

Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permited without express written permission.

Linux DNS (Domain Name Server) System Setup Checklist

by Martin Tremblay

This document will guide you through the complete setup for a secure name server on a Linux Red Hat computer. This procedure will probably work with several Red Hat versions. However, it was written based on version 7.1. We are only going to install the necessary elements to run a name server securely. We are also assuming that this system is critical, so no downtime is acceptable. All suggested software are open source. I strongly recommend that you install (and of course use) each one of them.

I will not discuss how to setup BIND in this document. I am assuming that you are already comfortable with BIND configuration.

Table of Contents

- Before you start
- Split-Horizon DNS
- System and Network Information
- Physical Security
- BIOS Security
- Linux Installation
  - Kickstart Install
  - Set Logging
  - Login Banner
  - Secure the System
  - Reboot
  - Patches
  - Compile the Kernel
  - Reboot
- Compile, Install and Configure Open Source Software
  - Perl
  - NTP (Network Time Protocol)
  - zlib
  - openSSL (Secure Socket Layer)
  - openSSH (Secure Shell)
  - BIND (Berkeley Internet Name Daemon)
- Linux Firewall
- Create a Bootdisk
- Staying up to date
- Resources and References
- Appendix A: Checklist

© SANS Institute 2000 - 2005 Author retains full rights.

Before you start

I have written all scripts (BASH scripts) required to setup your system. I suggest that you read them carefully and make sure that you understand everything before running them. You can also perform operation manually since this documentation is explicitly referring to the scripts.

For those who are not really familiar whith bash scripting language, I suggest some very useful links in the Resources and References section. You will definitely need to know shell script language anyway if you want to administer UNIX systems.

The best way to use those scripts is to copy them (including `ks.cfg` at the root level) on a single floppy disk (DOS format is ok). The floppy does not need to be bootable. Make sure that all files are in UNIX format (no CR/LF).

Here is the command to mount the floppy drive on a RedHat Linux system: **mount /dev/fd0 -t vfat /mnt/floppy**. You can then access the floppy in **/mnt/floppy**.

Split-Horizon DNS

"Split-horizon (sometimes called split-brain) DNS is a DNS configuration where an organization presents one set of DNS information to external organizations and reserves a second, separate set of DNS information for internal use. This is generally done by maintaining two different collections of name servers: an "external" set which publishes the limited amount of DNS information the external organizations need to interact with your company, and an "internal" set which holds your complete, rich set of DNS information. Note that the separate DNS zone database are generally maintained on two different sets of physical machines [...]

Note that when your internal name servers wish to resolve external host names they must contact root name servers and name servers at other Internet-connected sites. This can open your internal name servers to attack from the outside. For this reason, many organizations that run split-horizon DNS also employ a sort of DNS proxying (slave forwarding name servers) to "hide" their internal name servers completely from the outside world."

Pomeranz, Hal and Deer Run Associates, Running UNIX Applications Securely, SANS Institute, 2001

Considering the fact that split-horizon DNS is a "best practice", which is exactly what you want to do, you now have to determine where to place the DNS server. "Internal" servers should be on your local corporate network. "External" servers can be directly connected to the Internet. I suggest that you place the DNS server in your DMZ, behind a firewall if possible. This will provide another security layer, which is never bad if you can handle the configuration easily. "Internal" or "external" DNS server setups should be exactly the same, except for the BIND configuration. I'll discuss that later.

System and Network Information

Here is what you need to know about the system and the network before beginning the setup. Most of this information has been used in the scripts. It will be easy for you to change it in the "Variables" section of each script, according to your system and network.

System

Processor:	Intel Pentium-II 450Mhz
Memory:	256MB RAM

© SANS Institute 2000 - 2005 Author retains full rights.

Hard drive(s):	2 x 9GB (Raid-1)

Network

Hostname:	ns1.mydnsserver.com
Domain:	mydnsserver.com
IP:	123.12.1.3
Netmask:	255.255.255.0
Gateway:	123.12.1.1
Your ISP DNS server:	ns.myispdnsserver.com (123.12.1.234) (we need to access a well known working DNS to download sources before becoming our own DNS...)
Remote logging server:	log.mydnsserver.com
NTP servers:	ntp1.mydnsserver.com, ntp2.mydnsserver.com, ntp3.mydnsserver.com

Physical Security

You should keep in mind that anyone who has access to your server's console of your server can damage it. Any hacker who has access to your console can easily and rapidely have a root shell on it. You can trust me on that one. Therefore it is very important to limit access to your server to as few people as possible. You absolutely need to have a locked room if you pretend to have a high-avalability service. You can do a lot more here: UPS, redundant link, water-proof room, guardians, etc. The list is long and could be costly.

I will not want go any deeper on the physical security issue. Just keep in mind that someone who has a physical access to your server can also have a root shell on it. I'll let you imagine the rest...

Here is a short checklist that give a certain level of security. I've found it in Solaris Security, Version 2.0 from Hal Pomeranz (step 4.2).

Physical Security Checklist

1. ___ Place the server in a locked room with access controled by the administrator. Verify that drop-down ceilings and raised floors do not allow uncontrolled access.
2. ___ (Advanced) Porvide electronic access control and recording for the server room.
3. ___ Provide temperature and humidity controls sufficient to avoid damage to the equipment. One uninterruptible power supply (UPS) vendor provides an optional attachment that monitors temperature and humidity and can send administrative alerts and emails and can page the system administrator.
4. ___ (Advanced) Provide one or more halon-type automatic fire extinguishers.
5. ___ Install a UPS and associated software that enables the server to shut down automatically and safely when the power in the UPS is about to be exhausted.
6. ___ (Advanced) Use survaillance cameras to record who accesses the equipment.
7. ___ Lock the CPU case and set up a system to ensure the key is protected and yet easily available to the administrator. Make a back-up key and protect it off-site in a secure disaster recovery site or a safety deposit box or similarly protected place. Lock the server down with a cable or in a rack.
8. ___ Arrange the room so that the keyboard is hidden from prying eyes at windows or other vantage points.
9. ___ (Advanced) Consider providing additional shielding against electronic interference or eavesdropping.

© SANS Institute 2000 - 2005 Author retains full rights.

You should protect your computer BIOS for two main reasons. The most important one is that you don't want to allow anyone to modify BIOS settings. This could allow a hacker who has physical access to your server to reboot from the floppy or the CD-ROM, log as root, and then gain access to all your file system. You definitely need to set a password on your BIOS. Also, make sure that your system always boots from the hard drive (C:). Never allow it to boot from a removable media (floppy, CD-ROM, ZIP).

The second reason is that some "AUTO" BIOS settings can be incorrectly recognized by Linux. Those "AUTO" detections include hard disks (especially IDE drive), PnP IRQ settings, etc. This can result in a corrupted file system. You need to disable "AUTO" settings. This will make your system boot faster, which is always a good thing.

Note that some SCSI BIOS setups do not allow you to password protect it. If a hacker can enter the SCSI BIOS setup, the only way to make sure he will not boot from the SCSI CD-ROM is a locked CD-ROM cage.

Make sure to document all your BIOS settings since it is really easy to forget. Place it in a secure place accessible to other administrators. It is also a good idea to document the hard drive geometry and non-default system settings in the same document.

Using the Red Hat Kickstart installation method, you are going to install Red Hat Linux with minimal packages to safely run a name server. You then need to secure the system. To do so, you also need to re-compile a monolythic kernel and apply the latest patches.

Your system should not be connected to the network at this time. You need to restrict access before exposing your server to hackers. This is done by setting restricted access via tcp_wrappers when securing the system.

Kickstart Install

The Red Hat Kickstart installation method is useful if you want to install your system quickly. With this method, you can create a single file (`ks.cfg`) containing the answers to all questions that would normally be asked during a typical Red Hat Linux installation. The `ks.cfg` file should be on a floppy drive.

(`ks.cfg` : Section 1: Networking)
Make sure to change network parameters at the beginning of the file according to your settings.

(`ks.cfg` : Section 5: Partitions)
You also have to look at the "Partition" section. I know that partitioning is a religious debate among system administrators. I suggest the following four partitions: `/` (root), `swap`, `/usr` and `/var`. The point is that there are two partitions you must have: `/` (root) and `swap`. Let's say that the `/` (root) partition can be all "unused space" of your drive after you've decided of other partitions. For the `swap` partition, it is usually safe to allocate twice the amount of physical memory then what is usually allocated. With respect to performance issues, I recommend to create separate partitions for the OS files, `/usr`, and `/var` for the logging. For the Linux files partition, 1GB is enough but I prefer to make it 2GB just to make sure. For the logs partitions it depends on how many histories you will keep on your system, but with 1GB you should be OK for a while. I intentionaly will not create a `/home` partition simply because our system should only have a small number of users: root and some DNS administrators.

Note that if you want to make Raid-1 mirroring, you will also have to create a `/boot` partition. 64Mb will be more than

© SANS Institute 2000 - 2005 Author retains full rights.

you need.

Here is my suggestion for a name server system such as the the one defined earlier:

`/boot` (if Raid-1)	64MB
`/` (root)	6.5GB (or the rest of the drive)
`swap`	512MB
`/usr`	2048MB
`/var`	1024MB

(`ks.cfg` : Section 6: Packages)
The sample `ks.cfg` file will install RedHat "Base" packages. The package list included in the "Base" package can be found on the first RedHat CD-ROM in `/dev/cdrom/RedHat/base/comps`. I also add some essential packages (kernel-headers, bind-utils, iptables, iputils, pidentd, portmap, gmp, make, binutils, tcp_wrappers, gcc, cpp, glibc-devel, ncurses, ncurses-devel, byacc) and useful ones (ftp, traceroute, wget, indexhtml, lynx).

Booting from the Red Hat CD-ROM, you should type the following special command to launch the kickstart setup:
***boot:* linux ks=floppy**

Set Logging

(`improve-logging` : Section 2: Variables)
The improve-logging script will help you set logging facilities properly. All operations are commented. Make sure to read the "Variables" section of the script. It contains some parameters that you may like to change.

(`improve-logging` : Section 3: Configuring syslogd)
The script will generate a `/etc/syslog.conf` file better than Red Hat's default one. Mainly, it will add some kernel level logging and also add information to the `/var/log/syslog`.

(`improve-logging` : Section 4: Configuring logrotate)
You should also think about rotating your log files on a daily basis. You will then be able to get those archived log files and store them into a secure media (tape, CD-ROM, etc). You should also consider to check your logs regulary. To do so, I recommend Psionic Logcheck. It is easy to use and configure and you can monitor your logs "on the fly".

I strongly recommend that you use a remote log server. It will be a lot harder for a hacker to hide his actions.

Login Banner

You should notify all users that logged onto your system that you are monitoring them. It will be easier to prosecute them after an attack. Click here for an example of banner here. Just put your organization name at the proper location and copy the content in `/etc/motd` and **chmod 644 /etc/motd**.

Here is an example of banner: (largely inspired from the Computer Incident Advisory Center)

```
**************************************************************
*
*                        NOTICE TO USERS
*
*  This computer is the property of YOUR_ORGANIZATION_HERE. It is for
```

© SANS Institute 2000 - 2005 Author retains full rights.

```
*   authorized use only. Users (authorized or unauthorized) have no explicit
*   or implicit expectation of privacy.
*
*   Any or all uses of this system and all files on this system may be
*   intercepted, monitored, recorded, copied, audited, inspected, and disclosed
*   to authorized site, and law enforcement personnel, as well as authorized
*   officials of other agencies, both domestic and foreign. By using this
*   system, the user consents to such interception, monitoring, recording,
*   copying, auditing, inspection, and disclosure at the discretion of
*   authorized site or YOUR_ORGANIZATION_HERE personnel.
*
*   Unauthorized or improper use of this system may result in administrative
*   disciplinary action and civil and criminal penalties. By continuing to use
*   this system you indicate your awareness of and consent to these terms and
*   conditions of use. LOG OFF IMMEDIATELY if you do not agree to the conditions
*   stated in this warning.
*
**********************************************************************
```

Secure the System

The improve-security script will help you secure your system. All operations are commented. With tcp_wrappers you are going to reject every TCP/IP connection for TCP services compiled with the libwrap libraries. At the end of the script you will be asked to reboot. Don't forget to connect the sytem to the network before rebooting. You are going to download sources, so a valid network connection and a valid DNS are required.

(`improve-security` : Section 2: Delete useless users)
First, you should delete useless users on your system. With the **userdel** command, remove the following users: lp mail news uucp operator games gopher ftp rpc rpcuser.

(`improve-security` : Section 3: Disable useless services)
You should then disable unrequired services. From the **/etc/rc.d/rc3.d/** directory, disable the following services: kudzu ipchains portmap nfslock netfs apmd atd sendmail gpm anacron xinetd. Note that the xinetd service is now disabled. It is a good idea to disable it as a lot of security holes come from services running under xinetd. If you absolutly need to run it, please consider to run it under tcp_wrappers, which is described below.

(`improve-security` : Section 4: Tighten up setting in /etc/inittab)
In the `/etc/inittab` file, comment out the *ca::ctrlaltdel:/sbin/shutdown -t3 -r now* line to disable the Ctrl+Alt+Del reboot signal. You also may want to force a password before entering in single user mode. You can do so by adding **~~:S:wait:/sbin/sulogin** just the below the *si::sysinit:/etc/rc.d/rc.sysinit* line.

(`improve-security` : Section 5: Disable TELNET remote root logins)
You should only allow root login from the console. To do that, in the `/etc/securetty` file, ensure that every line only includes "TTYx" consoles. If remote users require root privileges on the system, they can simply login to their own account and use the **su -** command to become root.

(`improve-security` : Section 6: Restrict TCP connections)
You should now restrict TCP connexion access on your system. Create a `/etc/hosts.deny` file containing only this line: **ALL: ALL**. `/etc/hosts.deny` will reject every connections for TCP services compiled with the libwrap libraries, typically only those started by xinetd and the sshd daemons. You now need to provide yourself with remote access to your DNS server to administer it. You are going to install openSSH later, but tcp_wrapers will drop every SSH connexion. To be able to connect to the system, you should edit the `/etc/hosts.allow` file. You will have to add a line that could read like that: **sshd: 12.234.21.96/27**. This will allow the 12.234.21.96/27 subnet to establish tcp connexions with the sshd service. You could look at the hosts.allow manpages for more information on the syntax. Both files must be owned by "root" and must be "600"

© SANS Institute 2000 - 2005 Author retains full rights.

(`improve-security` : Section 7: Check for special files)
At this time, it is a good idea to note every SUID/SGID file on your system. This kind of files is a potential security risk for your system and should be monitored closely. If a hacker can copy a SUID shell executable, this will quickly and easily give him a root shell on your system. It is a good idea to frequently run **find / -type f \(-perm -04000 -o -perm -02000 \)** as root to get the complete list and compare it with previous results.

World-writable files, particulary system files, can also be a security hole. To locate world-writable files on your system, use **find / -perm -2 ! -type l -ls** and be sure to know why those files are writable. That last part of the command (! -type l) will not list files from `/dev` and symbolic links.

Reboot

It is now time to reboot your system. I will also have to connect it to the network to be able to download kernel sources.

Patches

At the time you install your Linux OS, many security patches are already released. It is a good habit to patch your system weekly. Updating a Red Hat Linux system is quite easy. You just have to downlaod the latest RPM packages from ftp://updates.redhat.com/7.1/386/ then "freshen" the packages that are already installed on your system with the **rpm -F [PatchesDir]/*** command.

You could update your system in many other ways. Two of them are the Red Hat Update Agent and AutoRPM. They are really easy to use.

Here is the way I do it:

```
[root]# mkdir /usr/local/updates
[root]# cd /usr/local/updates
[root]# wget -N ftp://updates.redhat.com/7.1/en/os/i386/*
[root]# rm -f kernel*
[root]# rpm -F *
```

Compile the Kernel

If you have never compiled the Linux kernel before, you should read the Kernel HOWTO. You must at least know that you should never delete your backup kernel that works until you are sure to have a working replacement. I personally never delete the original kernel, just in case...

The compile-kernel script will help you compile the kernel. All operations are commented. You should build a monolythic kernel, whitout any modules, to improve performance (there is no consensus on that one). It can also help from a security point of view. Imagine that a hacker accesses your system and gains access to your file system. He only has to copy his "rootshell" module and it will be automatically loaded into the kernel. This is easy for the hacker and hard to detect.

(`compile-kernel` : Section 3: Get source code)
(`compile-kernel` : Section 4: Compile sources)
Here is the list of options that you should change from the default options. I refer to the `/usr/local/linux/.config` file that is created with the **make menuconfig** command. Don't worry, it will be easy to find where to change variables from the menu. Variable names are really significative.

© SANS Institute 2000 - 2005 Author retains full rights.

Code maturity level options	CONFIG_EXPERIMENTAL=y
Loadable module support	CONFIG_MODULES is not set
Multi-device support (RAID and LVM)	CONFIG_MD=y CONFIG_BLK_DEV_MD=y CONFIG_MD_RAID1=y
Networking options	CONFIG_NETFILTER=y
IP: Netfilter Configuration	CONFIG_IP_NF_CONNTRACK=y CONFIG_IP_NF_IPTABLES=y CONFIG_IP_NF_MATCH_LIMIT=y CONFIG_IP_NF_MATCH_MULTIPORT=y CONFIG_IP_NF_MATCH_TCPMSS=y CONFIG_IP_NF_MATCH_STATE=y CONFIG_IP_NF_FILTER=y CONFIG_IP_NF_TARGET_REJECT=y CONFIG_IP_NF_TARGET_MIRROR=y CONFIG_IP_NF_NAT=y CONFIG_IP_NF_NAT_NEEDED=y CONFIG_IP_NF_MANGLE=y CONFIG_IP_NF_TARGET_LOG=y
Network device support	CONFIG_NETDEVICES=y
Ethernet (10 or 100Mbit)	CONFIG_EEPRO100=y (replace by your NIC here...)
File systems	CONFIG_MSDOS_FS=y CONFIG_VFAT_FS=y

(`compile-kernel` : Section 3: Get source code)
(`compile-kernel` : Section 4: Compile sources)
At the time this procedure was written, Kernel 2.4.7 (linux-2.4.7.tar.gz) was the latest stable kernel.

```
[root]# wget http://www.kernel.org/pub/linux/kernel/v2.4/linux-2.4.7.tar.gz
[root]# tar xzf linux-2.4.7.tar.gz
[root]# cd linux
[root]# make mrproper
[root]# make menuconfig
[root]# make dep
[root]# make bzImage
[root]# make install
```

(`compile-kernel` : Section 5: Updating LILO)
Don't forget to review the `/etc/lilo.conf` file. If you are not doing Raid-1, change /dev/md1 for /dev/hda1 in the `image=/boot/vmlinuz` section. If you correct it, don't forget to rebuild LILO (**/sbin/lilo**). You should also validate some links in **/boot**, `/boot/System.map` should point to `/boot/System.map-[NewKernelVersion]`, and `/boot/vmlinuz` should point to `/boot/vmlinuz-[NewKernelVersion]`.

© SANS Institute 2000 - 2005 Author retains full rights.

Reboot

It is now time to reboot your system.

A lot of the following tools can be installed as Red Hat RPM packages. It will however be easier to update them later if you compile them. I also prefer to work this way because I am working with more than one UNIX plateform and it is easier to administer.

Perl

Running install-perl, you will be able to download sources, compile and install Perl. You definitly need Perl to compile openSSL but also for scripting purposes.

(`install-perl` : Section 3: Get source code)
(`install-perl` : Section 4: Compile sources)
At the time this procedure was written, Perl 5.6.1 (stable.tar.gz) was the latest stable version.

```
[root]# wget http://www.perl.com/CPAN/src/stable.tar.gz
[root]# tar xzf stable.tar.gz
[root]# cd perl-5.6.1
[root]# ./configure -de -Uinstallusrbinperl
[root]# make
[root]# make instal
```

NTP

(Network Time Protocol)

With NTP, it is really easy to keep system clocks from many computers in sync. The protocol developped by David Mills of the University of Delaware is the Internet standard time synchronization protocol. It is robust and safe, works well with firewalls and requires little network bandwith.

"From a security perspective, effective prosecution of security incidents requires accurate "matching" timestamps on all log files. Any discrepancies will complicate or sabotage legal proceedings."
Pomeranz, Hal and Deer Run Associates, Running UNIX Applications Securely, SANS Institute, 2001

In my opinion, it is a good idea to run your organization NTP servers on the same computers as your name servers. The reason is that those servers are (supposed to be) secure and they should be available 24/7. Since running an NTP daemon is not resource consuming, I can't figure out why you would like to use another system to accomplish this task.

Using install-ntp, you can download sources, compile, install and configure an NTP client. Make sure to review the "Variables" section of the script, especially for your NTP servers.

(`install-ntp` : Section 3: Get source code)
(`install-ntp` : Section 4: Compile sources)
At the time this procedure was written, ntp-4.1.0.tar.gz was the latest version.

© SANS Institute 2000 - 2005 Author retains full rights.

```
[root]# wget http://www.eecis.udel.edu/~ntp/ntp_spool/ntp4/ntp-4.4.1.0.tar.gz
[root]# tar xzf ntp-4.1.0.tar.gz
[root]# cd ntp-4.1.0
[root]# ./configure –prefix=/usr/local
[root]# make
[root]# make instal
```

(install-ntp : Section 5: Configure NTP client)
Here is an example of an NTP client configuration file (`/etc/ntp.conf`):

```
driftfile /etc/ntp.drift
ntp1.mydnsserver.com
ntp2.mydnsserver.com
ntp3.mydnsserver.com
restrict default nomodify
```

Here is an example of NTP client synchronization cron job script (`/etc/cron.hourly/set-ntp`):

```
#!/bin/sh
CONFFILE=/etc/ntp.conf
SERVERS=`awk "/^server|peer/ {print \$2}" ${CONFFILE} | grep -v "^127"`
/usr/local/bin/ntpdate -sbu -t 3 ${SERVERS}
/sbin/hwclock –systohc
```

zlib

Running install-zlib, you will be able to download sources, compile and install zlib. This is a data compression software library required by openSSL.

(install-zlib : Section 3: Get source code)
(install-zlib : Section 4: Compile sources)
At the time this procedure was written, zlib-1.1.3.tar.gz was the latest version.

```
[root]# wget http://www.info-zip.org/pub/infozip/zlib/zlib-1.1.3.tar.gz
[root]# tar xzf zlib-1.1.3.tar.gz
[root]# cd zlib-1.1.3
[root]# ./configure –prefix=/usr/local
[root]# make
[root]# make install
```

openSSL

(Secure Socket Layer)

Running install-openSSL, you will be able to download sources, compile and install openSSL. This is an encryption library required by openSSH.

(install-openSSL : Section 3: Get source code)
(install-openSSL : Section 4: Compile sources)
At the time this procedure was written, openssl-0.9.6b.tar.gz was the latest version.

© SANS Institute 2000 - 2005 Author retains full rights.

```
[root]# wget http://www.openssl.org/source/openssl-0.9.6b.tar.gz
[root]# tar xzf openssl-0.9.6b.tar.gz
[root]# cd openssl-0.9.6b
[root]# ./config –prefix=/usr/local
[root]# make
[root]# make install
```

openSSH

(Secure Shell)

Running install-openSSH, you will be able to download sources, compile, install and configure openSSH. This is a secure shell that you will need in order to access the system remotely.

The configuring portion of the scripts will create an `sshd_config` file. Note that this script will not allow you to log as root with SSH. You will need to log with a normal user, then **su** to root. The script will also create a `/etc/rc.d/init.d/sshd` file to stop and start the SSH daemon.

(`install-openSSH` : Section 3: Get source code)
(`install-openSSH` : Section 4: Compile sources)
At the time this procedure was written, openssh-2.9p2.tar.gz was the latest version.

```
[root]# wget ftp://ftp.openbsd.org/pub/OpenBSD/OpenSSH/portable/openssh-2.9p2.tar.gz
[root]# tar xzf openssh-2.9p2.tar.gz
[root]# cd openssh-2.9p2
[root]# ./configure –prefix=/usr/local –with-ssl-dir=/usr/local –with-libwrap –with-tcp-wrappers –without-rsh
–disable-suid-ssh –disable-scp-stats –with-md5-passwords
[root]# make
[root]# make install
```

(`install-openSSH` : Section 5: Configure sshd daemon)
Here is an example of a secure sshd configuration file (`/usr/local/etc/sshd_config`):

```
Port 22
Protocol 2
ListenAddress 123.12.1.3
#
HostKey /usr/local/etc/ssh_host_key
HostKey /usr/local/etc/ssh_host_rsa_key
HostKey /usr/local/etc/ssh_host_dsa_key
#
ServerKeyBits 768
LoginGraceTime 600
KeyRegenerationInterval 3600
PermitRootLogin no
#
IgnoreRhosts yes
StrictModes yes
X11Forwarding no
X11DisplayOffset 10
PrintMotd yes
```

© SANS Institute 2000 - 2005 Author retains full rights.

```
PrintLastLog yes
KeepAlive yes
#
SyslogFacility AUTH
LogLevel INFO
#
RhostsAuthentication no
RhostsRSAAuthentication no
HostbasedAuthentication no
RSAAuthentication yes
#
PasswordAuthentication yes
PermitEmptyPasswords no
#
Subsystem sftp /usr/local/libexec/sftp-server
#
#End od configuration file
```

From the above configuration file, you should especially note a certain number of things. First, if you can, use only the SSH2 protocol (Protocol 2), which is more secure than the SSH1 protocol (you should know that the SSH3 protocol will be available soon in openSSH 3). Furthermore, you should never allow the root user to login (PermitRootLogin no) directly via SSH. Please make sure to review every configuration option in the openSSH man pages to make sure that it is really what you need.

Replace "r" programs with SSH:

```
[root]# ln -s /usr/local/bin/ssh /usr/bin/rsh
[root]# ln -s /usr/local/bin/slogin /usr/bin/rlogin
[root]# ln -s /usr/local/bin/scp /usr/bin/rcp
```

BIND

(Berkeley Internet Name Daemon)

(`install-BIND` : Section 2: Variables)
Running install-BIND, you will be able to download sources, compile, install and configure BIND. This is the most used DNS software on the Web. Make sure to change the "Variables" section of the script according to the domain you want to administer.

(`install-BIND` : Section 3: Get source code)
(`install-BIND` : Section 4: Compile sources)
At the time this procedure was written, bind-9.1.3.tar.gz was the latest release version.

```
[root]# wget ftp://ftp.isc.org/isc/bind9/bind-9.1.3.tar.gz
[root]# tar xzf bind-9.1.3.tar.gz
[root]# cd bind-9.1.3
[root]# ./configure –prefix=/usr/local
[root]# make
[root]# make install
```

(`install-BIND` : Section 5: Configure named daemon)
We are going to make a quick and easy setup of only one domain. The named service will run as named:named

© SANS Institute 2000 - 2005 Author retains full rights.

(user:group).

We need to configure BIND to operate in a "chroot jail", meaning that it cannot see or access files outside its own little directory tree.

"The idea behind chroot is fairly simple. When you run BIND (or any other process) in a chroot jail, the process is simply unable to see any part of the filesystem outside of the jail. For example, in this document, we'll set BIND up to run chrooted to the directory /chroot/named. Well, for BIND, the contents of this directory will appear to be /, the root directory. Nothing outside of this directory will be accessible to it. You've probably encountered a chroot jail before if you've ever used ftp to log into a public system."
Scott Wunch, Chroot-BIND HOWTO, 2001

Here is the directory structure of the "chrooted" environment, also known as the jailtree:

```
/var/named/
        dev/
        etc/
            named/
                slave/
        var/
            run/
```

Because it is possible to obtain the BIND version of a DNS server (**dig @ns1.mydnsserver.com version.bind txt chaos**), it is important to change the version that the name server show to the world. The script will place "DNS Server" in the `named.conf` file. You could however change it in the "Variables" section of the script. It is important to hide your BIND version to hackers. It will makes their work harder...

(`install-BIND` : Section 6: Create DNS zone files)
You need to create "zone" files to be able to run a name server. I recommend to setup BIND with the simplest configuration as possible at first. After you know that the name server is well installed, you can begin to "play" with the configuration.You will also have to download the `named.root` file from ftp://ftp.rs.internic.net/domain/named.root. This file holds the information on root name servers needed to initialize the cache of Internet domain name servers. In the `named.conf` file, this zone is referred to as the "." zone.

Linux Firewall

IPtables

You have to install a firewall on your DNS server. The IPtables package is already installed on your system. What you need is a bunch of rules to allow or deny trafic.

Just have to copy this rules file in **/etc/sysconfig/**. This file was written by James C. Stephens. Don't forget to change the IPADDR variable at the beginning of the script. Denied packets will be output to the console by default. When everything is fine, just comment out log entries at the end of the `/etc/sysconfig/iptables` script.

You also have to modify `/etc/rc.d/init.d/iptables`. Within the start() function, in the *Applying iptables firewall rules:* section, change the *grep [...] $IPTABLES_CONFIG | grep [...] | /sbin/iptables-restore -c &&* \ line with **$IPTABLES_CONFIG && **.

Create a Bootdisk

Don't forget to create a bootdisk of your healthy system. It could save you a lot of time later. Make sure the boot diskette

© SANS Institute 2000 - 2005 Author retains full rights.

is properly labeled and stored in a secure place. Since your kernel does not have any module at all, you will have to create the **/lib/modules/[KernelVersion]/** directory in order to create the bootdisk with the **mkbootdisk [KernelVersion]** command.

Staying up to date

"There is a lot of mailing lists that distribute information about computer security, newly discovered vulnerabilities and exploits, bug fixes, and updates. Even with an automated process for finding updated RPMs, it is important to stay informed about security-related problems for which updated software may not be available yet. At least you will know to turn a vulnerable service off until a fix is available."
Brotznam, Lee E., Allied Technology Group, Inc. and David A. Ranch, Trinity Designs, Securing Linux Steb by step, SANS Institute, 2001

On your DNS server, only two services are exposed to the rest of the world: openSSH and BIND. It is a good idea to pay a particular attention to those services and make sure to always use the latest recommended version for both of them.

See the Mailing Lists section in the Resources and References section below to get some useful mailing lists to subscribe to.

Resources and References

Links

Shell scripting	http://www.linuxgazette.com/issue57/okopnik.html http://www.linuxdoc.org/LDP/abs/html/
Split-Horizon DNS	http://people.oven.com/bet/lwd/lwd.html#split%20horizon http://hostmaster.rutgers.edu/inside-outside.html
Red Hat Kickstart	http://www.redhat.com/support/manuals/RHL-7.1-Manual/customization-guide/ch-kickstart2.html
Logging	http://www.linuxsecurity.com/feature_stories/feature_story-64.html http://coombs.anu.edu.au/~avalon/nsyslog.html
Banner	http://www.ciac.org/ciac/bulletins/j-043.shtml
Securing Linux System	http://www.ibiblio.org/pub/Linux/docs/HOWTO/Security-HOWTO http://www.securityportal.com/lasg/ http://www.sans.org/infosecFAQ/linux/linux_list.htm http://www.linuxdoc.org/HOWTO/Security-HOWTO-5.html
Linux Kernels	http://www.kernel.org http://www.kernelnotes.org http://www.linuxdoc.org/HOWTO/Kernel-HOWTO.html
Red Hat patches	http://www.redhat.com/apps/support/updates.html ftp://updates.redhat.com/ http://www.kaybee.org/~kirk/html/linux.html

© SANS Institute 2000 - 2005 Author retains full rights.

Perl	http://www.perl.com/CPAN/
NTP	http://www.ntp.org/
zlib	http://www.gzip.org/zlib/
openSSL	http://www.openssl.org/
openSSH	http://www.openssh.com/ http://www.openbsd.org/cgi-bin/man.cgi?query=sshd http://www-106.ibm.com/developerworks/library/l-keyc?t=gr,p=RSA-DSA http://www-106.ibm.com/developerworks/linux/library/l-keyc2/
BIND	http://www.securityportal.com/cover/coverstory20001113.html http://www.cymru.com/~robt/Docs/Articles/secure-bind-template.html http://www.linuxdoc.org/HOWTO/Chroot-BIND-HOWTO-2.html http://www.netsecurity.pl/www.boran.com/security/sp/bind_hardening.html
Linux Firewalls	http://www.linux-firewall-tools.com/linux/ http://www.sns.ias.edu/~jns/security/iptables/index.html http://netfilter.samba.org/
Mailing Lists	http://www.linuxsecurity.com/general/mailinglists.html http://www.redhat.com/mailing-lists/ http://listserv.securityportal.com/SCRIPTS/WA-SECURITYPORTAL.EXE?SUBED1=securityportal-l&A=1

Books

Brotznam, Lee E., Allied Technology Group, Inc. and David A. Ranch, Trinity Designs, Securing Linux Steb by step, SANS Institute, 2001
Pomeranz, Hal and Deer Run Associates, Running UNIX Applications Securely, SANS Institute, 2001
Cameron Newham and Bill Rosenblatt, Learning the Bash Shell, 2nd Edition, O'Reilly and Associates, 1998
Brotznam, Lee E. and Hal Pomeranz, Linux/Solaris Practicum, SANS Institute, 2001
Ablitz and Liu, DNS and BIND, 3rd Edition, O'Reilly and Associates, 1998
Pomeranz, Hal and Deer Run Associates, Solaris Security Step by step, Version 2.0, SANS Institute, 2001

Red Hat Linux DNS server checklist

Pre-installation steps

© SANS Institute 2000 - 2005 Author retains full rights.

1. ___ Determine DNS architecture
2. ___ Regroup system information (hostname, IP, netmask, gateway, DNS, ...)
3. ___ Determine the physical location for the system
4. ___ Review BIOS settings

Linux Installation

1. ___ Kickstart install
2. ___ Modify logging
3. ___ Put your own banner
4. ___ Secure Linux
5. ___ Reboot
6. ___ Patch system
7. ___ Compile the kernel
8. ___ Reboot

Compile, Install and Configure Open Source Software

1. ___ Compile and install Perl
2. ___ Compile, install and configure NTP
3. ___ Compile and install zlib
4. ___ Compile and install openSSL
5. ___ Compile, install and configure openSSH
6. ___ Compile, install and configure BIND

Linux Firewall

1. ___ Create IPtables rules script

Create a bootdisk

1. ___ Create a bootdisk

Comments

_____
_____
_____
_____
_____
_____

Printed Name:_____

Signature:_____ Date:_____

© SANS Institute 2000 - 2005 Author retains full rights.