SANS Institute
InfoSec Reading Room

Implementing a Bulletproof MTA

This paper provides comprehensive instructions for installing and setting up the qmail Mail Transfer Agent (MTA), chosen because of it's security, reliability, and functionality.

Implementing a Bulletproof MTA

Nick Reeves

Introduction:

Installing and setting up the qmail **M**ail **T**ransfer **A**gent can be difficult and complex. What follows below is an instructional how to. The qmail MTA has been chosen because of it's security, reliability, and functionality. In 1997 Dan Bernstein offered $500.00 to anyone finding a security hole in qmail. As of this writing the reward is still unclaimed, see http://cr.yp.to/qmail/guarantee.html . This is a huge advantage over the sendmail MTA, which has had many published exploits in the past eight years. Some good information on these vulnerabilities can be found here: http://www.google.com/search?q=published+sendmail+exploits&hl=en The sendmail application did not even restrict relaying by default until the release of version 8.9

The qmail MTA has proven to be a very fast and reliable MTA. "As of October 2001 qmail is the 2nd most widely used SMTP server on the internet and is the fastest growing SMTP server ever." (http://cr.yp.to/qmail.html) It's creator, Daniel Bernstein, is responsible for a short list of software, such as the djbdns package, ucspi-tcp, daemontools, and of course qmail itself.

First the Red Hat 7.2 machine will be altered to rid of some security holes that are found in the default installation. Next, ucpsi-tcp, deamontools and qmail will be installed and configured. The vpopmail package by Inter7 technologies will then be installed. The vpopmail package integrates with qmail and allows relay control, virtual users, and virtual domains, these features are crucial components in a secure email processing system. Finally, tests will be ran on the POP3 and SMTP server, restrictive relaying will be tested, and a brief external audit of the host will be performed.

This document will leave no room for MTA failure. By using qmail, any further vulnerabilities, weaknesses, or bugs already in sendmail will be avoided, such as mailbox delivery format and invalid SUID settings on applications. Adding users with shell access to the system will not be needed, and there will be no way to take advantage of the system as an open-relay.

These instructions assume a Red Hat 7.2 Linux system is installed as a "server" without an x-windows system. It is recommended you have at least six months of experience using linux before using this guide. Please adjust your syntax, paths, and commands accordingly.

Securing Red Hat 7.2:

The default Red Hat 7.2 Installation is not internet ready. The most Secure MTA's and configurations will do us no good if other precautions are not taken to lockdown our host. After unnecessary services are shutdown and configuration changes to the ssh daemon are made, we can continue on with the MTA installation. After setup of the MTA, a very basic audit of the host will be done to confirm that no extra services are listening.

Run the setup application that comes with Red Hat:

```
# /usr/sbin/setup
```

When the menu opens use the Cursor keys to select **System Services** from the menu then hit Enter. The services disabled here (by hitting spacebar on the selected service) are: apmd, gpm, lpd, portmap, netfs, sendmail, telnet, and xfs. Reboot the system to confirm none of the services are running in the process listing. After the reboot perform a **netstat –nap** on the machine, which will list what ports the system is listening on, only port 22 (ssh) should be listed. edit **/etc/ssh/sshd_config** and change **PermitRootLogin Yes** To **PermitRootLogin no** and uncomment the **Protocol** line and remove **1** from it so it reads **Protocol 2**. This will force users logging in to use the ssh2 protocol. Restart sshd by issuing the command **killall –HUP sshd** and the new sshd configuration will be in use.

A host-level firewall should be setup as well, Red Hat 7.2 comes with iptables to do this. Guides on setting up iptables can be found at :
http://www.linuxnewbie.org/nhf/intel/security/iptables_basics.html
http://www.boingworld.com/workshops/linux/iptables-tutorial
http://www.linuxdoc.org/HOWTO/Firewall-HOWTO.html

When using iptables the only ports open to the public should be port 25(smtp) and port 110(pop3). Filtered access to our ssh port(22) from the appropriate networks should be setup as well. Make sure to allow a backup network access in your iptables config just incase of any outages or other emergencies. If none of your users will be checking their email from home or from the road, you can even filter access to port 110(POP3) with iptables.

No root logins are permitted with ssh now, so add an additional user account:
/usr/sbin/useradd admin

Set a password for the user following some basic password requirements.
Passwords should be at least 8 characters in length, and must contain letters numbers and symbols. As root edit the **/etc/login.defs** file and change the **PASS_MIN_LEN** from **5** to **8**. Also change the **PASS_MAX_DAYS** value to **90**. Users will be forced to use at least an eight-digit password that must be changed every 90 days:

passwd admin (follow prompts)

If the host is no longer running any unnecessary applications, and the sshd daemon is configured properly. These are good steps in the right direction to securing the host. Linux security in itself is worthy of a another practical. Some other good practicals and links for securing linux are :
http://rr.sans.org/linux/sec_install.php
http://rr.sans.org/linux/hardening.php
http://www.boran.com/security/unix1.html
http://www.linuxdoc.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/

Preparation:

Download the source for the packages we will need:

cd /usr/local/src/
wget http://www.wyzo.net/files/qmail-1.03.tar.gz
wget http://www.wyzo.net/files/daemontools-0.76.tar.gz
wget http://www.wyzo.net/files/ucspi-tcp-0.88.tar.gz
wget http://www.wyzo.net/files/vpopmail-5.0.1.tar.gz

As root, do the following:

mkdir /package
chmod 755 /package
mv daemon* /package/
tar –zxvf qmail-1.03.tar.gz
tar –zxvf ucspi-tcp-0.88.tar.gz
tar –zxvf vpopmail-5.0.1.tar.gz
rm *.gz (optional if you need the space)
cd /package/
tar –zxvf daemontools-0.76.tar.gz

all of our packages are now uncompressed.

Installations:

ucspi-tcp:

The ucspi-tcp package consists of many applications written by Bernstein. In this installation tcpserver and rblsmtpd will be used. rblsmtpd and its capabilities will be explained later. The tcpserver application works similar to inetd or xinetd. It listens on a configured port and calls the application specified when you start tcpserver. You can specify IP, port number and any variables that will be passed to the program run by tcpserver. If the **–x** option is used when starting tcpserver, it will use the specified .cdb file that contains a list of allowed hosts that can connect to the service. All hosts not in the .cdb file will be denied access to the service tcpserver is controlling. tcpserver has no known vulnerabilities. Running daemons under tcpserver avoids additional bugs or weaknesses found in the xinetd daemon. A few security advisories regarding xinetd are:
http://www.safermag.com/html/safer40/alerts/15.html
http://www.linuxsecurity.com/advisories/redhat_advisory-1603.html

cd /usr/local/src/ucspi-tcp-0.88
make
make setup check

ucspi-tcp is now installed.

daemontools:

The daemontools package includes programs such as multilog and supervise. This package monitors UNIX services Set to run by having a **run** file in the **/service/program-name/** directory. If the program called in the **run** file dies, supervise will try to restart it. The other application in daemontools we will be using is multilog, it is logging application that will listen to the program called in the **/service/program-name/run** file and log messages from that program to a specified path. The multilog options are set in **/service/program-name/log/run** file.

```
# cd /package/admin/daemontools-0.76/
# package/install
```

Daemontools is installed.
Note : daemontools places the following text into the **/etc/inittab** file to start supervise on startup of the PC :

```
SV:123456:respawn:/command/svscanboot
```

qmail:

```
# cd /usr/local/src/qmail-1.03
# mkdir /var/qmail
```

Create the groups and users qmail needs to run, and create the directories qmaill will be logging too.

```
# /usr/sbin/groupadd nofiles
# /usr/sbin/groupadd qmail
# /usr/sbin/useradd -g nofiles -d /var/qmail/alias alias
# /usr/sbin/useradd -g nofiles -d /var/qmail qmaild
# /usr/sbin/useradd -g nofiles -d /var/qmail qmaill
# /usr/sbin/useradd -g nofiles -d /var/qmail qmailp
# /usr/sbin/useradd -g qmail -d /var/qmail qmailq
# /usr/sbin/useradd -g qmail -d /var/qmail qmailr
# /usr/sbin/useradd -g qmail -d /var/qmail qmails
# mkdir -p /var/log/qmail/smtpd
# chown qmaill /var/log/qmail /var/log/qmail/smtpd
```

Configure and build qmail.

```
# make setup check
# ./config-fast "your.full.hostname"
```

qmail is now installed.

Now setup the boot and supervise scripts for qmail, as well as a control script borrowed and modified from the Life With Qmail documentation project. Qmail comes with several boot scripts for different types of mail delivery formats. Since this installation will use the qmail-pop3d program later, the **./Maildir/** format will be used.

From my research, the **./Maildir/** format seems more reliable than the "standard" **mbox** format. the mbox format simply appends messages to a single **mbox** file in the user's directory and will put a marker in the file indicating where the message starts and ends. Since the messages are in one file, the file is locked when being read or written, so all other processes must wait until the file is no longer being accessed before they can download from or append to the **mbox** file. If the mail server crashes or the daemons die in mid-delivery, the stop points of the message are not determined and messages could be merged together or possibly corrupted. Mailbox format is also
Not compatible with NFS due to file locking.

./Maildir/ format saves the messages in individual files. This prevents any locking issues with the messages and makes the server capable of storing and deleting messages faster, more reliably, and over NFS.

Copy the qmail boot script :

```
# cp /var/qmail/boot/home /var/qmail/rc
```

Edit the **/var/qmail/rc** file
and replace the last line of the file, which reads :

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

with :

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

Doing this tells the qmail delivery agent that it will be delivering
The messages in Maildir format.

Create the supervise run scripts for the
qmail-send and qmail-smtpd daemon:

```
# mkdir –p /var/qmail/run/qmail-send/log
# mkdir –p /var/qmail/run/qmail-smtpd/log
```

Create the /var/qmail/run/qmail-send/run file which contains :

```
#!/bin/sh
exec /var/qmail/rc
```

For the /var/qmail/run/qmail-smtpd/run the qmaild user ID and group ID are required, find these by typing:

```
# id –u qmaild
# id –g qmaild
```

The nofiles GID on this test server is **507** and the qmaild UID is **514**. The run script below will reflect this following the **–u** and **–g** flags for tcpserver. Change your UID and GID accordingly. It will also need to determined how many simultaneous incoming smtp connections will be allowed to the server, input that number after the **–c** flag. In the example 30 is used. The default connection limit for tcpserver is 40. raise or lower this depending on the amount of mail traffic you anticipate. In the script the **–v –R -H –l** (ell not one) and **-x** options are also used for tcpserver. **–v** is verbose which will have tcpserver output any errors to syslog, **–R** will tell tcpserver not to gather IDENT or TAP information, **-H** tells tcpserver not to lookup the remote host in DNS, **-l 0** tells tcpserver not too lookup the localhost of the machine, and **–x** tells tcpserver to use the .cdb database specified by the path that follows the **–x** option. The **–H –R** and **–l** options are all used to speed up tcpserver. It will take less time for tcpserver to establish the connections if it's not required to do DNS or IDENT lookups.

Create the /var/qmail/run/qmail-smtpd/run file which contains :

```
#!/bin/sh
exec /usr/local/bin/tcpserver -v –R -H -l 0 -x /home/vpopmail/etc/tcp.smtp.cdb -c 40 \
      -u 514 -g 507 0 smtp /usr/local/bin/rblsmtpd\
         -r outputs.orbz.org  /var/qmail/bin/qmail-smtpd 2>&1
```

The **/var/qmail/bin/rblsmtpd -r outputs.orbz.org** portion of the script needs an explanation. If you have ever received unsolicited e-mail you realize that spam is an ongoing issue. It is one of the main motivations for this practical. Spam is abusive, annoying, and expensive. For years people have been putting wide open or "open-relay" smtp servers on the internet. An open relay is categorized as a server that will send e-mail to anyone, even if the sender or recipient is not local to the system. These servers are a direct target for someone wanting to send spam. The spammer will pipe thousands or even millions of messages through the open relay out to the internet at almost no cost to them. It puts all the bandwidth and CPU loads on the relay in use. And even worse, it avoids the spammer from taking the fall. All the spam they sent did not originate from their server. It originated from the open relay. which means the person responsible for the machine will receive all the complaints. A few groups decided upon a way to reduce this annoyance, A list of open-relays that were spamming to the public was created, and they blocked smtp access from hosts on that list. Right now there is over 10 RBL databases

with a listing of smtp servers from which spam has originated. By using the rblsmtpd daemon in our script we ask the RBL list hosted at outputs.orbz.org if the smtp server we are establishing a connection with is in the RBL black hole list. If it is decided that the host in question was spamming and the IP address for the host is listed. Our SMTP server will not accept smtp traffic from it, thus keeping spam from entering the network.

The mutlilog run script is very similar to the run scripts above. The **t** option is used for time stamping and the **s** option to specify the size at which mutlilog rotates logs. Multilog by default keeps 10 logfiles. In the example the logs will be about 1MB in size before they rotate.

Create the /var/qmail/run/qmail-send/log/run file which contains :

```
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill /usr/local/bin/multilog t s1000000 /var/log/qmail
```

The above run script calls the multilog application as the qmaill(qmail log) user, tells it to write to /var/log/qmail directory, to timestamp and rotate the logs after they reach 1000000 bytes in size.

Create the /var/qmail/run/qmail-smtpd/log/run file which contains :

```
#!/bin/sh
exec /usr/local/bin/setuidgid qmaill /usr/local/bin/multilog \
t s1000000 /var/log/qmail/smtpd
```

Identical to the above script accept we are logging to /var/log/qmai/smtpd.

Give the run files executable permissions:

```
# chmod 755 /var/qmail/run/qmail-send/run
# chmod 755 /var/qmail/run/qmail-smtpd/run
# chmod 755 /var/qmail/run/qmail-send/log/run
# chmod 755 /var/qmail/run/qmail-smtpd/log/run
```

Above sendmail was stopped in the services configuration for Red Hat, and will not be running if the machine was restarted. Issue the following commands to stop and completely remove sendmail from the system:

```
# killall sendmail
# rpm -e --nodeps sendmail
```

Many applications use sendmail as their default MTA. The qmail package comes with a replacement binary that accepts the same commands and variables as the sendmail binary. Put the replacement binary into place by issuing these commands:

```
# ln -s /var/qmail/bin/sendmail /usr/lib
```

ln -s /var/qmail/bin/sendmail /usr/sbin

vpopmail:

After review of the vpopmail documentation and INSTALL files included with the source, this is the configuration that will be used:

cd /usr/local/src/vpopmail-5.0.1
/usr/sbin/groupadd -g 89 vchkpw
/usr/sbin/useradd -g vchkpw -u 89 vpopmail

The configure options used are :
--enable-roaming-users=y
Tell the vchkpw program, which will run under tcpserver, to place the IP of the person connecting into the open-smtp and .cdb files we specify IF the user properly authenticates. The qmail-send run file uses the .cdb file to restrict relaying. This is how users who successfully check their e-mail will be allowed to send e-mail.
--enable-auth-logging=y will have the vchkpw program log failed authentications to syslog. A requirement to troubleshoot or audit POP3 access.
--enable-default-domain=test.com vpopmail is capable of controlling mutiple domains, the default username would be **user@domain.com** when authenticating, with this option, users from the test.com domain only use their usernames i.e. **user** when authenticating for POP3 access.

Configure vpopmail with the following line :

./configure --enable-roaming-users=y --enable-auth-logging=y --enable-default-domain=test.com

Once the configure completes successfully the vpopmail configure script will output the configure options you chose as well as the other default options such as home directory UID, GID etc. Finish the vpopmail install by typing:

make
make install-strip

Vpopmail is installed and ready. Add a test domain and test users. Remember to use the same password restrictions that we used when adding our linux user. The usernames and passwords shown below are strictly for demonstrative purposes only. These three commands will add our test domain and 2 users.

~vpopmail/bin/vadddomain test.com password
~vpopmail/bin/vadduser test@test.com password
~vpopmail/bin/vadduser sans@test.com password

When a domain is setup using vpopmail it adds the domain to the rcpthosts and virtualdomains files in **/var/qmail/control** as well as adds the following line to **/var/qmail/user/assign** which will tell qmail where to deliver the local mail for the domains.

+test.com-:test.com:89:89:/home/vpopmail/domains/test.com:-::

Setup the pop3 daemon with tcpserver and vchkpw, the vpopmail INSTALL file has a great example to use. create the /var/qmail/run/vchkpw/run file which will contain:

```
!/bin/sh
   exec /usr/local/bin/softlimit -m 2000000 \
      /usr/local/bin/tcpserver -v -R -H -l 0 0 110 /var/qmail/bin/qmail-popup \
    mail.test.com /home/vpopmail/bin/vchkpw /var/qmail/bin/qmail-pop3d Maildir 2>&1
```

Make sure to change your paths and domain name when you create the script file above. The above script calls the qmail-popup and qmail-pop3d application. There is a 2000000-byte memory limit to avoid excessive memory usage and in this case tcpserver will be running verbose on port 110 reading maildir formatted messages. qmail-pop3d is an excellent alternative to some other POP3 daemons such as qpopper. Qpopper like sendmail has an extensive history of vulnerabilities, here are a few links regarding those vulnerabilities :
http://www.ciac.org/ciac/bulletins/k-009.shtml
http://www.cert.org/advisories/CA-1998-08.html
http://www.geocrawler.com/archives/3/91/1999/11/0/2936466/

Make the file executable:

```
# chmod 755 /var/qmail/run/vchkpw/run
```

Make our qmail control script. open http://www.wyzo.net/files/qmail.txt in your browser and copy the contents into the **/usr/sbin/qmailctl** file and make it executable :

```
# chmod 755 /usr/sbin/qmailctl
```

Create the default qmail aliases for the system.

```
# echo test@test.com > /var/qmail/alias/.qmail-root
# echo test@test.com > /var/qmail/alias/.qmail-postmaster
# echo test@test.com > /var/qmail/alias/.qmail-mailer-daemon
```

In the above example, all mail will forward to the test account. These should be changed so that they forward to the appropriate person.

Since daemontools was already setup, supervise is already running. Once the links to the **/supervise** directory are created our run scripts will start, and the machine will be ready to send and receive e-mail. Create the links with the following commands:

```
# ln -s /var/qmail/run/qmail-send/ /service/qmail-send
# ln -s /var/qmail/run/qmail-smtpd/ /service/qmail-smtpd
# ln -s /var/qmail/run/vchkpw/ /service/vchkpw
```

Run **/usr/sbin/qmailctl stat** to confirm everything is running ok.

Testing:

Make sure the POP3 server is working correctly. Then make sure messages are correctly being delivered to local users. Then test restrictive relaying by trying to send a message to an outside domain from a host that has not authenticated via POP3.

Login to the POP3 server. In the example below the IP address of the mail server is used because we do not own the test.com domain to setup DNS records for it.

```
# telnet 192.168.1.210 110
Trying 192.168.1.210...
Connected to 192.168.1.210.
Escape character is '^]'.
+OK <13180.1011142521@mail.test.com>
```

type in **user** followed by your username and hit enter. The mail server will respond with:
+OK
type in **pass** followed by your password and hit enter. The mail server will respond with:
+OK
You can type in the **list** command and hit enter:
+OK
1 3937
.
Issue the **quit** command to logout
+OK
Connection closed by foreign host.

POP3 is working correctly and there is one message waiting. check the **/home/vpopmail/etc/open-smtp** file to make sure our host was granted relay access to the server:

```
# more /home/vpopmail/etc/open-smtp

192.168.1.187:allow,RELAYCLIENT="",RBLSMTPD=""   1011142473
```

Vchckpw added my IP address to the open-smtp file and rebuilt the tcp.cdb file to grant me relay access.

Test that e-mail can be sent to outside domains. In the following example, issue the commands that follow the # symbol:

```
# telnet 192.168.1.210 25
 Trying 192.168.1.210...
 Connected to 192.168.1.210.
 Escape character is '^]'.
 220 mail.test.com ESMTP
 #helo mail
 250 mail.practical.com
 #mail <test@test.com>
 250 ok
 #rcpt <nick@wyzo.net>
 2 50 ok
 #data
 354 go ahead
#TESTING 123..
#.
250 ok 1011144485 qp 13213
#quit
 221 mail.practical.com
 Connection closed by foreign host.
```

Look at the /var/log/qmail/current log to make sure the remote delivery went through :

```
# tail –f /var/log/qmail/current
```

This is the log output :

```
2002-01-15 17:47:46.017 new msg 262766
2002-01-15 17:47:46.017 info msg 262766: bytes 184 from <test@test.com> qp 13234
uid 502
2002-01-15 17:47:46.023 starting delivery 5: msg 262766 to remote nick@wyzo.net
2002-01-15 17:47:46.023 status: local 0/10 remote 1/20
2002-01-15 17:47:53.426 delivery 5: success:
66.119.194.82_accepted_message./Remote_host_said:_250_ok_1011145303_qp_32485/
2002-01-15 17:47:53.426 status: local 0/10 remote 0/20
```

Replace nick@wyzo.net with a local address and an outside address and make sure your messages arrived. Also make sure users who have not authenticated are not allowed to send e-mail. (once again type the commands followed by The # symbol). Telnet into port 25 on the mail server from a host that has not logged into the POP3 server.

```
# telnet 192.168.1.210 25
Trying 192.168.1.210...
Connected to 192.168.1.210.
Escape character is '^]'.
220 mail.practical.com ESMTP
# helo mail
250 mail.test.com
# mail <test@test.com>
250 ok
# rcpt <nick@wyzo.net>
553 sorry, that domain isn't in my list of allowed rcpthosts (#5.7.1)
```

Before being allowed to input our data for the remote message, the mail server warns that we are not allowed to send e-mail unless it's destination is a local address or we are in the .cdb file, and closes our connection. So it is confirmed that restricted relaying is functioning properly.

Before going live with the server make sure you do extensive testing. Use the guidelines above. Test deliveries to multiple outside domains, multiple local addresses, and even multiple virtual domains. Perform a port scan on the machine using at least 2 different port scanners to make sure the server is only listening on the required ports.

The following output is from 2 different port scanners. NmapNt 2.53 by eEye , and superscan 3.00 by foundstone.

```
C:\programs\Nmapnt>nmapnt -sS 192.168.1.210
Starting nmapNT V. 2.53 SP1 by ryan@eEye.com
eEye Digital Security ( http://www.eEye.com )
based on nmap by fyodor@insecure.org  ( www.insecure.org/nmap/ )

Interesting ports on  (192.168.1.210):
(The 1520 ports scanned but not shown below are in state: closed)
Port       State       Service
22/tcp     open        ssh
25/tcp     open        smtp
110/tcp    open        pop-3

Nmap run completed -- 1 IP address (1 host up) scanned in 22 seconds
```

SuperScan output:

```
+ 192.168.1.210
        |___   22  SSH Remote Login Protocol
               |___ SSH-2.0-OpenSSH_2.9p2.
        |___   25  Simple Mail Transfer
               |___ 220 mail.test.com ESMTP..
        |___  110  Post Office Protocol - Version 3
               |___ +OK <15229.1011303523@mail.test.com>..
```

Both scanners show that only POP3, SMTP and SSH ports are listening on the mail server.

Conclusion:

- The Red Hat 7.2 machine has been minimally "locked down".
  Disabling the startup of unnecessary applications, password policies and tuning of the sshd daemon has significantly improved the security of our linux host. There is still room for dramatic improvement by use of local and network firewalls, network intrusion detection (snort), and local system integrity checking (tripwire).
- Spam has been prevented from originating or entering our network.
  The use of restricted relaying and RBL lists have made this task simple and effective.
- Education. The document provides an in-depth look into the installation and configuration of qmail. The applications have been defined and explained, and the interactions between them have been observed. This document walks through the installation process describing each step on the way, and provides enough info to troubleshoot issues with message delivery, as well as test for functionality.
- Bug free software has been found and implemented.
  Using qmail in replacement for sendmail, qmail-pop3d as opposed to qpopper, and tcpserver in the place of xinetd we have put a stop to any further security issues in those pieces of software.

What next?

The server is setup and humming along, but there is still more to keep in mind. It is always a good idea to keep informed on all the software used on a system. You can do this by subscribing to the qmail, Red Hat, and other security mailing lists. The software might currently be bug free, but new vulnerabilities are being found every day. Keeping abreast of software developments is a crucial ingredient to overall system security.

Further improvements upon this installation can be made as well. Some ideas that can further increase security for the MTA are virus scanning and POP3 encryption. POP3 authentications in this scenario are sent in plain text. Anywhere along the way between the client and server the username and password to the POP3 account could be obtained. a good document on setting up encrypted POP3 authentication is:

http://www.linuxdoc.org/HOWTO/mini/Secure-POP%2BSSH.html . Virus protection can also be installed on the MTA to scan messages as they arrive. The qmail-scanner program (http://qmail-scanner.sourceforge.net/) is a great addition to a qmail MTA.

Sources / References of information:

Qmail install files :
INSTALL, INSTALL.maildir INSTALL.vsm

The qmail homepage
http://cr.yp.to/qmail.html

Vpopmail installation guide
http://www.inter7.com/vpopmail/INSTALL

How to Install ucspi-tcp
URL : http://cr.yp.to/ucspi-tcp/install.html

Benchmarking mbox versus maildir
URL : http://www.courier-mta.org/mbox-vs-maildir/

Life With Qmail
URL : http://www.lifewithqmail.org/lwq.html

Securing Your NuSphere Installation on Red Hat Linux
URL : http://www.nusphere.com/products/library/secure_install_redhat.pdf

Allowing controlled SMTP relaying in Sendmail 8.9
URL : http://www.sendmail.org/tips/relaying.html

Choose a right password
URL : http://linuxdoc.org/LDP/solrhe/Securing-Optimizing-Linux-RH-Edition-v1.3/chap5sec31.html

The rblsmtpd program
URL : http://cr.yp.to/ucspi-tcp/rblsmtpd.html

Open relay database
http://ordb.org/about/

Manual pages: sshd
URL : http://www.openbsd.org/cgi-bin/man.cgi?query=sshd

The tcpserver program
URL : http://cr.yp.to/ucspi-tcp/tcpserver.html

The multilog program
URL : http://cr.yp.to/daemontools/multilog.html

The supervise program
URL : http://cr.yp.to/daemontools/supervise.html

Last Updated: July 17th, 2018

Upcoming SANS Training

Click here to view a list of all SANS Courses

SANS Riyadh July 2018	Riyadh, SA	Jul 28, 2018 - Aug 02, 2018	Live Event
SANS Pittsburgh 2018	Pittsburgh, PAUS	Jul 30, 2018 - Aug 04, 2018	Live Event
Security Operations Summit & Training 2018	New Orleans, LAUS	Jul 30, 2018 - Aug 06, 2018	Live Event
SANS Hyderabad 2018	Hyderabad, IN	Aug 06, 2018 - Aug 11, 2018	Live Event
Security Awareness Summit & Training 2018	Charleston, SCUS	Aug 06, 2018 - Aug 15, 2018	Live Event
SANS Boston Summer 2018	Boston, MAUS	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS San Antonio 2018	San Antonio, TXUS	Aug 06, 2018 - Aug 11, 2018	Live Event
SANS August Sydney 2018	Sydney, AU	Aug 06, 2018 - Aug 25, 2018	Live Event
SANS New York City Summer 2018	New York City, NYUS	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS Northern Virginia- Alexandria 2018	Alexandria, VAUS	Aug 13, 2018 - Aug 18, 2018	Live Event
SANS Krakow 2018	Krakow, PL	Aug 20, 2018 - Aug 25, 2018	Live Event
Data Breach Summit & Training 2018	New York City, NYUS	Aug 20, 2018 - Aug 27, 2018	Live Event
SANS Chicago 2018	Chicago, ILUS	Aug 20, 2018 - Aug 25, 2018	Live Event
SANS Prague 2018	Prague, CZ	Aug 20, 2018 - Aug 25, 2018	Live Event
SANS Virginia Beach 2018	Virginia Beach, VAUS	Aug 20, 2018 - Aug 31, 2018	Live Event
SANS San Francisco Summer 2018	San Francisco, CAUS	Aug 26, 2018 - Aug 31, 2018	Live Event
SANS Copenhagen August 2018	Copenhagen, DK	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS SEC504 @ Bangalore 2018	Bangalore, IN	Aug 27, 2018 - Sep 01, 2018	Live Event
SANS Wellington 2018	Wellington, NZ	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Amsterdam September 2018	Amsterdam, NL	Sep 03, 2018 - Sep 08, 2018	Live Event
SANS Tokyo Autumn 2018	Tokyo, JP	Sep 03, 2018 - Sep 15, 2018	Live Event
SANS Tampa-Clearwater 2018	Tampa, FLUS	Sep 04, 2018 - Sep 09, 2018	Live Event
SANS MGT516 Beta One 2018	Arlington, VAUS	Sep 04, 2018 - Sep 08, 2018	Live Event
Threat Hunting & Incident Response Summit & Training 2018	New Orleans, LAUS	Sep 06, 2018 - Sep 13, 2018	Live Event
SANS Baltimore Fall 2018	Baltimore, MDUS	Sep 08, 2018 - Sep 15, 2018	Live Event
SANS Alaska Summit & Training 2018	Anchorage, AKUS	Sep 10, 2018 - Sep 15, 2018	Live Event
SANS Munich September 2018	Munich, DE	Sep 16, 2018 - Sep 22, 2018	Live Event
SANS London September 2018	London, GB	Sep 17, 2018 - Sep 22, 2018	Live Event
SANS Network Security 2018	Las Vegas, NVUS	Sep 23, 2018 - Sep 30, 2018	Live Event
SANS DFIR Prague Summit & Training 2018	Prague, CZ	Oct 01, 2018 - Oct 07, 2018	Live Event
Oil & Gas Cybersecurity Summit & Training 2018	Houston, TXUS	Oct 01, 2018 - Oct 06, 2018	Live Event
SANS Brussels October 2018	Brussels, BE	Oct 08, 2018 - Oct 13, 2018	Live Event
SANS Pen Test Berlin 2018	OnlineDE	Jul 23, 2018 - Jul 28, 2018	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced