Global Information Assurance Certification Paper

Securing a Corporate Mail Relay using Sendmail and RedHat 8.0
GCUX Practical v1.9

Chris Garringer

Summary

This paper covers the installation of a secure Sendmail relay. It is assumed the relay will be behind a firewall in a DMZ, however firewall configuration is not covered. This paper covers the Sendmail host only. The first main section covers the installation of a minimal Redhat 8.0 server with Sendmail. The second section covers post-install configuration including TCP Wrappers, iptables, and Tripwire. The final section covers ongoing maintenance and backup.

Table of Contents

Functional Requirements:
The company uses a commercial email system for internal Email. The SMTP interface provided by the email system is not fully compatible with other Internet mail systems. As a result a Sendmail system is required as a relay between the Internet and the internal system. This gives the highest level of compatibility between the company's internal Email system and Internet mail. The system will act only as a relay, there will be no local mail delivery. Mail not destined for, or sent from the company's domains will be refused.

Environment:
The company firewall is configured with a DMZ subnet where publicly accessible servers are installed. The Sendmail system will be installed there. The firewall will be configured to allow only SMTP (port 25) traffic to the sendmail box from the Internet. Secure Shell (port 22) traffic will be allowed from the internal network to the Sendmail system. The sendmail system will be able to send smtp traffic to the internal mail gateway only. This is the only direct port access through the firewall to the sendmail machine. Other ports on the firewall will be open, but only to the other servers on the DMZ subnet. The firewall is configured as a deny-all system. Only specifically authorized ports and destination hosts are available from the Internet. While direct connection over these ports cannot be made from the Internet, consideration must be given to the possibility that other DMZ servers may be compromised, allowing a full range of attacks on the Sendmail box. Other than the preceding statements the firewall configuration will not be covered in this document.

The company maintains a network monitoring system that checks the system's status (via ping) and the SMTP status (via attempted connections). This system will notify administrators of system reboots or SMTP failures independently of the Sendmail server itself.

The Sendmail system is required to relay only mail to/from the company's domain or domains. The sendmail system will also be used for spam filtering, virus-checking, and redirection of mail for specific users. Sendmail configuration will be discussed only for security purposes, general Sendmail configuration will not be discussed. No local mail delivery will be done on the system. All mail generated by the system will be sent to users on the internal mail system. System access will be limited to the IS administrators responsible for the mail system. The Sendmail server will be located in the company computer room, which has restricted access. Only IS personnel or people escorted by IS personnel are allowed in the computer room.

System Description:
The system hardware will be a dual 700Mhz processor system with 512MB of RAM and a 40GB hard drive. A CD drive and a floppy drive are also installed. This system is much more powerful than is

needed for mail relaying at this time, it is intended to allow virus-checking and spam protection in the near future and allow for company growth.

The system software is RedHat 8.0, with all current patches, running Sendmail 8.12, also the latest patch version. The RedHat provided Sendmail distribution will be used to allow use of Redhat's update mechanism per company standards. This reduces the risk of updates breaking the system as RedHat has already tested the update rpm's we will use on their system. Additional software to install and configure is:

iptables – this will provide a host-based firewall for a second level of protection after the corporate firewall. This is part of the RedHat install.

Tripwire – This program monitors the file system for changes to files. It is part of the RedHat install.

Logsentry – This program was available through psionic.com. After this paper was written, and before it was published, Cisco bought out Psionic. I have made several attempts to find out if Logsentry will continue without success. It is still included here because this is how the system is configured.
This is a program to monitor log files for security problems. Redhat's program for this is logwatch. Logwatch requires Perl programming knowledge for configuration, whereas logsentry's configuration is in text files. Both programs do the same job but I prefer logsentry for its ease of configuration. Logwatch is installed by default and will be removed after installation.

Openssh – This will be used for remote logins to the sendmail system. All of the xinetd services will be disabled. Openssh is part of the RedHat install.

Software used for configuration testing:

Nessus – vulnerability testing software see http://www.nessus.org for documentation and download.

Nmap – port scanner – see http://www.insecure.org/nmap for documentation.
Also included in the Redhat distribution

CIS – host-based security evaluation – see http://www.cisecurity.org for documentation and download.

:

Risk Analysis

Threats to the linux server include: remote compromise, specially crafted Email, denial-of-service, privilege escalation, local compromise and relaying unauthorized mail.
It cannot be assumed that all compromise attempts will be limited to the STMP or SSH ports. While the firewall restricts direct connections to the SSH (from internal systems) and SMTP (from external and internal systems) one of the other DMZ servers could be compromised and used to launch attacks. Also the firewall itself could be compromised, allowing attacks on any port. To mitigate this risk, iptables will be configured on the Sendmail system to restrict connections to the SSH and SMTP ports only. Iptables will also allow ICMP replies for all users. Also, a remote attack depends on having the server listening to a port that has a vulnerability.

The possibility of remote compromises can be mitigated by restricting the number of services

running on the machine. This server will be a dedicated sendmail relay, with limited access via SSH for administration. There is no reason for any other service to run.

Specially crafted Email: As seen in CERT Advisory CA-2003-02 (http://www.cert.org/advisories/CA-2003-07.html)
Email can be a vehicle for system compromise, even when the server is not running the program mailer. This is a risk which cannot be avoided, the entire purpose of the server is Email. The best that can be done is to keep the Sendmail version current.

There are several sendmail exploits that depend on the version of Sendmail running. These are most easily blocked by running the current version of Sendmail. Since the standard sendmail greeting message displays the version number, the greeting message should be changed to limit the amount of information given to possible attackers.

Denial of service: The server can be overloaded by a large number of emails, a large number of connections being opened then left unused, or attempting to overload the log files with multiple bad messages.

Privilege escalation: If an attacker can compromise a local account then they can attempt a number of attacks to get root access or disable the machine. This risk is limited by severely restricting the number of user accounts, since there is no reason to access the system except for administration, and removing many of the accounts that the RedHat install automatically creates. File system privileges will be tightened in accordance with Sendmail recommendations (Sendmail 3rd edition, Section 10.5).

Local compromise: If a intruder can get physical access to a system there are a number of ways to get root access. The easiest on RedHat Linux is to reboot the server and edit grub.conf to boot into single user mode. The risk of this is minimal as the system will be in a restricted access area. Also, the boot loader will be setup to require a password before you can edit the boot sequence. The system could be booted to a CD or floppy as the system has both. The systems position in an access restricted room is considered sufficient defense against this possibility. Also, these options require the system to be rebooted. A reboot would be noticed by the company's network monitoring system independent of any logs on the Sendmail system.

Spam relay: Since this is a mail relay, care must be taken to relay only mail to or from the company's domains. An additional level of protection is provided by the firewall as it will proxy mail through only if it is for the company's domains.

System Installation

Prerequisites:
Before beginning installation the following information is required:

System name
IP address

Netmask
Default router
Primary Name Server
Secondary Name Server
A floppy disk to create a boot disk.
A floppy disk to save installation files.
A CD with the current RedHat 8.0 RPM updates

For this discussion we will use the following:

System name	bernese
IP address	10.100.100.100
Netmask	255.255.255.0
Default router	10.100.100.80
Primary Name Server	10.100.50.50
Secondary Name Server	10.100.42.42

During installation the system will be connected to a hub with no other systems connected. The OS running during installation is not secure, so physical isolation is required.

The only RedHat package group installed will be the Sendmail package, all other packages will be selected from the Individual package list. The file system will have /boot,/,/usr,/var, /home and swap. This allows /usr to be mounted read-only and /var and /home to be mounted nosuid. These settings are per the recommendations in the SANS Securing Unix course, 6.5.

The current rpm updates for RedHat 8.0 should be obtained from either ftp.redat.com or a mirror site. Mirrors are listed at www.RedHat.com/mirrors. All updates from the 8.0 i386, i686 and noarch directories should be obtained. Note that there may be more than one kernel revision available in the i386 and i686 directories. Select only the most recent. The kernel rpms are the largest ones and downloading multiple versions will greatly extend the download time. Write the rpms to a CD for installation on the new system. This avoids the necessity of connecting the new system to another machine before security configuration is done.

Installation Procedure:
General notes: User input will be in italics. The TEXT installation method will be used. Many screens in the text install have a list of items to select and a set of boxes giving options below. Use the up/down arrow keys to highlite the selection in the list of items, and the TAB key to move from the list to each of the option boxes.

Boot to CD number 1 to start the installation. At the RedHat Installation screen you have multiple installation options. Since this machine will be a server and not running a GUI interface, enter *text* and press Return.

The first screen is a welcome screen. Select OK to continue.

The next 3 screens allow selection of the system's default language, keyboard and mouse. Make sure the language selection and keyboard selection are consistent. The mouse selection is NOT

meaningless, even though this machine will not run a GUI. Several programs will allow use of a mouse without a GUI. If you have a 2 button mouse be sure that the Emulate 3 Button option is selected. This should be done for you when a 2 button mouse is selected.

After the mouse selection screen is the Type of System to install. Select Custom here, in order to have more control over the packages installed. The less software installed and running the fewer vulnerabilities are possible and fewer updates are required.

The next screen is Disk Partitioning. There are several choices here, chose *Autopartition*. Redhat's autopartition sets up /boot, / and swap. The / partition is set to Fill available disk space making editing of it unnecessary. After selecting Autopartition you can easily customize the file system settings.

If the system hard drive had existing partitions you will see a screen with options on what to do with them. This is to be a dedicated server and a fresh install so select Remove all partitions. You will be asked to confirm this selection. If the original drive is not partitioned you will not see this screen.

Autopartitioning will setup /boot, swap and / for the Custom install. Swap will be set to physical memory size and is correct. /boot is set to 100MB, which is small. This is where the kernel images are kept for booting, and I prefer to keep the current image and 2 older images so set /boot to 200MB. The / partition is set to "Use all remaining space", so the other partitions we need do not require editing the / partition. We will create /var, /home and /usr partitions.

The /var partition is for logs and the sendmail queues so this partition will be 5GB to allow for logging and be large enough to resist denial of service attacks aimed at filling up the queues.

The /usr partition is for system binaries. It is a separate partition to allow mounting the system binaries as a read-only partition. This cannot be done if the binaries are part of the / partition. It will be set to 3GB.

The /home partition is for user directories. There will be only administrative users on the system so this does not need to be large. It will be set to 2GB.

On the partition screen, highlight the boot partition. Then *TAB* to the EDIT box and hit enter. On the details screen that appears:

Field Name	set to
Mount point	/boot (no change)
File System type	ext3 (no change)
Size	*200* (changed from 100)
Fixed Size	
Fill Maximum size of (MB)	Fixed Size (no change)

Leave the Force to be a primary partition and Check for Bad Blocks unchecked. Select OK

On the partition screen, *TAB* to NEW and hit enter. On the details screen that appears:

Field name	set to
Mount point	*/usr*

File System Type */ext3*
Size *3000MB*
Fixed Size
Fill Maximum size of (MB)
Fill all available space *Fixed Size*
Leave the Force to be a primary partition and Check for Bad Blocks unchecked. Select OK.

Create the next partition the same way. The settings for it are:

Field Name	set to
Mount Point	*/var*
File System Type	*/ext3*
Size	*5000MB*
Fixed Size	
Fill Maximum size of (MB)	
Fill all available space	*Fixed Size*

Leave the Force to be a primary partition and Check for Bad Blocks unchecked. Select OK.

The last partition created is /home with the following settings:

Field Name	set to
Mount Point	*/home*
File System Type	*/ext3*
Size	*2000MB*
Fixed Size	
Fill Maximum size of (MB)	
Fill all available space	*Fixed Size*

Leave the Force to be a primary partition and Check for Bad Blocks unchecked. Select OK.

At the partition screen check the partitions to make sure they are correct. When they are correct select OK to continue.

The next 5 screens are the boot loader selection and configuration. Select the grub boot loader, lilo is depreciated.

The 2nd screen asks for any special parameters. You do not enter anything here, select OK to continue.

The 3rd screen asks if you want to configure a boot loader password. Although this server is in a secure area, a boot loader password should still be configured. The boot loader password does not affect normal reboots, but stops someone from editing the boot parameters and getting root by booting into single user mode. Always set a boot loader password.

The next screen allows configuring Grub to boot other operating systems. This is a dedicated box so there is nothing to do here, go to the next screen.

The last boot loader configuration is the location. The default of putting the boot loader in the Master Boot Record is correct.

The next section is network configuration. The first screen sets up the network card. This server has a fixed address, so uncheck the Use bootp/dhcp box. Leave the "activate on boot" box checked. After unchecking the bootp/dhcp box you can enter the IP parameters. This screen attempts to be helpful and fill in parameters for you after you give the IP address. I have never found this to be useful, make sure you enter your own numbers. The fields that must be set are:

IP Address	*10.100.100.100*
Netmask	*255.255.255.0*
Default Gateway	*10.100.100.80*
Primary Nameserver	*10.100.50.50*
Secondary Nameserver	*10.100.42.42*

The next screen sets the hostname. This should be unique in your domain. Enter *bernese* for this system.

Next is the firewall configuration. The RedHat default is to setup ipchains during install. Iptables is the current ip filtering program, ipchains is an older program. RedHat supports both, but defaults to the old program. Iptables will be setup manually after the installation and update is done. In this screen, select *NONE*.

Language Support and Time zone selection are the next 2 screens. Add any languages you may need other that the default installed language. Be sure and set the timezone correctly for your area.

The next screen sets the root password. Set this according to your company standards. I recommend 8+ characters, no dictionary words, a mixture of case, and numbers and symbols. The entry screen will not allow passwords less than 6 characters and you cannot leave the screen until the 2 entry fields match.

The next screen allows adding a non-root user. Always add a user at this screen. After setup on this machine is complete root will not be allowed to login directly except at the console. An administrative user is required to login over the network.

After the initial non-root user is created you will be allowed to create additional users. This is a limited access box, create accounts only for the people tasked with controlling the sendmail system.

Authentication configuration is next. We are using MD5 and shadow passwords. NIS , LDAP, and Kerberos are not used and will not be discussed in this manual. If you are using one of the other authentication systems, see your setup documentation for the correct settings. The default is MD5, shadow passwords and nothing else. Select OK.

Next is the software selection section. The package groups include a lot of unnecessary software, so

most of this is packages to delete. Software selection fine-tuning is done next.

Packages groups to de-select
Gnome Desktop Environment
Graphical Internet
Graphics
Office/Productivity
Printing Support
Sound & Video
Text Based Internet
X Windows System

Package Groups to Add
Mail Server

Check the option to Select Individual Packages. At this point the only check marks should be next to Mail Server and Select Individual Packages. Select OK.

The next screen lists the individual packages grouped by purpose. Listed below are the groups and the changes to make in those groups.

Amusements/Games – Nothing selected

Amusements/Graphics – Nothing selected

Applications/Archiving – No Changes

Applications/CPAN – Nothing selected.

Applications/Communications
Clear lrzsz and minicom. This is modem support and there will be no modem on this machine.

Applications/Databases – Nothing selected

Applications/Editors – No changes

Applications/Emulators – Nothing selected

Applications/Engineering – Nothing selected

Applications/File
Select stat – This is a tool to get file information.

Applications/Internet
Select ethereal – this monitors network traffic

Applications/Multimedia – Nothing selected.

Applications/Office – Nothing selected.

Applications/Productivity – Nothing selected.

Applications/Publishing – Nothing selected.

Applications/System
Select arpwatch for monitoring arp
Select autorun for administrator convenience. The machine will be located in a secure area so automounting floppies and CD's is an acceptable risk.
Select tripwire – this is necessary for monitoring system changes
Select psacct to install system accounting.
Select procinfo

De-select idsn4k-utils
De-select setserial
De-select stat-serial

Applications/Text – No changes

Applications/Utilities – nothing selected.

Development/Debuggers
Select ltrace – for tracing program execution
Select strace – for tracing program execution
Select sysreport – for documenting system configuration

Development/Languages – Nothing selected.

Development/Libraries
Select libpcap – required for ethereal

Development/System – Nothing selected.

Development/Tools – Nothing selected.

Documentation
Select bash-doc
Select sendmail-doc

Networking/Mail – Nothing selected.

System Environment /Base
De-select lilo

System Environment/Daemons
De-select imap – no local mailboxes
De-select wvdial
De-select xinetd

System Environment/Kernel – No changes

System Environment/Libraries – No changes

System Environment/Shells – No changes

User Interface/Desktops – No changes

User Interface/X – No changes

User Interface/X Hardware Support – Nothing selected.

Select OK. The installation program will review the packages selected and look for missing dependencies. If any packages selected require other packages for support they will be listed. If any are found, select "Install packages to satisfy dependencies". This is the default, select OK. The installation of the OS will begin, you will be prompted to install CD's 2 and 3 as needed.

After all software is installed the system does a Post Installation Configuration and asks if you want to create a boot disk. A boot disk is an important recovery tool so go ahead and create it.

Select OK to reboot and remove the CD. The initial installation is done.

Post-Installation configuration

NOTE: All post-installation steps are done as root. The first step after installation is complete is to bring the system up to the current level on all installed software. Load the CD with the current RedHat 8.0 updates and mount the CD, *mount /mnt/cdrom.* You must be root to load rpms. Change directories to the CD directory with the rpms. The command to update software is *rpm –Fv *rpm>>/root/<filename>.* This command will go through all the rpm's in the current directory and update any installed rpm. It will not install new rpms. The v is verbose mode and will show the installation process. The output is redirected to a file so we have a record of what was installed. The output is appended in case you have the rpm's in multiple directories. If you created multiple directories on the update CD this command must be run in all directories.

Once the system is updated we need to clean up the users that Redhat automatically installs that we do not need. Redhat installs the following users : games, uucp, lp (no printing on this machine), ftp , news, gopher ???rpc???. These users should be deleted, even though they are not allowed to login by default. Deleting them makes the password file smaller so it is easier to spot changes. If they were left in, an attacker could change the shell and this is easier to overlook than a new user entry.

To delete these users run: *for i in games uucp lp ftp news gopher rpc;do /usr/sbin/userdel $i;done.*

There are several systems started by default that are not used. Although there may be no known security problems with some of them, only necessary services should be running. To turn off a service use the chkconfig command as follows:

```
/sbin/chkconfig --level  2345 nfslock off
/sbin/chkconfig --level 2345 pcmcia off
/sbin/chkconfig --level 345 autofs off
/sbin/chkconfig --level 345 portmap off
/sbin/chkconfig --level 345 netfs off
```

These commands turn off: nfs locking since NFS will not be used on this machine. There is no PCMCIA equipment on the server, so the subsystem is not needed. Autofs automounts NIS maps for NFS, neither of which is used. There is no need for portmap as sendmail does not use it. Netfs mounts network file systems and is not used in this system.

Iptables configuration

Installation - Iptables was part of the initial Redhat installation. The initialization script looks for the existence of the iptables rules file, /etc/sysconfig/iptables, and if it exists starts iptables with the specified configuration.

Since this server is a sendmail relay for incoming and outgoing Email, the iptable configuration should allow incoming sendmail connections from and to anywhere. System administration will be done using ssh from internal machines only, specifically the MIS subnets. Iptable rules will setup to allow ssh connections from the 2 MIS subnets only. ICMP pings will be allowed from anywhere. No other ports will be allowed to make connections to this server. This is an additional level of defense for the server in case the firewall is compromised. An additional level of protection for ssh will be provided by the tcp wrappers configuration below.

The rules to implement the iptable configuration are as follows:

```
/sbin/iptables -A INPUT  -m state  --state  ESTABLISHED,RELATED –j     ACCEPT
/sbin/iptables -A INPUT  -p icmp --icmp-type 8 -j ACCEPT
/sbin/iptables -A INPUT  -p tcp --dport 25 -j ACCEPT
/sbin/iptables -A INPUT  -p tcp --dport 22 -s 10.100.200.0/24 -j ACCEPT
/sbin/iptables -A INPUT  -p tcp --dport 22 -s 10.100.201.0/24  -j ACCEPT
/sbin/iptables -A INPUT  -p udp --sport 53 -s 10.100.50.50 -j ACCEPT
/sbin/iptables –A INPUT –p udp –sport 53 –s 10.100.42.42 –j ACCEPT
/sbin/iptables -A INPUT  -j LOG
/sbin/iptables -P INPUT DROP
```

Line 1 allows established and related connections to the server.
Line 2 allows other hosts to ping the sendmail server.
Line 3 allows SMTP connections to the server.
Line 4 and 5 allow ssh connections from the 2 MIS subnets.
Lines 6 and 7 allow the DNS name servers for the sendmail box to supply DNS resolution. If you have more than one DNS name server, add a line for each one.

Line 8 logs any connection attempts that are not specifically allowed.
Line 9 sets the default policy for the INPUT table to DROP. All connections not specifically allowed will be dropped. The logsentry program will be configured to consider any lines logged as security violations.
To keep the configuration across reboots, you must run iptables-save. Run the command as follows:

/sbin/iptables-save > /etc/sysconfig/iptables.

When the system boots up, the iptables file will be read and the configuration applied.

TCP Wrappers Configuration

TCP wrappers is a program that uses two files, /etc/hosts.allow and /etc/hosts.deny to allow or deny access to server daemons. There is also a tcp wrappers library to allow compiling tcp wrapper support directly into programs. The tcp wrapper program is used for daemons run from xinetd. Xinetd is not running on this server, but sendmail and openssh have tcp wrappers support compiled into the standard RedHat release of the programs.

Configuration - Edit /etc/hosts.allow and add the following lines:

```
sshd:          10.100.200.0/255.255.255.0,10.100.201.0/255.255.255.0
sshd-X11forward        10.100.200.0/255.255.255.0,10.100.201.0/255.255.255.0
sendmail:  ALL
```

The sshd line allows hosts in the 2 subnets 10.100.200.0/255.255.255.0 and 10.100.201.0/255.255.255.0 to connect to the ssh daemon. The sshd-X11forward line allows the 2 subnets to use X11 forwarding.
The sendmail line allows any host to connect to the sendmail port. Note that TCP Wrappers, as of the time this was written does not allow the classless netmask notation, you must list out the netmask as shown above.

The /etc/hosts.deny file will have one line as follows:

```
ALL:ALL
```

This is a default deny configuration, so anything not specifically allowed in the /etc/hosts.allow file will be rejected.

Syslog configuration

The /etc/syslog.conf file controls the system logging. There are three changes to make to this file. By default , RedHat puts auth messages to /var/log/messages and authpriv messages to /var/log/secure. These two are associated and I prefer them both in /var/log/secure. SSH logs are by default put into /var/log/messsages. It is much easier to see who is using ssh if the messages in a separate log. The local6 facility will be created to handle ssh logs. **NOTE**: In syslog ALL whitspace MUST be TABS. Do not use the space bar, your log configuration will not work as you intend unless you use TABS.

Syslog will not create a file, so the file for ssh logs must be created and set to allow root access only. Create the file using the following commands:

touch /var/log/ssh.log
chown root:root /var/log/ssh.log
chmod 700 /var/log/ssh.log

The following lines need to be edited in /etc/syslog.conf:

Add *auth.none* to the line for /var/log/messages. This will keep auth messages out of the /var/log/messages file. The original line is

*.info;mail.none;authpriv.none;cron.none /var/log/messages

The new line will be

.info;mail.none;authpriv.none;cron.none;auth.none* /var/log/messages

Add auth.* to the /var/log/secure line. All auth messages will now be in the /var/log/secure file. The original line is

authpriv.* /var/log/secure

The new line will be

authpriv.* *;auth.** /var/log/secure

To capture sshd messages to /var/log/ssh.log add the following line to the syslog.conf file.

*Local6.** */var/log/ssh.log*

To implement the changes you need to have syslog re-read its configuration file. Use the following command:

kill -HUP `cat /var/run/syslogd.pid`

SSH configuration

The default for RedHat is to allow root to login over ssh. This poor security as you cannot track who is coming into your system. For security auditing purposes it is better to have users attach as themselves and su to root. This will log the ssh connection in the ssh.log file and the su in /var/log/messages. To implement this the openssh configuration file, /etc/ssh/sshd_config, must be edited. Default settings for openssh are listed in the file, but commented out. This provides an easy method to check what settings are default and which are modified. When we change the defaults we will remove the comment (#) and change the setting.

Original Line	New Line
#Protocol 2,1	Protocol 2
SyslogFacility AuthPriv	SyslogFacility local6
#PermitRootLogin yes	PermitRootLogin no
#Banner /some/path	Banner /etc/issue.net

The first line deletes the version 1 protocol support. Version 1 is not as secure as version 2 so we change the system to allow version 2 only.

The second line changes the logging destination from authpriv (/var/log/secure) to local6 (/var/log/ssh.log). The configuration for local6 was done previously in the syslog.conf configuration

section.

The third line disallows root login over ssh. To get root a user must login and su to root. This allows logging of who is executing programs as root.

The fourth line sets the banner for ssh login. Edit the file /etc/issue.net and have it contain the following line only:

Authorized uses only. All activity may be monitored. (reference text)

To restart the ssh daemon with the new settings:

kill -HUP `cat /var/run/sshd.pid`

The process ID for the ssh daemon is written to /var/run/sshd.pid on daemon startup.

Logsentry installation and configuration

Logsentry is a program to monitor your logfiles. The program uses 4 configuration files, logcheck.hacking, logcheck.ignore, logcheck.violations, and logcheck.violations.ignore. Each file is a list of patterns to match against lines in the logfiles. The logfiles to check are listed in the logcheck.sh file, located in /usr/local/etc. The logtail program keeps the current position in each logfile, so only entries since the last run of logsentry are checked. Logsentry uses egrep to match patterns against the logfiles. The report is emailed to SYSADMIN (as defined in logcheck.sh). Logcheck.hacking is a list of patterns that match against possible system attacks. Logcheck.violation is a list of patterns that match possible security violations. Logcheck.violation.ignore filters logcheck.violation matches to reduce false positives. Logcheck.ignore is a list of patterns that are definitely not a problem. If a pattern is in logcheck.violation.ignore, you should have the same pattern in logcheck.ignore. Anything not matched in one of the logcheck files is added to the report as 'Unusual System Events'. This is easily configurable for your personal preferences.

RedHat 8.0 comes with a logwatch program to do the same thing. Logwatch is Perl-based and all configuration is done by modifiying multiple perl programs. This is a very difficult program to configure, and when mail is sent to the admins, the timestamp is removed. To remove logwatch run

rpm -e logwatch

Download and extract logsentry on another RedHat 8.0 system with the Redhat development environment. After extraction the logcheck-1.1.1 directory will be created in the directory you downloaded logsentry into. CD to the logcheck-1.1.1 directory and run make linux, as root. The logtail program will be installed in /usr/local/bin and the logcheck.sh, logcheck.ignore, logcheck.violations, logcheck.hacking, and logcheck.violations.ignore files are in /usr/local/etc. Copy all these files to floppy or CD and install the files in the same directories on the sendmail server. The files should be owner root, group root with mode 600, except for logtail and logcheck.sh which should be mode 700. The logcheck system uses a temporary directory during execution, which is defaulted to /usr/local/etc/tmp. This must be changed as /usr will be set to read-only at the end of the post-install configuration. Create the directory /var/local/etc/tmp using the following commands:

mkdir –p /var/local/etc/tmp

chown root:root /var/local/etc/tmp

chmod 700 /var/local/etc/tmp

Edit the logcheck.hacking file and add the lines
Dropped invalid comments from header address
kernel: IN=

The first line looks for the log entry made by sendmail when an Email attempting to exploit the sendmail vulnerability discovered recently (see CERT Vulnerability Note VU #398025). Note that this is not a guaranteed compromise attempt as some proprietary Email systems also put extraneous information in headers that may trigger this. But it should be examined immediately if this entry is seen in your mail logs. The second line looks for any packets that are not allowed by the iptables filter. If you get packets you did not expect you should immediately check the source system or the firewall.

Edit the logcheck.violations file and edit the su root line to be
*su.*session opened for user*
Add the line
su(pam_unix)
RedHat linux does not put su root in the logfile, it inserts su(pam_unix)... These lines will cause logcheck to send all su attempts, successful and not, to the network administrators. Since this system should be stable and accessed only by administrators this should not happen often.

Edit /usr/local/etc/logcheck.sh and change the SYSADMIN=root line to SYSADMIN=*youremailaddress*. Change the line starting TMPDIR from /usr/local/etc/tmp to /var/local/etc/tmp. The ssh.log file needs to be added to the logcheck.sh files. The files logcheck.sh will check for Linux systems are defined after the line # *Linux Red Hat Version 3.x, 4.x*. After the line $LOGTAIL /var/log/maillog >> $TMPDIR/check.$$ add the line:
$LOGTAIL /var/log/ssh.log >> $TMPDIR/check.$$

To keep from being inundated with ….. did not issue MAIL/EXPN… errors add the following line to both /usr/local/etc/logcheck.violations.ignore and /usr/local/etc/logcheck.ignore:
*sendmail.*did not issue MAIL*

The logcheck.sh program will be run every 3 hours. To configure this, run *crontab –e* and add the line: *0 0,3,6,9,12,15,18,21 * * * /usr/local/etc/logcheck.sh* .

Logrotate configuration
By default, RedHat 8.0 rotates logs weekly, and keeps 4 copies. We will do daily rotations and keep a month (30 days) of logs. Edit /etc/logrotate.conf and change:
weekly to *daily*
rotate 4 to *rotate 30*
Save the file.
Ssh.log must be added to the log rotation. It is controlled by syslog so edit /etc/logrotate.d/syslog and add /var/log/ssh.log to the line containing /var/log/messages. Make sure and maintain the space-separated format. It is easiest to add /var/log/ssh.log immediately before /var/log/messages.

Tripwire installation and configuration

Tripwire monitors changes to files and directories on a system. The policy file and database must be setup on a system you know to be uncompromised (immediately after installation). For further documentation on tripwire see http://www.tripwire.org. This is the website for the open-source version of tripwire, which is included in the Redhat installation. To install tripwire, *cd /etc/tripwire* and run *./twinstall.sh.* You will be asked for pass phrases to protect two keys, the site key and the local key. The site key is used to sign the configuration and policy files. The local key is used to sign the database. Follow the prompts to create the two keys. After you have entered the local pass phrase the install program will sign the initial configuration and policy files. The configuration file does not need to be changed, but the default policy file assumes a full installation of Redhat, and is very specific on the binary files to watch. We will configure a smaller policy file that watches the binary directories and not specific files.

The policy file configuration is changed by editing /etc/tripwire/twpol.txt. Rules in the twpol.txt file have two sections, the configuration section is between parentheses and the files list is between curly brackets. When you are instructed to delete a rule make sure you have deleted both sections and all delimiters. Edit twpol.txt and make the following changes:

Delete rule “File System and Disk Administration”
Delete rule “Kernel Administration Programs”
For rule “Networking Programs”
    Delete all lines whose path starts with /bin or /sbin
    Delete the line with /etc/sysconfig/network-scripts/ifdown-cipcb
    Delete the line with /etc/sysconfig/network-scripts/ifup-cipcb
Delete rule “System Administration Programs”
Delete rule “Hardware and Device control programs”
Delete rule “System Information Programs”
Delete rule “Application Information Programs”
Delete rule “Shell Related Programs”
Delete rule “Operating System Utilities”
Delete rule “Critical Utility Sym-Links”
For rule “User binaries
    Remove (recurse =1) from all lines in this rule.
Delete rule “Shell Binaries”
Delete rule ”Critical system boot files”
For rule “Critical Configuration files”
    Add the line
      /etc/mail -> $(SEC_BIN);
    Delete the line with /etc/httpd/conf
    Delete the line with /etc/named.conf
    Delete the line with /usr/sbin/fixrmtab
    Delete the line with /etc/samba/smb.conf
    Delete the line with /etc/xinetd.conf
Delete the line with ($TWLKEY)/$HOSTNAME)-local.key
From rulename “System boot change

There are multiple lines to delete here, all of them start with /var/lock/subsys. The filenames to delete are listed, separated by commas: apmd, arpwatch, autofs, bcm5820, bgpd, bootparamd, canna, cWnn, dhcpd, firewall, freeWnn, gated, httpd, identd, innd, ipchain, iptables, ipvsadm, irda, iscsi, isdn, junkbuster, kadmin, kprop, krb524, krb5kdc, kWnn, ldap, linuxconf, lpd, mars-nwe, mcserv, mysqld, named, netfs, nfs, nfslock, nscd, ntpd, ospf6d, osfpd, pcmcia, portmap, postgresql, pxe, radvd, rarpd, reconfig, rhnsd, ripd, ripngd, routed, rstatd, rusersd, rwalld, rwhod, smb, snmpd, squid, tux, tWnn, ups, vncserver, wine, xfs, xinetd, ypbind, yppasswdd, ypserv, ypxfrd, zebra. Also delete the line starting /var/log. The log files will change daily so monitoring them does not return any useful information.

For rule “Root config files”

Delete the line with /root/.esd_auth
Delete the line with /root/.gnome_private
Delete the line with /root/.gnome-desktop
Delete the line with /root/.gnome
Delete the line with /root/.ICEauthority
Delete the line with /root/.Xauthority.

The last two rules contain files that will change during system operation. The System boot change rule lists files that will change on each system boot. This is another check for when a system is rebooted. The Root config files rule lists some files that will change whenever root logs in or out. This is a check on whether root has been in the system.

After the twpol.txt file is updated, be sure you are in the /etc/tripwire directory, and run */usr/sbin/twadmin --create-polfile twpol.txt* to create the policy file. The site key phrase will be required. If you have made an error editing the twpol.txt file the twadmin program will exit and display the line number of the error. Note that the twadmin program exits on the FIRST error, so you may have to do this more than once. Once the policy file is created, run */usr/sbin/tripwire –init* to build the tripwire database.

The Redhat default is to run tripwire once per day at 4:20am. This machine is more critical to the company so we need to set tripwire to run more often. Run */usr/bin/crontab –e* and add the following line:

*30 0,8,12,16,20 * * * /usr/sbin/tripwire --check*

This sets tripwire to run every 4 hours, including the default time. The default Redhat installation emails tripwire reports to root. Add root to the /etc/mail/aliases file with an alias of your network administrator or administrator group.

Sendmail configuration

The sendmail configuration program was not installed on this machine. Sendmail configuration files must be generated on another Redhat 8.0 machine with the sendmail-cf rpm installed. The following lines are to be added to the *.mc file you use for the system. The only items covered are security related. The base sendmail configuration is up to the user. To have a known base configuration, the sendmail.mc and submit.mc files supplied by the default Redhat sendmail installation will be used. Copy the /etc/mail/sendmail.mc and /etc/mail/submit.mc files from bernese to the system you

have sendmail-cf on. Rename the files bernese.mc and bernesesubmit.mc. Edit bernese.mc as follows:

Remove the line DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl. This line blocks the system from listening to external machines on port 25, since this is a mail relay the line must be deleted.

Remove the line FEATURE (`accept_unresolvable_domains')dnl. This line allows sendmail to accept mail from domains that DNS cannot resolve. This is a bad idea, unresolvable domains are almost always spam and almost certainly indicates mail with bad intentions. For cases this might be desirable see Sendmail, 3rd Edition section 4.8.2.

Add the line *define (`confLOCAL_SHELL_PATH',`P=/bin/false')dnl* . The line must come before the MAILER lines. I put definitions at the top of the file after the standard Redhat definitions. This disables the program mailer. An attack vector for sendmail is to send mail to the program mailer to execute programs on the local machine. This machine is a relay and there is no reason for the program mailer to run. It is defined as part of the local mailer, so it cannot be removed but the shell can be changed to an invalid shell. For further information see Sendmail, 3rd Edition section 20.4.7

Modify the Privacy Flags line to be *define (`confPRIVACY_FLAGS',`needmailhelo, noexpn,novrfy,noverb,authwarnings, restrictmailq,restrictqrun')dnl*. This line requires the sending site to identify itself, does not allow a remote site to query for user names or mailing lists, disables verbose mode, adds a header to mail that may be forged, and restricts access to the mailq directories. This setting allows DSN (Delivery service notification) messages. If you do not want delivery service notifications sent to the sender the line becomes much simpler, as follows: *define (`confPRIVACY_FLAGS', `goaway,restrictmailq, restrictqrun')dnl*. See Sendmail, 3rd Edition sections 10.8.2.7 and section 24.9.80 for further explanations.

By default, sendmail will tell any system connecting to port 25 the version number of the running sendmail daemon. This is not information that should be easily available. To change this, add the following line to the bernese.mc file:

define(`confSMTP_LOGIN_MSG',`$j is ready')dnl

This displays the line <systemname> ESMTP is ready. This does not give out a lot of information and complies with the RFC's. Refer to Sendmail, 3rd Edition section 24.9.105.

Compile the .mc file to a .cf file using the instructions found in the README file in /usr/share/sendmail-cf on the machine you have setup for sendmail configuration. Transfer the .cf file to bernese in the /etc/mail directory and name it sendmail.cf.

There are 2 files to edit to restrict mail relaying. The only domains you want to relay are the domains you own. If mail is not to or from your domains it must be rejected. By default the Redhat configuration does not allow relaying. To allow relaying of your domains edit the /etc/mail/access file. This file controls relaying for the sendmail system. For each of your domains add a line as follows:

Mydomain.xxx RELAY

Save the file and run the following command to build the access database for sendmail:

/usr/bin/makemap hash access<access

The command must be run as root and the current directory must be /etc/mail.

As this machine is a relay into your network, the mail destination for your domain(s) is known. Assuming the name of your internal mail server is intmail.mydomain.domain, edit the file /etc/mail/mailertable and add the following line:

Mydomain.domain *smtp:[intmail.mydomain.domain]*

This directs all mail to mydomain.domain to the internal mail server. The brackets around the internal mail server name stop sendmail from doing a MX lookup on the name. This is to avoid mailing loops and is documented at http://www.sendmail.org/m4/mailertables.html. Save the file and run the following command, as root with current directory /etc/mail:

/usr/bin/makemap hash mailertable<mailertable

Edit the /etc/mail/access file and add your domain(s) to the file in the following form:

Yourdomain.domain *RELAY*
Yourdomain2.domain2 *RELAY*

Save the file and run the following command, as root with current directory /etc/mail:

/usr/bin/makemap hash access<access

This will allow mail to/from the listed domains to be relayed, mail not to or from your domains will be rejected.

Edit the /etc/mail/aliases file to add an alias for root, postmaster, and all administrative users to send their local mail to your main mail system. This will eliminate users having to log into the system to get mail. The format for the alias file is:

Local_name: *alias*

Restart sendmail with the following command: */sbin/service sendmail restart.*

Restricting crontab and at jobs

The only user that should be setting up jobs for automatic execution is root. To enforce this restriction create two files, /etc/cron.allow and /etc/at.allow. Both files will have one line as follows:

r*oot*

This will not stop cron or at jobs from being run by other users, but only root can run the crontab commands.

Configuring the file system

Once all the configuration and installation is completed, the file system needs to be configured. The /usr filesystem should be set to read-only, as no changes should be made after this point. The /var and /home filesystems should not have programs on them, especially not suid programs so they should be set nosuid. To implement these changes edit the /etc/fstab file and change the following lines:

```
LABEL=/usr          /usr          ext3    defaults        1 2
LABEL=/usr/local    /home         ext3    defaults        1 2
LABEL=/var          /var          ext3    defaults        1 2
/dev/fd0            /mnt/floppy   auto    noauto,owner,kudzu 0 0
```

Change to (changed item is in italics on each line)

```
LABEL=/usr          /usr            ext3   ro          1 2
LABEL=/home         /usr/local      ext3   nosuid      1 2
LABEL=/var          /var            ext3   nosuid      1 2
/dev/fd0          /mnt/floppy       auto   nosuid,noauto,owner,kudzu 0 0
```

Save the file and reboot to implement your changes. When you need to install updates, you must change the /usr filesystem back to read-write. To do this execute the following command, as root: */bin/mount /usr –o remount,rw*. Once the updates are done, set the filesystem back to read-only with the following command, also executed by root: */bin/mount /usr –o remount,ro.*

After the system is connected to the network, login as root and copy 2 files to a floppy disk, /root/install.log and /root/anaconda-ks.cfg. The install.log file is a log of the installation. The anaconda-ks.cfg file is a kickstart configuration file to recreate the machine. The anaconda-ks.cfg file is important if you have to rebuild the machine, you can setup a kickstart installation and you will automatically get a duplicate of the installation you just performed. Keep the floppy in a safe place.

Post-Configuration Testing

Security will be checked with three programs, the Center for Internet Security's (CIS) Linux benchmark, nmap, the port scanner, and nessus, the network security scanner. The CIS benchmark is a host test, nmap and nessus are both network tests.

The Center for Internet Security (http://www.cisecurity.org) has designed several benchmark tests for operating system security. We will run the Linux benchmark against our system configuration to see how we look against the CIS standards. The benchmark can be downloaded from the CIS WEB site and is installed using a rpm package. Go to http://www.cisecurity.org and follow the instructions to download Linux 1 (as of 4/2003). Copy the file to bernese and extract the programs. If you have re-booted the machine after post-installation you will need to make the /usr filesystem read-write again (see the Configuring the File System section). Install the scoring tool with the command *rpm –i CISscan-1.2.0-1.3.i386.rpm*. After the program is installed, run */usr/local/CIS/cis-scan.* The program will scan your system looking for what it considers good and bad security settings. The output is a numerical score and a report detailing what it found. The numerical score and the path to the report are displayed on the screen after the program runs. After following the installation instructions you should get a score of 7.5 out of 10. This will change over time as CIS adjusts its benchmark. Check the report for Negative comments to see what CIS considers a problem. In this case several negative comments came because system defined users such as smmsp and sshd had a "valid shell" of nologin. This is not a problem as the nologin "shell" will not allow the user to login, it is seen as a valid shell because it is in the /etc/shells file. The CIS benchmark also takes off for no /etc/ftpusers file. This is not a problem for bernese as there is no ftp server installed on the system. The system appears secure from this test.

The two network test, nmap and nessus, must be run from another computer on the same subnet as bernese. If the machines are not on the same subnet, the tests may not be complete as intervening routers and/or firewalls can block ports that may actually be open on the machine being tested. As

there are other machine on the DMZ thay may be compromised, bernese should be tested with the possibility of unlimited attack vectors in mind. I recommend using a Linux box for this testing, nmap and nessus work best from Linux.

NMAP is a port scanner from www.insecure.org. It will scan a range of IP addresses and look for open ports on each host it encounters. There are a number of scanning options available for nmap, if you want more information go to http://www.insecure.org/nmap for more information. Redhat makes nmap available in its standard distribution also. Make sure there are not restrictive routers or firewalls between your testing machine and bernese. Run nmap (as root) with the following command: */usr/bin/nmap –s S –O 10.100.100.100.* This uses a SYN scan and attempts to identify the operating system. The scan will take some time as it checks each possible port. When the scan is done the only open port it sees is port 25, sendmail. The SSH port does not appear to be open to the scanner as the iptables filter will allow only packets from the two privileged subnets to get to the ssh port. The OS identification is correct, although you get a disclaimer since only 1 port is found to test. This test shows that there are only necessary services (sendmail and SSHd) listening, greatly restricting the possible attack vectors on bernese. The SSH port is further restricted by both iptables and tcp wrappers to a limited range of addresses. While an attacker can spoof their source address, this is difficult from an Internet address as the 10.0.0.0 network is not Internet-routable. If a DMZ server is compromised then spoofing is easier, but the attacker still must know the correct address and the increase in difficulty makes it more likely the compromise will be detected. .

Nessus is a network security scanner. It will scan a range of hosts, and a range of ports on each host, and then look for known security holes with those ports. Nessus is available from www.nessus.org. Get the program and install it per the instructions on the website. Make sure the testing machine is on the same subnet as bernese to avoid having ports blocked.
Start the nessus client and login to the nessus server (both can be on the same machine). Under the Pref tab select ICMP ping and set the nmap scan to SYN. Under the Scan Options tab set the port range to 25. You have already done a port scan on bernese with nmap and you know that port 25 is the only available port. You could scan the ports again but this makes the test MUCH longer with the same results.

Check the sendmail greeting message by telnetting to port 25 as follows: *telnet bernese 25*. The message shown should be: bernese ESTMP is ready. If you see anything else then the SMTP_LOGIN_MESSAGE configuration in the Sendmail configuration section was not done. Go back and check this.

Check mail relaying by using telnet to connect to the sendmail port. Check the cases of: mail to your domain, mail from your domain, and mail that is not to or from your domain. The first 2 cases should succeed, the third case should fail. For mail to/from your domain you can use mail on bernese. To send mail to user@domain.com use the command */bin/mail –s TEST user@domain.com.* After you hit return, enter the test message you want. End the message with a line containing only a period. The CC: is optional. Send to your domain and an outside domain and make sure the mail arrives. You can check the maillog, /var/log/maillog, to see if the mail transferred. To check relaying use the telnet command to port 25, *telnet bernese 25*. You should get back *host@localdomain ESTMP is ready.* The commands will be as follows:
HELO bernese.domain250 localhost.localdomain

Hello [10.100.100.100], pleased to meet you
mail from:fake@fake.com
250 2.1.0 fake@fake.com... Sender ok
rcpt to:fake@fake.com
550 5.7.1 fake@fake.com... Relaying denied.
If you get relaying denied you have the sendmail relaying set correctly, if not check you sendmail.mc file , the /etc/mail/mailertable and the /etc/mail/access file.

Make sure you are receiving email from the logcheck program. Run */usr/local/etc/logcheck.sh* and check to see that you get mail. To make sure that there is something in the file use the */usr/bin/logger –p auth.info "Testing BAD"* command to send a message to syslog. This command sends the message Testing BAD to the auth loggin service. If successful this will appear in /var/log/secure and verify your syslog.conf changes, the operation of syslog, and thelogcheck.sh program should pick this up as a security violation. This will also be your most common false positive in the violations section, it is amazing how often the three letter sequence b a d will appear in the maillog.

Check tripwire operation by running */usr/sbin/tripwire –check.* Once you confirm it is running correctly add a file to /usr/bin (*touch /usr/bin/testtripwire)* and run tripwire again to make sure the new file is picked up. After you have verified the operation, delete the /usr/bin/testtripwire file. Note that to do this test you must remount the /usr filesystem as read-write (*/usr/bin/mount /usr –o remount,rw)*. Remount it as read-only when the test is finished.

Ongoing Operation

This system has very simple backup requirements. It is a mail relay, so there should be no mail on the system under normal circumstances. There are no user accounts other than for administration. The items that need to be backed up are:

/etc/mail/	This directory contains all the sendmail configuration files. The mc files are on another machine, do not forget to back these up as well. The files that will change on bernese are the database files, virtusertable, genericstable, access and mailertable.
/etc/tripwire/twpol.txt	This is the policy file for tripwire, listing the files to watch and what to watch for. This should not change often.
/var/lib/tripwire/bernese.twd	This is the current tripwire database and should not change often.
/etc/ssh/sshd_config	SSH daemon configuration file. This should never change.
/usr/local/etc/logcheck.*	These are the configuration files for logsentry. They will change as you decide what messages you can ignore. After a time they will become static files.
/etc/sysconfig/iptables	Iptables configuration file. This file should be static.
/etc/syslog.conf	Configuration file for syslog. This file should not change.
/etc/hosts.allow	

/etc/hosts.deny	Configuration files for TCP wrappers. These should change only as you change access rules for administrators.
/etc/cron.allow /etc/at.allow	These files restrict who can modify the automatically running system jobs. The files should not change.

To keep the system up-to-date on patches, I recommend paying for Redhat's basic support. This will allow you to get notifications of fixes for your system. Some people object to sending Redhat the system configuration as a security problem. I believe the advantage of easier updates and available patch notification outweigh the problem of letting a third-party know your configuration. Once you have joined the Redhat network (http:// ??????????????) you can register the system. To register the system, after you have in installed in the DMZ, run the *up2date* command on bernese. It will step you through registering your system. Once the system is registered you can run *up2date--list* to get a list of available patches. For further information on using the up2date command see the man pages.

Carefully monitor the email sent by the logsentry (logcheck) system. These will provide warnings of compromise attempts. You will have to adjust the logcheck.violations.ignore and logcheck.ignore files as you gain experience with your system. Edit the files with caution, you want to reduce the number of messages collected by logcheck without eliminating messages that might indicate problems. Adding lines to logcheck.violations should be simple, but put a lot of thought into what to ignore.

The tripwire email will let you know if anyone has changed files. Unless one of your administrators has changed the file, then this will tell you that you have been compromised. This is one email you really don't want to see problems with.

Set a schedule to monitor the server with nmap. I recommend at least once a month. The nmap run should show only port 25 open. If any other ports show open, then you have either been compromised or one of your administrators has failed to document a change.

References

Costales,Bryan with Allman, Eric. Sendmail, 3rd Edition . Sebastopol: O'Reilly & Associates 2003

Harker, Robert. Managing Internet Mail, Setting Up & Trouble Shooting Sendmail & DNS . Lincoln: Harker Systems 2002

Hunt, Craig. Linux Sendmail Administration, Alameda:, Sybex Inc. 2001

Russell, Rusty. Linux 2.4 Packet Filtering HOWTO. Version 1.26, 01/24/2002 URL:http://www.netfilter.org/documentation/HOWTO//packet-filtering-HOWTO.txt

Coulson,David. Mastering IPTables. May 2001 URL:http://davidcoulson.net/writing/lxf/iptables.pdf

Barrett, Daniel and Silverman, Richard. SSH The Secure Shell The Definitive Guide. Sebastopol: O'Reilly and Associates 2001

SANS Institute, Topics in Unix Security

SANS Institute, UNIX Practicum