Global Information Assurance Certification Paper

GIAC Certified UNIX Security Administrator (GCUX)

Practical Assignment

Version 1.9 (revised April 8, 2002)
Option 1 B Securing Unix Step by Step

Configuring an Email Server using RedHat Linux, Qmail, and OpenSSL

Tony Giordano

Table of Contents

Purpose

This document was created to partially fulfill the requirements for earning the GIAC Certified UNIX Security Administrator (GCUX) Certification. It details a step by step process for creating an email server that would be suitable for a small- to medium-sized organization.

This practical was written to focus on security-related aspects of the configuration. Details regarding mail server specific configurations are not covered in detail except where they relate directly to system security. However, many useful links and references are provided to aid in the configuration of a production level server similar to the one described here.

Description of System

Usage

This document provides a step-by-step guide for the installation, configuration, and maintenance of an email server for a small-to-medium sized organization. It is assumed that the organization has an internal network connected to the Internet and has a firewall system in place to provide the initial protections from external attacks. It is also assumed that the firewall configuration provides a DMZ (De-Militarized Zone) to house network services such as DNS, email, and web servers. This document does not involve the configuration of the firewalls or other servers except where their configuration directly affects the email system.

The selection of equipment, operating system, and application software was based on security, reliability, and cost. Also, the system is designed to make use of open source technologies in order to provide as much flexibility as possible when selecting related software such as email clients. The use of open source products increases overall system security and reliability by allowing any interested party to examine (and possibly improve) the software.

Whenever possible multiple levels of protection have been provided. For example, even though the network is protected by a firewall/DMZ configuration, the mail server employs a host-based firewall. Its always a good idea to use as much protection as possible and to only remove these layers when a justifiable business case is present. Even then, a risk analysis should be conducted to ensure that the potential benefits of relaxed security outweigh the risk of exploitation.

Hardware

The system used for the mail server consists of a Pentium II (266MHz) based PC with 128 MB of RAM. The computer utilizes a single 8 GB IDE hard drive. It contains a RealTek RTL-8029 10 Mbit ethernet card and is equipped with standard floppy and cdrom drives, a mouse and a keyboard.

Software

The software that will be used for the mail server is selected to provide maximum benefit to the customer. Security, reliability, price, and documentation availability were all factors in deciding which operating system and applications to install.

In order to achieve the minimal acquisition and maintenance costs, open source products were selected. The nature of open source programs means that literally thousands of programmers had the opportunity to review the code for security and performance issues. Also, the lack of profit-based motives ensures that all software is written thoroughly and its release is determined by its completeness and nothing else.

The operating system selected is RedHat Linux version 7.3. This release of RedHat uses the 2.4.18 version of the Linux kernel. In addition to the base linux install, the latest patches from RedHat were also installed on the system.

Remote access might be required for configuration and maintenance purposes. To ensure that only authorized users access the system and to protect the authentication information, the latest version of the secure shell daemon (ssh) was installed.

The heart of the system is the SMTP mail server software. Qmail was chosen over sendmail and postfix. This decision was based on security, functionality, and the fact that qmail is compatible with several other email-related and networking programs. All of which were written by the same author, Dan J. Bernstein.

The UC-SPI package of network monitoring tools was selected to replace such common services as inetd (or the newer xinetd) and to enhance tcp_wrappers. The ability to create and retrieve mail messages will be handled by an ssl-tunneled access to a POP3 (Post Office Protocol) daemon. The POP3 authentication mechanism will be checkpassword, a program also related to the qmail development project. The encryption of POP information will be handled by stunnel and openssl.

A host-based firewall system will be installed to provide even more protection against unauthorized protocols. Finally, we will ensure that system files are not modified by using the tripwire system integrity monitor. Tripwire was selected over AIDE because it is more widely known and thoroughly tested.

Risk Analysis

The potential for attack of the mail server is broken into two phases. The first is attacks that could occur while the system is being configured and is in its most vulnerable state. The second is for the time that the computer is functioning as a mail server.

During Installation

Since the machine will be in a vulnerable state while it is being configured, its best to physically disconnect it from the outside world until all software is installed, configured, and tested. Some studies have found that newly installed machines were attacked within minutes of being connected to the Internet[1]

The best approach is to connect the mail server to a stand-alone network in which all the computers are trusted and have no connections to the Internet or other corporate networks. Since the computer will not have any outside access, all software must be available on floppy disk or cdrom. Also, any testing from a remote computer will have to be done on the stand-alone network. The standalone network can be complicated or as simple as a laptop computer connected to the mail server via a cross-over cable. This is the process used for this project.

In Production

While the server is in production it will be vulnerable to malicious attacks and accidental misconfigurations. Furthermore, the potential for a Denial of Service attack is quite high because mail servers are highly "visible" machines from users on the Internet. Some possible attacks and the mitigating procedures are listed below.

Denial of Service

Swamping the server with a high number of messages could possibly fill hard drives with mail and logging information. Configurations within qmail will help minimize the amount of disk space used for messages and the RedHat logrotation capability will reduce the chances of using all of the /var partition. Qmail is also designed with some intelligent algorithms that reduce the number of bounced and resent messages. These can be further optimized to fit an organization's specific needs.

Network Eavesdropping

Eavesdropping can be broken into two areas, (1) sniffing the wire to retrieve password

[1] Lance Spitzner, founder of the Honeypot Project, writes: "I built the first honeypot in a spare bedroom early last year. Within 15 minutes it was scanned by a hacker looking for easy prey."

information and (2), sniffing to read email contents. We can virtually eliminate the password sniffing aspect by configuring all authentication to be handled via encrypted channels. However, protecting mail message contents is much more difficult because this type of traffic spans networks that are beyond our control. The best method to prevent the unauthorized release of information is to employ encryption within the mail clients via PGP or similar encryption techniques. Since this paper deals solely with the mail server configuration, PGP is not discussed.

Accidental Misconfiguration

Often a system is inadvertently comprised by a legitimate user. The chances for modifying critical files will be reduced by deploying a host-based integrity checker (tripwire) and by setting critical system files (/etc/shadow, /etc/passwd, /etc/sysctl.conf) to immutable. The tripwire system will notify the administrator of any file modifications and accidental changes will not be possible to the critical files without first removing the immutable setting.

Step by Step Installation

BIOS Configuration

The BIOS system allows the user to modify low-level hardware configuration settings. Many versions of BIOS's exist, however most have the same basic functionality and options. Probably the two most important configurations are the boot device(s) and the built-in password protections. Usually the password configuration can be configured to protect against modifications to the BIOS settings and also to prevent the system from booting without a proper password.

The primary boot device should be the hard drive. This will prevent anyone with physical access to the computer from inserting a bootable floppy or cdrom, bringing the system up in single-user mode, and mounting the file systems at will. However, hardware failures can be dealt with by anyone with the BIOS password.

The password protection settings should be set to require a password prior to making BIOS changes, but not for the boot process. This will allow the machine to reboot after a power outage automatically, yet only allow authorized people to make changes to the system.

Operating System Installation

This installation process assumes that the user has either purchased the RedHat 7.3 installation disks, or has created ISO images of them. To start the install, configure the BIOS to boot from the cdrom drive, insert the first cd into the drive and reboot the system.

Press ENTER at the first prompt to initiate the default (graphical) installation program. For the most part, the default options may be selected during the process. However, for thoroughness, all inputs are documented here -- sometimes with brief explanations or 'gotchas'. Any entries that were altered from the defaults are marked in **bold**.

Language Selection:
    English
Keyboard Configuration:
    Model: Generic 105 key (Intel) PC
    Layout; U.S. English
    Dead keys: Enable Dead Keys
Mouse Configuration;
    Generic 3-button Mouse (PS/2)
    Emulate 3 Buttons
Installation Type:
    Custom
Disk Partitioning Setup
    Manually partition with Disk Druid

Disk partitioning is highly dependent on the size and number of drives that will be used in the systems. However, a few important features of this install should be noted.

First, a separate partition for the /boot partition was created and forced to be a primary partition. Storing /boot on a dedicated partition is recommended by RedHat Inc.

"The /boot partition"
In order to avoid potential conflicts with the BIOS 1024 cylinder limit. Red Hat suggests that all files which are needed to bootstrap a system, such as the second stage LILO bootloader, any kernels and individual ramdisks, be kept in a small partition intentionally located near the front of the drive. This partition is referred to as the /boot partition.[2]

Second, although not necessary, its good practice to create separate partitions for /var and /home. Since these filesystems are writeable by normal users and by "system accounts" its possible for many files to be created in these directories. Once the file systems are filled to capacity, a denial of service state might result. Keeping these directories separated from system partitions will enable the administrator to correct any problems with minimal disruption.

Also, since we will be using disk quotas to minimize Denial of Service attacks. It will be necessary to have /home on a dedicated partition.

Finally, having /home and /usr/local partitions will enable the system to be upgraded without the need to copy user files, or third party software to a tape, cdrom, or remote computer. Simply electing to not format these partitions during a major system upgrade will preserve the data. Also, having the areas where user and third party data is stored on dedicated partitions simplifies the data backup process.

The file system format chosen for this particular installation is listed in the following table.

Partition	Mount Point	Size (MB)	File System Type
/dev/hda1[3]	/boot	32	ext3
/dev/hda2	/var	2048	ext3
/dev/hda3	/	2048	ext3
/dev/hda4	N/A	<extended>	

[2] RH300 -- Red Hat Certified Engineer Training and Certification Course, page 1-18.

[3] Forced to be a primary partition during the Disk Partitioning phase of the installation.

/dev/hda5	/usr/local	1024	ext3
/dev/hda6	N/A	250	swap
/dev/hda7	/home	remainder of drive (3567 MB)	ext3

Boot Loader Configuration:
Use GRUB as the boot loader
Install Boot Loader record on:
/dev/hda Master Boot Record (MBR)
Default Boot Image
Boot Label = RedHat Linux
Boot Loader Password Configuration:
Check Use a GRUB Password
Enter a password
Confirm the password
Network Configuration
Uncheck Configure using DHCP
Activate on Boot
IP Address: 192.168.1.2
Netmask: 255.255.255.0
Network: 192.168.1.0
Broadcast: 192.168.1.255
Hostname: rudolph.northpole.com
Gateway: 192.168.1.1
Primary DNS: 192.168.1.1

NOTE: The network settings should be adjusted to fit your particular network.

Firewall Configuration:
No Firewall (this will be completed later in the install process)
Additional Language Support:
default language set to English (USA)
Additional languages: none
Time Zone Selection:
UTC-06 US Central
Check Use Daylight Savings Time

NOTE: Again, time zone configuration should match your individual needs.

Account Configuration:
Root Password: *********
Confirm: *********

Probably the simplest and most important security consideration for any system is the root password. Any easily guessable (or written-down) password makes all other security configurations useless. Please select a password that you will remember, that is not easily guessable, and is unique from passwords on other systems. Also, use different passwords for the BIOS, Boot Loader, and root passwords.

Additional Account:
    none (these will be configured later)
Authentication Configuration:
    Enable MD5 passwords
    Enable shadow passwords
    NIS = disabled
    LDAP = disabled
    Kerberos = disabled
    SMB = disabled
Package Group Selection:
    Uncheck:
        Printing Support
        Classic X Window System
        X Window System
        GNOME
        Sound and Multimedia Support
        Dialup Support
    Leave Checked:
        Network Support
        Messaging and Web Tools
    Check:
        Utilities
        Software Development
        Kernel Development
    Select Individual Packages:
        turn off:
            Applications -> Communications -> efax
            System Environment -> Base -> yptools
            System Environment -> Base -> up2date
            System Environment -> Base -> rhn-register
            System Environment -> Daemons -> autofs
            System Environment -> Daemons -> nfs-utils
            System Environment -> Daemons -> portmap
            System Environment -> Daemons -> sendmail.cf
            System Environment -> Daemons -> sendmail-devel
            System Environment -> Daemons -> ypbind
        Leave all other settings in their current state.

The system will now format the drive partitions and install the specified packages. After several minutes you will be asked to insert the second RedHat cdrom. Since our system is configured to only use the minimal amount of services, we will not be prompted to insert the third RedHat cd. However, it is a good idea to keep the entire installation set available for possible future upgrades or changes to the mail server.

    Boot Disk Creation:
        Insert a blank floppy disk in the drive.

Conclusion:
Red Hat Linux 7.3 is now installed on the server. Remove all floppy disks and cdroms and reboot the system.

Operating System Patches

Red Hat maintains an ftp site with updates and patches for their software. At the time this project was completed there were six patches available. These were downloaded and written to a cdrom and them installed on the mail server using the following process:

```
mount /mnt/cdrom
rpm -Fvh /mnt/cdrom/patches/*rpm
umount /mnt/cdrom
```

The -F option to the rpm command is to ensure that new services are not added to the system as part of the patch operation. F, for freshen, will only install a patch if it is a newer version of a currently installed package. If the more commonly used option of U (for update) were used, all patches would be installed. This could lead to a system that was running services that are not necessary and probably not configured correctly.

As new patches become available, they will also need to be installed. The maintenance section of this document contains details for this procedure.

Tightening the Operating System

The operating system will be fairly tight because most services were not installed as part of the initial system installation. However, there are still several things that can be accomplished to decrease the likelihood of a remote or local exploit, to increase the ability to detect any security breaches, to prevent accidental misuse of the system, and to minimize the effects of Denial of Service attacks.

Disable CTRL-ALT-DEL system reboots

Many people work with Microsoft operating systems and have become accustomed to pressing CTRL-ALT-DEL to login and to recover from locked screen saver sessions. However, the CTRL-ALT-DEL combination will have a much different effect on this mail server -- it will reboot it. To prevent this accidental mishap we will completely disable the CTRL-ALT-DEL key combination.

1) open the /etc/inittab file in a text editor
2) find the line containing `ca||ctrlaltdel|/sbin/shutdown -t3 -r now`
3) comment this line by adding a `#` at the beginning
4) save the file
5) feed the updated information to the current init process with:

```
init q
```

Configure Automatic Logouts for Users[4]

To ensure that users do not forget to logout a timeout can be configured. To set this on a system-wide basis, add this line to the /etc/profile file.

1) search for a line containing: `export PATH USER LOGNAME....`
2) add a line immediately before this one that contains:

```
TMOUT 1800
```

3) append TMOUT to the list of variables that are exported on the next line.

SIDEBAR *SIDEBAR* *SIDEBAR* *SIDEBAR* *SIDEBAR*
The value of the TMOUT variable is the number of seconds of idle time before the user is Akicked off@ the system. Adjust this to fit your needs.

Remove shell command history files on logout

[4] **This only works for users that have a bash login shell.**

Add this line to the file /etc/skel/.bash_logout:

```
rm -r $HOME/.bash_history
```

Reduce information retained in history files

Reduce the chances of a hacker finding valuable information in shell command history files while the user is still logged in.

edit the /etc/profile file, set the HISTSIZE value to 10

Remove unnecessary system accounts and groups

By default, RedHat Linux installations are configured with many unnecessary system accounts. Some of these exist for services that our mail server will not be running and others are installed as a matter of convenience for the administrator. Most of these should be deleted.

Issue the following commands to remove user and groups accounts.

```
userdel mail
userdel adm
userdel lp
userdel sync
userdel shutdown
userdel halt
userdel news
userdel uucp
userdel operator
userdel games
userdel gopher
userdel ftp
userdel vcsa
userdel mailnull
userdel rpm
userdel nscd
userdel ident
userdel radvd
groupdel mail
groupdel adm
groupdel lp
groupdel news
groupdel uucp
groupdel games
groupdel gopher
groupdel dip
groupdel ftp
groupdel floppy
groupdel vcsa
groupdel mailnull
groupdel rpm
groupdel nscd
groupdel ident
groupdel radvd
```

Protect critical files from modification

The `chattr` command can be used to protect certain files from either accidental or intentional modification. Issue the following commands to set the immutable attribute of critical files.

```
chattr +i /etc/passwd
chattr +i /etc/shadow
chattr +i /etc/group
chattr +i /etc/gshadow
chattr +i /etc/grub.conf
```

NOTE: Anytime changes are required to these files you will need to first issue a `chattr -i <filename>` command. After the changes are in place, reset the immutable bit with `chattr +i <filename>`.

Protect against some Denial of Service/Malformed Packet attacks

Add the following lines to /etc/sysctl.conf

```
# protect against some DoS attacks
net.ipv4.icmp_echo_ignore_all = 1
net.ipv4.icmp_echo_ignore_broadcasts = 1
net.ipv4.tcp_syncookies = 1

# many hacks involve manipulating a packet's route
net.ipv4.conf.all.accept_source_route = 0
net.ipv4.conf.all.accept_redirects = 0

# fragmented packets are used to fool IDS's
net.ipv4.ip_always_defrag = 1

# packets with forged source IP's are probably malicious
net.ipv4.conf.all.rp_filter = 1

# log all suspicious packets
net.ipv4.conf.all.log_martians = 1
```

Create Operating System level accounts

Accounts for normal users should be created so that users do not need to log in to the system as root. Anytime that root access is required, the user should log in as a normal user, then use the `su` command to elevate his/her privileges to root. This increases system security in multiple ways.

First, the system logs will indicate which user switched to root in the /var/log/messages file. This information could be used to determine any malicious or accidental problems created by the root account.

Second, by requiring users to first log in as a normal user, the ability to log in directly as root can be limited to certain users without limiting system usability or functionality.

Restricting the ability to use the su command will be configured later.

For this example system only one user account was created. The account "tony" was created and configured using the following commands:

```
chattr -i /etc/passwd /etc/shadow /etc/group /etc/gshadow
useradd -m tony
passwd tony
Changing password for user tony.
New password: ********
Retype new password: ********
passwd: all authentication tokens updated successfully.
chattr +i /etc/passwd /etc/shadow /etc/group /etc/gshadow
```

Prevent Direct root Logins

Logging into the system as root directly is not good system admin or security practice. To prevent this action several changes are required to the mail server's configuration.

Step 1) edit the /etc/pam.d/su file
ensure that the first line is uncommented

```
(auth     sufficient /lib/security/pam_rootok.so)
```

uncomment the line:

```
auth required  /lib/security/pam_wheel.so use_uid
```

Step 2) Turn off the immutable attribute on system files

```
chattr -i /etc/passwd /etc/shadow /etc/group /etc/gshadow
```

Step 3) add users allowed to become the superuser to the wheel group
verify the GID of the wheel group

```
grep wheel /etc/group
```

the number in the third field is the GID, most likely it will be 10

Step 4) add the correct user(s) to this group

```
usermod -G10 tony
```

Of course you should replace the 10 with the GID of your wheel group and tony with the account name that you want to give su privileges to. Repeat this command for each account that will be added to the wheel group.

Step 5) reset immutable bits

```
chattr +i /etc/passwd /etc/shadow /etc/group /etc/gshadow
```

Configure Log Rotation

The default RedHat installation has the logrotate feature activated, but the frequency of rotations and amount of history to retain should be improved. Edit the

`/etc/logrotate.conf` file and make the following changes:

`weekly` becomes `monthly`
`rotate 4` becomes `rotate 6`
uncomment the `compress` line

These changes will allow the system administrator to have access to six month's worth of logs versus 4 weeks. As you can imagine, the increased amount of required disk space will be drastic. Compression has been employed to reduce the effects of these changes, however, a close watch should be kept on the /var partition and the log settings adjusted if necessary.

Modify the filesystem mount options

Edit the /etc/fstab file and make the following changes to the options field:

```
/boot     defaults
```

becomes

```
defaults,nosuid,nodev
```

```
/home defaults
```

becomes

```
nosuid,nodev,rw,exec,auto,nouser,async,usrquota
```

```
/var defaults
```

becomes

```
defaults,nosuid,nodev
```

Disk Space Quotas

One of the simplest denial of service attacks is to fill a system's disk space with "legitimate" files. Since this computer will function as a mail server, it would be quite simple to bombard the server with incoming email until its storage space was consumed. In order to prevent this type of attack we will use user-based disk quotas. Another protection for this type of denial of service will be employed when the mail server software is configured.

NOTE: the actual values for these quotas will depend on the number of users, the normal amount of email traffic, the storage space available, and the mail checking frequency and retention habits of the users. Adjust your settings to fit your specific needs.

Also note, the quotas set in this example server are extremely small. This was done intentionally to make it easier to verify that the quota limitations are working correctly.

Step 1) -- Ensure the Linux kernel has quotas available

Filesystem quotas are enabled by default with the RedHat 7.3 version of the linux kernel. If you are using a non-default kernel then you need to ensure that the quota modules are available.

Step 2) -- mount filesystems with usrquota option
This was accomplished previously when the fstab file was editted.

Step 3) -- Check quota configuration
/sbin/quotacheck -uvag

Step 4) -- Turn on Quota system
/sbin/quotaon -avug

Step 5) -- Create a user prototype
/usr/sbin/edquota -p tony
Replace tony with the user account that you want to use as your base prototype.

Step 6) -- Configure quotas for the prototype account
/usr/sbin/edquota tony
Adjust the values for soft and hard limits for both number of blocks and number of inodes that this user is allocated. This project used 20 and 25 for both sets of soft and hard limits.

Step 7) -- OPTIONAL -- adjust the grace period for which warnings are given when the soft limit is reached, the default is 7 days.

Tighten permissions on run-control files
In order to prevent unprivileged users from altering the system services, the permissions on files in the /etc/rc.d directory and its sub-directories should be changed.

```
chmod -R 700 /etc/rc.d/*
```

Application Software

Secure Shell Daemon
To provide remote access to the server for maintenance, sshd was installed. Recently, some vulnerabilities were discovered with ssh, but those problems have been corrected in the latest version of the software (openssh version 3.4). This version was downloaded from www.openssh.org and installed on the system. If a newer version is available, it should be used.

If you have been following this installation process exactly, then you will need to turn off

the immutable bit on the system files.

```
chattr -i /etc/passwd /etc/shadow /etc/group /etc/gshadow
```

Step 1) -- create privileged user area and account

```
mkdir /var/empty
chown root:sys /var/empty
chmod 755 /var/empty
groupadd sshd
useradd -g sshd -d /var/empty -s /bin/false sshd
```

Step 2) -- install ssh software

copy tar file to harddrive

```
cp /mnt/cdrom/openssh-3.4p1.tar.gz /usr/local/src
cd /usr/local/src
tar zxvf openssh-3.4p1.tar.gz
cd openssh-3.4p1
./configure && make && make install
```

Step 3) -- configure sshd

The secure shell daemon configuration settings are stored in /usr/local/etc/sshd_config. Edit this file to disable the outdated version and only allow connections to use Protocol 2. Also, change the default port (22) to a port that will only be shared with legitimate users. This will not stop any sophisticated hackers, but will prevent the scanning tools that are associated with large-scale attacks from finding the ssh service automatically.

uncomment the line containing Protocol 2,1
remove the comma 1
uncomment the line containing Port 22
change the 22 to a non-default port number (2112 for example)
save the file

Step 4) -- create startup/shutdown scripts

Create a file named `ssh` in `/etc/rc.d/init.d` similar to the one shown below:

```
#! /bin/bash
#
# ssh          Start/Stop the secure shell daemon.
#
# chkconfig: 2345 55 25
# description: ssh allows encrypted remote access

RETVAL=0



start() {
      echo -n $"Starting sshd: "
```

```
        /usr/local/sbin/sshd -g 60 &
        sleep 1
        PID=`ps -efw | grep sshd | grep -v grep | awk '{print $2}'`
        echo $PID > /var/lock/sshd
        echo
}

stop() {
        echo -n $"Stopping sshd: "
        PID=`cat /var/lock/sshd`
        kill $PID
        echo
}

restart() {
        stop
        start
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
  restart)
        restart
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart}"
        exit 1
esac

exit $?
```

Step 5) -- create the symbolic links in the run control directories with the command:

```
chkconfig --add ssh
```

Now the sshd daemon will start when the system enters run-levels 2, 3, 4, and 5; and will stop when the system enters states 0, 1, and 6.

Qmail Install

Before installing qmail, be sure that sendmail is completely uninstalled.

```
rpm -e --nodeps sendmail
```

Copy the program from cdrom to the harddrive

```
mount /mnt/cdrom
cp /mnt/cdrom/qmail-1.03.tar.gz /usr/local/src
```

Uncompress/Unpack qmail

```
cd /usr/local/src
tar zxvf /usr/local/src/qmail-1.03.tar.gz
```

This will create a new directory (/usr/local/src/qmail-1.03) and place source code, documentation, and install scripts in it. Following the instructions in the INSTALL file, we make a directory for qmail

```
mkdir /var/qmail
```

SIDEBAR SIDEBAR SIDEBAR SIDEBAR SIDEBAR
Qmail was intentionally designed to have all its binary files, its configuration and documentation files, and the outgoing mail queue in a single location. This increases reliability by ensure that all files will be available to the system However, it can lead to some confusion for the administrator because things aren't found in their standard locations (man pages, executables, etc). It is my opinion that a suitable tradeoff is to place the man pages in a standard directory, but all other "required" files in their designed locations. This can easily be accomplished by creating a symbolic link prior to installing qmail. `ln -s /usr/share/man /var/qmail/man`

Some system accounts are created to own and run the various parts of the qmail system. For Linux the following script was developed by editing the INSTALL.ids file that comes with the qmail package

```
(turn off immutable bits, if necessary)
groupadd nofiles
useradd -g nofiles -d /var/qmail/alias alias
useradd -g nofiles -d /var/qmail qmaild
useradd -g nofiles -d /var/qmail qmaill
useradd -g nofiles -d /var/qamil qmailp
groupadd qmail
useradd -g qmail -d /var/qmail qmailq
useradd -g qmail -d /var/qmail qmailr
useradd -g qmail -d /var/qmail qmails
(turn immutable bits back on, if necessary)
```

Next, we compile the qmail binaries

```
cd /usr/local/src/qmail-1.03
make setup check
```

Then we configure the control files that qmail will use when delivering and accepting messages.

```
./config rudolph.northpole.com
```

NOTE: Replace rudolph.northpole.com with the actual DNS name of your mail server.

The config script relies on a functioning DNS system to determine the IP address of the mail server. If one is not available when qmail is being installed (because the machine is not connected to a network for instance), then the config-fast script should be used with the host's fully qualified domain name as the only argument (`./config-fast rudolph.northpole.com`).

SIDEBAR *SIDEBAR* *SIDEBAR* *SIDEBAR* *SIDEBAR*
Users familiar with sendmail will notice that qmail does not use a central file to control email aliases (/etc/aliases). Instead, each user can control their own aliases, with the limitation that those aliases must begin with the actual username (user-alias). System wide aliases are configured by the administrator. The controls for these are kept in an account named "alias" which was created earlier.

Configure the alias account to accept mail messages

```
su - alias
/var/qmail/bin/maildirmake ~alias/Maildir
```

Create the standard mail server aliases

```
touch .qmail-postmaster .qmail-mailer-daemon .qmail-root[5]
chmod 644 .qmail*
```

SIDEBAR *SIDEBAR* *SIDEBAR* *SIDEBAR* *SIDEBAR*
With qmail, the root account never receives mail. This is a design consideration that improved security in two ways. First, any malicious code that might be sent will not go directly to root. Instead, it can be directed to a less powerful (and safer) account. Second, mundane tasks such as reading email should not be performed as a privileged user. Not having mail sent directly to root discourages this practice.

[5] These files will be modified later to redirect incoming messages to legitimate users.

Exit from the alias shell

```
exit
```

The next step is to configure how qmail will store incoming messages. Several options are available for this setting and the decision on which method to use depends on how the users will retrieve their mail, and on the email client's capabilities.

The simplest and most widely user method is to concatenate all message for each user into a single file. This file is stored in a central directory (typically `/var/spool/mail`). However, this technique can lead to a corruption of all messages when any one message has an invalid format. Also, the necessity for each user to access the central area can cause performance problems and bottlenecks.

To avoid the bottleneck issue, the idea of storing mail in each user's home directory was developed. This technique concatenates messages into a file typically called `Mailbox`. However, the concatenation/corruption issue is still present.

The method recommended by the qmail author is to keep each message in a separate file under the user's home directory. This technique has been called Maildir. The Maildir method removes the need for each user to access a central store and reduces the concatenation problems. However, it is not a perfect solution either. Since the messages are stored under /home, or wherever user accounts are maintained, a denial of service can result if one user's mail fills up the file system.. Also, many older mail clients are not compatible with this format.

Ultimately the system administrator must decide on the storage system to be employed. For this project the qmail recommendation of Maildir will be used.

Copy the /var/qmail/boot/home template file to the qmail run control script (preserving its ownership and permissions)

```
cp -p /var/qmail/boot/home /var/qmail/rc
```

Edit this file to configure the Maildir storage format by changing the line:

```
qmail-start ./Mailbox splogger qmail
```

to:

```
qmail-start ./Maildir splogger qmail
```

Configure the system to automatically configure new accounts for the Maildir format:

```
/var/qmail/bin/maildirmake /etc/skel/Maildir
echo ./Maildir/ > /etc/skel/.qmail
```

Instal UCSPI (Unix Client-Server program interface)

Ucspi-tcp is a replacement for the inetd and xinetd super daemons -- it will ensure that other daemons are started when connection requests are received. Ucspi-tcp provides improved reliability, decreased complexity, and more efficient coding to bring a stable and safe platform for managing network services. In fact, the Australian Computer Emergency Response Team (AUSCERT) recommends the use of tcpserver (an integral part of ucspi) for use with qmail instead of inetd. "It is recommended that qmail be used with tcpserver instead of inetd."[6]

Steps to installing and configuring ucspi:

Download the latest source from http://cr.yp.to/ucspi-tcp.html and store it in /usr/local/src.

uncompress the file

```
cd /usr/local/src
tar zxvf ucspi-tcp-0.88.tar.gz
```

compile the source code and install the binary version

```
cd ucspi-0.88
make && make setup check
```

The binaries that are part of the ucspi-tcp package will be installed in /usr/local/bin by default.

NOTE: All daemons related to SMTP, SMTPS, POP3, POP3S will be started and stopped using a single script. This script will be detailed later. For now, verify the SMTP installation using the following command.

Verify the installation:

```
tcpserver -v -u <UID or qmaild> -g <GID of qmaild> 0 smtp \
/var/qmail/bin/qmail-smtpd\ 2>&1 /var/qmail/bin/splogger smtpd \
3  &
```

The output of `netstat -an` should show a line with `0.0.0.0:25` in the fourth column which indicates that the server is accepting requests on port 25 from all IP's.

Install qmail-pop3d

Qmail-pop3d is a package that allows users connecting via the Post Office Protocol version 3 (POP3) to connect to the qmail system. The POP3 protocol is used to create and retrieve email messages whereas SMTP is designed to deliver message throughout the network. The pop3d configuration consists of two parts.

[6] www.auscert.org/information/auscert_info/papers/usc20.html

First, to allow users to send and retrieve email, some authentication must be done. However, normal POP3 configuration sends the authentication information in clear text through the network -- this problem will be remedied later. The standard authentication module for qmail-pop3d is called checkpassword. This tool relies on the `/etc/password` and `/etc/shadow` files to determine if a user's password is correct. This architecture removes the need to synchronize normal passwords with email passwords.

Download checkpassword
copy the file to `/usr/local/src`, uncompress it, and install it

```
cp /tmp/checkpassword-0.90.tar.gz /usr/local/src
cd /usr/local/src
tar zxvf checkpassword-0.90.tar.gz
make && make install
```

The second part of the installation is simply to start a daemon that will listen for incoming POP3 requests on the server.

```
tcpserver -v -R -H -l 0 0 110 /var/qmail/bin/qmail-popup \
rudolph.northpole.com /bin/checkpassword \
/var/qmail/bin/qmail-pop3d Maildir 2>&1 &
```

Again, checking with a `netstat -an` should show a line containing `0.0.0.0:110` to indicate that the POP daemon is running correctly.

Install stunnel

Stunnel is a tool that will "wrap" normal network communication in an encrypted tunnel using Secure Sockets Layer (SSL) technology. This package requires that the openssl libraries be in installed before it can be used.

Step 1: install openssl libraries
copy the openssl source code to the hard drive and compile it

```
cp /mnt/cdrom/openssl-0.9.6d.tar.gz /usr/local/src
cd /usr/local/src
tar zxvf openssl-0.9.6d.tar.gz
cd openssl-0.9.6d
./config
make && make test && make install
```

Step 2: install stunnel packages
copy the source to the hard drive and compile it also

```
cp /mnt/cdrom/stunnel-3.22.tar.gz /usr/local/src
cd /usr/local/src
tar zxvf stunnel-3.22.tar.gz
```

```
cd stunnel-3.22
./configure
make && make install
```

Step 3: create certificates
The system will ask you for information regarding your domain name, company name, location, etc. Answer all the prompts with proper information. When finished the certificate file will be named stunnel.pem and will be stored in the `/usr/local/src/stunnel-3.22` directory. This file should be moved to a secure location on the hard drive that will be accessible when the daemons are being started. I suggest putting the file in `/usr/share/ssl/certs` and protecting it by setting its ownership, permissions, and immutable bit as follows:

```
chown root:root /usr/share/ssl/certs/stunnel.pem
chmod 400 /usr/share/ssl/certs/stunnel.pem
chattr +i /usr/share/ssl/certs/stunnel.pem
```

It's also a good idea to keep a copy of this file on a floppy disk, stored in a secure off-site facility.

SIDEBAR	*SIDEBAR*	*SIDEBAR*	*SIDEBAR*	*SIDEBAR*
In order to start the stunnel daemon, the current working directory must be the same as where the stunnel.pem file is stored (/usr/share/ssl/certs). This will be seen in the run control script that is presented later in this document.				

Test stunnel wrapping POP3 by issuing this command:

```
stunnel -N pop3s -d pop3s -r localhost:pop3 2>&1 &
```

Verify that its running by issuing a `netstat -an` command and looking for a process listening on port 995.

Test stunnel wrapping SMTP by issuing this command:

```
stunnel -N smtps -d smtps -r localhost:stmp 2>&1 &
```

Verify that its running by issuing a netstat -an command and looking for a process listening on port 465.

Configuring the POP and SMTP daemons to not allow unencrypted connections from remote computers

As mentioned earlier, the older mail protocols did not protect authentication information

very well. In fact, they required users (or email client software) to send username and passwords through the system in clear text.

Our system will not allow clear-text transmissions of username/password combinations for use with the POP3 protocol. To restrict these types of connections we will use the tcprules program that is part of the ucspi package.

Tcprules is similar to the tcp_wrappers product. It relies on files that determine which types of connections to allow and to deny. We will configure the tcprules system using these steps:

Step 1) -- Create directory to store the rules database
```
mkdir /etc/tcprules
```

Step 2) -- Create a file to be used for POP3 connections
```
vi /etc/tcprules/pop.tcp
```
insert these lines to allow connections only from the 127.* network and to deny all other connections:
```
127.:allow
:deny
```

Step 3) -- Convert the text rules format to the binary database
cd /etc/tcprules
```
tcprules pop.tcp.cdb pop.tcp.tmp < pop.tcp
```

Step 4) -- Configure the POP3 daemon to use the tcprules system This requires a change to the tcpserver command that was used to start the POP daemon. This change is shown in the following section.

Create start/stop scripts for all the mail-related services

The following file was created to be used to start and stop the qmail daemons, the unencrypted POP3 and SMTP daemons and the encrypted POP3S and SMTPS daemons. Most UNIX-like operating systems have a standard directory for these run control scripts -- usually under the `/etc/rc.d` directory. Red Hat 7.3 keeps run control scripts in `/etc/rc.d/init.d`. It also maintains a set of symbolic links that determines which rc scripts to start and which ones to stop when changing run levels.

Creating and naming the various symbolic links was handled with the `chkconfig` command and by including the chkconfig and description lines as comments in the script. The syntax for chkconfig command is:
```
chkconfig --add qmail
```

```
#! /bin/bash
#
# qmail         Start/Stop the qmail daemon.
#
# chkconfig: 2345 80 30
# description: qmail is a replacement for the standard UNIX program sendmail. \
#                  It handles incoming and outgoing SMTP connections. \
#                  This script also starts the ssl wrappers for POP and SMTP \
#                  connections.

RETVAL=0



start() {
   QMAIL_UID=`id -u qmaild`
   QMAIL_GID=`id -g qmaild`
   MAX_PROC=40
   OLDPWD=`pwd`

          echo -n $"Starting qmail-smtpd: "
          tcpserver -v -u $QMAIL_UID -g $QMAIL_GID 0 smtp \
          /var/qmail/bin/qmail-smtpd 2>&1 /var/qmail/bin/splogger smtpd 3 &
          sleep 1
          PID=`ps -efw | grep tcpserver | grep qmail-smtpd | grep -v grep | awk '{print $2}'`
echo $PID > /var/lock/qmail-smtpd          echo

          echo -n $"Starting qmail-send: "
          csh -cf '/var/qmail/rc &'
          sleep 1
          PID=`ps -efw | grep qmail-send | grep -v grep | awk '{print $2}'`
          echo $PID > /var/lock/qmail-send
          echo

          echo -n $"Starting pop: "
          tcpserver -x /etc/tcprules/pop.tcp.cdb -v -R -H -l 0 0 110 \
          /var/qmail/bin/qmail-popup \
          rudolph.northpole.com /bin/checkpassword /var/qmail/bin/qmail-pop3d \
          Maildir 2>&1 &
          sleep 1
          PID=`ps -efw | grep tcpserver | grep pop | grep -v grep | awk '{print $2}'`
          echo $PID > /var/lock/pop
          echo

          echo -n $"Starting stunnel-pop: "
          cd /usr/share/ssl/certs
          stunnel -N pop3s -d pop3s -r localhost:pop3 2>&1 &
          sleep 1
          PID=`ps -efw | grep stunnel | grep pop | grep -v grep | awk '{print $2}'`
          echo $PID > /var/lock/stunnel-pop
          echo

          echo -n $"Starting stunnel-smtp: "
```

```
        cd /usr/share/ssl/certs
        stunnel -N smtps -d smtps -r localhost:smtp 2>&1 &
        sleep 1
        PID=`ps -efw | grep stunnel | grep smtp | grep -v grep | awk '{print $2}'`
        echo $PID > /var/lock/stunnel-smtp
        echo

        cd $OLDPWD
        return
}

stop() {
        echo -n $"Stopping qmail-smtpd: "
        PID=`cat /var/lock/qmail-smtpd`
        kill $PID
        echo

        echo -n $"Stopping qmail-send: "
        PID=`cat /var/lock/qmail-send`
        kill -TERM $PID
        sleep 2
        kill -CONT $PID
        echo
        echo -n $"Stopping pop3d: "
        PID=`cat /var/lock/pop`
        kill $PID
        echo

        echo -n $"Stopping stunnel-pop: "
        PID=`cat /var/lock/stunnel-pop`
        kill $PID
        echo

        echo -n $"Stopping stunnel-smtp: "
        PID=`cat /var/lock/stunnel-smtp`
        kill $PID
        echo

}


restart() {
        stop
        start
}

case "$1" in
  start)
        start
        ;;
  stop)
        stop
        ;;
```

```
  restart)
        restart
        ;;
  *)
        echo $"Usage: $0 {start|stop|restart}"
        exit 1
esac

exit $?
```

Step 5) -- create the symlinks in the run control directories with the command:
`chkconfig --add ssh`

Now the sshd daemon will start when the system enters run-levels 2, 3, 4, and and 5; and will stop when the system enters states 0, 1, and 6.

Finish Qmail Configuration

A few other modifications should be made to the Qmail install. First, the chances that the disk drives are filled due to large messages should be reduced by limiting the size of individual incoming messages that qmail will accept. Creating a file named databytes in the `/var/qmail/control` directory that contains the maximum allowable file size will provide this functionality. The actual limit should be adjusted to fit your needs, but 5MB is a good size to start with.

Second, if you notice many emails from spamming hosts or other undesirable sources, they can be stopped by creating another file in the control directory. A file named badmailfrom that contains domains and/or hosts to deny will create a `A`blacklist`@` of remote sites.

Lastly, the mail intended for accounts such as mailer-daemon, root, postmaster and others should be directed to an actual user. Adding tony to the ~alias/.qmail-root and other .qmail files in the ~alias directory will inform the qmail daemons as to the proper place to deliver this type of mail. The mail can be redirected to local users, remote users, or sent to programs. The options that are available are highly dependant on the mail storage system chosen. Refer to the qmail documentation for more information.

Install iptables

The linux kernel provided by RedHat 7.3 includes support for iptables. However, RedHat has chosen an overly complicated system for configuring the iptables rules and for starting/stopping the required services.

I recommend not using the default iptables scripts and replacing them with more configurable and more maintainable versions.

Create a script that will contain the statements that will set up our host-based firewall. First we will clear out any existing rules. Then configure the system to only accept the types of traffic that we specify.

Create a file in /etc/iptable/rules containing the following information:

```
# IPTABLES CONFIGURATION

# Flush tables
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
/sbin/iptables -P FORWARD DROP

# allow incoming DNS
/sbin/iptables -I INPUT -p tcp --dport 53 -j ACCEPT
/sbin/iptables -I INPUT -p udp --dport 53 -j ACCEPT

# allow pop3s from internal addresses
/sbin/iptables -I INPUT -p tcp -s 192.168.1.0/24 --dport 995  -j ACCEPT
# allow smtps from internal addresses
/sbin/iptables -I INPUT -p tcp -s 192.168.1.0/24 --dport 465  -j ACCEPT

# allow smtp from external addresses
/sbin/iptables -I INPUT -p tcp ! -s 192.168.1.0/24 --dport 25   -j ACCEPT

# allow ssh from internal addresses (on our unique port)
/sbin/iptables -I INPUT -p tcp -s 192.168.1.0/24 --dport 2112 -j ACCEPT

# allow all outgoing flows
/sbin/iptables -I OUTPUT -m state --state NEW,RELATED,ESTABLISHED -j ACCEPT

# allow incoming packets related to flow originated locally
/sbin/iptables -I INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
```

Set the executable bit on this script:

```
chmod 744 /etc/iptable/rules
```

Next, create a script to be run during boot by the rc system. This script will start the iptables process. Create the following file in `/etc/rc.d/init.d`

```
#! /bin/bash
#
# iptables          Start/Stop the iptables system.
```

```
#
# chkconfig: 2345 08 92
# description: iptables is a host-based firewall system to restrict network
access to our serverl

RETVAL=0


case "$1" in
        start)
                if [ -x /etc/iptables/rules ]
                then
                        /etc/iptables/rules
                fi
                ;;
        *)
                echo $"Usage: $0 {start}"
                exit 1
esac

exit $?
```

The final step is to deploy the new script using the chkconfig command:

```
chkconfig Badd iptables
```

Install tripwire

Tripwire is a software package that informs the administrator of any changes to critical system files. By first creating a database of information (files, access permissions, modification times, file sizes, etc), then conducting periodic scans of the system, tripwire will notice any changes that were made. These changes can be sent to the administrator for appropriate action.

Step 1) -- install tripwire rpm
copy the package to the harddrive

```
cp /mnt/cdrom/tripwire-2.3.1-10.i386.rpm /usr/local/src
cd /usr/local/src
rpm -Uvh tripwire-2.3.1-10.i386.rpm
```

Step 2) -- configure initial run

```
/etc/tripwire/twinstall.sh
```

answer prompts for site and local passphrases

SIDEBAR	*SIDEBAR*	*SIDEBAR*	*SIDEBAR*	*SIDEBAR*
Tripwire uses two passphrases to ensure proper system access. The first is to				

protect the configuration and policy settings from unauthorized access and the second is to ensure the integrity of the system databases. The phrases chosen should be different from each other, and different from other passwords used on the mail server.

Step 3) -- customize policy file
The `/etc/tripwire/twpol.txt` file should be edited to fit your particular installation/configuration. Also, some additions to the default install should be made to force tripwire to manage the qmail binaries and control files. Adding:

```
/var/qmail/bin $(SEC_CONFIG) (recurse=1);
```

and

```
/var/qmail/control $(SEC_CONFIG) (recurse=1);
```

to the end of the "Critical Configuration Files" section will accomplish this.

Step 4) -- clean out policy file (optional)
The default tripwire configuration will search for many files that are not installed are our system. To avoid misleading errors in the tripwire reports, first run a tripwire check. Next, remove any "missing" files from the tripwire policy file and recreate the database.

Step 5) -- recreate policy file

```
cd /etc/tripwire
twadmin --create-polfile ./twpol.txt
```

Step 6) -- remove plain text versions of files

```
rm /etc/tripwire/twpol.txt
rm /etc/tripwire/twcfg.txt
```

Step 7) - create database

```
tripwire --init
```

Ongoing Maintenance

Patches:

System maintenance will involve adding any security-related patches that might become available. Both Operating System and application patches should be installed. One of the best ways to stay up-to-date with patches is to subscribe to the various mailing lists that exist for Linux, RedHat, Qmail, etc. Which of these and how many is entirely up to the administrator.

When updates to RedHat linux are available they will be posted on ftp://updates.redhat.com. The rpm tool can be used to download and apply these patches in one action. By creating a cron job to run:

```
rpm -Fvh ftp://updates.redhat.com/current/en/os/i686/*
```

these patches can be installed automatically.

Another option is to subscribe to the RedHat Network. This system will allow patches to be installed with as much or as little user intervention as desired. However, a fee can be charged for this service.

Tripwire:

Besides keeping the software up-to-date several other monitoring jobs should be performed periodically. Most important of these is to monitor the tripwire reports. Since we installed tripwire from the RedHat rpm, a nightly cron job will be run to verify the integrity of the system files. This report should be read daily.

Also, tripwire can be configured to mail this report to a remote computer. This benefits overall system security in two ways. First, it makes monitoring these reports more convenient, and therefore, more likely to be accomplished. Second, it ensures that this important information is available in multiple places. On the other hand, transmitting this type of critical information (versions of applications installed, installation locations, operating system configuration, etc) via unsecure channels can create new problems for the security administrator.

Disk Space:

Even though we have implemented log rotation and disk quotas, its still possible for the hard drive to become full. It=s a good idea to create a script that will notify the administrator of a file system that is approaching its capacity before the problem becomes critical.

Creating a cron job that will run a simple `df` command will provide the administrator of any problems before the used capacity reaches 100%.

Log Files:

They exist for a reason. Many programmers spent many hours to make sure that all necessary information is stored in these files. Read them! Everyday!

The log files for most applications are usually kept in the `/var/log` directory. We have touched on the ability to configure log rotation, compression, deletion, and mail with the RedHat `logrotate` command. Further configuration using `logrotate` and `cron` are possible to customize your particular log file maintenance.

Check Configuration

Qmail Functionality:

To completely test the functionality of the mail server you should test the creation, sending, and receiving of mail messages. Use an email client that understands SSL-based authentication to mail servers. An excellent choice is the KMail client that is part of the KDE 3.0 Desktop Environment. Configure the client to connect to the email server and create the proper accounts. Mail composition can be tested by sending a message from one account to another. The retrieval of messages can then be tested by reading that message.

Qmail has many configuration, tuning, and additions that are not covered in this document. See the references for further reading on qmail.

Run Nmap:

Nmap is a tool that will determine which ports are listening on a remote machine (in addition to a million other things). Use it to list the open ports on the mail server. Run the following commands from a remote computer that is connected to the mail server via a cross-over cable, or is part of the installation stand-alone network.

check for TCP-based services:
```
nmap -sT 192.168.1.2 -p1-65535 -P0
```
anything other than 25, 110, 465, 995, and 2112 should be investigated

check for UDP-based services:
```
nmap -sU 192.168.1.2 -p1-65535 -P0
```
no UDP services should be found

check for IP-based services:
```
nmap -sO 192.168.1.2 -P0
```
only the basic IP services should be found (icmp, igmp,udp, and tcp)

Periodic checking with nmap will help ensure that no new services are started on the mail server. This is especially important after installing or upgrading any software on the mail server. Again, this step can be automated by created cron jobs that will run the nmap commands and report any unusual services to the administrator.

Run Nessus:

Nessus is a vulnerability scanning tool. It can be installed on a different machine on the stand-alone network and used to verify that no known vulnerabilities exist. Like nmap, this tool should be run from a remote computer.

Download latest source from www.nessus.org

run the `nessus-installer.sh` script

make the certificates with:
`/usr/local/sbin/nessus-mkcert`

create the users with:
`/usr/local/sbin/nessus-adduser`

start the nessus daemon:
`/usr/local/sbin/nessus -D &`

run nessus:
`/usr/local/sbin/nessus`

Use the GUI to select which vulnerabilities to scan for and to view the results of the scan. You should only have a few of the most minor types of vulnerabilities found (linux version determined, ssh daemon version determined, etc.). Correct or mitigate any other vulnerabilities found.

Files without valid owners:

Find any files that are not owned by a valid user. These often occur when files are downloaded and uncompressed using the tar command.

```
find / -nouser -o -nogroup
```

Correct any files found by either removing the file or changing its ownership via the `chown` and `chgrp` commands.

Run Control Scripts:

To check the status of run control scripts, issue the following command:

```
chkconfig Blist <script>
```

This will display the start/stop configuration for each of the Linux run levels. Use this command to verify that only the correct services are started for the run level in question.

Tripwire:

To verify whether tripwire will correctly identify changes to the system we will intentionally change a few files, then run a tripwire check.

```
touch /var/qmail/control/tony_was_here
chmod 777 /sbin/chkconfig
```

run a tripwire check
```
tripwire --check
```

The tripwire report should indicate that the two files that were changed and their parent directories where flagged.

Tcprules:

To verify that tcprules is configured correctly and not allowing any traffic on port 110 to access the mail server from a remote computer, follow this procedure:

Log in to a remote computer
Open a command window (Command Prompt, Xterm, etc)
Attempt to connect using the POP protocol

```
telnet <mailserver.domain.tld> 110
USER <username>
PASS <password>
LIST
QUIT
```

If any command beyond the `telnet` attempt is successful, then the mail server is allowing POP3 connections from remote machines. Double check all installation steps to find where the misconfiguration took place and correct the problem.

Ethereal:

Ethereal is a packet sniffer. This tool can be used to examine the contents of network traffic. Analyzing the traffic is the best way to ensure that user authentication information is being encrypted when retrieving or sending email between the server and the clients.

The upper window in the Ethereal interace will show high level (low-level in OSI terms) packet information including source IP address, destination IP address, and port/service. By selecting the packets that contain the actual authentication info, the username and password information will be in clear text if the stunnel wrappers are not working correctly. However, in our situation, this data will be encypted.

To try to read the clear text version of a packet=s payload, select the particular packet in the upper-most window and then read its contents in the lower-most window. This verification along with the service name given by ethereal (SMTPS or POP3S) will be a double-check of the encryption process.

References

Australian Computer Emergency Response Team, "UNIX Security Checklist v2.0"
www.auscert.org/information/auscert_info/papers/usc20.html

Bernstein, Dan J. AUCSPI-TCP@
http://cr.yp.to/ucspi-tcp.html

CERT Coordination Center. AUNIX Configuration Guidelines@
http://www.cert.org/tech_tips/unix_coinfiguration_guidelines.html

Fenzi, Kevin & Dave Wreski. ALinux Security Howto@, April 25,1999
http://www.linuxnow.com/docs/content/Security-HOWTO-html/Security-HOWTO.html@

Mourani, Gerhard. ASecuring and Optimizing Linux: Red Hat Edition@, June 7, 2000
http://www.openna.com/products/books/securing-optimizing-linux/solrhe.htm

Nessus Documentation
http://www.nessus.org/documentation.html

Newbigin, John. AHow to Serverify RedHat Linux@
http://uranus.it.swin.edu.au/~jn/llinux/redhatserver.htm

Nmap Documentation
http://www.insecure.org/nmap/nmap_documentation.html

Red Hat Inc., ARed Hat Certified Engineer Training and Certification Course@, 1999

Red Hat Inc., ARed Hat Linux 7.3 B The Official Red Hat Customization Guide@, 2002

Red Hat Inc., ARed Hat Linux 7.3 B The Official Red Hat Reference Guide@, 2002

Scambray, Joel, Stuart McClure, George Kurtz. AHacking Exposed, 2nd Edition@, 2001

Sharpe, Richard. AEthereal User=s Guide, V1.1 for Ethereal 0.8.19", 2001
http://www.ethereal.com/docs/user-guide

Sill, David. ALife With Qmail@, December 29, 2001
http://www.lwq.org

Thorton, Adam, "A Thumbnail Guide to System Security for Linux: Part III -- SSL Wrapping Services: A Working Example", March 2002

www.naspa.com/pdf/2002/0302%20pdf/TO203006.pdf

Van Dooren, Ralph. AQuota Mini-Howto@, April 2002
http://tldp.org/HOWTO/mini/QUOTA.html