POP-Toaster using Qmail, Vmailmgr, Courier, and Squirrelmail

Konstantin Ryabitsev

POP-Toaster using Qmail, Vmailmgr, Courier, and Squirrelmail

by Konstantin Ryabitsev

Published March 18, 2005

Abstract

This document is useful for people who are looking to set up an e-mail system with an easy-to-use client webmail front end and support for name-based virtual domains. It proposes a QVCS tie-in as the best small to mid-class server solution. This document is written for Centos systems running 3.

Table of Contents

Chapter 1. Introduction

Say, you are looking to start your own small-to-medium hosting business and you need to come up with the best solution for an e-mail server. The things you are looking for are:

- Security
- Reliability
- Lax hardware requirements
- Support for many virtual hosts, all sitting on one IP address (name-based hosting)
- SMTP relaying for your clients
- POP3 and IMAP mailbox access
- A nice webmail front-end for your clients

I was facing the same problem and I have found that one of the best solutions would be to use QVCS tie-in. It is very easy to configure, runs very reliably, and has very good security features.

This guide will help you configure and set up a similar system.

OSes, Packages, and Disclaimers

There are many flavors of UN*X software out there and it's very hard to write a uniform document that would work for every distribution. This guide is aimed at Centos, and is based on a set of binary packages provided with it. If you are running something other than Centos 3, or like building stuff from source, please consult the following document: http://megaz.arbuz.com/?p=qmail_howto [http://megaz.arbuz.com/?p=qmail_howto]

Important

A special word about the binaries. *You must understand, that although I have put a lot of effort into making and troubleshooting these packages, no guarantee WHATSOEVER is given that they will work for you. There is no warranty, no assurance, not even an implication that these packages are suitable for the task described in this document.* Having said that (this is a standard disclaimer, really. :)), I am quite convinced that a lot of people will find these packages quite useful and suitable.

Chapter 2. Installation: From Zero to 60 in 30 minutes

This section will walk you through a generic Centos installation process. The system we are going to install is going to aim *exclusively* at being a mail server running nothing but virtual servers and webmail interface (so-called "pop-toaster"). If you are planning to use your system for any other services, you can still glance through this installation part for hints and caveats, but your install will differ from the one outlined below.

Considering the hardware

This setup is aimed at low- to middle-low-end installations, hence we will be VERY relaxed about our hardware requirements. Nevertheless, there are several important things to consider. First of all, you need to make sure that your server is capable of handling peak loads, such as happen at times when a new outlook virus hits the Internet. Another thing to consider is how many clients you are planning to support, together with how much maximum space you are going to allow them to have.

Overall, we are looking at three different variables -- memory amount, processor speed, and hard drive space. Let's consider a setup with *500* clients max and look at each of these three variables.

HDD space

A sensible amount of mail quota to allow per client would be about 50Mb, so the amount of hard drive space you will require just for 500 clients worth of e-mails would be around 25Gb. That's not all, though, as it is also necessary to consider the amount of hard drive space needed for the mail queue.

Let's imagine that you have been hit with a virus that mails itself to a hundred people and all of your 500 clients got infected one way or the other. If the virus is around 100Kb in size, that means that the total amount of traffic a single client will generate will be around 10Mb. Multiply that by 500, and you will arrive at a staggering 5Gb of traffic just to handle that virus. Since qmail will spend a good deal of time making connections, you will want to make sure that there is plenty of space to queue all of these requests. What this means is that you will need to allow around 5-7Gb of space for queuing, which brings us to 30-35Gb of total space for the mail subsystem.

The OS itself will actually require very little space -- no more than 1Gb for everything, including virtual web-servers, preferences and other miscellaneous data.

After we allow about one more gigabyte for system swap space, we arrive at 35-40Gb overall HDD space needed for an installation with 500 clients. Re-calculate the requirements for your number of clients using the following formula:

HDD considerations

User space: N*50Mb
Qmail queue: N*10Mb
System and swap: 2GB

Whether you decide to choose SCSI or IDE is up to you, but you should consider that most common HDD activity will be accessing and moving small files, something that high-RPM SCSI drives do best. Depending on how redundant you want to be (which generally depends on how bemused your clients can get at the prospect of losing email or facing a significant downtime), you might consider creating a RAID array to mirror all your data.

If you do decide to go with RAID, then my advice would be to get 1 small IDE drive for the system, and 3 SCSI drives for a RAID-1 array (1 active, 1 mirror, and 1 spare). Granted, this setup will be more expensive, but believe me, you will sleep *much* better at night.

RAM requirements

The amount of RAM you will require depends on the number of simultaneous connections you are going to have to the server. This largely depends on the environment you are setting this up for.

If you are creating this setup for your company, then it's a good possibility that a good chunk of these 500 will be accessing your system simultaneously, especially around 9am in the morning when people first arrive at work and check their e-mail. If, however, you are an ISP and your clients are mostly home-users, then the amount of simultaneous connections your server is likely to experience would be lower, since people will tend to check their e-mail at various times during the day.

Let's approximate -- if you are setting up a server for your company, the likely peak usage would be around 90% of all your clients. The amount of memory each request will consume depends largely on what kind of connection it is -- smtp and imap require very small amounts of memory for each connection, within a few hundred kilobytes each (unless you are doing email filtering and virus checking, in which case the only official limit to the amount of memory you should put into your server is how much the motherboard can physically take. See the advanced section for more info). Webmail requests, however, are very memory-hungry and will likely gobble up a hefty chunk of RAM -- around 5Mb per each request. However, the good thing about webmail is that each request lasts only a few seconds, so even if 200 people decide to connect to your server at around the same time, it's unlikely that there will be any more than 50 http processes running simultaneously.

But let's be pessimistic and allow for freaky coincidences. Let's imagine that all of your 500 clients decided to connect to your server at roughly the same time, and our apache daemon spawned 150 processes, consuming 5Mb each. That brings the memory usage up to 750Mb. The system itself consumes about 50Mb of your memory, so at peak loads it will be consuming around 800Mb of RAM. If you want your server to be snappy at all times, you will need to have at least that much memory in your box, however, if you decide that such coincidence is not very likely and you'd rather save on extra memory, you can settle on 512Mb and let the swapping process catch the rest.

On the other hand, if you are an ISP with most clients being home-users, you are not likely to experience more than 10% of your clients trying to connect at the same time. The memory requirement would be more relaxed, and it is likely that 256Mb of memory will suffice for you. Nevertheless, it's always better to have more memory, than less, so you are still encouraged to use 512Mb for 500 clients.

In general, to calculate how much memory you will need use the following formulas:

RAM considerations

For a company: N/3*5+50
For an ISP with home-users: N/10*5+50

For 500 users these values will be 880Mb and 300Mb respectively. If you are going to rely on swapping, you can bring these values down to 512Mb and 256Mb.

CPU requirements

None of the processes are very CPU-intensive, actually, and you are not very likely to bottleneck at the processor level. Again, the only exception to this is if you are doing spam filtering or virus checking, which tends to tax the system very heavily both in terms of RAM and CPU. Overall, I would recommend using something like a 1.5 GHz and above system for 500 users, so the calculation formula would look something like so:

CPU Considerations

Lower end: N*1.5+800
Higher end: N*2+1000

I'm using the +800 method simply because I think that if you decide to use something less than a 800Mhz system, you are likely to be plagued by various problems related to aging hardware.

Other things

I am not covering networking environment and bandwidth, since you will likely have to stick with what you already have anyway. A common 100Base-T network card will suffice in terms of a NIC. However, you should consider implementing some sort of a backup solution to make sure that you don't lose your job or go out of business when your server catches on fire and you find it reduced to cinders when you come to work one lovely Monday morning. I have only good words to say about Amanda http://www.amanda.org/, or you may choose some of the many alternatives.

Refer to the "Backup" section further in the document for the list of directories to include in your backup run, and for restoring instructions.

Installing Centos 3

There are two ways to do it. One is to get installation CDs and go through the installation process yourself, and another one is to use kickstart for a network install of a cookie-cutter QVCS system. In any case you will need the following information.

Partitioning

You need to have at least 4 partitions: /, swap, /var, /home.

Use the calculations we just did in the previous section to come up with appropriate partition sizes, and create the "/home" partition last letting it use the rest of the remaining disk space. If you're making a RAID-1, utilize Disk Druid's nice RAID'ing features.

For our example, the partitions would look like so, for a 40Gb HDD:

```
/     - 1Gb
```

```
swap  - 1024Mb
/var  - 7Gb
/home - the rest
```

Installing Centos 3

Tip

This document uses http://ftp.osuosl.org/pub/centos/3/os/i386/ for the location of installable Centos media, but you can refer to the http://www.centos.org to see the list of available mirrors if this one is too slow or otherwise unsuitable for you.

This document doesn't cover the installation of Centos, but generally you can use several possible ways of installing: most commonly by installing from ISOs, or by doing it over the network.

Installing from ISOs

First you will need to download and burn the isos, which you can find on the Centos List of Mirrors [http://www.centos.org], then boot from the first binary ISO to start the installation process.

Installing from the Network

This installation process is similar to the previous one, except that you only have to download and burn a very small image called boot.iso. Download the boot.iso from the Centos mirror site [http://ftp.osuosl.org/pub/centos/3/os/i386//os/i386/images/], burn it, and boot from it.

Generic Installation Instructions

The install process is simple enough. Just follow the setup screens, paying attention to the partitioning scheme we have discussed above. When it gets to package installation select "Custom" and then *uncheck all groups in the selection list*. For this installation we only want the core of the operating system.

Once the installation is complete, reboot, login as root, and perform the following actions:

```
[root@mail root]# wget http://mirror.mricon.com/qvcs-guide/qvcs-init-centos

[root@mail root]# sh qvcs-init-centos
```

Qvcs-init will install the public keys, update your machine to the latest Centos errata for 3, download and install qmail source RPM (binary qmail RPMs cannot be distributed due to license restrictions), and then download and install QVCS related applications.

QVCS-install

Now, after the core of QVCS is installed, we need to run **qvcs-install** in order to configure the system for our purposes.

```
[root@mail root]# qvcs-install
```

This utility will configure the system software for some default settings, suitable for running the base QVCS install. The best thing about it is the fact that it will save backup copies of the files it overwrites into `/var/lib/qvcs` so you can always restore old configurations if you find it necessary.

Once this step is done, you are ready to configure your system for actual use.

Chapter 3. Basic Configuration

Let's go ahead and configure your system so it's suitable for your purposes.

Examples

For the sake of providing examples, I will be using the following virtual domains to make the narrative easier to follow: hogwarts.jk, theministry.jk, quibbler.jk (what Harry Potter addiction?).

Creating the first virtual domain

The first virtual domain requires some effort, but only relative to the others. Here is how to go about it.

Note

If you are getting "`command not found`" errors, make sure you are logged in as root. If you have used **su** to become root, make sure you use "**su -**" to enable the root environment.

```
[root@mail root]# addvirt hogwarts.jk
```

The **addvirt** script will ask you for a password. Remember it, as you will need it to enable the domain in vadmin. Make sure it's a good one, too, as it is a system password and though the account is marked as **/sbin/nologin** during creation, having poor passwords is one of the main reasons servers get cracked.

Now you need to create the first virtual user. To do that, you will need to switch to the domain "master user" and use the **vadduser** command to create the virtual account. If you look at the output of the **addvirt** command, you will notice something to the matter of "`Creating new domain user "hogwarts_jk"`. In the next command you will need to use the username reported by **addvirt** instead of "hogwarts_jk" (usually it just substitutes all dots for underscores in the domain name to arrive at the username). Oh, and make it something other than "albus," of course.

```
[root@mail root]# su -s /bin/bash - hogwarts_jk
[hogwarts_jk@mail hogwarts_jk]$ vadduser albus
[hogwarts_jk@mail hogwarts_jk]$ exit
```

Editing `/etc/vadmin/vadmin.conf`

Tip

The only editor that comes with your machine is **vi**. If it gives you the creeps, you can install **nano** by using yum. Nano is a successor to pico and inherits all of its shortcuts.

```
[root@mail root]# yum install nano
```

It is useful to know that calling nano with a "-w" flag will turn off automatic line wrapping. Good for files where you have to type a very long line without it wrapping:

```
[root@mail root]# nano -w filename.conf
```

Open `/etc/vadmin/vadmin.conf` in your editor and locate the `[auth]` section. Change the `elvis` parameter to reflect the virtual user that you have just added.

```
[auth]
   method = user
   force_https = yes
   elvis = albus@hogwarts.jk
```

Editing `/etc/httpd/conf.d/vadmin.conf`

This apache include file provides a secret hash string that will be used to encrypt your vadmin data. Right now it says "LLAMA" but go ahead and change it to something other than that. It can be any string of any length and contain any characters as long as they aren't quotes. Lines from your favorite songs or books are a good choice. For example:

```
<Directory "/usr/share/squirrelmail">
 SetEnv CRYPTO_HASH_LINE "Draco Dormiens Nunquam Titillandus"
 SetEnv MCRYPT_ALGO "rc4_builtin"
</Directory>
```

Tip

You can set the MCRYPT_ALGO to something other than "rc4_builtin" if you want stronger encryption than rc4. "Blowfish" is a good fast algorithm, but you may choose among the following: blowfish, twofish, tripledes, gost, serpent, and others. Consult libmcrypt documentation for more info.

What's in the name?

It is useful to check whether the qmail installer set your hostname correctly. Go into /etc/qmail/control and check what the file "me" says. It may be empty, or it may contain the FQDN of your server. You want to put the official name of your server in that file, e.g. "mail.yourisp.com" -- it should not remain empty, as that will cause some outgoing mail to bounce.

Reboot

Well, you're done! Reboot to enable the new configurations.

```
[root@mail root]# reboot
```

A note on DNS

DNS is not covered in this guide, but it would be as easy as pointing "mail.hogwarts.jk" to the IP address of your server. Same goes for all other mail.domainname.com settings -- as long as you point them at the IP address of your brand new QVCS system, you are set. Oh, and, of course, don't forget to **addvirt** them.

Tip

If you are just playing around with your system and don't feel like mucking with DNS quite yet, you can edit the resolver on your local computer to point to a certain IP address so your browser knows where to go. In Linux/UN*X this would be in /etc/hosts, while for windows the file is somewhere in C:\WINDOWS\system32. Google for "**/etc/hosts windows**" for more information.

Chapter 4. Administering your system

Logging in to Vadmin

Vadmin Plugin for Squirrelmail is a tool written to simplify mundane tasks such as adding and deleting users, activating domains, setting quotas, etc. To log in, surf to **`https://mail.hogwarts.jk/`** and log in as the user you have specified as `elvis` in vadmin configuration. Once you log in, click on "options" and find the "Administrator Interface" link presented somewhere on the page.

The administrator interface starts with a login screen. Type in your mailbox password (the same password you used to log in to Squirrelmail). The next screen will prompt you for the domain password -- it's the one you used when creating the virtual domain using the **addvirt** command. Once you submit the password, it will be stored on the server in an encrypted format.

Elvises, Admins, Cross-Admins, Oh My!

There are three levels of admins in Vadmin. There is a superuser (lovingly referred to as "elvis"), cross-admins, and "lowly" admins. Here are the main differences.

Elvis

Elvis has access to all virtual domains configured on the system -- it's the "root" in terms of system accounts. Elvis is also the only user who can administer cross-admins.

Cross-admins

Cross-admins are users who can administer more than one domain, just in case you have users who own several. Cross-admin setup tools in Vadmin allow you to set up who these users are and which domains they have access to.

Lowly Admins

This is the lowest form of administrators -- they can only administer one domain -- their own. You can give a user administrator privileges by checking "*can administer this domain*" in the "edit user" screen.

Domain Limits

This version of Vadmin introduces the option to limit how much control lower admins have over certain domains. For example, you as elvis can specify how many mailboxes there are allowed in a domain, how much maximum quota a user can have, how many messages they are allowed to have in their inbox, etc. There are two levels of domain limits -- the ones set up by an elvis, and another set up by a cross-administrator. The latter cannot override the master limits as specified by the superuser.

Note

If you are upgrading from an earlier version of Vadmin, note that domain limits do not apply retroactively. In other words, if you have a domain with users who have their quotas set to 200Mb, setting a domain-wide limit of 100Mb will not affect already existing accounts. Only when new accounts are created will the maximum quota limit be enforced.

Root Email

We need to set up the address for root, otherwise important system messages will go into the bit bucket. To do this, edit `/etc/aliases.qmail` and uncomment the last line, changing "mark" to some real address. Then do the following:

```
[root@mail root]# ln -s /etc/aliases.qmail /etc/aliases
[root@mail root]# newaliases
```

Remember to run **newaliases** every time you edit `/etc/aliases`, otherwise the system will be unaware of the changes. Also note that `/etc/aliases` can only be used for real users, not virtual users. Use vadmin to set up the aliases and forwards for the latter.

Removing domains

To remove domains, use "**rmvirt domainname.com**". It will optionally back up configurations for the domain before removing it entirely.

Automated Updates Using Yum

The tool we have used for installation -- yum is an automated installer/updater that is a free substitute for up2date. One of the most important aspects of running a server is keeping it constantly patched, so any security vulnerabilities are mitigated as soon as Red Hat issues fixes.

If your installation is more or less a vanilla setup of QVCS, then you might consider enabling automated nightly updates of your system, so any errata packages are applied as soon as they are released. To do so, run:

```
[root@mail root]# chkconfig yum on
[root@mail root]# service yum start
```

If you feel edgy about having an automated updater tool running on your system, you may leave auto-updating disabled, but then I would suggest putting a "yum check-update" run into your nightly cron run. The following will notify the root user whenever there are updates available for the system:

```
[root@mail root]# echo "yum -d 0 check-update" > /etc/cron.daily/yum-check.cron

[root@mail root]# chmod a+x /etc/cron.daily/yum-check.cron
```

Evaluate any updates and apply them. Don't let your server become a part of the sad Internet cracking statistic.

To update your system manually, run:

```
[root@mail root]# yum update
```

Keeping in Time

It is important for a mailserver to have its clock set correctly, otherwise there may be problems with messages being timestamped incorrectly. This will help keep your clock in sync with the central network time authority. Create a file /etc/cron.hourly/rdate.cron and put the following in it:

```
#!/bin/sh
# Synchronize the time with nist.gov
(/usr/bin/rdate -s time.nist.gov) && (/sbin/hwclock --systohc)
```

Then set the execute permissions:

```
[root@mail root]# chmod 755 /etc/cron.hourly/rdate.cron
```

Virtual Users

There are several ways your users can log in to check their email. All of the following forms are correct for the virtual user "albus" at the virtual domain "hogwarts.jk":

```
albus@hogwarts.jk
albus:hogwarts.jk
hogwarts_jk-albus
```

However, it is ill-advised to let your clients know of any other method besides the very first one. Just let them use their email address as the username, and you will be sure to avoid a lot of confusion.

Backup

There are many backup systems out there, so I will not cover them in this little foray. Instead, I will tell you which parts to back up, and it will be up to you to come up with a method.

The following files and/or directories need to be backed up in a cookie-cutter QVCS system. If you make additions or modifications, you will need to make sure they are reflected in this list.

Note

Some of the files in this list include the ones created or modified in the advanced section. If you did not add advanced features, your system may lack some of these entries.

```
/etc/passwd
/etc/shadow
/etc/group
/etc/sslcert.pem
/etc/sysconfig/spamassassin
/etc/sysconfig/iptables
/etc/hosts.*
/etc/xinetd.d/smtp
/etc/ssh/*_key*
/etc/httpd
/etc/vadmin
/etc/squirrelmail
/etc/qmail
/etc/courier-imap
/etc/vmailmgr
/var/lib/vadmin
/var/lib/squirrelmail
/var/qmail/queue
/home/dom
/root
```

Restore

If your system has crashed and you have to reinstall everything from scratch, here is how you would go about it.

1. *Install a vanilla system.* Just create a vanilla Centos 3 setup.

2. *Restore the backup files.* Restore them over the existing tree. For example, if your backup is in `/home/bak.tar.gz`, then you would restore it like this:

```
[root@mail root]# cd /
[root@mail root]# tar xzvf /home/bak.tar.gz
```

3. *Run qvcs-init-centos.* Refer to the install section of this guide. It will wordily complain about creating .rpmnew files, but that's exactly what you want.

4. *Run qvcs-install.* However, skip all sections except the last two -- where it enables/disables services and removes sendmail.

This should be it. After these steps are done, your system should be reinstalled.

Chapter 5. Advanced Configuration

At this point you have a system that provides the skeleton of a full email solution. However, you will probably want to take this further and add some features useful for a modern email service.

Encrypted Communication (SSL)

You will most likely want to configure SSL on your newly installed machine. It is already enabled for the most part, but not at all configured. First thing you will need is an SSL certificate.

Let's first of all create a test certificate to practise on. Perform the following actions:

```
[root@mail root]# cd /usr/share/ssl/certs
[root@mail root]# make stunnel.pem
```

The program will ask you some questions, the most important of which is "Common Name". That would be the host name of your server, but before we do that, let's have a bit of a segue.

SSL And Virtual Hosts

Doing SSL on virtual hosts is tricky because the client machine will check whether the hostname of the server matches the "common name" listed in the certificate it provides during the "SSL Handshake". If these two do not match, the client will either drop the connection, or present the user with a very large, very obnoxious, and very visible SSL certificate warning.

The solution is to pick a consistent host name for your mailserver that would be both convenient and reflect upon your company as the provider of the service. I.e. if you are known as "The Quibbler Data Express", you will want to make "mail.quibbler.jk" as the common name for your SSL certificate. This is the address you will give out to all your clients for their outgoing and incoming email (SSL interface for the webmail using vadmin redirects is discussed further down).

So, once you have decided on which domain name you are going to use as your main SSL host, go ahead and fill out the "Common Name" field in the test certificate. I'll use "mail.quibbler.jk" for my examples.

Once you're done, you will see a `stunnel.pem` in that directory. A good place for it to be is in `/etc/sslcert.pem` so it can be easily backed up.

```
[root@mail root]# mv stunnel.pem /etc/sslcert.pem
```

Enabling SSL in Qmail

Qmail never runs as user root, so we will need to change the ownership on the ssl certificate to that of user "qmaild":

```
[root@mail root]# chown qmaild /etc/sslcert.pem
[root@mail root]# chmod u-w /etc/sslcert.pem
[root@mail root]# ln -s /etc/sslcert.pem /etc/qmail/control/servercert.pem
```

```
[root@mail root]# service qmail restart
```

Enabling SSL in Courier-IMAP

Simple enough:

```
[root@mail root]# ln -s /etc/sslcert.pem /etc/courier-imap/sslcert.pem
[root@mail root]# service courier-imap restart
```

Enabling SSL in Apache

Almost the exact same set of actions for Apache.

```
[root@mail root]# cd /etc/httpd/conf
[root@mail root]# rm ssl.crt/server.crt ssl.key/server.key
[root@mail root]# ln -s /etc/sslcert.pem ssl.crt/server.crt
[root@mail root]# ln -s /etc/sslcert.pem ssl.key/server.key
[root@mail root]# service httpd restart
```

Vadmin And SSL Enforcement

You may wish to enforce SSL in vadmin, so all your clients are redirected to an SSL site. Open `/etc/vadmin/vadmin.conf` in your editor and locate a commented-out section called "`[redirect]`". Remove the semicolons and change it so it looks like so:

```
[redirect]
   https = yes
   host = mail.quibbler.jk
   path = /
```

Now if you go to mail.hogwarts.jk, it will transparently redirect you to https://mail.quibbler.jk/, thus ensuring that all your communication with the server is secured.

Obtaining a Real SSL Certificate

Depending on how serious you want to be, you might want to go ahead and obtain a real SSL certificate, as sold by the Certification Authorities. Obtaining an SSL certificate is usually a painful and expensive process -- they run for about $150 per year per hostname. Several companies provide CA services: for more information go to www.whichssl.com [http://www.whichssl.com/]. If you are not worried about your clients seeing warning messages in their browsers about unrecognized signing authorities, then you may skip this part -- the self-signed certificate you created by running **make stunnel.pem** is just as secure.

Trained monkeys working at the CA companies should be able to walk you through the process once you have decided that you want a real certificate and picked which company you want to spend money with. When you have the real certificate made out for the domain name that you have picked, you will need to make a .pem file out of the .crt and .key parts (unless they can give you a .pem file in the first place). This is done by simply concatenating the .crt and .key files together. E.g.:

```
[root@mail root]# cat server.key server.crt > sslcert.pem
```

If your key is protected by a passphrase, you will need to remove it before making a .pem, as otherwise every time the server restarts you will need to enter the passphrase manually, plus qmail SSL will simply not work. To remove the passphrase, perform the following actions:

```
[root@mail root]# openssl rsa -in server.key -out nopass.key
[root@mail root]# mv nopass.key server.key
```

Once you have the `sslcert.pem` file, just replace our self-signed certificate in `/etc/sslcert.pem` and restart the services (qmail, courier-imap, httpd). Congratulations, you've now officially sold your soul to big business.

Selective Relaying

Selective relaying is a method of allowing certain "trusted" incoming email messages to be sent further along to their final destination. You don't want *ALL* messages to be relayed, as that would quickly make your server the target for relaying spam, but you might want to enable this for your clients. If you want your users to be able to use your mailserver when they send outgoing email (not just via the webmail interface, that is), read this part.

Origin-based relaying

Let's say you have a certain range of IP addresses that your users send email from. This range of addresses is therefore a "trusted subnet" and we can configure our mailserver to accept email from this origin without any further questioning and relay the messages to wherever they need to go.

We will use tcp wrappers for selective relaying. Open the `/etc/hosts.allow` file in your editor: it should currently have the following entries:

```
tcp-env: 127.0.0.1 : setenv RELAYCLIENT
tcp-env: ALL
```

Let's say that we want everyone from our trusted network to send their outgoing e-mail through our mailserver. If our trusted network is `192.168.1.0/24`, then we would change `/etc/hosts.allow` as follows:

```
tcp-env: 127.0.0.1 192.168.1. : setenv RELAYCLIENT
```

```
tcp-env: ALL
```

If we only had a fraction of class C, we could change it as follows:

```
tcp-env: 127.0.0.1 192.168.1.0/255.255.255.128 : setenv RELAYCLIENT
tcp-env: ALL
```

or, we could limit it by domain name, like so:

```
tcp-env: 127.0.0.1 .hogwarts.jk : setenv RELAYCLIENT
tcp-env: ALL
```

This would mean that any host with IP address resolving to "somehost.hogwarts.jk" would be allowed to relay e-mail.

If you have a lot of relaying rules, keeping them all on one line might get messy. In this case you may create a separate file with all the allowed hosts and networks in it. For example, put all your rules in the file `/etc/relay.rules`, so it contains something like this:

```
127.0.0.1
.hogwarts.jk
192.168.1.0/255.255.255.128
rosmerta.hogsmeade.jk
```

and change `/etc/hosts.allow` to contain the following entries:

```
tcp-env: /etc/relay.rules : setenv RELAYCLIENT
tcp-env: ALL
```

For more information about various patterns read the manual page for tcp wrappers. You can view it by executing:

```
[root@mail root]# man hosts.allow
```

Authenticated SMTP

Naturally, if your clients tend to travel and bring their laptops with them, then specifying the allowed IP ranges is not going to be very useful. Authenticated SMTP allows relaying of email messages only for people who already have accounts on the server. In fact, this is the preferred way of relaying email these days.

Open /etc/xinetd.d/smtp in your favorite editor and modify the server-args line so it looks like so (*NOTE: The following is all on one line!*):

```
server_args = /var/qmail/bin/tcp-env -R /var/qmail/bin/qmail-smtpd mail.quibbler.jk
```

As usual, replace "mail.quibbler.jk" with the name of your mail server (the one specified in the SSL certificate). After you're done editing that file, run:

```
[root@mail root]# service xinetd restart
```

Email filtering

This seems to be a popular request, and QVCS is certainly capable of providing the infrastructure needed for this. However, let me start with a huge warning.

Huge Warning

Email filtering requires some BEEFY HARDWARE™. If your mail server sees some significant email traffic, and I'm talking upwards of 5-10 thousand emails a day, you will want to have some serious iron for hardware, especially in terms of RAM and processor speed. If you have less than 1G of high-speed memory, the server performance will degrade significantly, and anyone putting a less-than AMD/P4 2GHz for this will soon come to regret their foolishness. You have been forewarned.

Spamassassin

Now let's enable spamassassin. Since you are using virtual users, there are certain options you will need to set in order for the spamd daemon not to complain. Open /etc/sysconfig/spamassassin in your editor and change the SPAMDOPTIONS line to be the following:

```
SPAMDOPTIONS="-d -c -a -m30 -H -x -u nobody"
```

Now start it:

```
[root@mail root]# chkconfig spamassassin on
[root@mail root]# service spamassassin restart
```

Now let's tell qmail-scanner that it can use spamassassin. The following command will reconfigure it -- it will take forever to run, so don't fret if it's been sitting there for a while. Eventually it'll get finished.

```
[root@mail root]# cd /usr/share/qmail-scanner
[root@mail root]# ./configure --batch --debug no --install
```

Not done yet! Now you have to edit /etc/hosts.allow and change your tcp-env : ALL line as follows:

```
tcp-env: ALL : setenv QMAILQUEUE /var/qmail/bin/qmail-scanner-queue.pl
```

Now you've done it!

How to filter out spam

If you now look at the headers of your email messages, you will see something like this:

```
Received: from luna@quibbler.jk by peeves by uid 500 with qmail-scanner-1.22

    (spamassassin: 2.63. Clear:SA:0(0.4/5.0):.
    Processed in 3.495582 secs); 29 Jul 2004 02:53:49 -0000
X-Spam-Status: No, hits=0.4 required=5.0
```

The key here is the header X-Spam-Status. All you have to do is configure your email client to look for that header, and if it contains "Yes", either move the message into the *Junk* folder, or assign it a low priority. Simply deleting messages marked as "X-Spam-Status: Yes" is not at all advised, as any automated system will have false-positives, meaning that you can lose important email.

Virus filtering

You can also use qmail-scanner to set up virus scanning, but that is not covered here. Feel free to ask around on the support lists, perhaps someone has done it.

Chapter 6. Finalizing it all

Your mail system is set up. If you have encountered any problems during the install, then consult the documentation provided with the misbehaving component -- it will most likely tell you whom to contact for support. If everything is running smoothly and you are happy with your system, then congratulations -- you've got yourself one of the best solutions for a pop-toaster out there.

Why this is not recommended for large systems

The only reason this is not recommended for large systems is because SquirrelMail is currently not very scalable -- you cannot easily run it on a server farm, since both SquirrelMail and Vadmin save their preferences onto the HDD (a trade-off for not requiring a database engine). However, if you decide not to use SquirrelMail/Vadmin, then Qmail-VmailMgr-Courier is definitely a strong enough solution to be run on high-demand servers, but this has its own set of requirements and is not covered under this guide.

Subscribe to the mailing lists!

No, honestly, do so. Subscribe to the following two mailing lists:

- `<qvcs-guide-rpms@lists.sourceforge.net>`
- `<qvcs-guide-announce@lists.sourceforge.net>`

The first one will notify you when newer RPMs become available, and the second one will tell you of any other happenings. To subscribe to these lists please go to the qvcs-guide website, at http://mirror.mricon.com/qvcs-guide.

Corrections and Comments

If you've found a mistake in this document which you would like to correct, or would just like to comment on something, please send a message to `<qvcs-guide-list@lists.sourceforge.net>` so I can make the correction or read your comments. You may also check the qvcs-guide website at http://mirror.mricon.com/qvcs-guide for the latest version of this document.

Report your success

If you found this Guide useful, please let me know by sending this brief email (replacing {your locality} with the name of your town, state, country).

```
[root@mail root]# echo "Greetings from {your locality}." | mail qvcs-report@mricon.co
```

Please also consider expressing your gratitude by sending me a gift from my Amazon Wishlist, which you may find on the main website. This will help me leverage the time I put into maintaining this guide and the packages that come with it. After all, this software isn't free as in beer, but free as in "you are free to reward the author accordingly." :)

Chapter 7. Upgrading from Red Hat 9 to Centos 3

Using the power of yum

Upgrading from Red Hat 9 to Centos 3 is actually painfully straightforward. Use these simple instructions to get up to speed in no time.

Modifying yum.conf

Modify your yum.conf to look like this:

```
[main]
cachedir=/var/cache/yum
debuglevel=2
logfile=/var/log/yum.log
pkgpolicy=newest
tolerant=1

[centos-base]
name=Centos 3 Base
baseurl=http://ftp.osuosl.org/pub/centos/3/os/i386//os/i386/
gpgcheck=1

[centos-updates]
name=Centos 3 Updates
baseurl=http://ftp.osuosl.org/pub/centos/3/os/i386//updates/i386/
gpgcheck=1

[qvcs-guide]
name=QVCS Guide RPM Repository
baseurl=http://mirror.mricon.com/qvcs-guide/yum/centos-3/
gpgcheck=1
```

Tip

You can use a different Centos 3 URL if you wish -- they are all yum-enabled.

Import the GPG key

As with Red Hat 9, it's important to use GPG checking to make sure that the packages you're installing haven't been tampered with. Import the Centos key using the following command:

```
[root@mail root]# lftpget http://ftp.osuosl.org/pub/centos/3/os/i386//os/i386/RPM-G

[root@mail root]# rpm --import RPM-GPG-KEY-CentOS-3
```

Yum upgrade

Now run the following command to upgrade your system to Centos 3:

```
[root@mail root]# yum upgrade
```

The upgrade process is likely to take some time, so prepare to be patient.

Fix yum

The upgrade will screw up yum.conf a little, so you'll need to fix it back again. To do it, perform the following command:

```
[root@mail root]# mv /etc/yum.conf-SAVE /etc/yum.conf
```

Rebuild QMail

I cannot at all distribute binary packages for qmail in this release, so you will need to rebuild them once yum upgrade terminates. Use the following commands:

```
[root@mail root]# wget http://mirror.mricon.com/qvcs-guide/qmail.src.rpm

[root@mail root]# yum -y install rpm-build gcc openssl-devel
[root@mail root]# rm -f /usr/src/redhat/RPMS/i386/qmail*rpm
[root@mail root]# rpmbuild --rebuild --define 'allpatches 1' qmail.src.rpm

[root@mail root]# cd /usr/src/redhat/RPMS/i386
[root@mail root]# rpm -Uvh qmail*rpm
```

Vadmin DB Change

Centos does not come with gdbm storage enabled, so you will need to convert the database format from gdbm to db4. To do it, run the following command:

```
[root@mail root]# vadmin-convert-gdbm2db4
```

This should convert existing databases into the new .db4 format. Now open /etc/vadmin/vadmin.conf and edit the [storage] section so it looks like this:

```
[storage]
   type = dba
   flavor = db4
   suffix = .db4
   dir = /var/lib/vadmin
```

Miscellaneous

Courier-Imap relies on fam_sgi to monitor when files change, and in this version sgi_fam relies on the portmapper to run correctly, so enable portmapper in the list of services:

```
[root@mail root]# chkconfig portmap on
```

If you were using qmail-scanner to filter your incoming email, you will need to rerun the configuration program as well (and, just as previously, it will take forever to run):

```
[root@mail root]# cd /usr/share/qmail-scanner
[root@mail root]# ./configure --batch --debug no --install
```

Note

I have noticed that sometimes the upgrade process locks the password file, causing the upgrade for qmail-scanner to fail. If the configure command fails with the warning about user "qscand" not existing, reboot the system, then run the following commands (yes, the rpm -e command needs to be run twice):

```
[root@mail root]# rpm -e --noscripts qmail-scanner
[root@mail root]# rpm -e --noscripts qmail-scanner
[root@mail root]# yum -y install qmail-scanner
```

This should create the necessary qscand user. After that you can re-run the configure script for qmail-scanner.

Clean up

This is it! To tidy up, run the following command:

```
[root@mail root]# yum clean
```

If you have encountered problems during the upgrade, please report them to the mailing list.

Appendix A. Description of Packages

Let me explain in more detail what we just installed. There are overall 14 packages that constitute the qvcs system:

- qmail: This is the package with all main qmail binaries. Qmail is an MTA and MDA, which stands for "Mail Transport Agent" and "Mail Delivery Agent". It was written with security in mind and hasn't had a single security exploit in many years. Moreover, the author of this package has set up a prize of $1000 to anyone who can find a security flaw in qmail -- this prize has gone unclaimed in years.
- qmail-initscripts: This package contains initialization and xinetd scripts for qmail, written specifically for Centos.
- courier-imap: Courier-Imap is a very well-done IMAP server which was written specifically to work with "Maildir" mail storage system used by qmail. It is very fast, very standards compliant, and takes very little space in your computer's memory.
- vmailmgr: This is the Virtual Mail Manager for qmail -- it is also an MDA and allows you to have "virtual" e-mail users without giving said users shell access on your system, which can often lead to security compromises.
- vmailmgr-courier-imap: This small package adds an authentication module to courier-imap which allows it to work with virtual users set up by vmailmgr.
- vmailmgr-daemon: A small package containing a special binary which lets vmailmgrd communicate with other daemons, like perl or php in our case.
- ucspi-unix: This is a support package for vmailmgr-daemon and allows creating UNIX sockets on the system for communication between daemons.
- libmcrypt: This is a set of encryption libraries used by vadmin plugin. Vadmin can optionally use libmcrypt to encrypt the passwords before storing them on the hard drive for enhanced security. By default it uses a builtin rc4 function.
- libmcrypt-devel: This package is not installed by default and is only provided for the sake of completeness.
- php-mcrypt: A shared library file which ties libmcrypt to php and provides php encryption functions.
- squirrelmail: This is a great IMAP-based php webmail system.
- squirrelmail-vadmin: Vadmin is a plugin for squirrelmail which makes administering vmailmgr virtual domains a part of squirrelmail. It has some very nice features like the ability to add/remove users, set quotas or account expiration dates, etc.

- qmail-autoresponder: This package allows setting up autoresponders through the squirrelmail (vadmin) interface.

- qvcs-helpers: This package has a few helper scripts which come with this guide.

- bglibs: This package is not installed by default, but is needed to build several other packages. Unless you rebuild some packages from source RPMs, you do not need this.

- maildrop: Part of the qvcs-filter package set, it is used by qmail-scanner.

- tnef: A small application that will unpack Microsoft-style attachments. Useful for virus and spam scanning. Part of the qvcs-filter set.

- qmail-scanner: An alternative qmail-queue implementation that allows invoking spamassassin and various virus scanners. Part of the qvcs-filter package set.