

QMAIL ROCKS

Step by Step guide to setup Qmail server

Prerequisites for configuration:-

1. CentOS 4.3 installed with package installation type complete.
2. All current packages must be updated by using **yum -y update** command.
3. A public IP and DNS entry.
4. At least 15-20GB of space on Hard disk.
5. Connection to Internet.
6. Sufficient knowledge of Linux server configuration.

About this documentation:-

The documentation is a step by step guide for configuring a full featured qmail server. This is helpful newbie as well as experts. This documentation is divided into 20 steps for configuring Qmail-Rocks server. This installation has been tested on Redhat 7.x,8 and 9 as well as RHEL 3,4 and Fedora Core 1, 2 ,3,4. For further information on Qmail-Rocks please visit http://www.qmailrocks.org/install_rh.htm. For queries and contribution regarding this documentation please contact friendlyogi@gmail.com .

Here is a brief summary of what this Qmail installation includes:-

1. **qmail** – Installing qmail packages from source.
2. **ucspi-tcp** - The ever important tcpserver components.
3. **qmail's daemontools** - The invincible daemon management software.
4. **qmail's pop3d** - For those ever popular POP services.
5. **qmail SMTP authentication patch** –
6. **qmail SMTP TLS patch** - Encryption for your remote smtpd connection.
7. **ezmlm mailing list** - A mailing list program.
8. **autoresponder** – To add auto responding feature .
9. **qmailanalog** - Analyze qmail logs and get automatic nightly reports.
10. **vpopmail** - Virtual domains, made easy. Available **without** mysql integration.
11. **vqadmin** - A handy web interface for administering vpopmail domains as well as qmail control files.
12. **maildrop** - Apply filters to mail. Somewhat hard to use, but a worthwhile endeavor.
13. **qmailadmin** - A nice web interface for managing vpopmail domains in detail.
14. **courier IMAP & IMAP SSL** – Provides IMAP services.
15. **Squirrelmail webmail client** - "webmail" Great plugins and a snap to install.
16. **qmail-scanner** - An alternative queue program that allows exciting plug-in.
17. **qms-analog** - Taking qmail-scanner's functionality to new heights.
18. **Spamassassin** - Via Qmail-Scanner. Tags SPAM for easy disposal with 95% efficiency or better.
19. **Clam anti-virus** - Also via Qmail-Scanner. Scans all incoming mail and quarantines all that virus ridden e-mail.
20. **qtrap** - A domain level, word based mail filter with whitelist/blacklist functionality.

Installation has two parts:-

A) Pre Installation Checklist:-

The most important piece of advice before you begin the installation:-

“READ EVERY WORD ON EVERY STEP OF THE INSTALLATION”

A successful QMR Qmail installation requires certain packages be installed and certain configurations be present on your server. I've put together this page to provide a **general** checklist for visitors to use before they begin the installation. Keep in mind that, since setups will vary from server to server, you may find some requirements that are not listed here.

How much disk space should I have available on my server?

The following is a GENERAL estimate. Actual needs may vary from person to person and machine to machine. A safe amount of disk space would be about 80-90MB. This includes the download of the qmailrocks.tar.gz software bundle and its extraction. After a successful installation, the total amount of needed space for a safely operating mail server could be brought down to about 15MB after the qmailrocks.tar.gz content is removed. If your server is multiple partitions:-

Partition Information:- This is what I had on my system with 20GB of Hard disk:-

```
[root@mail ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/sda3	6.2G	164M	5.7G	3%	/
/dev/sda1	99M	12M	83M	13%	/boot
None	252M	0M	252M	0%	/dev/shm
/dev/sda6	2.0G	35M	1.8G	2%	/home
/dev/sda5	2.9G	1.3G	1.5G	48%	/usr
/dev/sda2	7.7G	438M	6.9G	6%	/var

What software packages should I already have installed on my server?

1. The Apache Web Server - You can use either version 1.3.x or version 2.x. It shouldn't make that big of a difference. *On my system it was httpd.i386.2.0.52-41.ent.2.centos.*

2. PHP - Version 4 or higher. You will probably want to make sure that it's either compiled with imap and mysql support, or if you are installing from RPMs, install the php-imap and php-mysql packages alongside the php package. *On my system it was php.i386.4.3.9-3.22.12*

3. Perl - I use version *perl.i386.3:5.8.5-36.el4_6.3*, but any version of 5 should work.

4. GCC - The gcc compiler. You should already have it installed, but if you don't you'd better. *On my system it was libgcc.i386.3.4.6-10.*

5. MySQL - MySQL is only REALLY needed if you intend to use it with vpopmail. Also, you may run into trouble installing some packages if you don't have it installed. All in all, it's a good idea to have mysql server installed. Version 4.x works just fine, but 3.x will work too. *On my system it was mysql.i386.4.1.22-2.el4.*

6. OpenSSL - Version 0.9.5a or higher. *On my system it was openssl.i686.0.9.7a-43.17.el4_6.1.*

7. OpenSSL-devel - For Redhat products and Fedora users. *On my system it was openssl-devel.i586.0.9.7a-43.17.el4_6.1.*

8. libssl-dev, for Debian users.

9. wget - Downloading packages and software is alot easier with wget. *On my system it was wget.i386.1.10.2-0.40E.*

10. patch & patchutils - Available via RPM for Redhat, the ports collection for FreeBSD or apt-get for Debian. You'll need these packages to apply the needed patches along the way during the install. Only patch will be sufficient. *On my system it was patch.i386.2.5.4-20.*

A special note on SELinux :- If you have the "selinux" package installed, disable it first. The selinux package interferes with vpopmail and vqadmin's ability to function correctly. Make sure you disable selinux before proceeding with this installation guide.

What software packages should NOT be installed?

1. Postfix - Redhat often will have Postfix installed by default. If it's installed on your server, you will need to either uninstall it or disable it.

2. Any POP service - This includes Qpopper or any POP service that may be running out of xinetd. If your're server has a POP service running, you will need to disable it.

3. Any SMTP services

4. Sendmail - It's ok to have Sendmail installed, because we'll uninstall it during the qmail installation.

What Perl modules should be installed?

This list may vary depending on your setup, but here goes:-

Digest::SHA1

Digest::HMAC

Net::DNS

Time::HiRes

HTML::Tagset

HTML::Parser

We shall discuss later, how to install these modules and how to check pre-installed modules.

I'm running a firewall on my server. What ports should I open?

Outbound ports (tcp) :-

25 - SMTP

110 - POP services

143 - IMAP

783 - Spamassassin

993 - IMAPS

Inbound Ports (tcp) :-

25 - SMTP

80 - HTTP

110 - POP services

143 - IMAP

443 - HTTPS

783 - Spamassassin

993 - IMAPS

It's done for now, we may proceed to Installation section now ☺

B) 20 Steps for Installation:-

1. Download All the Needed Items for the Qmail installation.....	5
2. Installing Qmail 1.03 itself.....	6
3. Installing EZmlm 0.53 and EZmlm-idx.....	9
4. Installing Autoresponder 2.0.5.....	10
5. Installing Vpopmail 5.4.13.....	11
6. Installing VQadmin 2.3.6.....	12
7. Installing maildrop 1.6.3.....	14
8. Installing QmailAdmin 1.2.9.....	15
9. Finalizing the qmail installation.....	16
10. Uninstalling Sendmail/Postfix.....	18
11. Starting up qmail.....	20
12. Installing Courier-imap/imapd with Courierpassd.....	22
13. Installation of the Squirrelmail web mail program.....	27
14. Clam Anti Virus & SpamAssassin.....	32
15. Installing qmail-scanner w/qms-analog.....	38
16. Installing Qmailanalog & Qlogtools.....	45
17. Installing Qtrap.....	47
18. Maintaining your qmail server.....	49
19. Mail client configuration.....	54
20. Feedback.....	57

Let's Begin with First Part,

Part 1 - Download the Software:-

The first thing you'll need to do to get started is to download all the needed software packages for the entire qmail installation process. To make things easier for everyone, I've combined all the needed packages into 1 giant tarball (.tar.gz) bundle that you can download in one easy step. Most of the included packages are the latest versions as of **March, 2005** (there are a few intentional exceptions) and I will make efforts keep the software bundle up to date as new versions are released. If you're going to be using the Qmailrocks installation guide to install qmail, I would strongly recommend that you download my software bundle. I've included several ready-made scripts and so forth, so everything will make more sense if you're on the same page as I am when going through the install.

So let's start the installation by getting the needed software. You will notice that below I create a new directory called **/downloads** and I place the Qmailrocks tarball in that directory before unpacking it. I would strongly recommend you do this as the rest of the instructions on this site are geared toward this source directory structure. Anyway, let's get down to business.

```
mkdir /downloads
```

```
cd /downloads
```

Now download the Qmailrocks.org software bundle (Size=25MB). The command below will download the bundle from the qmailrocks.org main server in Texas, USA. You can also download the bundle from a [qmailrocks mirror site](http://www.qmailrocks.org/downloads/qmailrocks.tar.gz), if you wish)

```
wget http://www.qmailrocks.org/downloads/qmailrocks.tar.gz
```

Output:-

```
--23:44:40-- http://www.qmailrocks.org/downloads/qmailrocks.tar.gz  
=> `qmailrocks.tar.gz'
```

```
Resolving www.qmailrocks.org... 64.246.60.29
```

```
Connecting to www.qmailrocks.org[64.246.60.29]:80... connected.
```

```
HTTP request sent, awaiting response... 200 OK
```

```
Length: 26,425,994 (25M) [application/x-gzip]
```

```
100%[=====>] 26,425,994 34.16K/s ETA 00:00
```

```
23:53:24 (49.54 KB/s) - `qmailrocks.tar.gz' saved [26425994/26425994]
```

Once you've downloaded qmailrocks.tar.gz, were going to place in a directory called "downloads" and then unpack it...

```
tar zxvf qmailrocks.tar.gz
```

Part 2 - Installing Qmail Itself

Now that you've downloaded all the needed packages, we can start the install. At this point you should have a qmailrocks source directory located at /downloads/qmailrocks. If you don't, go back to step 1. This step involves the setup of the very heart of your new qmail server. In this step, we'll install qmail itself, ucspi-tcp and daemontools. These 3 packages are the core of the qmail server and will be the foundation on which we build everything else. To start things off, I've created a handy little shell script that takes care of the first portion of getting qmail, ucspi-tcp and daemontools installed. Simply run this script from the command prompt. The script will tell you what it's doing along the way.

/downloads/qmailrocks/scripts/install/qmr_install_linux-s1.script

Output :- All steps completed!

If all goes well, you should have all the needed **user and groups** created as well as all the needed **directories, permissions and ownership settings** needed for the installation of qmail, ucspi-tcp and daemontools.

Before we start to compile and install qmail, ucspi-tcp and daemontools, we're going to apply a group of patches to qmail. These patches will build all sorts of cool functionality directly into qmail before we install it. In total, we're going to add around 15 patches, but fortunately [John Simpson](#) has combined all but one of these patches into one giant patch file. But it gets even easier because I've thrown together a shell script that applies ALL the patches in one quick step.

Let's apply these mega-patches and get this patching business out of the way...

/downloads/qmailrocks/scripts/util/qmail_big_patches.script

Output :- All done!

Now we build Qmail...

cd /usr/src/qmail/qmail-1.03

make man && make setup check

NOTE: - In my case it shows error :-

./compile qmail-local.c

./compile: line 3: exec: cc: not found

make: * [qmail-local.o] Error 127**

Reason:- GCC Libraries not installed properly, Some of them are missing so it can't compile .C files.

Solution:- Install GCC Libraries,

yum install lib*

This may install nearly 187 Packages, like :-

gcc, gcc4, gd-devel, glib2-devel, glibc-devel, glibc-headers, glibc-kernheaders

After installing Libraries, again try to Make.

./config-fast your_fqdn_hostname (ex: ./config-fast mail.friendlyogi.com)

NOTE:- It will show following message,

Now qmail will refuse to accept SMTP messages except to mail.friendlyogi.com.

Make sure to change rcpthosts if you add hosts to locals or virtualdomains!

We shall discuss it later, how to add rcpthosts.

OK, qmail itself is now built and installed. Now let's generate a secure certificate that will be used to encrypt your server's TLS encrypted SMTP sessions...

make cert

When you run the above command you will be asked a series of questions regarding the generation of your certificate. They are non-technical questions...such as your location, business name, organization name, common name and so forth. If you've ever generated an SSL cert before, this should be familiar stuff to you. If you haven't, simply follow the directions.

Here's a sample of my cert cert configs:-

```
Country Name (2 letter code) [GB]:IN
State or Province Name (full name) [Berkshire]:Andhra Pradesh
Locality Name (eg, city) [Newbury]:Hyderabad
Organization Name (eg, company) [My Company Ltd]:Friendlyyogi Ltd.
Organizational Unit Name (eg, section) []:Mail
Common Name (eg, your name or your server's hostname) []:mail.friendlyyogi.com
Email Address []:postmaster@friendlyyogi.com
```

If the cert is successfully generated it will be automatically installed at /var/qmail/control/servercert.pem, along with a symlink to that cert at /var/qmail/control/clientcert.pem

Now we set the right ownership for the newly create cert...

```
chown -R vpopmail:qmail /var/qmail/control/clientcert.pem
/var/qmail/control/servercert.pem
```

Now we build ucspi-tcp...

```
cd /usr/src/qmail/ucspi-tcp-0.88/
```

RH 9/RHEL/Fedora/Slackware users: You will need to patch ucspi-tcp with an additional errno patch:

```
patch < /downloads/qmailrocks/patches/ucspi-tcp-0.88.errno.patch
```

```
make && make setup check
```

If you don't get any errors, that's it for ucspi-tcp!

Now we build the daemontools....

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

RH 9/RHEL/Fedora/Slackware users: You will need to patch daemontools with an additional errno patch:

```
cd /package/admin/daemontools-0.76/src
patch < /downloads/qmailrocks/patches/daemontools-0.76.errno.patch
cd /package/admin/daemontools-0.76
```

package/install

If no errors are reported, you've successfully compiled the daemontools package!

All done for now...

If you run take a look at the running processes on your server at this point, you should see the daemon "svscanboot" running. You can usually do this with a "ps -aux" command.

OK, Qmail is almost totally installed but we're going to pause right here and install a bunch of handy tools and features that will make Qmail pretty and fun!

Part 3 – Ezmlm

EZmlm is a nice mailing list add-on to Qmail. I've used it several times myself and its actually one of the better mailing list programs out there. When we install Qmailadmin later on, you'll see that EZmlm integrates seamlessly into Qmailadmin to provide a very user friendly mailing list management interface. As an added bonus, Vpopmail (which we will install as well later on) will let you control what users can and cannot use mailing lists! So let's install it...

```
cd /downloads/qmailrocks/  
tar zxvf ezmlm-0.53-idx-0.41.tar.gz  
cd ezmlm-0.53-idx-0.41  
make && make setup
```

If you don't get any errors, then ezmlm is all set up and ready to go!

Part 4 – Autoresponder

Auto responder does exactly what you think it does. It allows us to set up auto responders for mailboxes and so forth so that when you are away, it will send automatic reply to one who sends you some email.

So let's install it...

```
cd /downloads/qmailrocks  
tar zxvf autorespond-2.0.5.tar.gz  
cd autorespond-2.0.5  
make && make install
```

If you don't get any errors, then autoresponder is all set up and ready to go!

Part 5 - Vpopmail

Now we shall install Vpopmail without MySQL support. For installing with MySQL, please visit qmail-rocks' online guide.

Vpopmail is one of the major components of this installation. Vpopmail allows us to do virtual domain mail hosting. It's got a lot of built in tools and features that make it a dream to work with. Even if you don't want to host mail for multiple domains, I would still recommend installing Vpopmail. It just makes the whole mail game easier.

So let's install it...

```
cd /downloads/qmailrocks  
tar zxvf vpopmail-5.4.13.tar.gz  
cd vpopmail-5.4.13
```

Now let's configure vpopmail...

```
./configure --enable-logging=p  
make && make install-strip
```

Output :-

```
make[3]: Leaving directory `/downloads/qmailrocks/vpopmail-5.4.13'  
make[2]: Leaving directory `/downloads/qmailrocks/vpopmail-5.4.13'  
make[1]: Leaving directory `/downloads/qmailrocks/vpopmail-5.4.13'
```

If you don't get any errors, Vpopmail is done now.

Part 6 – Vqadmin

Vqadmin is simply a nice web based interface that will let us manage Vpopmail. Through the interface we can create new domains, new users, net quotas, enable services and much more. So let's install it...

```
cd /downloads/qmailrocks
tar zxvf vqadmin-2.3.6.tar.gz
cd vqadmin-2.3.6
./configure --enable-cgibindir=/path/to/your/cgi-bin --enable-
htmldir=/path/to/your/html/directory
```

(Example: `./configure --enable-cgibindir=/var/www/cgi-bin --enable-htmldir=/var/www/html`. You can use `updatedb` then locate “file name” to search)

make && make install-strip

Output:-

```
make[3]: Leaving directory `/downloads/qmailrocks/vqadmin-2.3.6'
make[2]: Leaving directory `/downloads/qmailrocks/vqadmin-2.3.6'
make[1]: Leaving directory `/downloads/qmailrocks/vqadmin-2.3.6'
```

If the installation is successful, Vqadmin should install itself in the cgi-bin directory of your default website. Unless you tell it otherwise, that usually defaults to `/var/www/cgi-bin`. You can specify another location in the `./configure` command above.

Now you will need to add the following to your server's Apache configuration file (usually `httpd.conf`)

```
<Directory "/path/to/your/cgi-bin/vqadmin">
deny from all
Options ExecCGI
AllowOverride AuthConfig
Order deny,allow
</Directory>
```

In addition, within the Apache master config file you will want to set the "AllowOverride" option to "All". Example: **AllowOverride All**

```
cd /path/to/your/cgi-bin/vqadmin
like /var/www/cgi-bin/vqadmin
```

Now you will want to create a `.htaccess` file to password protect the Vqadmin interface. There should already be a `.htaccess` file in the `vqadmin` directory, so all you need to do is configure it. We'll use the ever useful [vi](#) editor for this.

vi .htaccess

```
AuthType Basic
AuthUserFile /path/to/where/you/want/to/store/the/password/file/.htpasswd
{ /var/www/cgi-bin/vqadmin/.htpasswd }
AuthName "Log-In to Friendlyogi's vQadmin"
require valid-user
satisfy any
```

chown apache .htaccess (you may need to change the chown to either "nobody", "apache" or "www" etc., depending on what user your installation of Apache is running as)

chmod 644 .htaccess

Now you need to create a corresponding .htpasswd file that's going to contain the username and encrypted password for the Vqadmin administrator...

```
htpasswd -bc /path/to/where/you/want/to/store/the/password/file/.htpasswd admin  
admin_password
```

like ,

```
htpasswd -bc /var/www/cgi-bin/vqadmin/.htpasswd admin admin123
```

```
chmod 644 /path/to/where/you/want/to/store/the/password/file/.htpasswd
```

Now restart Apache...

```
apachectl stop
```

```
apachectl start
```

If all has gone well, you should now be able to browse (in your web browser) to:

<http://www.yourdomain.com/cgi-bin/vqadmin/vqadmin.cgi>

Like, <http://202.53.64.189/cgi-bin/vqadmin/vqadmin.cgi>

Enter the user "admin" and whatever password you assigned it {admin123 in this case}.

You should now see the Vqadmin interface. Go ahead and add a new domain to your server! Pretty cool, huh? The "postmaster" user serves as the admin user for any new domain and we will use it to log into Qmailadmin, when we install that. As soon as we complete the install, that domain will be instantly able to get mail (assuming the MX is correctly pointing to your server). The nice thing about Vpopmail and Vqadmin is that you do not need to restart anything after you add a domain. Once you add it, it simply works! We're almost done!

Part 7- Maildrop

Maildrop is a mail filtering agent which can be used to filter messages as they arrive on the server. You will probably notice, once this installation is complete, that you don't really use maildrop. However, it's a cool tool and it's worth having around if you ever decide to get crazy with filtering your incoming mail. You can find documentation on maildrop [right here](#). Let's install it...

```
cd /downloads/qmailrocks
tar zxvf maildrop-1.6.3.tar.gz
cd maildrop-1.6.3
./configure --prefix=/usr/local --exec-prefix=/usr/local --enable-maildrop-uid=root
--enable-maildrop-gid=vchkpw --enable-maildirquota
```

NOTE:- On my system it shows error:-
checking how to run the C++ preprocessor... /lib/cpp
configure: error: C++ preprocessor "/lib/cpp" fails sanity check
See `config.log' for more details.
configure: error: /bin/sh './configure' failed for maildir
Reason:- gcc-c++ package not installed.
Solution:- Install the package and re-compile.
yum install gcc-c++

```
make && make install-strip && make install-man
```

If you didn't get any errors, maildrop should be all set!

Part 8 – Qmailadmin

Qmailadmin is going to provide us with a nice web based interface for administering mail accounts once they are setup through Vpopmail(text based tool) or Vqadmin(graphical tool). From Qmailadmin we can create mailboxes, aliases, forwards, mail robots, mailing lists. You'll also find a few other handy functions as well. Qmailadmin is sort of the icing on the Qmail cake.

Let's install it...

```
cd /downloads/qmailrocks  
tar zxvf qmailadmin-1.2.9.tar.gz  
cd qmailadmin-1.2.9  
./configure --enable-cgibindir=/path/to/your/cgi-bin --enable-  
htmldir=/path/to/your/html/directory  
Like, ./configure --enable-cgibindir=/var/www/cgi-bin --enable-htmldir=/var/www/html  
make && make install-strip
```

That's it! Now browse to <http://www.yourdomain.com/cgi-bin/qmailadmin> Like, <http://202.53.64.189/cgi-bin/qmailadmin> and you should see the login screen. Login with the postmaster account and password for the domain that you created a while back using Vqadmin. Pretty cool, isn't it? Go ahead and create some additional mailboxes for your domain(s).

If you didn't get any errors, Qmailadmin should be all set!

Part 9 - Finalizing Qmail

Ok, we've installed some utilities for qmail server. Now it's time to wrap up the configuration for qmail itself. After that, we will stop and remove Sendmail from the server and then it's time to crank qmail up!

The first thing we're going to do is create the qmail supervise scripts, create the the qmail rc and qmailctl scripts and then set the needed permissions on all these scripts. Lucky for you, I've created a script to do all this for you. The script will give you a breakdown of what it is doing while it's running. If any errors occur, you'll see them. However, if you've configured everything right up until now, you shouldn't have any problems.

So let's run the script...

/downloads/qmailrocks/scripts/finalize/linux/finalize_linux.script

Hey, that was easy. Now there are just a couple tweaks to make to these new scripts we just created...

vi /var/qmail/supervise/qmail-pop3d/run

Find "**mail.example.com**" and change it to your server's hostname. For example: mail.friendlyogi.com.

vi /var/qmail/supervise/qmail-smtpd/run

Find "**mail.example.com**" and change it to your server's hostname. For example: mail.friendlyogi.com

Next, we'll kill any running qmail processes so that we can implement some final configurations.

qmailctl stop

We setup selective relaying for localhost...

echo '127.:allow,RELAYCLIENT=""' >> /etc/tcp.smtp
qmailctl cdb

NOTE:- Cross check this entry in tcp.smtp file, otherwise while using smtp by outlook it will show 553, relay error.

Now we create the common system aliases. These aliases are going to tell Qmail what to do with common server-generated mails. Stuff like bouncebacks, cron daily output and various other systemic sources. It's a good idea to redirect these aliases to a mailbox that you are going to check on a regular basis. You don't want to have your systemic mails piling up in some deep dark corner of your server doing no good and slowly filling your disk up.

echo some_address > /var/qmail/alias/.qmail-root

where "some_address" is the system user or email address (like postmaster@friendlyogi.com) you want these addresses aliased to.

echo some_address > /var/qmail/alias/.qmail-postmaster

where "some_address" is the system user or email address where addresses aliased to.


```
echo some_address > /var/qmail/alias/.qmail-mailer-daemon
```

where "some_address" is the system user or email address where addresses aliased to.

```
ln -s /var/qmail/alias/.qmail-root /var/qmail/alias/.qmail-anonymous
```

```
chmod 644 /var/qmail/alias/.qmail*
```

Alright. We've got qmail ready to go. One of the last things we need to do is to disable/uninstall Sendmail on the server and replace the Sendmail binary with a symlink to qmail, so that our server won't freak out with Sendmail being gone.

Part 10 - Uninstalling Sendmail

Well, the moment you've been waiting for is finally here. We're going to uninstall the current MTA from the server. On some systems, this means removing Sendmail while on many systems it involves removing Postfix (another popular MTA). Either way, we're going to remove the current MTA and replace it with our new qmail installation. However, since Sendmail (or rather the path to Sendmail) is such a commonly used item among tons of server operations and cronjobs, you will see that, after we uninstall Sendmail, we will actually make an artificial Sendmail that is nothing more than a direct injection into Qmail.

Anyway, let's do it...

Since this installation concerns Redhat, I'm going to assume that either have Sendmail or Postfix installed as an RPM. If you have it installed from source instead, we'll cover that as well.

Removing Sendmail

First, you'll want to check and see if your Sendmail installation is an RPM installation or a source installation. A good way to tell, of course, is to simply ask....

```
rpm -qa | grep sendmail
```

If Sendmail is installed as an RPM, you will see an output similar to this:

```
sendmail-x.x.x.x  
sendmail-doc-x.x.x.x  
sendmail-devel-x.x.x.x  
sendmail-cf-x.x.x.x
```

If it's installed as an RPM, you can uninstall Sendmail quite easily by running the following commands:

```
/etc/rc.d/init.d/sendmail stop (to stop Sendmail)  
rpm -e --nodeps sendmail-x.x.x.x  
rpm -e --nodeps sendmail-doc-x.x.x.x  
rpm -e --nodeps sendmail-devel-x.x.x.x  
rpm -e --nodeps sendmail-cf-x.x.x.x
```

That's it! Sendmail is gone! Keep in mind that if you had an rpm installation and you ever want to return to Sendmail, you can simply install the RPMs again with no trouble (assuming you remove Qmail).

However, if the RPM query produces no output, you most likely have a source install of Sendmail. In that case...

```
/etc/rc.d/init.d/sendmail stop (to stop Sendmail)  
mv /usr/lib/sendmail /usr/lib/sendmail.old  
mv /usr/sbin/sendmail /usr/sbin/sendmail.old  
chmod 0 /usr/lib/sendmail.old /usr/sbin/sendmail.old
```

Removing Postfix

First, let's see if you actually have Postfix installed...

```
rpm -qa | grep postfix
```

If it's installed, you'll get an output similar to this:

postfix-x.x-x

So let's remove it...

```
/etc/rc.d/init.d/postfix stop  
rpm -e --nodeps postfix-x.x-x
```

Establishing an "artificial" Sendmail path

Now we will need to set up an "artificial" Sendmail, which is just a symbolic link to Qmail's Sendmail. This is needed to ensure that the myriad of systemic mail scripts are still able to send mail! Qmail's "Sendmail" is nothing more than a direct injection into Qmail itself..

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

That's it! If all has gone well, Sendmail or Postfix should be uninstalled and the Qmail should be in its place.

Part 11 - Starting qmail

Alright, qmail should be ready to go! Let's run a script that will check the key components of the installation and make sure everything is alright.

This script does NOT check the CONTENT or SYNTAX of your scripts, but only for the scripts' existence and their ownership/permissions settings. If you've screwed up the syntax of on the run scripts, this tool will not detect it.

/downloads/qmailrocks/scripts/util/qmr_inst_check

If you get a "Congratulations, your Qmailrocks.org Qmail installation looks good! " type of message, you're all set. If you get some errors, just follow the directions to fix the errors and then re-run the script until you get all errors corrected and you get a "congratulations" message. Assuming, you've passed the installation check script, let's crank Qmail up!

qmailctl stop

qmailctl start

You can find out how things are running by:

qmailctl stat

You should see an output like this:-

```
/service/qmail-send: up (pid 28659) 1 seconds
/service/qmail-send/log: up (pid 28660) 1 seconds
/service/qmail-smtpd: up (pid 28678) 0 seconds
/service/qmail-smtpd/log: up (pid 28664) 1 seconds
/service/qmail-pop3d: up (pid 28676) 1 seconds
/service/qmail-pop3d/log: up (pid 28677) 1 seconds
messages in queue: 0
messages in queue but not yet preprocessed: 0
```

Congratulations, Qmail is now officially up and running and you should be able to send and receive mail on the server. Before going further, please check the open port list in Pre Installation List.

Let's test your new server's POP3 service...

telnet localhost 110

you should see something like this:

```
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
+OK <28948.1228166339@mail.friendlyogi.com>
user postmaster@friendlyogi.com
+OK
pass 123456
+OK
quit
+OK
Connection closed by foreign host.
```

This is the sign of a successful POP connection to the server!
Now try sending mail to that same user from another location. Telnet to 110 again and run the "list" command and you should see the message that you send...

```
C:\Documents and Settings\welcome> telnet 202.53.64.189 110
```

```
+OK <29522.1228166457@mail.friendlyogi.com>
```

```
user postmaster@friendlyogi.com
```

```
+OK
```

```
pass 123456
```

```
+OK
```

```
list
```

```
+OK
```

```
.
```

```
quit
```

```
+OK
```

```
Connection to host lost.
```

```
C:\Documents and Settings\welcome>
```

And now let's test your server's SMTP service to make sure the TLS functionality is there...

```
telnet localhost 25
```

```
[root@mail courier-authlib-0.55]# telnet localhost 25
```

```
Trying 127.0.0.1...
```

```
Connected to localhost.localdomain (127.0.0.1).
```

```
Escape character is '^'.
```

```
220 mail.friendlyogi.com ESMTP
```

```
ehlo localhost
```

```
250-mail.friendlyogi.com
```

```
250-AUTH LOGIN CRAM-MD5 PLAIN
```

```
250-AUTH=LOGIN CRAM-MD5 PLAIN
```

```
250-STARTTLS
```

```
250-PIPELINING
```

```
250 8BITMIME
```

```
starttls
```

```
220 ready for tls
```

```
quit
```

```
quit
```

```
Connection closed by foreign host.
```

Part 12 - Installing Courier-imap/imap with Courierpassd

Now that you have qmail up and running, we're going to add a few extras onto it. For starters, we're going to install Courier-imap/imap along with Courierpassd. We will also be installing the Courier-authlib package to enable proper authentication through courier-imap. Installing IMAP will, obviously, enable IMAP connections to the mail server and it is a necessary ingredient for most popular web based mail clients such as Horde, SQuwebmail and Squirrelmail. Courier-imap is the preferred IMAP server to install because it has built in support for the vchkpw mail user setup that Vpopmail utilizes. In short, Courier IMAP works with Vpopmail and virtual domains. In addition to installing Courier-imap, we're going to install Courierpassd. Courierpassd is a utility that allows users to change their mailbox passwords remotely. This will come in handy when we install Squirrelmail in the next step of the installation. Courierpassd will allow your mail users to change their passwords using the Squirrelmail interface. This will give your users more power over their account settings and, more importantly, keep them from pestering you whenever they want to change their passwords. Keep in mind that when we get to compiling courier-imap, it will have to be compiled by a NON-ROOT USER.

So let's start by installing courier-authlib...

```
cd /downloads/qmailrocks/
```

```
tar jxvf courier-authlib-0.55.tar.bz2
```

```
cd courier-authlib-0.55
```

```
./configure --prefix=/usr/local --exec-prefix=/usr/local --with-authvchkpw --without-authldap --without-authmysql --disable-root-check --with-ssl --with-authchangeprawdir=/usr/local/libexec/authlib --with-redhat
```

Note : In my system it shows error :-

Library gdb not found...

Solution : - gdb is already installed but gdb-devel is not yet installed.

Install gdb-devel

```
yum install gdbm-devel
```

```
make && make check
```

```
make install-strip && make install-configure
```

Now we will add a startup command for authdaemond to the /etc/rc.local file to ensure startup on boot...

```
vi /etc/rc.local
```

Add the following line:

```
/usr/local/sbin/authdaemond start
```

Now let's install courier-imap/imap...

Remember, courier imap needs to be compiled by a NON-ROOT USER. For the purposes of this guide, I am going to use a NON ROOT user called "yogi".

```
[root@mail qmailrocks]# useradd yogi
[root@mail qmailrocks]# passwd yogi
Changing password for user yogi.
New UNIX password:
BAD PASSWORD: it is based on a dictionary word
Retype new UNIX password:
passwd: all authentication tokens updated successfully.
```

```
cd /downloads/qmailrocks/
```

```
tar jxvf courier-imap-4.0.2.tar.bz2
```

```
chown -R yogi:wheel courier-imap-4.0.2
```

```
cd /downloads/qmailrocks/courier-imap-4.0.2
```

```
su yogi
```

```
./configure --prefix=/usr/local --exec-prefix=/usr/local --with-authvchkpw --without-authldap --without-authmysql --disable-root-check --with-ssl --with-authchangeptdir=/usr/local/libexec/authlib --with-redhat
```

```
make && make check
```

```
Output :-
```

```
make[2]: Leaving directory `/downloads/qmailrocks/courier-imap-4.0.2'
```

```
make[1]: Leaving directory `/downloads/qmailrocks/courier-imap-4.0.2'
```

Now we will exit out of our NON-ROOT USER and go back to being root...

```
exit
```

```
make install-strip && make install-configure
```

```
Output :-
```

```
make[1]: Leaving directory `/downloads/qmailrocks/courier-imap-4.0.2'
```

Now let's create an SSL certificate for the IMAP-SSL server...

```
/usr/local/sbin/mkimapdcert
```

This will start an automated process that creates a self-signed imap-ssl X.509 certificate called `imapd.pem`. It should create this new certificate at `/usr/local/share/imapd.pem`. If the certificate already exists, the "mkimapdcert" tool will not let you overwrite it.

```
vi /usr/local/etc/imapd.cnf
```

Change `postmaster@example.com` an administrative email address

```
postmaster@friendlyyogi.com
```

Save and exit

```
vi /usr/local/etc/imapd
```

Make sure that the following configuration exists: **IMAPDSTART=YES**

```
vi /usr/local/etc/imapd-ssl
```

Make sure that the following configuration exists: **IMAPDSSLSTART=YES**

Make sure that the following configuration exists:
TLS_CERTFILE=/usr/local/share/imapd.pem

Save and exit the file.

```
vi /usr/local/etc/authlib/authdaemonrc
```

Around line 27, you should see the "authmodulelist" setting. Make sure that "authvchkpw" is the only module listed. Like so:

```
authmodulelist="authvchkpw"
```

Save and exit the file.

Now we create the startup scripts...

```
cp /usr/local/libexec/imapd.rc /etc/rc.d/init.d/imap
```

```
cp /usr/local/libexec/imapd-ssl.rc /etc/rc.d/init.d/imapssl
```

Now let's start up Authdaemon, IMAP and IMAPSSL. To be safe we'll stop each service before starting it...

```
/usr/local/sbin/authdaemon stop
```

```
/usr/local/sbin/authdaemon start
```

```
/etc/rc.d/init.d/imap stop
```

```
/etc/rc.d/init.d/imapssl stop
```

```
/etc/rc.d/init.d/imap start
```

```
/etc/rc.d/init.d/imapssl start
```

If you run "nmap localhost", you should see both 143 and 993 now open and listening.
Now let's test it...

```
telnet localhost 143
```



```
[root@mail courier-imap-4.0.2]# telnet localhost 143
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^'.
* OK [CAPABILITY IMAP4rev1 UIDPLUS CHILDREN NAMESPACE
THREAD=ORDEREDSUBJECT THREAD=REFERENCES SORT QUOTA IDLE ACL
ACL2=UNION STARTTLS] Courier-IMAP ready. Copyright 1998-2005 Double Precision,
Inc. See COPYING for distribution information.
a login postmaster@friendlyogi.com 123456
a OK LOGIN Ok.
a logout
* BYE Courier-IMAP server shutting down
a OK LOGOUT completed
Connection closed by foreign host.
```

Hint: The "a" that you see before my login commands is required.

If you were able to log in , as in the example above, you're all set. IMAP is installed! For further testing, you can configure a mail client such as Outlook to test both the IMAP and IMAP-SSL connection to your server. IMAPS runs on port 993. Now that Courier-imap is installed, let's install Courierpassd. Remember, Courierpassd is going to allow us to enable your mail users to change their own mail passwords via the Squirrelmail interface.

Note: Courierpassd will require that port 106 be open to at least local traffic (traffic from 127.0.0.1)

```
cd /downloads/qmailrocks
tar zxvf courierpassd-1.1.0-RC1.tar.gz
cd courierpassd-1.1.0-RC1
./configure
make && make install
```

OK. Courierpassd is installed now. Next, we are going to configure Xinetd/Inetd to run courierpassd.

If your server uses Xinetd, here's how you integrate Courierpassd into it:

```
cd /etc/xinetd.d
```

Here we create the xinetd script for courierpassd...

```
vi courierpassd
```

```
service courierpassd
{
port = 106
socket_type = stream
protocol = tcp
user = root
server = /usr/local/sbin/courierpassd
```

```
server_args = -s imap
wait = no
only_from = 127.0.0.1
instances = 4
disable = no
}
```

Note: You may want to add additional IP's to the "only_from" setting above, depending on your needs.

Save and exit.

Now let's add the Courierpassd service to the system's services file:

```
vi /etc/services
```

Append to following line to the /etc/services file:

```
courierpassd 106/tcp #for /etc/xinetd.d/courierpassd
```

If your system uses Xinetd, then we now want to restart Xinetd:

```
/etc/rc.d/init.d/xinetd restart
```

Now let's test Courierpassd by trying to reset the password for a mail account. Here's what a successful test should look like:

```
[root@mail xinetd.d]# telnet localhost 106
Trying 127.0.0.1...
Connected to localhost.localdomain (127.0.0.1).
Escape character is '^]'.
200 courierpassd v1.1.0-RC1 hello, who are you?
user postmaster@friendlyogi.com
200 Your password please.
pass 123456
200 Your new password please.
newpass 654321
200 Password changed, thank-you.
quit
200 Bye.
Connection closed by foreign host.
```

If the above session is successful for you, Courierpassd is working correctly!

Now that we've got Courier-imap and Courierpassd installed, let's install the webmail client - Squirrelmail.

Part 13 - Installing Squirrelmail

Now that we have Qmail running with IMAP, we can install a webmail client to make mail accessible via a web browser. My choice for this was [Squirrelmail](#). Squirrelmail is both easy to install and it has lots of nice plugins to broaden its abilities. I know that a lot of people out there like to use [Horde](#). I like Horde myself and I've installed it on my Qmail server alongside Squirrelmail.

Anyway, let's install Squirrelmail...

The first order of business to make sure PHP is installed and correctly configured. So let's get that out of the way...

In order for Squirrelmail to work correctly, you'll need to check a couple things about your PHP installation:

1. First of all, make sure some rendition of PHP 4 is installed. If it's not, kick yourself in the ass and then go install it. Sorry, I'm not going to give a PHP installation tutorial. To be safe, you will want the following config options to be active in your PHP installation.

```
--enable-track-vars
--enable-force-cgi-redirect
--with-gettext
--with-mysql
```

If you're running Redhat, however, PHP can be easily installed as an RPM either manually or with "up2date". A default RPM installation of PHP will usually cover you. The only extra RPMs you'll want to install is "php-mysql".

2. Make sure you have PHP uploads turned ON. This is done by editing a line in your php.ini file. The location of the php.ini file can vary, but it's usually located at /etc/php.ini. If it's not, don't panic. Just run the old "locate php.ini" command. ;) Here's the line you will want to check/edit:

```
file_uploads = On
```

That's it for the PHP setup. Now let's download Squirrelmail...

You can download the latest stable version of Squirrelmail from:

<http://www.squirrelmail.org/download.php>

I recommend downloading the .tar.gz version of the latest release.

Now change directories to the web directory of the website you want to serve Squirrelmail off of. In my case, I used /var/www/html

```
[root@mail html]# wget http://www.squirrelmail.org/countdl.php?fileurl=http%3A%2F%2Fprdownloads.sourceforge.net%2Fsquirrelmail%2Fsquirrelmail-1.4.17.tar.gz
```

tar xzvf /path/to/squirrelmail-x.x.x.tar.gz (enter whatever version you downloaded)

```
[root@mail html]# tar xvfz squirrelmail-1.4.17.tar.gz
```

Now rename the untarred folder to something more friendly...

mv squirrelmail-x.x.x webmail

```
[root@mail html]# mv squirrelmail-1.4.17 webmail
```

And now let's configure Squirrelmail...

```
mkdir /var/sqattachments
chown -R apache:apache /var/sqattachments (or whatever user apache runs as)
cd webmail
chown -R apache:apache data (or whatever user apache runs as)
cd config
./conf.pl
```

This will run the Squirrelmail setup script which will allow you to customize the installation as well as set your server settings.

SquirrelMail Configuration : Read: config.php (1.4.0)

Main Menu --

1. Organization Preferences
2. Server Settings
3. Folder Defaults
4. General Options
5. Themes
6. Address Books
7. Message of the Day (MOTD)
8. Plugins
9. Database
10. Languages

D. Set pre-defined settings for specific IMAP servers

C Turn color on
S Save data
Q Quit

Command >>

Most of the important settings are in area #2, which is dubbed "Server Settings". Here are the specs I recommend:

General

- ```

1. Domain : mail.friendlyogi.com
2. Invert Time : false
3. Sendmail or SMTP : SMTP
```

#### IMAP Settings

- ```
-----
4. IMAP Server : localhost
5. IMAP Port : 143
6. Authentication type : login
7. Secure IMAP (TLS) : false
```

8. Server software : other**9. Delimiter : detect****SMTP Settings****4. SMTP Server : localhost****5. SMTP Port : 25****6. POP before SMTP : false****7. SMTP Authentication : login****8. Secure SMTP (TLS) : false**

Note :

If you want to configure Organization's logo, name, URL then configure section 1

Like,

To change logo keep in mind that it will take path from /var/www/html/webmail/config folder, because ./config.pl is running from this location. So here Logo file is present at /var/www/html/webmail/config folder and so we have to configure it like this:-

2. Organization Logo : logo-small.png

If you want to configure data and attachment directories then configure section 4

Like,

1. Data Directory : /var/www/html/webmail/data/**2. Attachment Directory : /var/sqattachements/**

After completing configuration, save and quit.

Then visit <http://yourdomain/webmail/src/configtest.php>

Like <http://202.53.64.189/webmail/src/configtest.php>

to test your squirrel mail configuration.

It will show **“Congratulations, your SquirrelMail setup looks fine to me!”**

Once you've configured Squirrelmail to your liking, it's time to configure Apache to serve our new webmail interface...

Note : On my system there is no need to configure virtual host on apache because I use single domain hosting.

There are probably about a million ways to do this, but here's what I do. I edit the httpd.conf **Apache configuration file** and add the following block:

```
<VirtualHost 1.2.3.4:80>
ServerName mail.mydomain.com
ServerAlias mail.*
ServerAdmin postmaster@mydomain.com
DocumentRoot /var/www/html/webmail
</VirtualHost>
```

Here's a breakdown of what's above:

<VirtualHost 1.2.3.4:80> - This indicates I'm setting up my mail interface as a virtual host, rather than IP based. Obviously, you're going to want to replace 1.2.3.4 with the IP address of your web server. Additionally, what you have here may vary from server to server and is dependent on how you have your Apache configured. Be cautious!

ServerName mail.mydomain.com - The official name of the webmail server virtual host.

ServerAlias mail.* - This line establishes a wildcard serveralias called mail.*. With this setup, any domain that is pointed to your server and that has an A record called "mail", will be able to get to the webmail interface by simply going to `http://mail.whateverdomain.com`. This is a pretty cool little feature and makes accessing the webmail interface easy for all of your users.

ServerAdmin `postmaster@mydomain.com` - The server administrative contact. This is not required, but I like to include it.

DocumentRoot `/var/www/html/webmail` - The document root of your webmail interface. This will vary, depending on where you chose to install Squirrelmail. In this example, you can see that I installed it at `/var/www/html/webmail`

</VirtualHost> - The closing tag to the virtualhost.

Make sure you restart Apache after making the above changes.

Ok, now that Apache is all configured, let's test the new webmail interface...

`http://www.yourdomain.com/webmail`

We'll sign in with the postmaster account under the domain you should have created earlier using Vqadmin...

Username: `postmaster@yourdomain.com`

Password: `your_password`

If all has gone well, Squirrelmail should log you right into your account! From here you will be able to both send and receive mail as well as a host of other additional functions. Again, Squirrelmail has tons of really cool plugins, and you can check them out at [Squirrelmail's plugins page](#). Installing the plugins is pretty easy and their site can help you out.

Now, I'm going to cover the addition of 1 Squirrelmail plugin. Keep in mind, there are tons of other plugins available. We're going to install the "change_pass" plugin which will allow our mail users to change their passwords from the Squirrelmail interface. This is made possible by the installation of Courierpassd that we did when we installed Courier-imap in the previous step.

So here goes...

`cd /path/to/squirrelmail_directory/plugins (example: cd /var/www/html/webmail/plugins)`

Download the module...

`wget http://squirrelmail.org/countdl.php?fileurl=http%3A%2F%2Fwww.squirrelmail.org%2Fplugins%2Fchange_pass-2.7-1.4.x.tar.gz`

Unpack the module...

`tar zxvf change_pass-2.7-1.4.x.tar.gz`

Remove the tarball of the module...

`rm -rf change_pass-2.7-1.4.x.tar.gz`

Now let's go and add the module into Squirrelmail...

```
cd /path/to/squirrelmail_directory/config
```

Run the Squirrelmail configuration tool...

```
./conf.pl
```

Choose the option for "plugins". On my version of Squirrelmail, this was option 8. Once you are in the modules menu you should see the "change_pass" module on the list of available, but inactive, modules. You can add the "change_pass" module by simply typing the number associated with the module and then hitting enter. Once the module appears on the active module list, go ahead and save the configuration changes and then exit out of the configuration tool.

Alright! You should be all set now. All that's left to do log into Squirrelmail and try out the password change tool!

That's it for Squirrelmail. Now let's move on to the next step.

Part 14 - Clam Antivirus & SpamAssassin

Alright, the next 2 steps are going to be the trickiest of the entire installation. First, on this page, we're going to install Clam Antivirus and Spamassassin. Then, on the next page, we will install qmail-scanner which will tie ClamAV and Spamassassin into the operation of our qmail server.

First, let's make sure you have all the required perl modules and required packages. All of the perl modules below are required for Spamassassin and Qmail-Scanner to work. I'll talk more about this further down in the install.

You will need these Perl Modules:

Digest::SHA1
Digest::HMAC
Net::DNS
Time::HiRes
HTML::Tagset
HTML::Parser
Pod::Usage
Parse::Syslog
Statistics::Distributions

You will also need these other packages:

perl-suidperl
unzip

If you don't have perl-suidperl or unzip installed, you will find RPM's of these 2 packages included in the Qmailrocks package.

- Helpful Hints -

Checking/Installing Perl Modules

How do I know if my server has these perl modules?

The following script will check your system for the above modules.

You MUST run this script as a NON-ROOT user!!

/downloads/qmailrocks/scripts/util/check_perlmods.script

```
[root@mail qmailrocks]# su yogi
[yogi@mail qmailrocks]$ /downloads/qmailrocks/scripts/util/check_perlmods.script
QMR check_perlmods v1.1
Checking for the existence of needed perl modules...
checking for Time::HiRes...
/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/Time/HiRes.pm
checking for Net::DNS...
```



```

/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/Net/DNS.pm
checking for Digest::SHA1...
/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/Digest/SHA1.pm
checking for Digest::HMAC...
/usr/lib/perl5/vendor_perl/5.8.5/Digest/HMAC.pm
checking for HTML::Tagset...
/usr/lib/perl5/vendor_perl/5.8.5/HTML/Tagset.pm
checking for HTML::Parser...
/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/HTML/Parser.pm
checking for Mail::SpamAssassin...
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin.pm
checking for Pod::Usage...
/usr/lib/perl5/5.8.5/Pod/Usage.pm
checking for Parse::Syslog...
No documentation found for "Parse::Syslog".
checking for Statistics::Distributions...
No documentation found for "Statistics::Distributions".
Check Complete.

```

Here it shows last two modules are yet not installed.
You will need to install them.

Important Note: The **Parse::Syslog** and **Statistics::Distributions** perl modules are currently not available in RPM package format (at least I couldn't find any).

```

[root@mail qmailrocks]$ cd /downloads/qmailrocks/perlmods/source/
[root@mail source]$ ls
Digest-HMAC-1.01.tar.gz      Net-DNS-0.47_01.tar.gz
Digest-SHA1-2.10.tar.gz     Parse-Syslog-1.03.tar.gz
HTML-Parser-3.36.tar.gz     Statistics-Distributions-1.02.tar.gz
HTML-Tagset-3.03.tar.gz     Time-HiRes-1.59.tar.gz
Mail-SpamAssassin-2.63.tar.gz

```

```

[root@mail source]# tar xvfz Parse-Syslog-1.03.tar.gz
[root@mail source]# tar xvfz Statistics-Distributions-1.02.tar.gz
[root@mail source]# cd Parse-Syslog-1.03
[root@mail Parse-Syslog-1.03]# perl Makefile.PL
[root@mail Parse-Syslog-1.03]# make
[root@mail Parse-Syslog-1.03]# make test
[root@mail Parse-Syslog-1.03]# make install
[root@mail Parse-Syslog-1.03]# cd ..
[root@mail source]# cd Statistics-Distributions-1.02
[root@mail Statistics-Distributions-1.02]# perl Makefile.PL
[root@mail Statistics-Distributions-1.02]# make
[root@mail Statistics-Distributions-1.02]# make test
[root@mail Statistics-Distributions-1.02]# make install

```

Now again check the installed module by running same script as above ,

```
[root@mail Statistics-Distributions-1.02]# su yogi
[yogi@mail Statistics-Distributions-1.02]$
/downloads/qmailrocks/scripts/util/check_perlmods.script
QMR check_perlmods v1.1
Checking for the existence of needed perl modules...
checking for Time::HiRes...
/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/Time/HiRes.pm
checking for Net::DNS...
/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/Net/DNS.pm
checking for Digest::SHA1...
/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/Digest/SHA1.pm
checking for Digest::HMAC...
/usr/lib/perl5/vendor_perl/5.8.5/Digest/HMAC.pm
checking for HTML::Tagset...
/usr/lib/perl5/vendor_perl/5.8.5/HTML/Tagset.pm
checking for HTML::Parser...
/usr/lib/perl5/vendor_perl/5.8.5/i386-linux-thread-multi/HTML/Parser.pm
checking for Mail::SpamAssassin...
/usr/lib/perl5/vendor_perl/5.8.5/Mail/SpamAssassin.pm
checking for Pod::Usage...
/usr/lib/perl5/5.8.5/Pod/Usage.pm
checking for Parse::Syslog...
/usr/lib/perl5/site_perl/5.8.5/Parse/Syslog.pm
checking for Statistics::Distributions...
/usr/lib/perl5/site_perl/5.8.5/Statistics/Distributions.pm
Check Complete.
```

Now it shows perl modules are installed.

Ok, so at this point you should have all of the above Perl modules installed on your system. In addition, as noted above, you will also need to have "unzip" and "perl-suidperl" packages installed. So let's install Clam Anti Virus and Spammassin..

Installing Clam Anti Virus...

```
cd /downloads/qmailrocks/
```

```
rpm -Uvh clamav-x.x-x.x.rpm
```

```
rpm -Uvh clamav-devel-x.x-x.x.rpm
```

Now we'll do a few configuration settings to the /etc/clamd.conf file.

```
vi /etc/clamd.conf
```

"Example" - should already be commented out. However, if it is not, make sure that it is commented out (#)

"LogFile" - should be set to /var/log/clamav/clamd.log

"LogTime" - should be uncommmented.

"LogSyslog" - should be uncommmented.

"User" - should be set to **qscand**

"ScanMail" - should be uncommented.

Now we are going to create the qscand user that ClamAV will run as. This user is really for qmail-scanner, which will be installed on the next step, but we'll go ahead and create it now so that we can start the ClamAV daemon...

```
useradd -c "Qmail-Scanner Account" -s /bin/false qscand
```

Note:- Please go through following steps properly, these are not documented at qmailrocks.org

You need to change ownership of various files from clamav to qscand, otherwise it will generate errors later on.

```
chown qscand:qscand /var/log/clamav -R
```

```
vi /etc/logrotate.d/clamd
```

```
edit following lines:-
```

```
create 640 qscand qscand
```

```
vi /etc/logrotate.d/freshclam
```

```
edit following lines:-
```

```
create 640 qscand qscand
```

```
vi /etc/freshclam.conf
```

```
edit following lines :-
```

```
DatabaseOwner qscand
```

```
chown -R qscand:qscand /var/run/clamav
```

```
chown -R qscand:qscand /var/lib/clamav
```

```
chkconfig freshclam on
```

```
chkconfig clamd on
```

Alright, now let's start Clamd...

```
/etc/init.d/clamd stop
```

```
/etc/init.d/clamd start
```

Ok, Clam Antivirus is now installed and running. Now let's go ahead and set it up so that it will auto-update every night with the latest virus definitions! First we will want to set up the proper logging for the updater....

```
/usr/bin/freshclam -l /var/log/clamav/clam-update.log
```

If the server is able to get updates, you should see an output similar to this:

```
Downloading daily.cvd [*]
```

```
daily.cvd updated (version: 8741, sigs: 34669, f-level: 38, builder: guitar)
```

```
WARNING: Your ClamAV installation is OUTDATED!
```

```
WARNING: Current functionality level = 6, recommended = 38
```

```
DON'T PANIC! Read http://www.clamav.net/faq.html
```

Database updated (472641 signatures) from db.in.clamav.net (IP: 219.106.242.51)
Clamd successfully notified about the update.

Woohoo! You're updated with the latest virus definitions from the Clam database!

Now we just set a crontab to run every night, which will run the auto-update procedure! In the example below, I've set mine to run every day at 1:25 AM. The odd run time came recommended from ClamAV, if you're wondering. Running freshclam at times other than the top of an hour reduces the load on their servers. so feel free to adjust the time to something even more offbeat, like 1:26 AM or something. The guys at ClamAV will thank you.

crontab -e (make sure you run this command as root)
25 1 * * * /usr/bin/freshclam --quiet -l /var/log/clamav/freshclam.log

Now save your new crontab and exit.

That's it! We're all done with Clam Anti Virus! You will now have a server-wide e-mail virus scanner that updates itself every night with the latest and greatest virus definitions!

Now let's install Spamassassin...

Note: Spamassassin utilizes port 783 to properly scan and release e-mail. If you have a firewall on your server, you will need to open up port 783 going in and out.

You may find, when installing Spamassassin, that a version of it is already installed on the system. If so, I would recommend uninstalling that RPM and installing the ones listed here.

```
[root@mail rpms]# rpm -e spamassassin
```

If you don't have the Mail::Spamassassin Perl module installed on your system, let's install it now...

```
cd /downloads/qmailrocks/perlmods/rpms/  
rpm -Uvh perl-Mail-SpamAssassin-3.0.2-1.i386.rpm
```

Now let's install the Spamassassin RPMs

```
cd /downloads/qmailrocks/  
rpm -Uvh spamassassin-3.0.2-1.i386.rpm spamassassin-tools-3.0.2-1.i386.rpm
```

```
[root@mail qmailrocks]# cd /downloads/qmailrocks/ [root@mail  
qmailrocks]# rpm -Uvh spamassassin-3.0.2-1.i386.rpm spamassassin-tools-3.0.2-1.i386.rpm  
error: Failed dependencies:
```

```
perl(Parse::Syslog) is needed by spamassassin-tools-3.0.2-1.i386
```

```
perl(Statistics::Distributions) is needed by spamassassin-tools-3.0.2-1.i386
```

Important Note: When you run the RPM installation command above, you may end up getting an RPM dependency error telling you that the **Parse::Syslog** and **Statistics::Distributions** Perl modules need to be installed first. Now, if you followed instructions you should have already installed these 2 modules from source. Since they were installed from source and *not* from RPM, the RPM installer can't see them. That's ok. If you are sure the perl modules are installed, you can simply override the dependency setting and

install the Spamassassin packages with the following command shown below. If you see any other dependency errors than the 2 perl modules listed here, you will need to address those seperately.

```
rpm -Uvh --nodeps spamassassin-3.0.2-1.i386.rpm spamassassin-tools-3.0.2-1.i386.rpm
```

Alright. Now we're going to add a user/group called "spamd" under which Spamassassin will be run...

```
groupadd spamd  
useradd -g spamd -s /home/spamd spamd
```

And now let's set some config options...

```
vi /etc/sysconfig/spamassassin
```

If the above file exists, replace its contents with the following line. If the file does not exist, create it and add the following line:

```
SPAMDOPTIONS="-x -u spamd -H /home/spamd -d"
```

Save and exit from the file.

```
vi /etc/mail/spamassassin/local.cf
```

Add the following line...

```
required_hits 5
```

Save and exit from the file.

Now start up Spamassassin...

```
/etc/rc.d/init.d/spamassassin start
```

Output:-

```
spamd 23516 9.1 1.1 28684 22944 ? Ss 20:12 0:00 /usr/bin/spamd -x -u spamd -H /  
home/spamd -d
```

If all has gone well, both Spamassassin and Clam Anti Virus should now be installed! With both of these programs installed.

Important Note: You will want to make sure that both clamd and Spamassassin are set to start up on boot. If you don't do this, your mail server will freak out the next time your reboot. On most Redhat systems, running the following command will allow you to edit the startup programs:

```
setup
```

One in the setup menu, scroll down and select the "**system services**" option. From the system services menu, scroll down to "clamd". Make sure that there is an asterisk (*) next to clamd, indicating it is set to start on boot. If no asterisk is present, press the spacebar to add it.

Repeat the procedure for Spamassassin. Exit out of the setup menu and you should be good to go.

Now we can now install Qmail-Scanner...

Part 15 - qmail-scanner w/qms-analog

If you will recall, when we compiled qmail earlier in this installation, we applied a patch to qmail called "qmailqueue.patch". This patch allows qmail to be configured to run with a substitute queuing mechanism. That's exactly what we're about to do here. We're going to tell qmail to use Qmail-Scanner as the queuing mechanism. Qmail-scanner is going to allow us to integrate Clam Antivirus and SpamAssassin into our qmail server's mail queue. Once qmail-scanner is installed, there will be a master script that is filled with configuration options that help you to tailor the functionality of Clam Antivirus and SpamAssassin to your needs. To expand the number of configuration options, we are also going to apply a patch to qmail-scanner. For this patch, we will be using [Mark Teel's qms-analog patch](#). Qms-analog incorporated the widely used qmail-scanner-st patch but it also adds some cool reporting functionality as well which we will utilize later in this installation guide. So let's get on it!

```
cd /downloads/qmailrocks
```

Unpack qmail-scanner...

```
tar zxvf qmail-scanner-1.25.tgz
```

Now unpack qms-analog...

```
tar zxvf qms-analog-0.4.2.tar.gz
```

Install qms-analog itself. This will come in handy in the next step when we install Qmailanalog.

```
cd qms-analog-0.4.2
```

```
make all
```

Next, we copy needed qms-analog files to the qmail-scanner source directory...

```
cp qmail-scanner-1.25-st-qms-YYYYMMDD.patch /downloads/qmailrocks/qmail-scanner-1.25/
```

Now, let's apply the qms-analog patch...

```
cd /downloads/qmailrocks/qmail-scanner-1.25
```

```
patch -p1 < qmail-scanner-1.25-st-qms-YYYYMMDD.patch
```

Now we will configure qmail-scanner and install it. So let's do it...

First, you need to configure the script for your needs...

```
cd /downloads/qmailrocks/qmail-scanner-1.25
```

```
vi qms-config
```

You will notice several fields that need to be customized to fit your needs.

```
#!/bin/sh
if [ "$1" != "install" ]; then
INSTALL=
else
INSTALL="--install"
fi
./configure --domain yourdomain.com \
--admin postmaster \
--local-domains "yourdomain.com,yourotherdomain.com" \
--add-dscr-hdrs yes \
--dscr-hdrs-text "X-Antivirus-MYDOMAIN" \
--ignore-eol-check yes \
--sa-quarantine 0 \
--sa-delete 0 \
--sa-reject no \
--sa-subject ":SPAM:" \
--sa-delta 0 \
--sa-alt yes \
--sa-debug no \
--notify admin \
"$INSTALL"
```

Now save and exit out of the config file. That was easy, wasn't it.

And now we will run a test config for qmail-scanner...

```
chmod 755 qms-config
./qms-config
```

Answer YES to all questions. If you get no errors, you can then run the script in "install" mode and this will install qmail-scanner on your server.

If you didn't get any errors on the test run above, then you should be ok to run the "real" installation script below. If it shows error, please scroll down and watch out for solution, So let's do it...

```
./qms-config install
```

Again, answer YES to all questions. If you get no errors, you can then run the script in "install" mode and this will install qmail-scanner on your server. If you do get errors, check out these [troubleshooting tips](#).

And now all that's left for qmail-scanner is to initiate the version file and the perlscanner database...

First, we'll initialize the version file. This command also helps to keep your server's /var/spool/qmailscan folder clear of rogue files that can develop when SMTP sessions are dropped. You may want to stick this command into your server's crontab and run it once a day. You'll see more on this in the "maintaining your qmail server" step near the end of this tutorial.. So let's run it...

```
setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -z
```

And now we will generate a new perlscanner database for qmailp-scanner. For future reference, it's a good idea to run this next command whenever you upgrade qmail-scanner. You'll see more on this in the "maintaining your qmail server" step near the end of this tutorial. So let's do it...

```
setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -g
```

A successful database build should produce the following output:

```
perlscanner: generate new DB file from /var/spool/qmailscan/quarantine-attachments.txt  
perlscanner: total of 9 entries.
```

And now one final ownership check...

```
chown -R qscand:qscand /var/spool/qmailscan
```

Woohoo, qmail-scanner is installed! Now it's time to tie qmail-scanner into qmail itself.

```
vi /var/qmail/supervise/qmail-smtpd/run
```

To instruct Qmail to use Qmail-Scanner as the alternative queuing mechanism, we add the following line to the SMTP "run" script right under the first line (#!/bin/sh):

```
QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue.pl" ; export QMAILQUEUE  
..and we change the "softlimit" in that same script...
```

```
change softlimit to 40000000
```

Note: It is absolutely vital that you change the "Softlimit" setting in this script. If you don't, qmail may fail to deliver mail!!!

So now the qmail-smtp/run file should look like this:

```
#!/bin/sh  
QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue.pl" ; export QMAILQUEUE  
QMAILDUID=`id -u vpopmail`  
NOFILESGID=`id -g vpopmail`  
MAXSMTPD=`cat /var/qmail/control/concurrencyincoming`  
LOCAL=`head -1 /var/qmail/control/me`  
if [ -z "$QMAILDUID" -o -z "$NOFILESGID" -o -z "$MAXSMTPD" -o -z "$LOCAL" ];  
then  
echo QMAILDUID, NOFILESGID, MAXSMTPD, or LOCAL is unset in  
echo /var/qmail/supervise/qmail-smtpd/run  
exit 1  
fi  
if [ ! -f /var/qmail/control/rcpthosts ]; then  
echo "No /var/qmail/control/rcpthosts!"  
echo "Refusing to start SMTP listener because it'll create an open relay"  
exit 1  
fi  
exec /usr/local/bin/softlimit -m 40000000 \
```



```
/usr/local/bin/tcpserver -v -R -l "$LOCAL" -x /etc/tcp.smtp.cdb -c "$MAXSMTPD" \
-u "$QMAILDUID" -g "$NOFILESUID" 0 smtp \
/var/qmail/bin/qmail-smtpd your_domain.com \
/home/vpopmail/bin/vchkpw /usr/bin/true 2>&1
```

Once you've got the qmail-smtpd file modified, save the changes and exit from the file. Now we will finalize the qmail-scanner installation by going over some post-install configuration options. After that, we'll fire everything up and take qmail-scanner for a test drive!

If the config test produced any error messages, stop here!

If you got a "setuid" related error when you ran the above script it is most likely due to you Perl installation not supporting setuid functionality.

Testing suid nature of /usr/bin/perl...

Whoa - broken perl install found.

Cannot even run a simple script setuid

**You will either have to correct this
or use the C-wrapper within the ./contrib dir**

Error was:

Can't do setuid (cannot exec sperl)

Redhat hosts need to have perl-suidperl installed to get setuid support

See FAQ for further details

Hit RETURN to continue, Ctr-C to exit and fix:

If your Perl installation does NOT allow for setuid functionality:-

```
cd /downloads/qmailrocks/qmail-scanner-1.25/contrib  
make install
```

Now we will customize the qmail-scanner configuration script...

```
cd /downloads/qmailrocks/qmail-scanner-1.25  
vi qms-config-cwrapper
```

You will notice several fields that need to be customized to fit your needs. Let's have a look.

```
#!/bin/sh  
if [ "$1" != "install" ]; then  
INSTALL=  
else  
INSTALL="--install"  
fi  
./configure --domain yourdomain.com \  
--admin postmaster \  
--local-domains "yourdomain.com,yourotherdomain.com" \  
--add-dscr-hdrs yes \  
--dscr-hdrs-text "X-Antivirus-MYDOMAIN" \  

```

```
--ignore-eol-check yes \  
--sa-quarantine 0 \  
--sa-delete 0 \  
--sa-reject no \  
--sa-subject ":SPAM:" \  
--sa-delta 0 \  
--sa-alt yes \  
--sa-debug no \  
--notify admin \  
--skip-setuid-test \  
"$INSTALL"
```

Now save and exit out of the config file. That was easy, wasn't it.
And now we will run a test config for qmail-scanner...

```
chmod 755 qms-config-cwrapper  
./qms-config-cwrapper
```

Answer YES to all questions. If you get no errors, you can then run the script in "install" mode and this will install qmail-scanner on your server. If you do get errors, check out these [troubleshooting tips](#).

```
./qms-config-cwrapper install
```

Again, answer YES to all questions. If you get no errors, you can then run the script in "install" mode and this will install qmail-scanner on your server. If you do get errors, check out these [troubleshooting tips](#).

```
vi /var/qmail/bin/qmail-scanner-queue.pl
```

Then change the first line of /var/qmail/bin/qmail-scanner-queue.pl to "#!/usr/bin/perl (in other words, remove the "-T" from the perl call.)

```
chmod 0755 /var/qmail/bin/qmail-scanner-queue.pl
```

And now all that's left for qmail-scanner is to initiate the version file and the perlscanner database...

First, we'll initialize the version file. This command also helps to keep your server's /var/spool/qmailscan folder clear of rogue files that can develop when SMTP sessions are dropped. You may want to stick this command into your server's crontab and run it once a day. You'll see more on this in the "maintaining your qmail server" step near the end of this tutorial.. So let's run it...

```
/var/qmail/bin/qmail-scanner-queue -z
```

And now we will generate a new perlscanner database for qmailp-scanner. For future reference, it's a good idea to run this next command whenever you upgrade qmail-scanner. You'll see more on this in the "maintaining your qmail server" step near the end of this tutorial. So let's do i t...

```
/var/qmail/bin/qmail-scanner-queue -g
```

A successful database build should produce the following output:

```
perlscanner: generate new DB file from /var/spool/qmailscan/quarantine-attachments.txt
```

```
perlscanner: total of 81 entries.
```

```
And now one final ownership check...
```

```
chown -R qscand:qscand /var/spool/qmailscan
```

Woohoo, qmail-scanner is installed! Now it's time to tie qmail-scanner into qmail itself.

```
vi /var/qmail/supervise/qmail-smtpd/run
```

To instruct Qmail to use Qmail-Scanner as the alternative queuing mechanism, we add the following line to the SMTP "run" script right under the first line (#!/bin/sh):

```
QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue" ; export QMAILQUEUE
```

..and we change the "softlimit" in that same script...

```
change softlimit to 40000000
```

Note: It is absolutely vital that you change the "Softlimit" setting in this script. If you don't, qmail may fail to deliver mail!!!

So now the qmail-smtp/run file should look like this:

```
#!/bin/sh
QMAILQUEUE="/var/qmail/bin/qmail-scanner-queue" ; export QMAILQUEUE
QMAILDUID=`id -u vpopmail`
NOFILESGID=`id -g vpopmail`
MAXSMTPD=`cat /var/qmail/control/concurrencyincoming`
LOCAL=`head -1 /var/qmail/control/me`
if [ -z "$QMAILDUID" -o -z "$NOFILESGID" -o -z "$MAXSMTPD" -o -z "$LOCAL" ];
then
echo QMAILDUID, NOFILESGID, MAXSMTPD, or LOCAL is unset in
echo /var/qmail/supervise/qmail-smtpd/run
exit 1
fi
if [ ! -f /var/qmail/control/rcpthosts ]; then
echo "No /var/qmail/control/rcpthosts!"
echo "Refusing to start SMTP listener because it'll create an open relay"
exit 1
fi
exec /usr/local/bin/softlimit -m 40000000 \
/usr/local/bin/tcpserver -v -R -l "$LOCAL" -x /etc/tcp.smtp.cdb -c "$MAXSMTPD" \
-u "$QMAILDUID" -g "$NOFILESGID" 0 smtp \
/var/qmail/bin/qmail-smtpd your_domain.com \
/home/vpopmail/bin/vchkpw /usr/bin/true 2>&1
```

Once you've got the qmail-smtpd file modified, save the changes and exit from the file. Now we will finalize the qmail-scanner installation by going over some post-install configuration options. After that, we'll fire everything up and take qmail-scanner for a test drive!

To activate all the changes we just made, we're going to have to completely stop and restart qmail.

Stop, Start, and Stat it...

```
qmailctl stop  
qmailctl start  
qmailctl stat
```

Now it's time to test the whole damn thing to see if Qmail-Scanner, Spamassassin and Clam AV are all working correctly. Fortunately, Qmail-Scanner comes with it's own testing script that does a fantastic job. So let's test it!

```
cd /downloads/qmailrocks/qmail-scanner-1.25/contrib  
chmod 755 test_installation.sh  
./test_installation.sh -doit
```

A successful test should produce the following output. 2 messages should be quarantined by Clam Antivirus in /var/spool/quarantine/new and 2 messages should be set to whatever mailbox you specified in the Qmail-scanner configuration script.

If you get 2 messages in your inbox and you see 2 messages in the quarantine folder, it's time to crack open a cold one! You've successfully installed all 3 packages! Woohoo!

Part 16 - QmailAnalog w/qlogtools & qms-analog

Qmailanalog performs some basic log analysis on those qmail log files and then outputs them to a desired location. In my case, I run qmailanalog every night and output the results to e-mail. Along with qmailanalog, we're going to install the "qlogtools" package. Qlogtools, as its name implies, provides an array of tools which can be used to analyze the qmail logs.

We're going to use one of the qlogtool packages, tai64n2tai, to convert the timestamps on the log files from a machine readable format to a human readable format which will come to us when we get the finished report. After we've installed both Qmailstats and Qlogtools, we will create a script which you can run on a nightly basis to generate e-mail stats. The script will also incorporate qms-analog, which we installed with qmail-scanner previously. The qms-analog output will give add qmail-scanner stats to our nightly report.

First, let's install qmailanalog...

```
cd /downloads/qmailrocks/
```

```
tar zxvf qmailanalog-0.70.tar.gz
```

```
cd qmailanalog-0.70
```

```
patch < /downloads/qmailrocks/patches/0.70-errno.patch  
make && make setup check
```

That's it. Qmailanalog is installed!
Now let's install qlogtools...

```
cd /downloads/qmailrocks/
```

```
tar zxvf qlogtools-3.1.tar.gz
```

```
cd qlogtools-3.1
```

```
patch < /downloads/qmailrocks/patches/qlogtools_errno.patch
```

```
mkdir /usr/local/man (if directory already exists, you're good to go)
```

```
make
```

```
./installer
```

OK. The qlogtools library of tools should now be installed.

Now we will implement a script to run Qmailanalog and then you can hook that script into the server's crontab to get stats generated every night.

The script below is a solid script that sends an email to the server administrator with both the qmailanalog output as well as qms-analog's readout of qmail-scanner's activities.

```
cp /downloads/qmailrocks/qms-analog-0.4.2/qmailstats /var/qmail/bin
```

```
vi /var/qmail/bin/qmailstats
```

```
## build the email message header
echo "To: your_postmaster@yourdomain.com" > $EMAILMSG
echo "From: your_postmaster@yourdomain.com" >> $EMAILMSG
echo "Subject: Nightly Qmail Stats Report for $DATE" >> $EMAILMSG
echo "" >> $EMAILMSG
echo "" >> $EMAILMSG
```

Now set the script executable...

```
chmod 750 /var/qmail/bin/qmailstats
```

Now run the script...

```
/var/qmail/bin/qmailstats
```

Check your email and you should get a report with some pretty cool stuff in it!

OK, if the qmailstats script is working, you will now want to create a crontab entry to run this script every night.

So, as the "root" user let's set up a cron entry...

```
crontab -e
```

```
0 3 * * * /var/qmail/bin/qmailstats 1>/dev/null 2>/dev/null
```

Save and exit from the crontab editor and you should be all set. The above entry will run the qmailstats script every night at 3:00AM.

Part 17 - Installing Qtrap

Our final ingredient in this installation is going to be a domain level word filter, which I've named "Qtrap". This script is applied on a per domain basis and serves as a "bad word" scanner to catch any spam that Spamassassin may have missed. This filter serves as the last defense against SPAM before it arrived in your inbox. I like this filter because it helps to get rid of any SPAM that happens to make it by Spamassassin.

Any emails that are scanned and contain a banned word will be automatically deleted and logged by the qtrap script. A whitelist feature now exists so that individual addresses or domains can be exempt from the qtrap scan.

So let's install it...

```
cd /home/vpopmail
```

```
mkdir -p qtrap/logs
```

```
cd qtrap
```

```
cp /downloads/qmailrocks/scripts/qtrap/qtrap-2.0.0 ./qtrap.sh
```

Defining your whitelist:

```
vi qtrap.sh
```

You will see a block of code for the whitelist that looks like this:

```
whitelist_check () {  
case $WHITELIST in  
address@somewhere.com|address@somewhereelse.com|*entiredomain.com)  
echo $SENDER found in whitelist on `date "+%D %H:%M:%S" >>  
/home/vpopmail/qtrap/logs/qtrap.log  
exit 0;;  
*)  
;;  
esac  
}
```

The email addresses in the **bold red text** above should be substituted with any email addresses that you wish to whitelist against the qtrap filter process. Whitelisted addresses will be allowed to send you mail that contains "banned" words. Un-whitelisted address will be scanned and their message deleted if it contains a banned word. As you can see above, you can specify an individual address (address@somewhere.com) or you can simply whitelist an entire domain (*entiredomain.com).

Defining your "banned word" list:

within the qtrap.sh script you should see another section, below the whitelist section of code, that looks like this:

```
checkall () {  
case $BANNED_WORDS in  
porn|PORN|Sex|SEX)  
printout $BANNED_WORDS
```

```

echo MESSAGE DROPPED from $SENDER because of $BANNED_WORDS on `date "+%D %H:
%M:%S" >> /home/vpopmail/qtrap/logs/qtrap.log
exit 99;;
*)
;;
esac
}

```

Now let's set up the logging directory...

```
touch /home/vpopmail/qtrap/logs/qtrap.log
```

```
chown -R vpopmail:vchkpw /home/vpopmail/qtrap
```

```
chmod -R 755 /home/vpopmail/qtrap
```

Now we will add this script into the mail path for a domain on our server.

```
cd /home/vpopmail/domains/yourdomain.com
```

```
vi .qmail-default
```

add the following line above the line that is already there

```
| /home/vpopmail/qtrap/qtrap.sh
```

Here's an example:

.qmail-default before:

```
| /home/vpopmail/bin/vdelivermail " delete
```

.qmail-default after:

```
| /home/vpopmail/qtrap/qtrap.sh
| /home/vpopmail/bin/vdelivermail " delete
```

Save these changes and that should be it. You don't have to restart anything. To test this last rule, try sending an e-mail to your mailbox and make sure that the test e-mail contains one of the words that you entered into the "bad word" list in the Qtrap script. If the filter is working right, the message should NOT arrive in your inbox. You should then be able to view the log file at /home/vpopmail/qtrap/logs/qtrap.log and see a log of the dropped message corresponding to the time at which you sent the test message. The drop log should look something like this:

```

MESSAGE DROPPED from someone@somewhere.com because of some_banned_word on on
06/13/03 02:37:51

```

If the test was successful, then that's it! Congratulations, you've completed the Qmail installation. Have fun. The next couple steps discuss cleanup as well as some closing notes and suggestions.

Part 18 - Maintaining your qmail server

Making sure that all services start on boot

If you've installed qmail correctly, it should already automatically start when you boot your server. However, you will want to make sure that all of the other needed services start as well.

--For Redhat users--

Starting Courier-imap on boot - make sure the following 2 lines exist in your server's /etc/rc.local file:

```
/etc/rc.d/init.d/imap start  
/etc/rc.d/init.d/imapd start
```

Starting SpamAssassin on boot

Try running the "setup" command and check the system services area to see if SpamAssassin is listed and marked with a "*" to start on boot. If SpamAssassin is not present in the system's "setup" tool, you can start it on boot by adding the following line to the server's /etc/rc.local file

```
/etc/rc.d/init.d/spamd start
```

Starting Apache on boot

Try running the "setup" command and check the system services area to see if the "httpd" service is listed and marked with a "*" to start on boot. If it is not present in the system's "setup" tool, you can start it on boot by adding the following line to the server's /etc/rc.local file

```
/etc/rc.d/init.d/httpd start
```

Maintaining the qmail queue

It's usually not a bad idea to keep your qmail server's queue in check. Your qmail server's queue is located at /var/qmail/queue. However, it's just about impossible to look directly at the queue folders and be able to tell what's going on.

qmailctl stat - This, of course, shows you your qmail server's current status. Included in the stats is the current condition of the queue. You can use this to see how many messages are sitting in the queue.

qmHandle - This add-on tool allows more in depth viewing of the queue and it also allows you to perform administrative functions on the queue. You can find instructions about installing qmHandle [here](#). Once qmHandle is properly installed, running the command "qmHandle" will provide a listing of all the possible commands possible. I'll go over a couple of them right now. The "qmHandle -l" command will give you a complete listing all every

message in the queue and a summary containing the date the message was sent, the sender and the intended recipient. The output for a single message might look like this:

6406395 (195, R)
Return-path: bob@somewhere.com[]
From: Bob Smith <bob@somewhere.com>
To: Frank Smith <frank@wherever.com>
Subject: Re: This weekend
Date: Mon, 16 Feb 2004 12:14:31 -0700
Size: 1482 bytes

The message number, 6406395, can be seen at the top of the message. If we wanted to delete this single message from the queue, we could do so with a command of "qmHandle -d6406395". The entire queue can be cleared out with the command "qmHandle -D".

queuelifetime - The "queuelifetime" setting for qmail determines how long messages can stay in the queue. By default, your qmail server will keep messages in the queue for 604,800 seconds, or 7 days. However, you can set a custom queuelifetime by creating a file called /var/qmail/control/queuelifetime. The content of that file is a single line containing a number which represents the number of seconds the queue will hold any given message. If you want to set a custom queuelifetime, you might want to whip out the old calculator.

A little advice on handling your queue

You need to keep in mind that a queued message is NOT necessarily a BAD thing. The only time a message is queued for any length of time is when it is undeliverable at the time it is originally sent. A message is usually undeliverable for 1 of 2 reasons:

- 1) The receiving mail server is offline. If this is the case, when that mail server comes back online at a later time your qmail server will then be able to deliver that message. A perfect example of this is the Qmailrocks.org mailing list. At any given time there lots of messages in the mailing list server's queue. However, these messages usually get delivered eventually. As you can imagine, some people on the mailing list are probably using a new qmail server to house the mailbox with which they have signed up on the list with. Well, of course, since that person is new the qmail there is a chance that their server may go down for periods of time while they are working on it and perfecting their qmail skills. It is at these times when their server is unreachable and when my list server then queues the message for later delivery. No big deal.
- 2) The recipient address is bogus or incorrect. If you get hit with a lot of spam on your server, or if you have a spammer on your server, this will probably happen to you at some point. Your queue gets filled with message bound for bogus addresses OR it gets filled with bounce messages that your server is trying to deliver back to spammers who sent spam to bogus addresses on your server. A good way to cut down on this is to set the domains on your server to "delete" catchall mail instead of bouncing it. This can be done from within the Qmailadmin interface. Setting a domain's catchall setting to "bounce" is a bad idea in my experience as it only results in a queue full of bounced bounce messages. If you don't need a catchall for your domain, do yourself a favor and set it to "delete". This is also the case if you are running Qmail-scanner. Qmail-scanner has an option to notify the sender when a virus is found in an e-mail. Bad idea. Most of the time, the address from which that virus laden e-mail came is

bogus. So trying to "notify" the sender usually results in nothing but a bunch of double bounces flying all over your server.

Fortunately, since your qmail server has a built in queue lifetime, messages will eventually drop out of the queue if they are undeliverable. The last thing I want to mention is that it is a common misconception that if your queue is full, mail being sent presently will not be delivered quickly. This is a misconception. As I said, the queue is a repository for messages that are not immediately deliverable. If a message is being sent to a valid address it will get sent immediately, regardless of the size of the queue. So, in summary, you don't need to panic every time you have messages in your queue. Most of the time, the best thing to do is to just let your queue take care of itself. However, there are time when you may deem it necessary to clean out the queue or take other administrative action, and that's what the "qmHandle" tool above is good for.

Maintaining qmail-scanner

Qmail-scanner is fairly easy to maintain. Once you get it configured how you want it to be, the main items your going to want to worry about are 1) the log files and 2) the virus quarantine area. Here's some info on both.

1) qmail-scanner logs - Qmail-scanner, when Clam Anti Virus and SpamAssassin are hooked into it, logs the virus scanning activities to /var/spool/qmailscan/qmail-queue.log. This log file can get REALLY big, so you will want to keep it in check. You may want to set up a rotation schedule for this log file or some other sort of scheduled task that deals with this log file on a routine basis. If this log file exceeds the linux file size limit of 2GB, your mail server will start freaking out and all hell will break lose. So do yourself a favor and keep an eye on this log file.

2) The virus quarantine area - When qmail-scanner pipes the mail out to Clam Anti Virus and virus is found, the virus laden message is quarantined at /var/spool/qmailscan/quarantine/new. Those e-mails will usually not pose any threat to your server, since there are very few Linux/Unix viruses and since they are not being executed. However, on a busy mail server, that folder can get filled up pretty quickly, so you may want to keep an eye on it and have it emptied on a routine basis. I empty mine out with a crontab that runs once a week.

Anytime you upgrade qmail-scanner, it's a good idea to refresh the qmail-scanner perlscanner database:

Redhat/Fedora/RHEL:

```
setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -g or /var/qmail/bin/qmail-scanner-queue -g (for non setuid setups)
```

Maintaining SpamAssassin

SpamAssassin is relatively maintenance free one you get it up and running. It logs it's activities to /var/log/maillog, so you can always reference the logs for any investigations. New versions of SpamAssassin are released fairly often, so you may want to occasionally check <http://www.spamassassin.org> to see if there have been any new releases. In my experience, I've always been able to install the new version over the older version with no

problems. If you upgrade, just make sure the `/etc/mail/spamassassin/local.cf` file still contains the setting you want and you should be in good shape.

Anytime you upgrade SpamAssassin, it's a good idea to refresh the qmail-scanner perlscanner database:

Redhat/Fedora/RHEL:

```
setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -g or /var/qmail/bin/qmail-scanner-queue -g (for non setuid setups)
```

Maintaining Clam Antivirus

The only things you really need to do to maintain Clam AV are:

1) Make sure your server is running the automatic virus definition updates on a regular basis. I run the following command out of crontab on a routine basis:

```
/usr/bin/freshclam --quiet -l /var/log/clamav/clam-update.log
```

2) Keep your version relatively current. New releases are put out fairly often, so keep an eye on it. Again, in my experience I've been able to install the newer version over the older one with no problems.

3) Anytime you do decide to upgrade Clam Antivirus, you will need to update qmail-scanner's version file. This is easily done by running the following command:

Redhat/Fedora/RHEL:

```
setuidgid qscand /var/qmail/bin/qmail-scanner-queue.pl -z or /var/qmail/bin/qmail-scanner-queue -z (for non setuid setups)
```

Maintaining the qmail logs

Fortunately, qmail's logs take care of themselves. They automatically rotate all on their own, so you never have to worry about them. The only thing worth noting is that you CAN customize the rotation schedule for the logs. This is done the "logs" supervise script for each supervised function. Confused? Ok, I'll explain.

In the QMR qmail installation, there are 3 supervise scripts and, subsequently, 3 logs for those scripts.

```
/var/qmail/supervise/qmail-pop3d/run operated the pop server and is logged via /var/qmail/supervise/qmail-pop3d/log/run
```

```
/var/qmail/supervise/qmail-smtpd/run operates the smtp server and is logged via /var/qmail/supervise/qmail-smtpd/log/run
```

```
/var/qmail/supervise/qmail-send/run operates the mail processor and is logged via /var/qmail/supervise/qmail-send/log/run
```

Each of the above "log/run" scripts tells the server how it wants those activities logged. Let's take the /var/qmail/supervise/qmail-pop3d/log/run logging script as an example:

```
#!/bin/sh
PATH=/var/qmail/bin:/usr/local/bin:/usr/bin:/bin
export PATH
exec setuidgid qmaill multilog t s100000 n20 /var/log/qmail/qmail-pop3d 2>&1
```

The last line of this script sets some of the logging options. We can break that last line into several parts:

```
exec setuidgid qmaill multilog t s100000 n20 /var/log/qmail/qmail-pop3d 2>&1
```

So let's break it down:

exec setuidgid qmaill multilog - run the multilog program as the "qmail" user.

t - inserts an @, a precise timestamp, and a space in front of each line.

The above "t" is why the rotated logs take on names like this:

@40000000402d1c562cbf3534.s

s100000 - logs will rotate when they reach 100000 bytes.

n20 - number of rotations to keep on hand before they fall off.

/var/log/qmail/qmail-pop3d 2>&1 - the directory to where the logs will be written, silently.

Backing up your qmail server

Backing up a qmail server is relatively easy. While different people may give you slightly different recommendations, you can ensure a safe backup of your qmail server if you backup the following 2 directories on a routine basis.

/home/vpopmail - backs up all your domain information, including mailboxes, passwords and the messages themselves.

/var/qmail - backs up all of your qmail settings. The /var/qmail/control directory is the most important directory in there to back up, but it won't hurt to just back up the whole damn qmail directory.

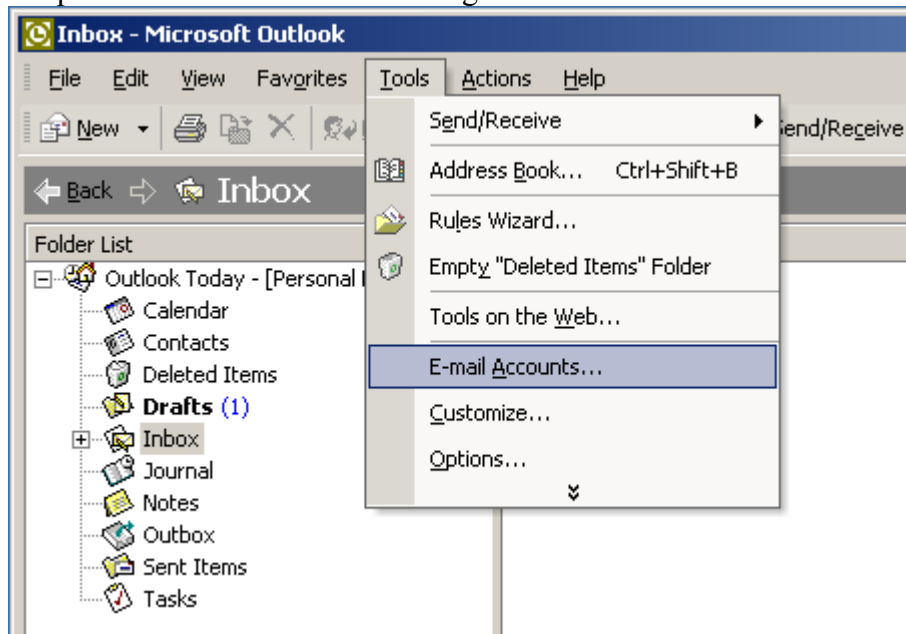
Part 19 - Mail Client Configuration

So now you've got a fully functioning mail server. Well, now it's time to make use of that mail server and configure a remote mail client to retrieve mail from the server and send mail through the server. For my example, I've chosen Microsoft's Outlook XP. Outlook Express varies slightly, but you should be able to get the general idea. Likewise, if you using some other mail client you should be able to get the idea. Please do not e-mail me asking for instructions on another mail client. This is the only one I'm going to provide for now. Anyway, let's configure Outlook...

For this example, let's set the following pre-conditions:

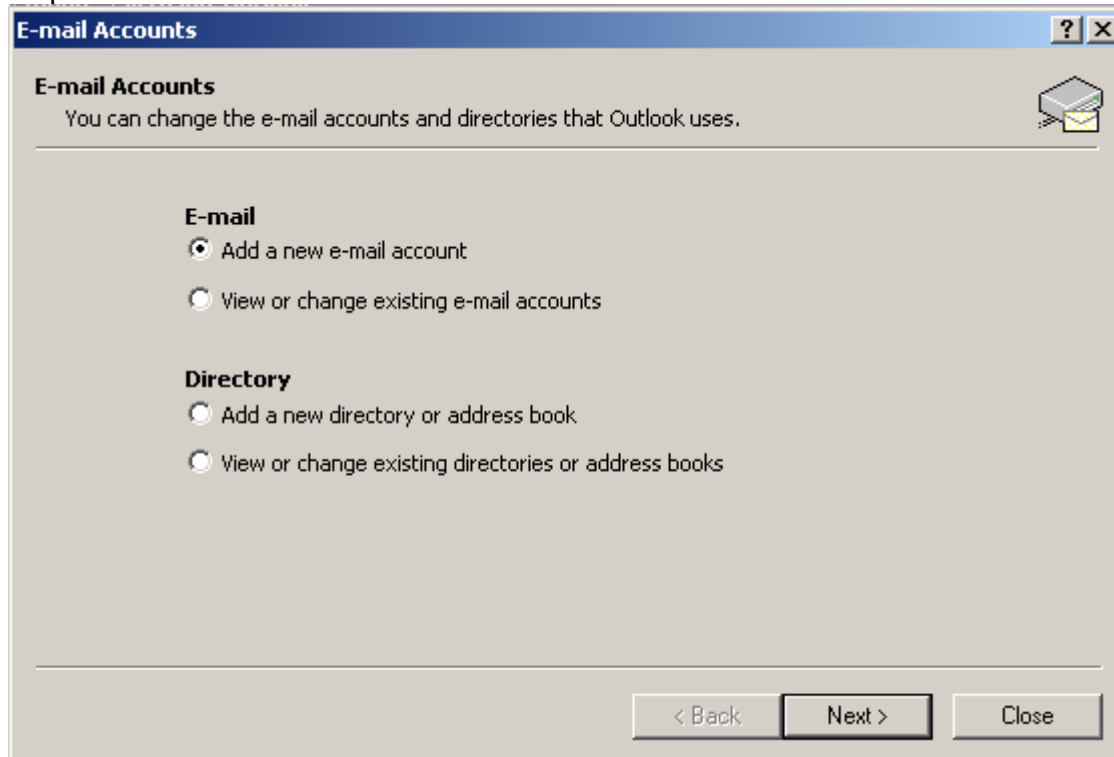
1. I have used Vqadmin to set up a new domain on my server called "domain.com"
2. Since I have set the domain up using Vqadmin, there should already be a main user for the domain called "postmaster@domain.com".
3. I'm going to set up Outlook to check mail for and "postmaster@domain.com" and will also use that user for SMTP authentication, so that I can send mail.

Step 1: Go to E-mail account configuration



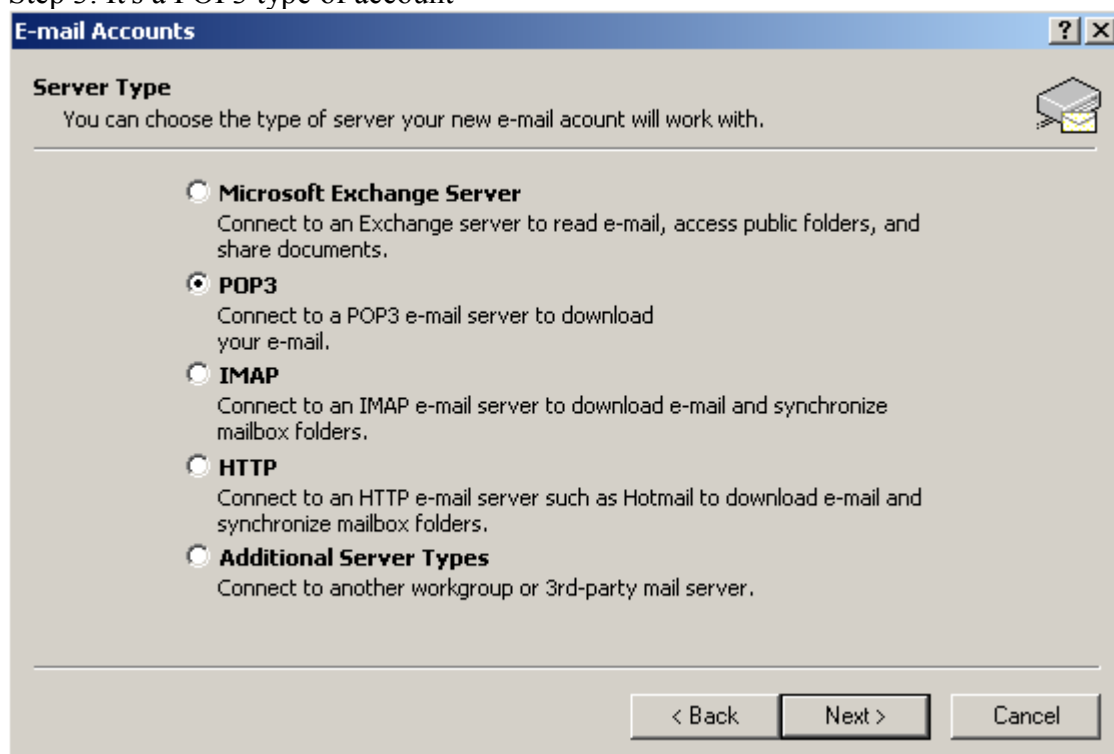
Select "E-mail Accounts" under Tools.

Step 2: Add a new E-mail account



When finished, click "Next".

Step 3: It's a POP3 type of account



When finished, click "Next".

Step 4: Configure the account

E-mail Accounts [?] [X]

Internet E-mail Settings (POP3) [Envelope Icon]

Each of these settings are required to get your e-mail account working.

User Information

Your Name:

E-mail Address:

Server Information

Incoming mail server (POP3):

Outgoing mail server (SMTP):

Logon Information

User Name:

Password:

Remember password

Log on using Secure Password Authentication (SPA)

Test Settings

After filling out the information on this screen, we recommend you test your account by clicking the button below. (Requires network connection)

Click on "More Settings" once you've filled out the above information.

Step 5: Enable SMTP Authentication (under the "more settings" area):

Internet E-mail Settings [?] [X]

General | **Outgoing Server** | Connection | Advanced

My outgoing server (SMTP) requires authentication

Use same settings as my incoming mail server

Log on using

User Name:

Password:

Remember password

Log on using Secure Password Authentication

Log on to incoming mail server before sending mail

Save all changes and give it a go! You should be all set.

FeedBack

Please post your valuable feedback at friendlyogi@gmail.com