian could not be used because it needed to be patched. After a successful authentication the according users have to be chrooted into the jail environment.

To distinguish between normal system users and users that have to be change-rooted the home directory needs a special marker. A normal system users' home directory defined in `/etc/passwd` is e.g. `/home/user1` while a change-rooted users' home directory looks `/jail/./home/jailusr1` alike. This tells the patched Openssh to chroot into `/jail` once the authentication was successful.

4.4.1 Getting, verifying, unpacking and patching

The sources for Openssh have to be downloaded and verified with GnuPG. There are also already patched versions of Openssh available on http://chrootssh.sourceforge.net.

```
buildhost# wget ftp://ftp.de.openbsd.org/pub/unix/OpenBSD/OpenSSH/portable/openssh-3.8.1p1.tar.gz
buildhost# wget ftp://ftp.de.openbsd.org/pub/unix/OpenBSD/OpenSSH/portable/openssh-3.8.1p1.tar.gz.si
buildhost# gpg openssh-3.8.1p1.tar.gz.sig
buildhost# wget http://chrootssh.sourceforge.net/download/osshChroot-3.8.1p1.diff
buildhost# wget http://chrootssh.sourceforge.net/download/MD5.txt
buildhost# grep osshChroot-3.8.1p1.diff MD5.txt |md5sum -c


buildhost# tar xjf openssh-3.8.1p1.tar.gz
buildhost# patch -p0 < osshChroot-3.8.1p1.diff
```

4.4.2 Configuring and making

There are a lot of possible `configure` options available. For the jumphost:

- `--prefix` was set to `/usr`. The binaries and the man pages will be installed in `/usr/bin` respectively in `/usr/man`.
- As a habit the configuration files are expected to be in `/etc/ssh` which can be controlled with `--sysconfdir`.
- The `--with-MD5-passwords` flag had to be enabled in order to read the MD5 hashed passwords.
- As IPV6 is not used yet it was be disabled with `--without-4in6`.
- PAM was not enabled because it seems to be broken when used with `Privilege Separation`.

```
buildhost# cd openssh-3.8.1p1
buildhost# ./configure \
--prefix=/usr \
--sysconfdir=/etc/ssh \
--with-md5-passwords \
--without-4in6
```

will produce the following summary:

```
OpenSSH has been configured with the following options:
                     User binaries: /usr/bin
                   System binaries: /usr/sbin
               Configuration files: /etc/ssh
                   Askpass program: /usr/libexec/ssh-askpass
                      Manual pages: /usr/man/manX
                          PID file: /var/run
  Privilege separation chroot path: /var/empty
            sshd default user PATH: /usr/bin:/bin:/usr/sbin:/sbin
                    Manpage format: doc
                       PAM support: no
                  KerberosV support: no
                  Smartcard support: no
                      S/KEY support: no
              TCP Wrappers support: no
              MD5 password support: yes
        IP address in DISPLAY hack: no
           Translate v4 in v6 hack: no
                  BSD Auth support: no
              Random number source: OpenSSL internal ONLY

              Host: sparc64-unknown-linux-gnu
          Compiler: gcc
    Compiler flags: -g -O2 -Wall -Wpointer-arith -Wno-uninitialized
Preprocessor flags:
      Linker flags:
         Libraries:   -lresolv -lcrypto -lutil -lz -lnsl  -lcrypt
```

After making Openssh an `init` script was added. In addition to start the daemon it creates a RSA and DSA public/private key-pair when run for the first time. The script included in the Debian Openssh binary package is doing exactly that.

```
buildhost# make
buildhost# mkdir tmp
buildhost# make install DESTDIR=tmp
buildhost# mkdir tmp/etc/init.d
buildhost# vi tmp/etc/init.d/sshd
buildhost# chmod 755 tmp/etc/init.d/sshd
buildhost# mkdir tmp/DEBIAN
buildhost# vi tmp/DEBIAN/control
# --------------------------- DEBIAN/control----------------------------------
Package: openssh-cc
Version: 3.8.1p1
Section: net
Priority: standard
Architecture: sparc
Maintainer: intern
Description: OpenSSH
```

```
# ------------------------------------------------------------------------------
buildhost# fakeroot dpkg-deb --build tmp openssh-cc_3.8.1p1_sparc.deb
```

4.4.3 Configuring Openssh

The initial SSH server configuration file, found in `/etc/ssh/sshd_config`, had to be customized for the jumphost. There is not much point in restricting the client configuration file `/etc/ssh/ssh_config` because each user is allowed to have his own `~/.ssh/config` which overrides the defaults.

The following settings had been changed compared with the initial `sshd_config`. The complete `sshd_config` can be found in Appendix D.

- By default `Protocol` allows SSH version 1 and 2. For the jumphost only 2 was allowed.
- The `LogLevel` was changed to `VERBOSE` instead of `INFO`.
- `root` logins had been disabled.
- The version 1 RSA authentication had been disabled.
- By default the `authorized_keys` file is maintained by the users. For the jumphost it is controlled by the jumphost administrators. This was enforced by setting `AuthorizedKeysFile` to `/etc/ssh/authorized_keys.%u`. The `%u` is replaced by the user login name.
- `X11Forwarding` has to be enabled for jumphost users.
- `Banner` is set to /etc/issue which will be used system wide.

```
# -------------------------- sshd_config -----------------------------------
Protocol 2
LogLevel VERBOSE
PermitRootLogin no
RSAAuthentication no
AuthorizedKeysFile      /etc/ssh/authorized_keys.%u
X11Forwarding yes
Banner /etc/issue
# --------------------------------------------------------------------------
```

4.5 Quota-tools

The Quota-tools (http://sourceforge.net/projects/linuxquota/) were used to restrict available disk space for jailed users. Because these users are permitted to use SCP to transfer

files from and to the jumphost it is advisable to set limits. The jumphost is often used in emergency cases and it is therefore crucial to have enough available disk space for each user.

I encountered problems using the Quota-tools with `ext3` and while looking for solutions I found that the `xfs` file system has built-in support for quotas, which means that no additional init script needed to be run when the system starts up. To add the `usrquota` option to `/etc/fstab` for the `xfs` file system is sufficient to turn quota support on. A newer version of the user level Quota-tools (3.01-pre2 onward) is recommended in `/usr/share/doc/xfsprogs/README.qu` That was the reason why the default Quota-tools provided by Debian were not used.

4.5.1 Getting, making and packaging

```
buildhost# wget http://switch.dl.sourceforge.net/sourceforge/linuxquota/quota-3.12.tar.gz
buildhost# tar xzf quota-3.12.tar.gz
buildhost# cd quota-tools
buildhost# ./configure --prefix=/usr
buildhost# make
buildhost# mkdir -p tmp/DEBIAN
buildhost# make install ROOTDIR=tmp
buildhost# vi tmp/DEBIAN/control
# --------------------------- DEBIAN/control---------------------------------
Package: quota-tools-cc
Version: 3.12
Section: admin
Priority: extra
Architecture: sparc
Maintainer: intern
Description: QuotaTool
# -------------------------------------------------------------------------
buildhost# fakeroot dpkg-deb --build tmp quota-tools-cc_3.12_sparc.deb
```

The creation of the `xfs` file system on the second hard disk, the mount options and the setting of quotas per user will be discussed in section 6.

4.6 Monit

In order to let the jailed users change their passwords a workaround had to be found. The problem with the SSH user Chroot environment is that the user authenticates himself to the `/etc/password` and `/etc/shadow` file. As soon as the authentication was successful he finds himself in the `/jail` Chroot environment. The user is only allowed to change her password within the jail (`/jail/etc/passwd` and `/jail/etc/shadow`) which will not have any effect on future logins.

A mechanism to timely and safely synchronize passwords from the jail to the system password files had to be put in place. The idea is that a daemon running in the main system environment is listening for changes of the `/jail/etc/shadow` file. Monit (http://www.tildeslash.com/monit/) provides besides other features this functionality. Monit is not in the stable Debian tree it had to be built from source.

4.6.1 Getting, making and packaging

```
buildhost# wget http://www.tildeslash.com/monit/dist/monit-4.3.tar.gz
buildhost# wget http://www.tildeslash.com/monit/dist/monit-4.3.tar.gz.md5
buildhost# md5sum -c monit-4.3.tar.gz.md5
buildhost# tar xzf monit-4.3.tar.gz
buildhost# cd monit-4.3
buildhost# ./configure --prefix=/usr --sysconfdir=/etc --without-ssl
buildhost# make
buildhost# mkdir -p tmp/DEBIAN
buildhost# make install DESTDIR=tmp
buildhost# vi tmp/DEBIAN/control
# --------------------------- DEBIAN/control----------------------------------
Package: monit-cc
Version: 4.3
Section: admin
Priority: extra
Architecture: sparc
Maintainer: intern
Description: Monit
# --------------------------------------------------------------------------
buildhost# mkdir -p tmp/etc/init.d
buildhost# vi etc/init.d/monit
buildhost# chmod 755 etc/init.d/monit
buildhost# fakeroot dpkg-deb --build tmp monit-cc_4.3_sparc.deb
```

The `etc/init.d/monit` script uses the standard Debian `start-stop-daemon` procedure to start and stop the `/usr/sbin/monit`.

4.6.2 Configuring

After installing the `monit` package on the jumphost and enabling it to run at system boot (`update-rc.d monit defaults`) the configuration file `/etc/monitrc` had to be created. The `monit` daemon polls every 10 seconds if the checksum of the `/jail/etc/shadow` has changed. If this is the case it executes `/usr/local/bin/sync_shadow.sh` with the parameter `monit`.

```
# --------------------------- /etc/monitrc ----------------------------------
```

```
set daemon  10    # Poll  intervals
check file shadow with path /jail/etc/shadow
    if failed checksum then exec "/usr/local/bin/sync_shadow.sh monit"
# ----------------------------------------------------------------------------
```

4.6.3 Synchronizing passwords

The security risks involved in the synchronization of passwords had to be considered. A jailed user could try to abuse the synchronization process to get the `root` password copied into the jail environment or to set the root password to his password. Several security measures had to be put in place to mitigate these risks:

- Checks are done before synchronization can start.
- Only users belonging to the group `jail` are synchronized.
- The `pam` configuration within the jail environment is very restrictive.

The entire `/usr/local/bin/sync_shadow.sh` script can be found in Appendix E. This script is able to perform two tasks:

System-to-Jail If the administrator (with root privileges) sets the password of a jailed user the new password has to be copied from the system password files to the jail password files. Before doing this task two requirements have to be fulfilled.

1. `/etc/shadow` has to be newer than `/jail/etc/shadow`.
2. The script was called without any parameters.

If the checks are OK:

1. All the entries in `/jail/etc/passwd` belonging to the group `1000` are copied into `/jail/etc/passwd`. Mark that `/jail/etc/passwd` is rewritten. The home directory field needs to be ridden of the string "`/jail/.`".
2. For all the members of group 1000 the `/etc/shadow` entries are copied to `/jail/etc/shadow`.

Jail-to-System If a jailed user changes his password with the command `passwd` the `monit` daemon will detect the changed `/jail/etc/shadow` file and run `sync-shadow.sh monit`. Two requirements have to be fulfilled to let the script update the system `/etc/passwd` and `/etc/shadow`.

1. `/jail/etc/shadow` has to be newer than `/etc/shadow`.
2. The script was called with the parameter `monit`.

If the checks are OK the `usermod -p` is run for every user in `/jail/etc/shadow` after checking that the user is a member of the group `jail`.

5 Packages and services

The configuration of the self-compiled packages has been discussed earlier. In this section all the other services and packages will be described.

5.1 Configuring Apt

`apt-get` is one of the package management tools provided by Debian to keep packages up-to-date. `apt-setup` configures how the Debian archives should be accessed. Possible choices are `ftp`, `http`, `cdrom` or `filesystem`. The sources for the Debian archives can also be modified by hand in the following file. It is important to add the IP addresses of these hosts to the Iptables control file `/etc/iptables.accept`.

```
# --------------------------- /etc/apt/sources.list --------------------------
deb http://sunsite.cnlab-switch.ch/ftp/mirror/debian/ stable main non-free contrib
deb-src http://sunsite.cnlab-switch.ch/ftp/mirror/debian/ stable main non-free contrib
deb http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb-src http://non-us.debian.org/debian-non-US stable/non-US main contrib non-free
deb http://security.debian.org/ stable/updates main contrib non-free
# ----------------------------------------------------------------------------
```

The last entry pointing to security.debian.org is crucial in order to keep the system up-to-date. As soon as Debian releases a security update it can be fetched with `apt-get <package-name>`.

After configuring the Debian archive sources it is time to update all the current packages with the command `apt-get update` and `apt-get upgrade`. `apt-get` with the parameter `update` command fetches the current package lists while the parameter `upgrade` will upgrade all the packages with newer versions if available.

5.2 Remove and add some packages

The fewer packages installed the less possible vulnerabilities threaten the system. The base installation of Debian is already very small. There are still some packages that can be removed:

Package	Description	Removal reason
at	Delayed job execution	Only regular jobs are executed. Cron can do that.
cpio	Archive management	Rarely used on modern systems. Not needed on the jumphost.
dhcp-client	DHCP client	IP address will be configured statically.
ed	Editor for dinosaurs	Vi is used to modify files.
eject	CD operator	Not necessary. CDROM will be detached after installation anyway.
exim	Mail transfer agent	Exim will be replaced with ssmtp.
fdutils	Floppy utilities	No floppy drive available.
info	Info documentation browser	Man pages are sufficient.
ipchains	Firewalling	Iptables is used instead.
libldap2	LDAP libraries	Was needed by exim which was removed.
libpcre3	Perl regular expression library	Not needed by any other package.
modconf	GUI for configuring modules	Jumphost Kernel does not support modules.
modutils	Module utilities	Jumphost Kernel does not support modules.
nano	Simple editor	Vi is used to modify files.
ppp, pppconfig, pppoe	Point-to-point protocol	Not used on this system.

Some packages need to be added to the jumphost. `apt-get install <package name>` can be used.

Package	Description	Install reason
ethtool	Change Ethernet settings	Used to set 100 Mbit/s full duplex and disable auto negotiation.
integrit	Integrity verification	Monitor file integrity (configuration will be described in section 5.8)
libgpmg1	General purpose mouse library	Is a vim dependency.
libssl0.9.6	SSL shared libraries	Used by Openssh.
logcheck, logcheck-database	Monitors system logfiles	Used to mail unusual system events to administrator (configuration will be explained in section 5.7).
logtail	Returns parts of logfiles	Logcheck depends on it.
lsof	Swiss army knife diagnostic tool	To get a quick overview of the system state.
ntpdate	Time update utilities	Update system time from NTP server (configuration will be explained in section 5.5).
perl, perl-modules	Scripting language	Logtail dependency.
ssmtp	Simple MTA	Replacing Exim.
vim	Vi improved	Some administrators are lost without it.
xfsprogs	XFS filesystem utilties	Used to create xfs file systems.
zlib1g	Compression library	Openssh dependency.

In addition the self-compiled packages have been installed:

Package	Description	Install reason
iptables-cc	Patched Iptables	Firewalling.
monit-cc	Daemon	Watch for shadow file changes.
openssh-cc	Patched Openssh	SSH daemon including user Chroot functionality.
quota-tools-cc	Disk quotas	Used to restrict available disk space for jumphost users.

The complete list of installed packages can be found in Appendix C.

5.3 Ethernet settings

`ethtool` was used to configure Ethernet settings. In the productive environment speed 100 Mbit/s and full duplex had to be set.

```
# --------------------------- /etc/init.d/networking -------------------------
..
start)
..
    echo -n "set duplex Full speed 100Mb/s "
    /usr/sbin/ethtool -s eth0 autoneg off
    /usr/sbin/ethtool -s eth0 speed 100 duplex full
..
# ----------------------------------------------------------------------------
```

5.4 Disable unused services

Per default on Debian only the `inetd` services are running. It was disabled because those services are not needed and would be a security risk. It was done with the command `update-rc.d -f inetd remove`. There was another unnecessary service available in `/etc/init.d` which has been removed as well: `update-rc.d -f audioctl remove`.

After that, `netstat -l` showed one listening port only, namely `ssh`:

```
jumphost# netstat -l
Proto Recv-Q Send-Q Local Address           Foreign Address         State
tcp        0      0 *:ssh                   *:*                     LISTEN
Active UNIX domain sockets (only servers)
..
```

5.5 Configure ntpdate

It is important for the security of the system that the system has the correct time, especially if log files have to be cross-compared with other systems. The correct time can be fetched from NTP (Network Time Protocol) servers around the world. After the installation of the `ntpdate` package an init script to run `ntpdate` at system startup was put in place. The IP address of the NTP servers have to be defined in `/etc/default/ntp-servers`.

To adjust the time at least once every day an entry in `crontab` is required. The following line tells cron to run `ntpdate` every day at two o'clock in the morning. Iptables had to be configured to allow the outgoing NTP request.

```
jumphost# crontab -e
# --------------------------- /var/spool/cron/crontabs/root ------------------
# synchronize clock
0 2 * * * /etc/init.d/ntpdate start >/tmp/out 2>&1
# ----------------------------------------------------------------------------
```

5.6 Harden mail

The jumphost does not require to act as a mail server and there is no need to deliver mail to local users. `exim` could do all of that and much more. Instead of hardening `exim` to only send out mails via mail hub it is also possible to install `ssmtp` (secure SMTP) which does exactly that and nothing more. It was also possible to remove additional packages `exim` depended on.

The configuration of `ssmtp` can be found in `/etc/ssmtp/ssmtp.conf`. The `mailhub` had been set to the mail hub used in the enterprise.

5.7 Configure logcheck

Logcheck is a handy tool to detect unusual entries in the system logfiles and send them via email to the administrator of the server. Depending on the requirements, Logcheck sends too many or too few system events. Anyway it has to be tailored to suit a specific system. The configuration is in `/etc/logcheck`. In the installation dialog the email address the reports will be sent to and the report level can be chosen. For the jumphost `server` was the configuration to start with.

The goal of the logcheck configuration is to not get any emails except an intruder really made it to break in. Because the machine is facing the Internet there will be many port scans etc. which will produce Iptables logs. An attacker could cause a certain damage with filling up the administrators mailbox without actually breaking in. An administrator will also ignore or filter regular mails and not pay attention to them. For these reasons the usage of Logcheck has to be considered closely. The Logcheck facility should not introduce a new security hole.

Usually an administrator does not check the system log files on a regular basis. Therefore an intruder has not to fear to get discovered even he would produces a lot of suspicious log entries. Remounting file systems, adding new users or booting the system etc are definitely events we want the administrator be informed about. That are the tasks, Logcheck has to serve on the jumphost.

The `logcheck` utility is ran at system boot and on a regular basis every hour. The `logcheck` cron job is defined in `/etc/cron.d/logcheck`.

To ignore all Iptables log entries starting with the string `ACCEPT` or `DROP OUT` a new file had been created in `/etc/logcheck/ignore.d.server`.

```
# ---------------------------- /etc/logcheck/ignore.d/iptables -----------------
kernel: ACCEPT .*
```

```
kernel: DROP OUT:.*
# ------------------------------------------------------------------------
```

Similar entries have been made for SSH logs that are not unusual:

```
# -------------------------- /etc/logcheck/ignore.d/iptables ----------------
sshd.*: Accepted publickey for .* from .* port .*
sshd.*: Connection closed by .*
sshd.*: Closing connection to .*
sshd.*: Connection from .* port .*
sshd.*: Found matching DSA key: .*
sshd.*: Postponed publickey for .* from .* port .* ssh2
sshd.*: Accepted password for .* from .* port .* ssh2
# ------------------------------------------------------------------------
```

After the Logcheck package had been installed the behavior of a normal user was imitated (log in, jump, change password etc). Logcheck then sent a mail containing a list of unusual system event. The report had to be examined closely and all the entries Logcheck should not consider as unusual had to be copied into the corresponding files in `/etc/logcheck`. Specific information like IP address or user name had to be be replaced with `.*`.

If logcheck wrongly considered an event as a security violation the same configuration steps as with the `ignore.d` were repeated with `violations.ignore.d`. It took some time and patience to configure logcheck to the needs of the jumphost server but without configuring it correctly it would not do any good.

5.8 Configure Integrit

Another layer of security had been added with the installation of the integrity checking tool Integrit. If files located in directories that are not supposed to change nevertheless have changed Integrit will send a mail to the system administrator. Integrit had been installed as a normal Debian package. A daily cronjob (`/etc/cron.daily/integrit`) compares the `current.db` with the `known.db`. Emails should only be sent if there were changes. This again is to ensure administrators do not ignore the mails because of regularity. Both databases are stored in `/var/lib/integrit`.

The configuration files for Integrit can be found in `/etc/integrit`. Configuration can either be done via editing `integrit.conf` first and then running `dpkg-reconfigure integrit` to enter the email address etc. interactively or modifying `integrit.debian.conf` manually.

All the directories or files starting with `!` should not be checked for changes. These are files expected to change regularly. Directories starting with `=` should be checked but not descended into.

```
# ---------------------------- /etc/integrit/integrit.conf --------------------
root=/
known=/var/lib/integrit/known.cdb
current=/var/lib/integrit/current.cdb
!/lost+found
!/boot/lost+found
!/dev
!/etc/adjtime
!/etc/iptables.allow
!/etc/network/ifstate
!/etc/passwd
!/etc/passwd-
!/etc/shadow
!/etc/shadow-
!/home
!/home/lost+found
!/jail/etc/passwd
!/jail/etc/shadow
!/jail/home
!/proc
!/root
!/tmp
!/tmp/lost+found
!/usr/lost+found
!/var
!/var/lost+found
=/usr/doc
=/usr/info
=/usr/share
# -----------------------------------------------------------------------------
```

The `integrit.debian.conf` is read by the daily cronjob.

```
# ---------------------------- /etc/integrit/integrit.debian.conf -------------
CONFIGS="/etc/integrit/integrit.conf"
EMAIL_RCPT="adminemail@company.com"
EMAIL_SUBJ="[integrit] report on changes in the filesystem"
SAVECYCLE=10
# -----------------------------------------------------------------------------
```

This configuration of `integrit` should not be trusted. This is not the weakness of `integrit` itself but of human and technical nature. Some issues should be considered.

Setting/Situation	Advantage	Disadvantage
Reports are only sent if there were changes.	Administrators will not filter or ignore the mails.	An intruder might prevent emails from being sent.
Database is stored locally.	Easy configuration.	An intruder could change files and copy the current.db to the known.db.
Integrit binary is stored locally.	Easy configuration.	An intruder could modify the binary.
Cronjob will update the database if there was a change.	No manual interaction necessary (copying current.db to known.db)	If the administrator ignores the mail no further mails will be sent the following days.

Of course these problems could be solved or worked around. It would be possible to mount the database and the integrity binary from a CDROM or via NFS. Cronjob could be instructed to not overwrite the known.db. Unfortunately if a skilled intruder becomes `root` he would be able to circumvent all these measures, except all the file systems would be mounted and checked by another system.

In the case of the jumphost `integrit` adds a layer of security without a lot of additional work. It is also meant to protect the administrators from themselves. Sometimes files are unintentionally changed by administrators. `integrit` informs accordingly.

After the installation and configuration of the jumphost a copy of `known.db` had been stored on the build machine. In case of doubt the `current.db` should be compared to this one.

6 Configure Jail (Chroot-Environment)

In this section the setup of the Chrooted environment for the jailed users will be described. It is also common to call the system within the system a “jail” to emphasize the situation of the user. He is meant to roam the jail only.

The setup of a Chrooted environment involved a lot of trial and error work. For this reason I wrote a script which sets up the environment.

6.1 Prepare partition

The home directories of the jumphost users had been put on their own partition in order to use disk quotas and to mount it with restrictive options. As discussed earlier the `xfs` filesystem had been chosen for this because of its convenient (no init scripts) fashion to

support quotas.

Problems arose when defining the `xfs` partition as the first partition of a hard disk. After a reboot the xfs partition was always lost. As a workaround the first partition consisting of one cylinder only with filesystem type ext3 had been created without using it any further.

```
Disk /dev/sdb (Sun disk label): 16 heads, 135 sectors, 3880 cylinders
Units = cylinders of 2160 * 512 bytes

   Device Flag    Start       End    Blocks   Id  System
/dev/sdb1             0         1      1080   83  GNU/Linux native
/dev/sdb2  u          1      3880   4189320   83  GNU/Linux native
/dev/sdb3             0      3880   4190400    5  Whole disk


jumphost# mkfs.ext3 /dev/sdb1
jumphost# mkfs.xfs /dev/sdb2
```

6.2 Setting up Jail

The `create_jail.sh` script does all the work to setup a brand new jail. It can be found in Appendix F. The script:

1. Sets umask to 666 which means to create files with permissions 000.
2. Removes the existing jail including unmounting `/jail/home`.
3. Defines all the binaries that should be included in the jail (bash, passwd, ls, mv, rm, ssh, ping, scp, telnet, ftp).
4. Creates the basic directory structure within `/jail` (etc, bin, lib, dev).
5. Creates device files in `/jail/dev` (null, urandom, tty).
6. Creates empty passwd and shadow files.
7. Creates `/jail/etc/services` with entries for `ftp`, `telnet` and `icmp`.
8. Creates `/jail/etc/pam.conf`. More details about this later in this section.
9. Copies `/etc/resolv.conf` to `/jail/etc/resolv.conf`.
10. Copies binaries and corresponding libraries (detected with `ldd`) to `bin` and `lib`.
11. Copies name service and PAM libraries to `/jail/lib`.

12. Sets strict permissions (lib[555], bin[111], etc[444], shadow[640], passwd[644]) and change group of `shadow` to `shadow`.

13. Sets `user-ID-on-execution-bit` (SUID Bit) on `ping` and `passwd`. Both `ping` and `passwd` need to be SUID to root. This means the user will run the binaries with root permissions.

The complete jail environment can be found in Appendix G.

6.3 Jail PAM configuration

`PAM` (Pluggable Authentication Modules) had to be configured within the jail in order to let the jail users change their password. To disable any other `PAM` services the configuration needs to be restrictive.

The first line of the jumphost `/jail/etc/pam.conf` configuration allows the passwd service. It requires the user to enter his current password first. The new MD5 password has to be between six and twenty characters long. The `obscure` option enables additional checks on the password to enforce strong passwords. All other `pam` services are denied with help of `pam_deny.so`.

```
# --------------------------- /jail/etc/pam.conf -----------------------------
passwd  password        required        /lib/pam_unix.so obscure min=6 max=20 md5
other   auth            required        /lib/pam_deny.so
other   account         required        /lib/pam_deny.so
other   password        required        /lib/pam_deny.so
other   session         required        /lib/pam_deny.so
# ----------------------------------------------------------------------------
```

6.4 Add jailed users

To let an administrator adding jailed users correctly a script was required. It can be found in Appendix H. The script requires two parameters: login name and comment. Running the script without any parameters it will display the syntax. The comment field consists of first name, last name, company and email address. In the following example the user `exampleusr` is added with the help of the script.

```
jumphost# addjailuser.sh exampleusr "John Miller XYcompany john.miller@xycompany.com"
```

Besides adding the user to the system the disk quota is set. The first parameter (80000) sets the soft limit to 80 MBytes. The hard limit (100MBytes) will be enforced after the grace period expired. The grace period can be set with `edquota -t`. The default setting are 7 days. `sync_shadow.sh` (see section 4.6.3 will update the jail `passwd` and `shadow` file. For the example user `exampleusr` the script executes the following commands:

```
jumphost# useradd -g jail -d "/jail/./home/exampleusr" \
 -c "John Miller XYcompany exampleusr@xycompany.com" -m exampleusr
jumphost# chmod 700 /jail/home/exampleusr
jumphost# setquota exampleusr 80000 100000 0 0 /jail/home
jumphost# passwd exampleusr
jumphost# sync_shadow.sh
```

7 System hardening

Besides disabling services, removing packages and application hardening the system, additional hardening can be done on the system. This will be described in this section. Most tasks have been done according to the SANS "Securing UNIX/GNU/Linux" documentation.

7.1 Users

7.1.1 Administrators and root

In previous sections the jail environment has been restricted. The system users need to be considered as well. On the jumphost only administrators are allowed to login with their personal account. Once logged in they are allowed to become `root` with issuing the `su` (switch user) command. The reason why the administrators have personal logins is to know who and when somebody logged into the system.

For the jumphost only one file needed to be put in the user skeleton, namely `.bash_profile`. It sets the `umask` to the value `066` which will create files with the permission 600. The UMASK environment variable should also be set to 066 in `/etc/login.defs`. On Debian the umask is also defined in `/etc/profile`. It should changed to 066 there as well.

The `PATH` environment variable is set to `/bin:/usr/bin`. It is not necessary to add `/usr/local/bin` where the `addjailuser.sh` and `sync_shadow.sh` scripts reside because they are only executable as root. Because there are only a few users on the jumphost and all of them are allowed to be root it is not necessary to install and configure `sudo` at this point. If more users with different roles would have access to the jumphost `sudo` would definitely make sense.

```
# --------------------------- /etc/skel/.bash_profile ------------------------
umask 066
export PATH=/bin:/usr/bin
export PS1='\u@\h:\$ '
# ----------------------------------------------------------------------------
```

There are two ways an administrator can log into the system. Usually she will use SSH. In order to enforce personal login in SSH instead of root the `PermitRootLogin` option in `sshd_config` had been set to `no`. If the user will login on the console which is possible via terminal server he will be allowed to login as root. It would be possible to prohibit root login on the console by editing `/etc/securetty`. To not break with the personal login policy it is not possible to login on the terminal server as root.

Administrators and jumphost users should be presented with a warning banner before logging into the system. The banner should state that only authorized users are allowed to login. The text is put into `/etc/issue` than linked to `/etc/motd`. In addition the `Banner` option in `/etc/ssh/sshd_config` had been set to `/etc/issue`. This will display the warning banner when presenting the login screen and again after a successful login.

According to SANS standards the limits set by the PAM module `pam_limits.so` should be set to reasonable values. After uncommenting `session required pam_limits.so` in `/etc/pam.d/login` the settings had to be adjusted in `/etc/security/limits.conf`. Unfortunately these settings are not enforced for SSH logins because it was compiled without PAM support because this would break the `PrivilegeSeparation` option.

```
# ---------------------------- /etc/pam.d/login ------------------------------
..
session    required   pam_limits.so
..
# ----------------------------------------------------------------------------


# --------------------------- /etc/security/limits.conf ----------------------
*       soft    nproc   65
*       hard    nproc   128
*       soft    nofile  256
*       hard    nofile  1024
# ----------------------------------------------------------------------------
```

7.1.2 System users

The `/etc/passwd` contains a lot of users that may be used by a daemon installed at a later point of time. On the jumphost no additional daemons will be installed and it is therefore a good idea to remove unused users. If there is uncertainty if a users owns any files on

the system it can be checked with `find`. Users that cannot be removed, but will never be expected to log into the system, should be locked and the home directory set to `/dev/null`.

```
jumphost# # find which users own files
jumphost# for i in `cut -d: -f1 /etc/passwd`; do echo $i
 find / -user $i 2>/dev/null |wc -l; done
jumphost# # remove some users
jumphost# for i in sync games lp list mail news uucp proxy  \
 postgres www-data backup irc gnats; do userdel $i; done
jumphost# # lock some users
jumphost# for i in daemon bin sys man operator list
 do usermod -s /dev/null -L $i; done
```

After this tasks has been finished `/etc/passwd` contains the following entries (the administrator accounts are not listed):

```
# --------------------------- /etc/passwd ----------------------------------
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/dev/null
bin:x:2:2:bin:/bin:/dev/null
sys:x:3:3:sys:/dev:/dev/null
man:x:6:100:man:/var/cache/man:/dev/null
operator:x:37:37:Operator:/var:/dev/null
nobody:x:65534:65534:nobody:/home:/bin/sh
sshd:x:100:100::/var/empty:/bin/bash
# --------------------------------------------------------------------------
```

7.1.3 Restrict cron

It is advisable to restrict the usage of `cron`. On the jumphost only `root` should be allowed to use the `cron` facilities. This can be enforced with the file `/etc/cron.allow`. If this file is not empty everybody not listed will not be permitted to use `cron`. In addition the file permissions of the directories read by cron should be restricted.

```
jumphost# echo root > /etc/cron.allow
jumphost# chown -R root:root /etc/cron* /var/spool/cron
jumphost# chmod -R go-rwx /etc/cron* /var/spool/cron
```

7.2 SUID/SGID bits and mount options

7.2.1 Executable with SUID Bit set

Before setting the correct mount options it is important to know where the SUID/SGID executables are located. Anyway it is good to have an idea which SUID/SGID executables are installed on a system and what they are supposed to do. `find` can be used to list all of them.

Only directories in `/var/cache/man` are SGID to root. The group of new directories created within `/var/cache/man` will be set to root as well. Only directories can get SGID. They are no security threat (see http://lists.debian.org/debian-devel/2000/10/msg01889.html).

```
jumphost# # Find exectutables with SGID bit set to root
jumphost# find / -group root -perm +g+s
/var/cache/man
/var/cache/man/...
```

The "danger" lies in the executables with SUID set to root. It would be possible to remove the SGID from all of these executables except `su`, `login` and the two jail binaries `passwd` and `ping`. This would be possible on the jumphost because all the administrators are allowed to become superusers and therefore could `ping`, `mount`, change their password etc after becoming root. There are reasons why this is not done on the jumphost:

- Next time a package, containing one of these files, will be upgraded, the permissions will be reverted. It could be more dangerous to wrongly assume to not have any SUID executables then to keep in mind that there are SUID on the system.
- Administrators should not be encouraged to become root to do simple tasks like pinging or changing their own password. This for example would circumvent the PAM `obscure` option.

```
jumphost# # Find executables with SUID bit set to root
jumphost# find / -user root -perm +g+s
/usr/bin/newgrp
/usr/bin/chfn
/usr/bin/chsh
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/crontab
/usr/bin/traceroute
/usr/lib/pt_chown
```

```
/usr/libexec/ssh-keysign
/sbin/unix_chkpwd
/bin/login
/bin/su
/bin/mount
/bin/umount
/bin/ping
/jail/bin/passwd
/jail/bin/ping
```

Even if no changes had been made, it is still wise to find all the SUID executables after a system installation just to make sure no unknown binaries have this bit set.

7.2.2 Mount options

After the SUID executables had been located mount options can be adjusted. Per default all the file systems are mounted `rw` without additional options except `/` which has been mounted with the option `rw,errors=remount-ro`.

- All the file systems which do not have device files (character special files) can be mounted `nodev`. `/dev` and `/jail/dev` are the only directories containing device files. All the other partitions are mounted with the option `nodev`.
- `/cdrom`, `/var` and `/home` are not allowed to run any SUID executables and can therefore be mounted with the option `nosuid`.
- `/usr` should not be written to. It can be mounted with the option `ro`. Before doing a system upgrade `/usr` has to be remounted `rw` with the command `mount -o remount,rw`.
- File execution is not allowed from `/boot`, `/tmp`, `/jail/home` and `/proc`. This can be configured with the mount option `noexec`.
- `/jail/home` has to be mounted with the option `usrquota` in order to enable the disk quota restrictions.
- `/boot` and `/cdrom` do not need to be mounted on system startup. The option `noauto` can be used.

To enable the mount-options after the next reboot `/etc/fstab` has to be modified accordingly.

```
# /etc/fstab: static file system information.
#
```

```
# <file system> <mount point>   <type>  <options>               <dump>  <pass>
/dev/sda2       /               ext3    errors=remount-ro       0       1
/dev/sda7       none            swap    sw                      0       0
proc            /proc           proc    rw,noexec,nodev         0       0
/dev/cdrom      /cdrom          iso9660 ro,nosuid,nodev,noauto  0       0
/dev/sda1       /boot           ext2    nodev,noauto,noexec     0       2
/dev/sda4       /usr            ext3    nodev,ro                0       2
/dev/sda5       /var            ext3    nodev,nosuid            0       2
/dev/sda6       /tmp            ext3    nodev,noexec            0       2
/dev/sda8       /home           ext3    nodev,nosuid            0       2
/dev/sdb2       /jail/home      xfs     usrquota,noexec,nodev   0       0
```

After remounting `mount -o remount,<options> /mount-point` respectively rebooting the system the mount table looks like this:

```
jumphost# mount
/dev/sda2 on / type ext3 (rw,errors=remount-ro)
proc on /proc type proc (rw,noexec,nodev)
/dev/sda4 on /usr type ext3 (ro,nodev)
/dev/sda5 on /var type ext3 (rw,nosuid,nodev)
/dev/sda6 on /tmp type ext3 (rw,noexec,nodev)
/dev/sda8 on /home type ext3 (rw,nosuid,nodev)
/dev/sdb2 on /jail/home type xfs (rw,noexec,nodev,usrquota)
```

7.3 Other hardening tasks

7.3.1 Network configuration parameters

SANS recommends reasonable `sysctl` settings. With the command `/sbin/sysctl -a` the currently set values were compared with the SANS recommendations. The settings that were different had been added to `/etc/sysctl.conf`.

```
# --------------------------- /etc/sysctl.conf -------------------------------
net.ipv4.conf.all.log_martians = 1
net.ipv4.conf.all.send_redirects = 0
net.ipv4.conf.all.secure_redirects = 0
# ----------------------------------------------------------------------------
```

In Debian spoof protection is enabled when running the `/etc/init.d/networking` script. Spoof protection compares incoming packets with the routing table. Because the jumphost only has one interface `rp_filter` is not really necessary, except for the case when a packet with the loopback address is coming in via eth0. But this case is already handled by Iptables.

7.3.2 Anonymous shutdowns

Per default the key combination Ctrl+Alt+Del causes a system to reboot. This behavior had to be disabled. Instead of a reboot the event will be logged.

```
# --------------------------- /etc/inittab ----------------------------------
..
::ca:12345:ctrlaltdel:/usr/bin/logger 'ctrl-alt-del trapped'
..
# ---------------------------------------------------------------------------
```

8 Maintenance procedures

The maintenance of the system is done by the system administrators. There are three areas the maintenance tasks can be divided into.

8.1 Updates and patches

The system administrator is responsible to keep the system up-to-date in terms of security. Operating system, software and security related mailing lists inform the administrator of security relevant updates respectively vulnerabilities that require action.

In general it is not necessary to update packages when new features have been added. Remote exploits have to be fixed immediately, local exploits within a reasonable time frame because users may have to be informed.

- A Debian package which fixes a security hole can be installed very easily. First the newest package lists have to be downloaded using the command `apt-get update`. Once this is done `apt-get install <package-name>` will replace the old package with the newer version. Note that the `/usr` partition has to be remounted `read-write` to allow installation of new packages in `/usr`.

  ```
  jumphost# mount -o remount,rw /usr
  jumphost# apt-get update
  jumphost# apt-get install <package-name>
  jumphost# mount -o remount,ro /usr
  ```

- Self-compiled software, namely Kernel, Openssh, Iptables, Monit and Quota-tools have to be handled according to the procedures described in the section 4. Before installing these packages on the production jumphost they have to be tested on a jumphost clone in the lab.

8.2 Users

There are several tasks that have to be done on a regular basis by the system administrator.

8.2.1 SSH Keys

As `sshd_config` set the `AuthorizedKeysFile` option to `/etc/ssh/authorized_keys.%u` (see 4.4.3) the keys need to be maintained by the administrator. For every user who is authorizing himself with the keys a file is created in `/etc/ssh`. The name of the file is `authorized_keys` followed by the user name.

As an additional security measure this file should belong to root and should only be readable by root. Anyway this should automatically be the case because only root can create files in `/etc/ssh` and `umask` had been set to `066`.

```
jumphost# vi /etc/ssh/authorized_keys.exampleusr
jumphost# chmod 600 /etc/ssh/authorized_keys.exampleusr
```

8.2.2 Add and remove jailed users

Because there are only a few administrators the adding and removing of administrators will not be detailed here.

The adding and removing of jailed users differs from the standard procedures.

This is the way to add a new user. The script to do this task has been described in section 6.4. To allow the jailed users to connect to any servers they want, the servers and ports have to be added to `/etc/iptables.accept` accordingly (see 4.2.4).

```
jumphost# addjailuser.sh exampleusr "John Miller XYcompany exampleusr@xycompany.com"
jumphost# vi /etc/iptables.accept
```

The removal of a jailed user is standard. The `-r` flag will also remove the home directory. `sync_shadow.sh` (see 4.6.3) has to be run in order to update the jail password files. The entries in `/etc/iptables.accept` belonging to the removed user have to be cleaned out.

```
jumphost# deluser -r exampleusr
jumphost# sync_shadow.sh
jumphost# vi /etc/iptables.accept
```

8.3 Monitoring

The administrator is responsible to read mails sent to him by `logcheck` (see 5.7) and `integrit` (see 5.8). If log entries sent to the administrator which are not security relevant the `logcheck` configuration files need to be adjusted. The same is valid for `integrit`. If files are allowed to change without noticing, the `integrity` configuration files need to be changed.

9 Testing

The following tests have been done to ensure the jumphost server is configured as expected:

9.1 System

- Except for the Kernel processes only `syslogd`, `klogd`, `monit`, `sshd`, `cron` and `bash` of the currently logged in user showed up.

```
jumphost# ps -ef
..
20642 ?        00:00:09 syslogd
11942 ?        00:15:21 klogd
24357 ?        00:41:06 monit
14076 ?        00:00:00 sshd
 8322 ?        00:00:15 cron
 6659 ?        00:00:22 sshd
 1349 ttyp0    00:00:00 bash
28903 ttyp0    00:00:00 bash
```

- `netstat -l` lists all the listening ports. Only `ssh` was in the `LISTEN` state.
- It's always good to get a second option. `lsof` had been run. The output is listed in Appendix I.
- `nmap -P0 -sX -sU -p 1-65535 jumphost` was run from another host within the same network (no firewalls in between).
- `apt-get update` and `/etc/init.d/ntpdate start` were run. If this does not work, probably `/etc/iptables.accept` was not properly configured.
- A file Integrit is monitoring was changed (added an empty line to `/etc/fstab`). A mail was sent indicating that `/etc/fstab` has been changed. To force integrit to check now instead of waiting for cron to do it (even though this is better to test cron as well) `/etc/cron.daily/integrit` can be issued.

```
changed: /etc/fstab   s(invalidchecksum454g16c5ffd53f5faa13deejls89044543f8f)
```

- Remounted `/usr` with running `mount -o rw /usr`. This created a log event, logcheck picked it up and sent a mail. The impatient could run `/etc/cron.d/logcheck` to get informed immediately.

```
jumphost kernel: grsec: .. mount /dev/sda4 to /usr by /bin/mount..
```

9.2 Jailed users

An important security test is to check if the jailed users can only do what they are allowed to. To test this behavior a test user with name testusr was created. Note the difference of the `passwd` entry on the system and on the jail.

```
jumphost# addjailuser.sh testusr "test"
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
User testusr added successfully.
jumphost# grep testusr /etc/passwd
testusr:x:1003:1000:test:/jail/./home/testusr:/bin/bash
jumphost# grep testusr /jail/etc/passwd
testusr:x:1003:1000:test:/home/testusr:/bin/bash
```

After the user logged in he finds himself jailed. The user was able to ping the target machine `10.1.1.1` but failed the SSH connection attempt failed because `/etc/iptables.accept` did not have entries for this user yet. As soon as `iptables.accept` had been modified and `iptables` restarted by root, testusr could successfully connect to `10.1.1.1`.

```
jumphost(testusr)# pwd
/home/testusr
jumphost(testusr)# ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1): 56 data bytes
64 bytes from 10.1.1.1: icmp_seq=0 ttl=253 time=171.4 ms
jumphost(testusr)# ssh 10.1.1.1
ssh: connect to host 10.1.1.15 port 22: Connection timed out

jumphost# echo "testusr:10.1.1.1:22" >> /etc/iptables.accept
jumphost# /etc/init.d/iptables restart
jumphost# iptables -L |grep testusr
ACCEPT  tcp  ..  10.1.1.1  ..  dpt:ssh ..  OWNER UID match testusr

jumphost(testusr)# ssh 10.1.1.1
testusr@10.1.1.1's password:

jumphost# grep testusr /var/log/messages
Sep 18 18:11:32 jumphost kernel: grsec: .. chdir to /jail/home/testusr by /usr/sbin/sshd ..
Sep 18 18:24:09 jumphost kernel: ACCEPT TCP22 (testusr): .. DST=10.1.1.1 .. PROTO=TCP .. DPT=22
```

The user changed his password. In the background `monit` detected that the MD5 hash of `/jail/etc/shadow` had changed and ran `sync_shadow.sh` which updated `/etc/shadow`.

```
jumphost(testusr)# passwd
Changing password for testusr
(current) UNIX password: oldpassword
Enter new UNIX password: simplepassword
Retype new UNIX password: simplepassword
Bad: new password is too short
Enter new UNIX password: strongpassword
Retype new UNIX password: strongpassword
passwd: password updated successfully
jumphost(testusr)# exit
testusr@jumphost's password: newpassword
jumphost(testusr)#
```

A jailed user is allowed to fill up maximum of 100 MBytes of disk space. For testing purposes this is was reduced to 2 MBytes and a file bigger than 2 MBytes was copied. As a result not the whole file had been copied but only the first 2 MBytes.

```
jumphost# repquota /jail/home
*** Report for user quotas on device /dev/sdb2
Block grace time: 7days; Inode grace time: 7days
                        Block limits                File limits
User            used    soft    hard  grace    used  soft  hard  grace
----------------------------------------------------------------------
root      --       0       0       0              3     0     0
testusr   --      24   80000  100000              5     0     0
jumphost# setquota testusr 2000 2500 0 0 /jail/home
jumphost# repquota /jail/home |grep testusr
testusr   --      24    2000    2000              5     0     0
jumphost(testusr)# scp security@10.1.1.1:/var/adm/messages .
messages: Disk quota exceeded
jumphost(testusr)# ls -l
-rw-------    1 testusr  1000      1966080 Sep 18 17:04 messages
jumphost# repquota /jail/home |grep testusr
testusr   --    1944    2000    2000              6     0     0
```

Last but not least a common breakout of jail was tried. A jailed user is not allowed to execute files from his home (mounted with noexec option). Still, for a test `/jail/home` has been remounted without `noexec`. Without Grsecurity compiled into the Kernel the following code compiled and ran by a jailed user was able to break out of the Chrooted environment. With Grsecurity this was not possible anymore.

```
/* Using Chroot Securely
   By Anton Chuvakin, Ph.D.
```

```
   2/10/2002 17:19
   http://www.linuxsecurity.com/feature_stories/feature_story-99.html
*/
#include <stdlib.h>
#include <stdio.h>
#include <unistd.h>
#include <sys/stat.h>
#include <sys/types.h>
int main(void)
{
    int i;
    mkdir("breakout", 0700);
    chroot("breakout");
    for (i = 0; i < 100; i++)
      chdir("..") ;
    chroot(".");
    execl("/bin/sh", "/bin/sh",NULL);
}
```

10 Backup and clone

The jumphost is meant to be a system without important user data on it. If the hard disks of a jumphost crash, a new jumphost will be put into place. Because the jumphost user data is stored on a another disk than the passwords either one of them will be lost. New passwords respectively new empty home directories would be created.

The work that has been done for this jumphost only pays of if it can be cloned. One reason to choose this hardware was that spare machines are available. Here comes a short description of how the jumphost was cloned.

1. The whole system (ca. 200MBytes) was archived and transfered to another machine where it was burned on a CD.

   ```
   debian# cd /
   debian# tar cf /home/jumphost.tar .
   debian# scp /home/jumphost.tar myburner:
   myburner# mkisofs -V jumphost -o jumphost.iso jumphost.tar
   myburner# cdrecord dev=0,0,0 -data jumphost.iso
   ```

2. The new hardware had been booted with the Debian installation CD. The first installation steps (swap, partitioning and mounting) have been done as usual. Then "Execute a shell" was chosen from the installation menu. The Debian installation CD was ejected and the jumphost CD inserted. After the jumphost CD had been mounted `jumphost.tar` was extracted in `/target`.

   ```
   debian# mount -o ro /dev/scd0 /mnt
   debian# cd /target
   debian# tar xf /mnt/jumphost.tar
   ```

3. The new system was installed. The good thing about using `tar` (instead of `dd` for example) is that the partitioning and mount points do not need to be identical on the new jumphost.

   It is important to adjust `/target/etc/fstab` if the partitioning is not identical to the "originating" jumphost. The next step was to make the system bootable. This has been achieved by chrooting into the new system, modifying `/etc/silo` and running `silo -f`.

```
debian# cd /target
debian# chroot .
debian# vi /etc/fstab
debian# vi /etc/silo.conf
debian# silo -f
debian# reboot
```

11 References

- GNU/Linux manual pages and /usr/share/doc
- Hal Pomeranz, Copyright 2000-2004, SANS GCUX course material
- Debian GNU/Linux, Debian developers
  http://www.debian.org
- Linux Kernel, Copyright 1997-2003 The Kernel.Org Organization, Inc.
  http://www.kernel.org
- Grsecurity, Brad Spengler and Michael Dalton
  http://www.grsecurity.net
- Iptables/Netfilter, Netfilter Core Team
  http://www.netfilter.org
- Iptables/Netfilter Owner-socketlookup patch, Patrick McHardy
  http://www.netfilter.org/patch-o-matic/pom-extra.html#pom-extra-owner-socketlook
- Openssh, Openssh developers
  http://www.openssh.org
- james@firstaidmusic.com, Chroot SSH,
  http://chrootssh.sourceforge.net/index.php
- Quota-tools, Linuxquota developers
  http://sourceforge.net/projects/linuxquota
- Monit, The monit project group
  http://www.tildeslash.com/monit/

- Logcheck, Rami Dass
  http://www.astro.uiuc.edu/~r-dass/logcheck/
- Integrit, Integrit developers
  http://integrit.sourceforge.net/
- 1996 Bruce Perens, 1996, 1997 Sven Rudolph, 1998 Igor Grobman, James Treacy 1998-2002 Adam Di Carlo, "Installing Debian GNU/Linux 3.0 For SPARC"
  http://www.debian.org/releases/stable/sparc/install
- http://www.spi-inc.org/SPI, Copyright 1997-2004, "Downloading Debian CD images via HTTP/FTP"
  http://www.debian.org/CD/http-ftp
- XFS filesystem FAQ, "Do quotas work?",
  http://oss.sgi.com/projects/xfs/faq.html#quotaswork
- nathans@sgi.com, XFS quota Readme, "QUOTA on XFS", /usr/share/doc/xfsprogs/README.quota.gz
- Ethan Benson ¡erbenson@alaska.net¿, "Re: Why is /var/cache/man/ sgid root?"
  http://lists.debian.org/debian-devel/2000/10/msg01889.html

A Iptables init script

```
# ------------------------- /etc/init.d/iptables -----------------------------
#!/bin/sh

stop() {
    PROG="/usr/sbin/iptables"
    $PROG -P INPUT ACCEPT
    $PROG -P FORWARD DROP
    $PROG -P OUTPUT ACCEPT
    $PROG --flush
}

start() {

PROG="/usr/sbin/iptables"
IPT_ACCEPT="/etc/iptables.accept"
CUT="/usr/bin/cut"
GREP="/bin/grep"
IPADDR=192.168.254.10
DNSSRV=192.168.254.20
NTPSRV=192.168.254.30
ILLEGAL_TCP_FLAGS="SYN,FIN PSH,FIN SYN,ACK,FIN SYN,FIN,PSH SYN,FIN,RST
  SYN,FIN,RST,PSH SYN,FIN,ACK,RST SYN,ACK,FIN,RST,PSH ALL RST"
```

```
$PROG -P INPUT DROP
$PROG -P FORWARD DROP
$PROG -P OUTPUT DROP
$PROG --flush

# INPUT

# accept LOOPBACK on loopback, log and drop on eth0
$PROG -A INPUT -i lo -j ACCEPT
$PROG -A INPUT -d 127.0.0.1 -i eth0 -j LOG --log-level 6 --log-prefix \
  "DROP INVALID_SOURCE: "
$PROG -A INPUT -d 127.0.0.1 -i eth0 -j DROP

# log and drop INVALID RANGES
$PROG -A INPUT -s 224.0.0.0/240.0.0.0 -i eth0 -j LOG --log-level 6 \
  --log-prefix "DROP INVALID_SOURCE: "
$PROG -A INPUT -s 224.0.0.0/240.0.0.0 -i eth0 -j DROP
$PROG -A INPUT -s 240.0.0.0/248.0.0.0 -i eth0 -j LOG --log-level 6 \
  --log-prefix "DROP INVALID_SOURCE: "
$PROG -A INPUT -s 240.0.0.0/248.0.0.0 -i eth0 -j DROP

# log and drop ILLEGAL FLAGS
for flag in $ILLEGAL_TCP_FLAGS; do
    $PROG -A INPUT -i eth0 -p tcp --tcp-flags ALL $flag \
          -j LOG --log-level 6 --log-prefix "DROP ILLEGAL_TCP_FLAGS: " \
          --log-ip-options --log-tcp-options
    $PROG -A INPUT -i eth0 -p tcp --tcp-flags ALL $flag -j DROP
done
$PROG -A INPUT -i eth0 -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m state \
  --state NEW -j LOG --log-level 6 --log-prefix "DROP ILLEGAL_TCP_FLAGS: " \
  --log-ip-options --log-tcp-options
$PROG -A INPUT -i eth0 -p tcp -m tcp ! --tcp-flags SYN,RST,ACK SYN -m state \
   --state NEW -j DROP

# log and drop INVALID STATE
$PROG -A INPUT -i eth0 -m state --state INVALID -j LOG --log-prefix \
  "DROP INVALID_STATE: " --log-level 6
$PROG -A INPUT -i eth0 -m state --state INVALID -j DROP

# log and drop ICMP
$PROG -A INPUT -d $IPADDR -i eth0 -p icmp -f -j LOG --log-prefix \
  "DROP ICMP_IN: " --log-level 6
$PROG -A INPUT -d $IPADDR -i eth0 -p icmp -f -j DROP

# accept ESTABLISHED
$PROG -A INPUT -d $IPADDR -i eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT
# log and accept incoming ssh
$PROG -A INPUT -d $IPADDR -i eth0 -p tcp -m tcp --sport 1024:65535 --dport 22 \
  -m state --state NEW -j LOG --log-prefix "ACCEPT SSH_IN: " --log-level 6
$PROG -A INPUT -d $IPADDR -i eth0 -p tcp -m tcp --sport 1024:65535 --dport 22 \
   -m state --state NEW -j ACCEPT

# log all incoming dropped traffic
#$PROG -A INPUT -j LOG --log-prefix "DROP IN: " --log-level 6
```

```
# OUTPUT

# GENERAL
# -------

# accept LOOPBACK
$PROG -A OUTPUT -o lo -j ACCEPT

# log and drop INVALID STATE
$PROG -A OUTPUT -o eth0 -m state --state INVALID -j LOG --log-prefix \
  "DROP INVALID_STATE: " --log-level 6
$PROG -A OUTPUT -o eth0 -m state --state INVALID -j DROP

# accept ESTABLISHED
$PROG -A OUTPUT -s $IPADDR -o eth0 -m state --state RELATED,ESTABLISHED \
  -j ACCEPT

# accept and log outgoing ICMP (echo request)
$PROG -A OUTPUT -s $IPADDR -o eth0 -p icmp -m icmp --icmp-type 8 -m state \
  --state NEW -j LOG --log-prefix "ACCEPT ICMP_OUT: " --log-level 6
$PROG -A OUTPUT -s $IPADDR -o eth0 -p icmp -m icmp --icmp-type 8 -m state \
  --state NEW -j ACCEPT

# accept and log outgoing TRACEROUTE (root only)
$PROG -A OUTPUT -s $IPADDR -o eth0 -p udp -m udp --sport 32769:65535 --dport \
  33434:33523 -m state --state NEW -m owner --uid-owner root -j LOG --log-prefix \
  "ACCEPT TRACEROUTE_OUT: " --log-level 6
$PROG -A OUTPUT -s $IPADDR -o eth0 -p udp -m udp --sport 32769:65535 --dport \
  33434:33523 -m state --state NEW -m owner --uid-owner root -j ACCEPT

# accept and log outgoing NTP (root only)
$PROG -A OUTPUT -s $IPADDR -d $NTPSRV -o eth0 -p udp --sport 32768:61000 \
  --dport 123 -m state --state NEW,ESTABLISHED --uid-owner root -j LOG --log-prefix \
  "ACCEPT NTP_OUT: " --log-level 6
$PROG -A OUTPUT -s $IPADDR -d $NTPSRV -o eth0 -p udp --sport 32768:61000 \
  --dport 123 -m state --state NEW,ESTABLISHED --uid-owner root -j ACCEPT

# accept and log outgoing DNS
$PROG -A OUTPUT -s $IPADDR -d $DNSSRV -o eth0 -p udp --sport 32768:61000 \
  --dport 53 -m state --state NEW,ESTABLISHED -j LOG --log-prefix \
  "ACCEPT DNS_OUT: " --log-level 6
$PROG -A OUTPUT -s $IPADDR -d $DNSSRV -o eth0 -p udp --sport 32768:61000 \
  --dport 53 -m state --state NEW,ESTABLISHED -j ACCEPT

# USER $PECIFIC
# -------------

# accept and log outgoing SSH
IFS='
'
for i in `cat $IPT_ACCEPT |$GREP -v ^\# | $GREP -v ^$`; do
    owner="`echo $i |$CUT -d: -f 1`"
    dest="`echo $i |$CUT -d: -f 2`"
```

```
    port="`echo $i |$CUT -d: -f 3`"
    $PROG -A OUTPUT -s $IPADDR -d $dest -o eth0 -p tcp -m tcp \
        --sport 32768:61000 --dport $port -m state --state NEW  \
        -m owner --uid-owner $owner  \
        -j LOG --log-prefix "ACCEPT TCP$port ($owner): " --log-level 6
    $PROG -A OUTPUT -s $IPADDR -d $dest -o eth0 -p tcp -m tcp \
        --sport 32768:61000 --dport $port -m state --state NEW  \
        -m owner --uid-owner $owner  \
        -j ACCEPT
done
IFS=

# LOG ALL
# -------

# LOG ALL outgoing dropped traffic
$PROG -A OUTPUT -j LOG --log-prefix "DROP OUT: " --log-level 6

}


case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        stop
        start
        ;;
    *)
        echo "Syntax: $0 start|stop|restart"
        ;;
esac
# ---------------------------------------------------------------------------
```

B Kernel configuration

```
CONFIG_EXPERIMENTAL=y
CONFIG_BBC_I2C=y
CONFIG_VT=y
CONFIG_VT_CONSOLE=y
CONFIG_SPARC64=y
CONFIG_HAVE_DEC_LOCK=y
CONFIG_RWSEM_XCHGADD_ALGORITHM=y
CONFIG_SBUS=y
CONFIG_SBUSCHAR=y
CONFIG_BUSMOUSE=y
CONFIG_SUN_MOUSE=y
CONFIG_SERIAL=y
CONFIG_SUN_SERIAL=y
CONFIG_SERIAL_CONSOLE=y
CONFIG_SUN_KEYBOARD=y
CONFIG_SUN_CONSOLE=y
CONFIG_SUN_AUXIO=y
CONFIG_SUN_IO=y
CONFIG_PCI=y
CONFIG_RTC=y
CONFIG_NET=y
CONFIG_BSD_PROCESS_ACCT=y
CONFIG_SYSCTL=y
CONFIG_KCORE_ELF=y
CONFIG_SPARC32_COMPAT=y
CONFIG_BINFMT_ELF32=y
CONFIG_BINFMT_ELF=y
CONFIG_PROM_CONSOLE=y
CONFIG_SUN_OPENPROMIO=y
CONFIG_SUN_MOSTEK_RTC=y
CONFIG_BLK_DEV_LOOP=y
CONFIG_PACKET=y
CONFIG_PACKET_MMAP=y
CONFIG_NETFILTER=y
CONFIG_UNIX=y
CONFIG_INET=y
CONFIG_SYN_COOKIES=y
CONFIG_IP_NF_CONNTRACK=y
CONFIG_IP_NF_FTP=y
CONFIG_IP_NF_IPTABLES=y
CONFIG_IP_NF_MATCH_LIMIT=y
CONFIG_IP_NF_MATCH_MAC=y
CONFIG_IP_NF_MATCH_MULTIPORT=y
CONFIG_IP_NF_MATCH_STATE=y
CONFIG_IP_NF_MATCH_CONNTRACK=y
CONFIG_IP_NF_MATCH_OWNER=y
CONFIG_IP_NF_FILTER=y
CONFIG_IP_NF_TARGET_REJECT=y
CONFIG_IP_NF_TARGET_LOG=y
CONFIG_IP_NF_TARGET_ULOG=y
CONFIG_IP_NF_ARPTABLES=y
CONFIG_IP_NF_ARPFILTER=y
CONFIG_IP_NF_ARP_MANGLE=y
CONFIG_SCSI=y
CONFIG_BLK_DEV_SD=y
CONFIG_SD_EXTRA_DEVS=40
CONFIG_BLK_DEV_SR=y
CONFIG_SR_EXTRA_DEVS=2
CONFIG_CHR_DEV_SG=y
CONFIG_SCSI_SYM53C8XX_2=y
CONFIG_SCSI_SYM53C8XX_DMA_ADDRESSING_MODE=1
CONFIG_SCSI_SYM53C8XX_DEFAULT_TAGS=16
CONFIG_SCSI_SYM53C8XX_MAX_TAGS=64
CONFIG_SCSI_QLOGIC_ISP=y
CONFIG_SCSI_QLOGIC_FC=y
CONFIG_SCSI_QLOGIC_FC_FIRMWARE=y
CONFIG_NETDEVICES=y
CONFIG_NET_ETHERNET=y
CONFIG_HAPPYMEAL=y
CONFIG_EXT3_FS=y
CONFIG_JBD=y
CONFIG_TMPFS=y
CONFIG_RAMFS=y
CONFIG_ISO9660_FS=y
CONFIG_PROC_FS=y
CONFIG_EXT2_FS=y
CONFIG_XFS_FS=y
CONFIG_XFS_QUOTA=y
CONFIG_MSDOS_PARTITION=y
CONFIG_SUN_PARTITION=y
CONFIG_LOG_BUF_SHIFT=0
CONFIG_CRYPTO=y
CONFIG_CRYPTO_HMAC=y
CONFIG_CRYPTO_NULL=y
CONFIG_CRYPTO_MD4=y
CONFIG_CRYPTO_MD5=y
CONFIG_CRYPTO_SHA1=y
CONFIG_CRYPTO_SHA256=y
CONFIG_CRYPTO_SHA512=y
CONFIG_CRYPTO_DES=y
CONFIG_CRYPTO_BLOWFISH=y
CONFIG_CRYPTO_AES=y
CONFIG_CRYPTO_CAST5=y
CONFIG_CRYPTO_CAST6=y
CONFIG_CRYPTO_ARC4=y
CONFIG_ZLIB_INFLATE=y
CONFIG_ZLIB_DEFLATE=y
CONFIG_GRKERNSEC=y
CONFIG_CRYPTO=y
CONFIG_CRYPTO_SHA256=y
CONFIG_GRKERNSEC_CUSTOM=y
CONFIG_GRKERNSEC_PAX_EI_PAX=y
CONFIG_GRKERNSEC_PAX_PT_PAX_FLAGS=y
CONFIG_GRKERNSEC_PAX_NO_ACL_FLAGS=y
CONFIG_GRKERNSEC_PAX_NOEXEC=y
```

```
CONFIG_GRKERNSEC_PAX_PAGEEXEC=y
CONFIG_GRKERNSEC_PAX_MPROTECT=y
CONFIG_GRKERNSEC_PAX_EMUPLT=y
CONFIG_GRKERNSEC_PAX_DLRESOLVE=y
CONFIG_GRKERNSEC_PAX_ASLR=y
CONFIG_GRKERNSEC_PAX_RANDUSTACK=y
CONFIG_GRKERNSEC_PAX_RANDMMAP=y
CONFIG_GRKERNSEC_PAX_RANDEXEC=y
CONFIG_GRKERNSEC_KMEM=y
CONFIG_GRKERNSEC_PROC_MEMMAP=y
CONFIG_GRKERNSEC_HIDESYM=y
CONFIG_GRKERNSEC_ACL_HIDEKERN=y
CONFIG_GRKERNSEC_ACL_MAXTRIES=3
CONFIG_GRKERNSEC_ACL_TIMEOUT=30
CONFIG_GRKERNSEC_PROC=y
CONFIG_GRKERNSEC_PROC_USER=y
CONFIG_GRKERNSEC_PROC_ADD=y
CONFIG_GRKERNSEC_LINK=y
CONFIG_GRKERNSEC_FIFO=y
CONFIG_GRKERNSEC_CHROOT=y
CONFIG_GRKERNSEC_CHROOT_MOUNT=y
CONFIG_GRKERNSEC_CHROOT_DOUBLE=y
CONFIG_GRKERNSEC_CHROOT_PIVOT=y
CONFIG_GRKERNSEC_CHROOT_CHDIR=y
CONFIG_GRKERNSEC_CHROOT_CHMOD=y
CONFIG_GRKERNSEC_CHROOT_FCHDIR=y
CONFIG_GRKERNSEC_CHROOT_MKNOD=y
CONFIG_GRKERNSEC_CHROOT_SHMAT=y
CONFIG_GRKERNSEC_CHROOT_UNIX=y
CONFIG_GRKERNSEC_CHROOT_FINDTASK=y
CONFIG_GRKERNSEC_CHROOT_NICE=y
CONFIG_GRKERNSEC_CHROOT_SYSCTL=y
CONFIG_GRKERNSEC_AUDIT_GROUP=y
CONFIG_GRKERNSEC_AUDIT_GID=1000
CONFIG_GRKERNSEC_EXECLOG=y
CONFIG_GRKERNSEC_RESLOG=y
CONFIG_GRKERNSEC_CHROOT_EXECLOG=y
CONFIG_GRKERNSEC_AUDIT_CHDIR=y
CONFIG_GRKERNSEC_AUDIT_MOUNT=y
CONFIG_GRKERNSEC_AUDIT_IPC=y
CONFIG_GRKERNSEC_SIGNAL=y
CONFIG_GRKERNSEC_FORKFAIL=y
CONFIG_GRKERNSEC_TIME=y
CONFIG_GRKERNSEC_EXECVE=y
CONFIG_GRKERNSEC_DMESG=y
CONFIG_GRKERNSEC_RANDPID=y
CONFIG_GRKERNSEC_TPE=y
CONFIG_GRKERNSEC_TPE_GID=1000
CONFIG_GRKERNSEC_RANDNET=y
CONFIG_GRKERNSEC_RANDISN=y
CONFIG_GRKERNSEC_RANDID=y
CONFIG_GRKERNSEC_RANDSRC=y
CONFIG_GRKERNSEC_RANDRPC=y
CONFIG_GRKERNSEC_SOCKET=y
CONFIG_GRKERNSEC_SOCKET_SERVER=y
CONFIG_GRKERNSEC_SOCKET_SERVER_GID=1000
CONFIG_GRKERNSEC_FLOODTIME=10
CONFIG_GRKERNSEC_FLOODBURST=4
```

C List of installed packages

```
adduser
apt
apt-utils
base-config
base-files
base-passwd
bash
bsdmainutils
bsdutils
console-common
console-data
console-tools
console-tools-
cron
debconf
debianutils
diff
dpkg
e2fsprogs
ethtool
fileutils
findutils
ftp
gettext-base
grep
groff-base
gzip
hostname
ifupdown
integrit
iptables
iptables-cc
klogd
less
libc6
libcap1
libdb2
libdb3
libgdbmg1
libgpmg1
libident
liblockfile1
libncurses5
```

```
libnewt0
libpam-modules
libpam-runtime
libpam0g
libpcap0
libperl5.6
libpopt0
libreadline4
libsasl7
libssl0.9.6
libstdc++2.10-
libwrap0
logcheck
logcheck-datab
login
logrotate
logtail
lsof
mailx
makedev
man-db
manpages
mawk
monit-cc
mount
ncurses-base
ncurses-bin
net-tools
netbase
netkit-inetd
netkit-ping
ntpdate
nvi
openssh-cc
passwd
pciutils
perl
perl-base
perl-modules
procps
quota-tools-cc
sed
shellutils
silo
slang1
sparc-utils
ssmtp
strace
sysklogd
sysvinit
tar
tasksel
tcpd
telnet
textutils
traceroute
util-linux
vim
whiptail
xfsprogs
zlib1g
```

D SSH daemon configuration file

```
# ------------------------- /etc/ssh/sshd_config -----------------------------
#       $OpenBSD: sshd_config,v 1.68 2003/12/29 16:39:50 millert Exp $

# This is the sshd server system-wide configuration file.  See
# sshd_config(5) for more information.

# This sshd was compiled with PATH=/usr/bin:/bin:/usr/sbin:/sbin

# The strategy used for options in the default sshd_config shipped with
# OpenSSH is to specify options with their default value where
# possible, but leave them commented.  Uncommented options change a
# default value.

#Port 22
Protocol 2
#ListenAddress 0.0.0.0
#ListenAddress ::

# HostKey for protocol version 1
#HostKey /etc/ssh/ssh_host_key
# HostKeys for protocol version 2
#HostKey /etc/ssh/ssh_host_rsa_key
#HostKey /etc/ssh/ssh_host_dsa_key

# Lifetime and size of ephemeral version 1 server key
#KeyRegenerationInterval 1h
#ServerKeyBits 768

# Logging
#obsoletes QuietMode and FascistLogging
#SyslogFacility AUTH
LogLevel VERBOSE

# Authentication:

#LoginGraceTime 2m
PermitRootLogin no
#StrictModes yes

RSAAuthentication no
#PubkeyAuthentication yes
AuthorizedKeysFile      /etc/ssh/authorized_keys.%u

# For this to work you will also need host keys in /etc/ssh/ssh_known_hosts
#RhostsRSAAuthentication no
# similar for protocol version 2
#HostbasedAuthentication no
# Change to yes if you don't trust ~/.ssh/known_hosts for
# RhostsRSAAuthentication and HostbasedAuthentication
#IgnoreUserKnownHosts no
# Don't read the user's ~/.rhosts and ~/.shosts files
```

```
#IgnoreRhosts yes

# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no

# Change to no to disable s/key passwords
#ChallengeResponseAuthentication yes

# Kerberos options
#KerberosAuthentication no
#KerberosOrLocalPasswd yes
#KerberosTicketCleanup yes
#KerberosGetAFSToken no

# GSSAPI options
#GSSAPIAuthentication no
#GSSAPICleanupCredentials yes

# Set this to 'yes' to enable PAM authentication (via challenge-response)
# and session processing. Depending on your PAM configuration, this may
# bypass the setting of 'PasswordAuthentication' and 'PermitEmptyPasswords'
#UsePAM no

#AllowTcpForwarding yes
#GatewayPorts no
X11Forwarding yes
#X11DisplayOffset 10
#X11UseLocalhost yes
#PrintMotd yes
#PrintLastLog yes
#TCPKeepAlive yes
#UseLogin no
#UsePrivilegeSeparation yes
#PermitUserEnvironment no
#Compression yes
#ClientAliveInterval 0
#ClientAliveCountMax 3
#UseDNS yes
#PidFile /var/run/sshd.pid
#MaxStartups 10

# no default banner path
Banner /etc/issue

# override default of no subsystems
Subsystem       sftp    /usr/libexec/sftp-server
# ------------------------------------------------------------------------------
```

E Password synchronization script

```
# ------------------------- /usr/local/bin/sync_shadow.sh -------------------
#!/bin/sh

#
# Author:      Roland Mathis
# Description: This script synchronizes /jail/etc/shadow with
#              /etc/shadow, but only for the users of the group
#              jail
# Version history:
# 2004-08-23   Roland Mathis   initial version
#

# variable definitions
passwd=/etc/passwd
shadow=/etc/shadow
jpasswd=/jail/etc/passwd
jshadow=/jail/etc/shadow

# umask
umask 066

# if (system shadow is newer than jail shadow and no param was given)
# then synchronize from system to jail
# ---------------------------------------------------------------------

if [ $shadow -nt $jshadow -a "x$1" == "x" ]; then

    # search system-passwd for users who belong to group 1000
    # and rewrite jail-passwd accordingly
    awk ' BEGIN { FS=OFS=":"; } {
        if ($4 == 1000) {
            split($6,home,"\/\.");
            print $1, $2, $3, $4, $5, home[2], $7;
        }
    } ' $passwd >$jpasswd

    # get shadow-entry for each jail-user from system-shadow and rewrite
    # jail-shadow accordingly
    rm $jshadow
    for i in `cat $jpasswd |cut -d: -f 1`; do
        grep "^$i:" $shadow >>$jshadow
    done


# if (jail shadow is newer than system shadow and param 'monit' was given)
# then synchronize from jail to system
# ---------------------------------------------------------------------

elif [ $jshadow -nt $shadow -a "$1" == "monit" ]; then
    for usr in `cut -d: -f 1 $jshadow`; do
        # is usr a member of group jail?
```

```
        if [ "`groups $usr |grep jail`" ]; then
            usermod -p `grep ^$usr $jshadow |cut -d: -f 2` $usr
        fi
    done

# syntax
# ------------------------------------------------------------------

else
    echo
    echo "ERROR: No sync was done!"
    echo
    echo "1) to synchronize from $passwd to $jpasswd "
    echo "   condition 1: $passwd is newer than $jpasswd"
    echo "   condition 2: user is a member of group jail"
    echo "   Syntax: $0"
    echo
    echo "2) to synchronize from $jpasswd to $passwd "
    echo "   condition 1: $jpasswd is newer than $passwd"
    echo "   condition 2: user is a member of group jail"
    echo "   Syntax: $0 monit"
    echo
fi
# ------------------------------------------------------------------------
```

F Create jail environment script

```
# ------------------------- create_jail.sh ---------------------------------
#!/bin/bash

#
# Author:      Roland Mathis
# Description: This script creates a new jail environment in /jail
# Version history:
# 2004-08-12   Roland Mathis   initial version
#

# create jail
umask 666
jail=/jail
umount $jail/home
if [ $? -ne 0 ]; then
    echo "abort, cannot umount $jail/home"
    exit 1
fi
rm -rf $jail
mkdir -p $jail/home
mount /jail/home
rm -rf $jail/home/*
```

```
# binaries in the jail
bins="/bin/bash /usr/bin/passwd /bin/ls /bin/mv /bin/rm /usr/bin/ssh
  /bin/ping /usr/bin/scp /usr/bin/ssh /usr/bin/telnet /usr/bin/ftp"

# create skel
cd $jail
mkdir etc
mkdir bin
mkdir lib
mkdir dev

# mknod
mknod $jail/dev/null c 1 3
chmod 666 $jail/dev/null
mknod $jail/dev/urandom c 1 9
chmod 444 $jail/dev/urandom
mknod $jail/dev/tty c 5 0
chmod 666 $jail/dev/tty

# prepare passwd and shadow (empty)
touch $jail/etc/passwd
touch $jail/etc/shadow

# ftp and telnet need services
echo "ftp       21/tcp" >$jail/etc/services
echo "telnet    23/tcp" >>$jail/etc/services
# icmp
echo "icmp    1       ICMP" >>$jail/etc/protocols

# pam.conf
echo "passwd    password        required        /lib/pam_unix.so obscure min=6 max=20 md5
other   auth            required        /lib/pam_deny.so
other   account         required        /lib/pam_deny.so
other   password        required        /lib/pam_deny.so
other   session         required        /lib/pam_deny.so
" >$jail/etc/pam.conf

# resolv.conf
cp /etc/resolv.conf $jail/etc

# copy files and libs
for i in $bins;  do
    cp $i bin
    ldd $i >/dev/null
    libs=‘ldd $i | grep -v "not a dynamic executable" |awk ’{ print $3 }’‘
    for l in $libs; do
        cp $l lib
    done
done

# nameservices libs
cp /lib/libnss_compat.so.2 /lib/libnss_dns.so.2 /lib/libnsl.so.1 \
  /lib/libnss_files.so.2 /lib/security/pam_unix.so /lib/security/pam_deny.so $jail/lib
```

```
# adjust perms
chmod 555 $jail/lib/*
chmod 111 $jail/bin/*
chmod 444 $jail/etc/*
chmod 640 $jail/etc/shadow
chmod 644 $jail/etc/passwd
chgrp shadow $jail/etc/shadow

# ping and passwd needs suid :/
chmod u+s $jail/bin/ping
chmod u+s $jail/bin/passwd
# ------------------------------------------------------------------------
```

G Jail environment

```
/jail/:
total 4
d--x--x--x    2 root     root         1024 Sep 11 18:51 bin
d--x--x--x    2 root     root         1024 Sep 11 18:51 dev
d--x--x--x    2 root     root         1024 Sep 11 18:51 etc
drwxr-xr-x    2 root     root            6 Sep 11 18:51 home
d--x--x--x    2 root     root         1024 Sep 11 18:51 lib

/jail/bin:
total 1387
---x--x--x    1 root     root       631484 Sep 11 18:51 bash
---x--x--x    1 root     root        75808 Sep 11 18:51 ftp
---x--x--x    1 root     root        55316 Sep 11 18:51 ls
---x--x--x    1 root     root        52332 Sep 11 18:51 mv
---s--x--x    1 root     root        86692 Sep 11 18:51 passwd
---s--x--x    1 root     root        21248 Sep 11 18:51 ping
---x--x--x    1 root     root        33668 Sep 11 18:51 rm
---x--x--x    1 root     root        38924 Sep 11 18:51 scp
---x--x--x    1 root     root       238396 Sep 11 18:51 ssh
---x--x--x    1 root     root       167100 Sep 11 18:51 telnet

/jail/dev:
total 0
crw-rw-rw-    1 root     root       1,   3 Sep 11 18:51 null
crw-rw-rw-    1 root     root       5,   0 Sep 11 18:51 tty
cr--r--r--    1 root     root       1,   9 Sep 11 18:51 urandom

/jail/etc:
total 4
-r--r--r--    1 root     root          229 Sep 11 18:51 pam.conf
-rw-r--r--    1 root     root            0 Sep 11 18:51 passwd
-r--r--r--    1 root     root           21 Sep 11 18:51 protocols
-r--r--r--    1 root     root           68 Sep 11 18:51 resolv.conf
-r--r--r--    1 root     root           25 Sep 11 18:51 services
-rw-r-----    1 root     shadow          0 Sep 11 18:51 shadow
```

```
/jail/home:
total 0

/jail/lib:
total 3750
-r-xr-xr-x    1 root     root       118262 Sep 11 18:51 ld-linux.so.2
-r-xr-xr-x    1 root     root      1304548 Sep 11 18:51 libc.so.6
-r-xr-xr-x    1 root     root        83804 Sep 11 18:51 libcrypt.so.1
-r-xr-xr-x    1 root     root       856492 Sep 11 18:51 libcrypto.so.0.9.6
-r-xr-xr-x    1 root     root        67520 Sep 11 18:51 libdl.so.2
-r-xr-xr-x    1 root     root       365076 Sep 11 18:51 libncurses.so.5
-r-xr-xr-x    1 root     root       136716 Sep 11 18:51 libnsl.so.1
-r-xr-xr-x    1 root     root       101000 Sep 11 18:51 libnss_compat.so.2
-r-xr-xr-x    1 root     root        12716 Sep 11 18:51 libnss_dns.so.2
-r-xr-xr-x    1 root     root        39472 Sep 11 18:51 libnss_files.so.2
-r-xr-xr-x    1 root     root        37796 Sep 11 18:51 libpam.so.0
-r-xr-xr-x    1 root     root        13336 Sep 11 18:51 libpam_misc.so.0
-r-xr-xr-x    1 root     root        94565 Sep 11 18:51 libpthread.so.0
-r-xr-xr-x    1 root     root       185660 Sep 11 18:51 libreadline.so.4
-r-xr-xr-x    1 root     root       119188 Sep 11 18:51 libresolv.so.2
-r-xr-xr-x    1 root     root        84596 Sep 11 18:51 librt.so.1
-r-xr-xr-x    1 root     root         7896 Sep 11 18:51 libutil.so.1
-r-xr-xr-x    1 root     root       121676 Sep 11 18:51 libz.so.1
-r-xr-xr-x    1 root     root         3696 Sep 11 18:51 pam_deny.so
-r-xr-xr-x    1 root     root        47876 Sep 11 18:51 pam_unix.so
```

H Script to add jailed users

```
# -------------------------- /usr/local/bin/addjailuser.sh --------------------
#!/bin/sh

#
# Author:      Roland Mathis
# Description: This script adds a new jail user.
#
# Version history:
# 2004-08-12   Roland Mathis   initial version
#

# check params, display syntax
if [ $# -ne 2 ]; then
    echo "syntax: $0 <loginname> \"<firstname lastname company email@address>\""
    exit 1
fi

# create home dir, passwd and shadow entry
useradd -g jail -d "/jail/./home/$1" -c "$2" -m $1
if [ $? -ne 0 ]; then
    echo "ERROR: cannot useradd. Exit."
```

```
    exit 1
fi

# adjust dir perms
chmod 700 /jail/home/$1

# set quota (80MB soft, 100MB hard)
setquota $1 80000 100000 0 0 /jail/home
if [ $? -ne 0 ]; then
    echo "ERROR: cannot set quota. Exit."
    exit 1
fi

# set initial password
check=1
while [ $check -eq 1 ]; do
    passwd $1
    if [ $? -eq 0 ]; then check=0; fi
done

# get passwd and shadow entries
/usr/local/bin/sync_shadow.sh
if [ $? -ne 0 ]; then
    echo "ERROR: cannt sync to jail. Exit."
    exit 1
fi

echo "User $1 added successfully. "
# ------------------------------------------------------------------------
```

I Output of lsof

```
COMMAND     PID   USER   FD   TYPE             DEVICE    SIZE      NODE NAME
init          1   root  cwd    DIR                8,2    1024         2 /
init          1   root  rtd    DIR                8,2    1024         2 /
init          1   root  txt    REG                8,2   93852     12323 /sbin/init
init          1   root  mem    REG                8,2  118262     26640 /lib/ld-2.2.5.so
init          1   root  mem    REG                8,2 1304548     26643 /lib/libc-2.2.5.so
init          1   root   10u  FIFO                8,2              46314 /dev/initctl
keventd       2   root  cwd    DIR                8,2    1024         2 /
keventd       2   root  rtd    DIR                8,2    1024         2 /
ksoftirqd     3   root  cwd    DIR                8,2    1024         2 /
ksoftirqd     3   root  rtd    DIR                8,2    1024         2 /
kswapd        4   root  cwd    DIR                8,2    1024         2 /
kswapd        4   root  rtd    DIR                8,2    1024         2 /
bdflush       5   root  cwd    DIR                8,2    1024         2 /
bdflush       5   root  rtd    DIR                8,2    1024         2 /
kupdated      6   root  cwd    DIR                8,2    1024         2 /
kupdated      6   root  rtd    DIR                8,2    1024         2 /
xfsbufd       7   root  cwd    DIR                8,2    1024         2 /
```

```
xfsbufd       7   root  rtd    DIR              8,2    1024         2 /
xfslogd/0     8   root  cwd    DIR              8,2    1024         2 /
xfslogd/0     8   root  rtd    DIR              8,2    1024         2 /
xfsdatad/     9   root  cwd    DIR              8,2    1024         2 /
xfsdatad/     9   root  rtd    DIR              8,2    1024         2 /
scsi_eh_0    10   root  cwd    DIR              8,2    1024         2 /
scsi_eh_0    10   root  rtd    DIR              8,2    1024         2 /
kjournald    11   root  cwd    DIR              8,2    1024         2 /
kjournald    11   root  rtd    DIR              8,2    1024         2 /
less       1233   root  cwd    DIR              8,2    1024         2 /
less       1233   root  rtd    DIR              8,2    1024         2 /
less       1233   root  txt    REG              8,4  169316      4346 /usr/bin/less
less       1233   root  mem    REG              8,2  118262     26640 /lib/ld-2.2.5.so
less       1233   root  mem    REG              8,2  365076     26685 /lib/libncurses.so.5.2
less       1233   root  mem    REG              8,2 1304548     26643 /lib/libc-2.2.5.so
less       1233   root    0r  FIFO              0,5          7354634 pipe
less       1233   root    1u   CHR              3,0            45320 /dev/ttyp0
less       1233   root    2u   CHR              3,0            45320 /dev/ttyp0
less       1233   root    3r   CHR              5,0            45105 /dev/tty
lsof       2133   root  cwd    DIR              8,2    1024         2 /
lsof       2133   root  rtd    DIR              8,2    1024         2 /
lsof       2133   root  txt    REG              8,4  173368      8210 /usr/sbin/lsof
lsof       2133   root  mem    REG              8,2  118262     26640 /lib/ld-2.2.5.so
lsof       2133   root  mem    REG              8,2 1304548     26643 /lib/libc-2.2.5.so
lsof       2133   root    0u   CHR              3,0            45320 /dev/ttyp0
lsof       2133   root    1w  FIFO              0,5          7354633 pipe
lsof       2133   root    2u   CHR              3,0            45320 /dev/ttyp0
lsof       2133   root    3r   DIR              0,2        0       1 /proc
lsof       2133   root    4r   DIR              0,2        0 139788296 /proc/2133/fd
lsof       2133   root    5w  FIFO              0,5          7354640 pipe
lsof       2133   root    6r  FIFO              0,5          7354641 pipe
kjournald  2811   root  cwd    DIR              8,2    1024         2 /
kjournald  2811   root  rtd    DIR              8,2    1024         2 /
kjournald  4706   root  cwd    DIR              8,2    1024         2 /
kjournald  4706   root  rtd    DIR              8,2    1024         2 /
kjournald  7320   root  cwd    DIR              8,2    1024         2 /
kjournald  7320   root  rtd    DIR              8,2    1024         2 /
cron       8322   root  cwd    DIR              8,5    4096     29280 /var/spool/cron
cron       8322   root  rtd    DIR              8,2    1024         2 /
cron       8322   root  txt    REG              8,4   26740      8186 /usr/sbin/cron
cron       8322   root  mem    REG              8,2  118262     26640 /lib/ld-2.2.5.so
cron       8322   root  mem    REG              8,2   37796     26687 /lib/libpam.so.0.72
cron       8322   root  mem    REG              8,2 1304548     26643 /lib/libc-2.2.5.so
cron       8322   root  mem    REG              8,2   67520     26646 /lib/libdl-2.2.5.so
cron       8322   root  mem    REG              8,2   83804     26644 /lib/libcrypt-2.2.5.so
cron       8322   root  mem    REG              8,2  101000     26649 /lib/libnss_compat-2.2.5.so
cron       8322   root  mem    REG              8,2  136716     26648 /lib/libnsl-2.2.5.so
cron       8322   root    0r   CHR              1,3            45098 /dev/null
cron       8322   root    1w   CHR              1,3            45098 /dev/null
cron       8322   root    2w   CHR              1,3            45098 /dev/null
cron       8322   root    3u   REG              8,5        5   29291 /var/run/crond.pid
sed       11180   root  cwd    DIR              8,2    1024         2 /
sed       11180   root  rtd    DIR              8,2    1024         2 /
sed       11180   root  txt    REG              8,2   27604      4177 /bin/sed
```

```
sed       11180   root  mem    REG                8,2  118262     26640 /lib/ld-2.2.5.so
sed       11180   root  mem    REG                8,2 1304548     26643 /lib/libc-2.2.5.so
sed       11180   root    0r  FIFO                0,5          7354633 pipe
sed       11180   root    1w  FIFO                0,5          7354634 pipe
sed       11180   root    2u   CHR                3,0            45320 /dev/ttyp0
sshd      11326   root  cwd    DIR                8,2    1024        2 /
sshd      11326   root  rtd    DIR                8,2    1024        2 /
sshd      11326   root  txt    REG                8,4  342708     8243 /usr/sbin/sshd
sshd      11326   root  mem    REG                8,2  118262    26640 /lib/ld-2.2.5.so
sshd      11326   root  mem    REG                8,2  119188    26655 /lib/libresolv-2.2.5.so
sshd      11326   root  mem    REG                8,4  856492    38736 /usr/lib/libcrypto.so.0.9.6
sshd      11326   root  mem    REG                8,2    7896    26657 /lib/libutil-2.2.5.so
sshd      11326   root  mem    REG                8,4  121676    38734 /usr/lib/libz.so.1.1.4
sshd      11326   root  mem    REG                8,2  136716    26648 /lib/libnsl-2.2.5.so
sshd      11326   root  mem    REG                8,2   83804    26644 /lib/libcrypt-2.2.5.so
sshd      11326   root  mem    REG                8,2 1304548    26643 /lib/libc-2.2.5.so
sshd      11326   root  mem    REG                8,2   67520    26646 /lib/libdl-2.2.5.so
sshd      11326   root  mem    REG                8,2  101000    26649 /lib/libnss_compat-2.2.5.so
sshd      11326   root    0u   CHR                1,3            45098 /dev/null
sshd      11326   root    1u   CHR                1,3            45098 /dev/null
sshd      11326   root    2u   CHR                1,3            45098 /dev/null
sshd      11326   root    3r  FIFO                0,5          7339721 pipe
sshd      11326   root    4u  IPv4             7339526              TCP 10.16.21.65:ssh->62.2.22.178
sshd      11326   root    5w  FIFO                0,5          7339721 pipe
sshd      11326   root    6u   CHR                2,0            45319 /dev/ptyp0
sshd      11326   root    7u   CHR                2,0            45319 /dev/ptyp0
sshd      11326   root    8u  unix 0xffffff8002fddaea0          7339727 /tmp/ssh-HUVDV11326/agent.11
sshd      11326   root    9u   CHR                2,0            45319 /dev/ptyp0
klogd     11942   root  cwd    DIR                8,2    1024        2 /
klogd     11942   root  rtd    DIR                8,2    1024        2 /
klogd     11942   root  txt    REG                8,2   25376    12339 /sbin/klogd
klogd     11942   root  mem    REG                8,2  118262    26640 /lib/ld-2.2.5.so
klogd     11942   root  mem    REG                8,2 1304548    26643 /lib/libc-2.2.5.so
klogd     11942   root    0r   REG                0,2       0     4110 /proc/kmsg
klogd     11942   root    1u  unix 0xffffff8002fdd9c20           212 socket
bash      13466 intern  cwd    DIR                8,8    4096    32705 /home/romath
bash      13466 intern  rtd    DIR                8,2    1024        2 /
bash      13466 intern  txt    REG                8,2  631484     4134 /bin/bash
bash      13466 intern  mem    REG                8,2  118262    26640 /lib/ld-2.2.5.so
bash      13466 intern  mem    REG                8,2  365076    26685 /lib/libncurses.so.5.2
bash      13466 intern  mem    REG                8,2   67520    26646 /lib/libdl-2.2.5.so
bash      13466 intern  mem    REG                8,2 1304548    26643 /lib/libc-2.2.5.so
bash      13466 intern  mem    REG                8,2  101000    26649 /lib/libnss_compat-2.2.5.so
bash      13466 intern  mem    REG                8,2  136716    26648 /lib/libnsl-2.2.5.so
bash      13466 intern    0u   CHR                3,0            45320 /dev/ttyp0
bash      13466 intern    1u   CHR                3,0            45320 /dev/ttyp0
bash      13466 intern    2u   CHR                3,0            45320 /dev/ttyp0
bash      13466 intern  255u   CHR                3,0            45320 /dev/ttyp0
sshd      14076   root  cwd    DIR                8,2    1024        2 /
sshd      14076   root  rtd    DIR                8,2    1024        2 /
sshd      14076   root  txt    REG                8,4  342708     8243 /usr/sbin/sshd
sshd      14076   root  mem    REG                8,2  118262    26640 /lib/ld-2.2.5.so
sshd      14076   root  mem    REG                8,2  119188    26655 /lib/libresolv-2.2.5.so
sshd      14076   root  mem    REG                8,4  856492    38736 /usr/lib/libcrypto.so.0.9.6
```

```
sshd      14076   root  mem    REG                8,2    7896     26657 /lib/libutil-2.2.5.so
sshd      14076   root  mem    REG                8,4  121676     38734 /usr/lib/libz.so.1.1.4
sshd      14076   root  mem    REG                8,2  136716     26648 /lib/libnsl-2.2.5.so
sshd      14076   root  mem    REG                8,2   83804     26644 /lib/libcrypt-2.2.5.so
sshd      14076   root  mem    REG                8,2 1304548     26643 /lib/libc-2.2.5.so
sshd      14076   root  mem    REG                8,2   67520     26646 /lib/libdl-2.2.5.so
sshd      14076   root    0u   CHR                1,3               45098 /dev/null
sshd      14076   root    1u   CHR                1,3               45098 /dev/null
sshd      14076   root    2u   CHR                1,3               45098 /dev/null
sshd      14076   root    3u  IPv4             51174                 TCP *:ssh (LISTEN)
bash      17965   root  cwd    DIR                8,2    1024         2 /
bash      17965   root  rtd    DIR                8,2    1024         2 /
bash      17965   root  txt    REG                8,2  631484      4134 /bin/bash
bash      17965   root  mem    REG                8,2  118262     26640 /lib/ld-2.2.5.so
bash      17965   root  mem    REG                8,2  365076     26685 /lib/libncurses.so.5.2
bash      17965   root  mem    REG                8,2   67520     26646 /lib/libdl-2.2.5.so
bash      17965   root  mem    REG                8,2 1304548     26643 /lib/libc-2.2.5.so
bash      17965   root  mem    REG                8,2  101000     26649 /lib/libnss_compat-2.2.5.so
bash      17965   root  mem    REG                8,2  136716     26648 /lib/libnsl-2.2.5.so
bash      17965   root    0u   CHR                3,0               45320 /dev/ttyp0
bash      17965   root    1u   CHR                3,0               45320 /dev/ttyp0
bash      17965   root    2u   CHR                3,0               45320 /dev/ttyp0
bash      17965   root  255u   CHR                3,0               45320 /dev/ttyp0
kjournald 18006   root  cwd    DIR                8,2    1024         2 /
kjournald 18006   root  rtd    DIR                8,2    1024         2 /
lsof      18643   root  cwd    DIR                8,2    1024         2 /
lsof      18643   root  rtd    DIR                8,2    1024         2 /
lsof      18643   root  txt    REG                8,4  173368      8210 /usr/sbin/lsof
lsof      18643   root  mem    REG                8,2  118262     26640 /lib/ld-2.2.5.so
lsof      18643   root  mem    REG                8,2 1304548     26643 /lib/libc-2.2.5.so
lsof      18643   root    4r  FIFO                0,5           7354640 pipe
lsof      18643   root    7w  FIFO                0,5           7354641 pipe
xfssyncd  19129   root  cwd    DIR                8,2    1024         2 /
xfssyncd  19129   root  rtd    DIR                8,2    1024         2 /
syslogd   20642   root  cwd    DIR                8,2    1024         2 /
syslogd   20642   root  rtd    DIR                8,2    1024         2 /
syslogd   20642   root  txt    REG                8,2   31936     12349 /sbin/syslogd
syslogd   20642   root  mem    REG                8,2  118262     26640 /lib/ld-2.2.5.so
syslogd   20642   root  mem    REG                8,2 1304548     26643 /lib/libc-2.2.5.so
syslogd   20642   root  mem    REG                8,2   39472     26651 /lib/libnss_files-2.2.5.so
syslogd   20642   root    0u  unix 0xffffff8002fdd9780               163 /dev/log
syslogd   20642   root    2w   REG                8,5   25674     44027 /var/log/auth.log
syslogd   20642   root    3w   REG                8,5    7514     43928 /var/log/syslog
syslogd   20642   root    4w   REG                8,5     374     44003 /var/log/daemon.log
syslogd   20642   root    5w   REG                8,5   14148     44032 /var/log/kern.log
syslogd   20642   root    6w   REG                8,5       0     43985 /var/log/lpr.log
syslogd   20642   root    7w   REG                8,5    1528     44036 /var/log/mail.log
syslogd   20642   root    8w   REG                8,5       0     43987 /var/log/user.log
syslogd   20642   root    9w   REG                8,5       0     43988 /var/log/uucp.log
syslogd   20642   root   10w   REG                8,5    1528     44014 /var/log/mail.info
syslogd   20642   root   11w   REG                8,5       0     43990 /var/log/mail.warn
syslogd   20642   root   12w   REG                8,5       0     43991 /var/log/mail.err
syslogd   20642   root   13w   REG                8,5       0     43976 /var/log/news/news.crit
syslogd   20642   root   14w   REG                8,5       0     43977 /var/log/news/news.err
```

```
syslogd   20642   root   15w   REG              8,5        0     43978 /var/log/news/news.notice
syslogd   20642   root   16w   REG              8,5        0     44012 /var/log/debug
syslogd   20642   root   17w   REG              8,5    23110     44043 /var/log/messages
syslogd   20642   root   18u  FIFO              8,2              46331 /dev/xconsole
monit     24357   root   cwd   DIR              8,2     1024         2 /
monit     24357   root   rtd   DIR              8,2     1024         2 /
monit     24357   root   txt   REG              8,4   292740      4351 /usr/bin/monit
monit     24357   root   mem   REG              8,2   118262     26640 /lib/ld-2.2.5.so
monit     24357   root   mem   REG              8,2    94565     26659 /lib/libpthread-0.9.so
monit     24357   root   mem   REG              8,2    83804     26644 /lib/libcrypt-2.2.5.so
monit     24357   root   mem   REG              8,2   119188     26655 /lib/libresolv-2.2.5.so
monit     24357   root   mem   REG              8,2   136716     26648 /lib/libnsl-2.2.5.so
monit     24357   root   mem   REG              8,2  1304548     26643 /lib/libc-2.2.5.so
monit     24357   root   mem   REG              8,2   101000     26649 /lib/libnss_compat-2.2.5.so
monit     24357   root    0u   CHR              1,3              45098 /dev/null
monit     24357   root    1u   CHR              1,3              45098 /dev/null
monit     24357   root    2u   CHR              1,3              45098 /dev/null
```

Last Updated: April 21st, 2019

Upcoming Training

SANSFIRE 2019	**Washington, DC**	**Jun 15, 2019 - Jun 22, 2019**	**Live Event**
Community SANS New York SEC506	**New York, NY**	**Jul 15, 2019 - Jul 20, 2019**	**Community SANS**
SANS Network Security 2019	**Las Vegas, NV**	**Sep 09, 2019 - Sep 16, 2019**	**Live Event**
SANS London October 2019	**London, United Kingdom**	**Oct 14, 2019 - Oct 19, 2019**	**Live Event**
SANS OnDemand	**Online**	**Anytime**	**Self Paced**
SANS SelfStudy	**Books & MP3s Only**	**Anytime**	**Self Paced**