

Qmail-Vmailmgr-Courier-SquirrelMail Installation Guide (BSD Systems)

Konstantin Riabitsev
icon@duke.edu

Version 2.0: February 2002

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 Qmail-Vmailmgr-Courier-SquirrelMail tie-in as the best small to mid-class server solution. This document is written for BSD systems.

Copyright (c) 2001-2002 by Konstantin Riabitsev <icon@duke.edu>. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being (no invariant sections), with the Front-Cover Texts being (none), and with the Back-Cover Texts being (none). A copy of the license is available to you in *Appendix B*.

Contents

1	The problem	2
2	Conventions used in this Guide	2
3	The Software	2
3.1	Operating System Software	2
3.2	Stuff You Will Need	3
4	Setting up Qmail	3
4.1	Preparing the scene	3
4.2	Compiling	4
4.3	Configuring qmail	5
4.4	Setting the sendmail wrapper	5
4.5	Startup scripts	6

5	Setting up ucspi-tcp	6
5.1	Compiling ucspi-tcp	7
5.2	Setting up tcprules	7
5.3	Running tcpserver for SMTP	8
5.4	Checking to see if it worked	8
6	Useful Pointers for Qmail and Tcpserver	9
7	Configuring VmailMgr	9
7.1	Building and installing VmailMgr	9
7.2	Configuring Virtual Domains	9
7.3	Helper shell scripts	10
7.4	Creating a virtual domain	10
7.5	DNS settings	10
7.6	Creating a sample user	11
7.7	Setting qmail system aliases	11
8	Configuring Courier-IMAP	12
8.1	Taking care of inetd	12
8.2	Building and installing Courier	12
8.3	Configuring Courier-IMAP	13
8.4	etc/imapd	13
8.5	etc/pop3d	13
8.6	Starting Courier-IMAP	14
8.7	SSL pop and imap	14
8.8	Testing your setup	14
8.9	Vmailmgr login handles	15
9	Setting Up Webmail	15
9.1	Compiling and installing Apache	16
10	PHP HTML Pre-processor	16
10.1	Compiling and installing PHP	16
11	Setting up SquirrelMail	17
11.1	Untarring and setting up directories.	17
11.2	Configuring SquirrelMail	18
12	Apache Configuration	19
12.1	Minimalistic configuration	19
12.2	Other possible configurations	20
12.3	SquirrelMail plugins	20

13 Administration Front-end	20
13.1 Installing Ucspi-unix	21
13.2 Installing libmcrypt	21
13.3 Running vmailmgrd	21
13.4 Vmailmgr Admin plugin for SquirrelMail	21
13.4.1 Recompiling PHP4	22
13.4.2 Getting and installing Vadmin	22
13.5 Adding domains to Vadmin	23
13.5.1 Using domain_magic.sh script	23
13.6 Adding more domains	24
14 Finalizing it all	25
14.1 Why this is not recommended for large systems	26
14.2 Corrections and Comments	26
14.3 Thank you and good luck! ;)	26
A Scripts	27
A.1 edrelays.sh	27
A.2 addvirt.sh	28
A.3 rmvirt.sh	31
B GNU Free Documentation License	34
B.1 Applicability and Definitions	34
B.2 Verbatim Copying	35
B.3 Copying in Quantity	36
B.4 Modifications	36
B.5 Combining Documents	38
B.6 Collections of Documents	38
B.7 Aggregation With Independent Works	39
B.8 Translation	39
B.9 Termination	39
B.10 Future Revisions of This License	39

1 The problem

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 Qmail-Courier-Vmailmgr-SquirrelMail 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.

2 Conventions used in this Guide

For an easier read, the following conventions will be used throughout this guide. File names and directory paths will be shown inline in **typewriter** font, e.g. “`/usr/libexec`”. Names of commands will be presented in **bold** typeface, for example “**ps auxww | grep qmail**”. Software names and other random bits and pieces that I find worthy of highlighting will be marked in *italics* typeface.

Scripts and verbatim commands that you will need to execute will be presented in **typewriter** font and placed in paragraphs of their own. For example:

```
mail# echo 'Hello World' | wc -l
mail# clear
```

Scripts can be found in the Appendix.

3 The Software

3.1 Operating System Software

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. Currently, I have two versions of the guide – one aimed at *Red Hat Linux* people (since that's

what I use as well), and one guide aimed at a more generic BSD-ish system, be it *FreeBSD*, *OpenBSD*, *NetBSD*, or *SlackWare*. You are reading the latter version. It has been tested on *OpenBSD-3.0* and is written with OpenBSD in mind, although it should work on other BSD systems without any significant issues.

This document assumes that you have successfully configured your system and are ready to put the e-mail stuff on it. Please refer to your BSD makers' web site for any installation instructions.

3.2 Stuff You Will Need

This shows latest versions at the time of writing. If newer versions are available, you should probably get them, unless otherwise instructed.

- *qmail-1.03* (<http://www.qmail.org/>)
- *ucspi-tcp-0.88* (<http://cr.yp.to/>)
- *courier-imap-1.4.2* (<http://www.courier-mta.org/>)
- *vmailmgr-0.96-9* (<http://www.vmailmgr.org/>)
- *apache-1.3.23* (<http://httpd.apache.org/>)
- *php-4.0.6* (<http://www.php.net/>)
- *squirrelMail-1.2.4* (<http://www.squirrelmail.org/>)
- *libmcrypt-2.4.18* (<http://mcrypt.hellug.gr/>)
- *ucspi-unix-0.34* (<http://untroubled.org/ucspi-unix/>)
- *vadmin-v1.0* (<http://mricon.com/xmlparse/SM/>)

4 Setting up Qmail

Setting up Qmail is rather trivial, but here are most basic steps you need to take.

4.1 Preparing the scene

Before you do anything, you will need to remove any traces of *Sendmail* from your system. Depending on the OS, sendmail could have been installed as a separate package, or as a part of the system. Run “**pkg_info**” and see if you have sendmail installed as a package. If you in fact do, run “**pkg_delete**” on it to remove the traces of this beast from your system. Otherwise just do the following (or something like this). Kill the “sendmail” process first – run “**ps ax | grep sendmail | grep -v grep**”. If this command returned any lines, look

at the first field and then run “**kill NNNN**” where “NNNN” is the number in the first field.

After the sendmail daemon has been stopped, we need to move the original sendmail binaries somewhere where they don't bother us (it's ok if some of the following commands return “*no such file or directory*”, but check your input for typo's!).

```
mail# chmod 0 /usr/sbin/sendmail
mail# chmod 0 /usr/lib/sendmail
mail# chmod 0 /usr/libexec/sendmail
mail# chmod 0 /usr/lib/sendmail.mx
mail# mv -f /etc/mail /etc/mail.orig
mail# mv -f /usr/sbin/sendmail /usr/sbin/sendmail.orig
mail# mv -f /usr/lib/sendmail /usr/lib/sendmail.orig
mail# mv -f /usr/libexec/sendmail /usr/libexec/sendmail.orig
mail# mv -f /usr/lib/sendmail.mx /usr/lib/sendmail.mx.orig
```

It is also possible that your system is configured to run cron scripts monitoring sendmail or the mail queue. Turn those off by running “**crontab -e**” and commenting out the line that calls sendmail.

Next, create a `/var/qmail` directory. You need to remember, that this is where qmail will store its queued e-mail, so if you want to be ready for some big traffic (remember the *iloveyou!*), you might want to make sure you have a reasonable amount of space available on that partition.

Qmail has very tight security features, and it achieves them by using six separate user accounts to handle various functions. You will need to create these special accounts before you can compile and install qmail.

How you do it depends on your OS, but here is the process that should be suitable for most modern BSD systems (note that it's “>>”, not “>” – VERY important):

```
mail# cd /etc
mail# echo "nfiles*:5000:" >> group
mail# echo "qmail*:5001:" >> group
mail# echo "alias*:5000:5000::0:0::/var/qmail/alias:" >> master.passwd
mail# echo "qmaild*:5001:5000::0:0::/var/qmail:" >> master.passwd
mail# echo "qmail1*:5002:5000::0:0::/var/qmail:" >> master.passwd
mail# echo "qmailp*:5003:5000::0:0::/var/qmail:" >> master.passwd
mail# echo "qmailq*:5004:5001::0:0::/var/qmail:" >> master.passwd
mail# echo "qmailr*:5005:5001::0:0::/var/qmail:" >> master.passwd
mail# echo "qmails*:5006:5001::0:0::/var/qmail:" >> master.passwd
mail# pwd_mkdb -p /etc/master.passwd
```

4.2 Compiling

Now that you have created the necessary users and made sure you have a sensible amount of free space in `/var/qmail`, you are ready to compile and install. This

is very trivial and takes no time at all.

I usually like to keep the sources in `/usr/local/src`, so let's create that directory and do all of our compilations in there.

```
mail# mkdir -p /usr/local/src
mail# cd /usr/local/src
```

Now copy the `qmail-1.03.tar.gz` file into `/usr/local/src` and get on with compiling:

```
mail# tar xzvf qmail-1.03.tar.gz
mail# cd qmail-1.03
mail# make && make setup check
```

If everything compiled without errors (which it should have), you are ready to configure qmail.

4.3 Configuring qmail

For the sake of providing examples I am going to assume that your hostname is "`mail.hogwarts.jk`". You, of course, should use whatever is the domain name of your future mail server.

Go to `/var/qmail/control` directory:

```
mail# cd /var/qmail/control
mail# echo mail.hogwarts.jk > me
mail# echo mail.hogwarts.jk > defaultdomain
mail# echo mail.hogwarts.jk > rcpthosts
mail# echo mail.hogwarts.jk > locals
mail# echo mail.hogwarts.jk > plusdomain
mail# echo localhost >> locals
mail# echo localhost >> rcpthosts
```

This covers the basic installation. We will get to virtual domains setup at a later step, since it's much easier done with a simple script, than by hand.

4.4 Setting the sendmail wrapper

Since most systems cannot function without sendmail, qmail provides a sendmail wrapper. To activate it do this:

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

This should make every program totally oblivious to the fact that you're not really using sendmail any more.

4.5 Startup scripts

Go to `/var/qmail/boot` directory, locate the file called “home” and move it to `/var/qmail/rc`:

```
mail# cd /var/qmail/boot
mail# mv home ../rc
```

Now we need to edit the `rc` file and make sure that it delivers into Maildirs, and not into Mailboxes, as it is still by default (by some horrible cosmic oversight, I guess). Open `/var/qmail/rc` in your favorite editor, and change “./Mailbox” to “./Maildir”.

To run qmail all you have to do is execute:

```
mail# /var/qmail/rc &
```

To have it started automatically you need to put “`/var/qmail/rc &`” into your `/etc/rc.local` file. Check if you have `/etc/rc.local` first – it might not be available by default on your flavor of BSD, but all of them check for the presence of this file during the boot-up, and will execute any commands in your `rc.local` automatically. If you **don’t** have a `/etc/rc.local`, then run:

```
mail# echo '#!/bin/sh' > /etc/rc.local
mail# chmod 0755 /etc/rc.local
```

Now let’s tell the server that it needs to start qmail during the boot-up:

```
mail# echo '/var/qmail/rc &' >> /etc/rc.local
```

Just to make sure that `rc.local` gets executed during the boot-up, reboot your machine. When the machine comes back up, run the following command:

```
mail# ps auxww | grep qmail
```

and find the “*qmail-lspawn*” process together with a few others. Test the system by sending a simple message like:

```
mail# uname -a | mail myself@remoteserver.com
```

Check your remote server and make sure you have received this e-mail. If something didn’t work, check your `/var/log/mail` or `/var/log/maillog` (whichever) for a possible explanation. Sorry, troubleshooting is out of the scope of this document. Visit <http://www.qmail.org/> for any support you need.

5 Setting up ucspi-tcp

Your mail server is no good if it can’t accept SMTP connections. This is best done by using the *ucspi-tcp* package available from the same developers who wrote qmail. Get the tar.gz and put it into `/usr/local/src`.

5.1 Compiling ucspi-tcp

No sweat, as always. Simply do:

```
mail# tar xzvf ucspi-tcp-0.88.tar.gz
mail# cd ucspi-tcp-0.88
mail# make && make setup check
```

This will compile and install ucspi-tcp files in your `/usr/local` directory.

5.2 Setting up tcprules

We want to make sure that nobody but our own clients are able to use our server to send outgoing mail. Otherwise we will have what is called an “*open relay*” – everyone will hate us and we’ll end up on various spam blacklists. Not something we want.

To enable “*selective relaying*”, we need to create a file called `relay.rules` somewhere where it makes sense, like in `/var/qmail/control` directory. The general file format goes something like this:

```
127.0.0.1:allow,RELAYCLIENT=''
192.168.1.:allow,RELAYCLIENT=''
:allow
```

What this does is instructs qmail that it should allow outgoing e-mail only for clients connecting from the local machine and our trusted subnet, which in this case is “192.168.1.0/24”. You, of course, will need to change it to your own subnet if you want this to work. You may add as many subnets in, following the given example and putting each on a separate line, to make sure that all your clients are covered. Don’t forget the last “:allow” on the line, or we won’t accept any incoming mail for our clients either.

Once your `/var/qmail/control/relay.rules` file is completed, you will need to compile it so it’s ready to be used. Do this by executing:¹

```
mail# /usr/local/bin/tcprules /var/qmail/control/relays.cdb \
/var/qmail/control/relays.tmp < /var/qmail/control/relay.rules
```

The command will compile your rules and put them in a binary file `relays.cdb`, located in the qmail control directory. Remember that you will need to recompile the rules if you add any more subnets in. I found it useful to use a no-brain script whenever I needed to edit the rules, called `edrelays.sh`. It is included

¹General note about scripts in this Guide: In order to make the scripts look sane on a displayed/printed page, I am using backslashes to go to the next line. You can either omit the “\” and just continue typing without hitting “return” until you get to the very end of the command, or you may follow my lead and use backslashes. In this case make sure you hit “return” right after you typed the “\”. What this does is it escapes the newline symbol and the system accepts the command as if it was all on one line. Most shells will show a “>” symbol after you hit return to denote that you may continue input.

in the *Appendix A.1* and is available for download from my website (read the introduction to the *Appendix A*).

Now editing rules is as simple as running *edrelays.sh* (don't forget to set the execution bits).

5.3 Running tcpserver for SMTP

Running tcpserver is no easy task. ;) Well, it is, it just needs to be placed in *rc.local* so we don't ever have to touch it again. The command line to execute tcpserver looks like this:²

```
mail# /usr/local/bin/tcpserver -R -x/var/qmail/control/relays.cdb \
-u5001 -g5000 0 smtp /var/qmail/bin/qmail-smtpd 2>&1 \
| /var/qmail/bin/splogger &
```

A bit of explanation of what this does. The first flag, “-R” tells the server not to try to establish an auth connection to the client, since it's mostly useless nowadays, when nearly everyone is running firewalls and/or silly operating systems. *-x/var/qmail/control/relays.cdb* flag makes sure we use the relaying rules we have just compiled. *-u5001* instructs tcpserver to run as user “qmaild” *-g5000* is group “nofiles”, 0 means that we want tcpserver to listen on all interfaces, *smtp* means we want it to bind to port 25 (the standard SMTP port used for mail exchange), */var/qmail/bin/qmail-smtpd* instructs the tcpserver to run qmail-smtpd program after establishing a connection, *2>&1* makes sure there isn't any garbage spewing onto the console and it all goes into the logs, and */var/qmail/bin/splogger* makes certain we log things in */var/log/maillog*. Quite simple, really. ;)

Since we want the SMTP service to start during the system boot as well, we need to add the monstrous line depicted above into our */etc/rc.local*. Break out your favorite editor and do so now.

5.4 Checking to see if it worked

Check to see if you have done everything correctly by doing this:

```
mail# telnet mail.hogwarts.jk smtp
```

You should see:

```
Trying 192.168.1.1...
Connected to mail.hogwarts.jk.
Escape character is '^]'.
220 mail.hogwarts.jk ESMTP
```

²If you get an “Ambiguous output redirect” message, then you are using *csh* and should be beaten by big surly men. Run “*sh*” and then execute the command. You may “*exit*” afterwards if you're that enamoured with *csh*. :)

Try saying hello. Type in

```
HELO hogsmeade.jk
```

It should say:

```
250 mail.hogwarts.jk
```

Type “quit” to close connection. We will do serious testing later.

6 Useful Pointers for Qmail and Tcpsrvr

Qmail and tcpsrvr are a very flexible and extendable combination. You may tweak the system to your liking – visit <http://www.qmail.org/> for a very lengthy list of software and enhancements available.

7 Configuring VmailMgr

Get the latest version of vmailmgr from <http://www.vmailmgr.org/>. At the time of writing it is vmailmgr-0.96-9.tar.gz.

7.1 Building and installing VmailMgr

Untar the distribution and cd into the vmailmgr-0.96-9 directory. Now configure and build it:

```
mail# ./configure
mail# make && make install
```

That’s it. If you encounter any problems, try using “**gmake**” instead of “**make**”, and if that doesn’t work, please refer to help docs and any info on the vmailmgr website, as troubleshooting is out of the scope of this document.

7.2 Configuring Virtual Domains

It is best of all to add and remove virtual domains by using helper scripts, otherwise things get extensively verbose and tedious. Before we start, however, let’s create a `/home/dom` directory where we will put all virtual domain directories, and add a “*dom*” group. This is done to organize everything nicely.

```
mail# mkdir /home/dom
mail# echo "dom:*:5002:" >> /etc/group
```

Note that `/home/dom` is where all your incoming users’ e-mail messages will reside, so depending on how many users you are going to support, you might want to make sure that you have plenty of space on this partition.

7.3 Helper shell scripts

It takes a lot of tedious steps to create or delete a virtual domain, therefore I found it useful to write a couple of shell scripts to do all the work for me.

The `addvirt.sh` script is presented to you in *Appendix A.2*. It will help you to significantly lessen the amount of typing you have to do in order to add a virtual domain. The `rmvirt.sh` script is located in *Appendix A.3* and is just as useful.

Copy these scripts into `/usr/local/bin` directory and set permissions to execute. You might want to check if your `/usr/local/bin` is in your `$PATH` if you don't want to type in the full path of each script every time.

7.4 Creating a virtual domain

Let's go ahead and create a virtual domain for our use. Since we will most likely use this mail server to handle all e-mail for *hogwarts.jk*, let's go ahead and add it as a virtual domain.

```
mail# addvirt.sh hogwarts.jk
Base Username [hogwarts]:
Creating new domain 'hogwarts.jk'.
Domain base user created in /home/dom/hogwarts.jk.
Please provide domain password for VmailMgr.
Changing local password for hogwarts.
New password:
Retype new password:
Adding domain to control/virtualdomains...done
Adding records to control/rcpthosts...done
Setting up the domain dir for vmailmgr...
vsetup: created users directory.
vsetup: wrote '.qmail-default' file.
vsetup: added alias 'mailer-daemon'
vsetup: added alias 'postmaster'
vsetup: added alias 'root'
...done
Restarting Qmail...done

All done! Domain hogwarts.jk created.
mail#
```

As you see, the script created a base user for qmail, added all needed entries in qmail control files, and even restarted qmail for you so changes can take place. Isn't that nice?

7.5 DNS settings

We need to make sure that all mail sent to *whomever@hogwarts.jk* is be handled by our mail server. This is done outside of qmail, in the DNS settings for

hogwarts.jk, specifically by setting an MX record looking something like this:

```
hogwarts.jk. IN MX 10 mail.hogwarts.jk.
```

When other domains are added to our system in the future, they will all have very similar DNS settings to the one above, like:

```
hogsmeade.jk. IN MX 10 mail.hogwarts.jk.  
diagonalley.jk. IN MX 10 mail.hogwarts.jk.  
theburrow.jk. IN MX 10 mail.hogwarts.jk.
```

For more information check with manuals for your DNS server, such as BIND or djbdns – I am not going to go in any details on the matter of DNS settings in this guide.

7.6 Creating a sample user

After the *hogwarts.jk* domain is added, we are ready to start adding users. Since we don't have *vmailmgrp* configured yet, we can't use a web front-end for this, therefore we will need to do it by hand. Do this by executing the following:

```
mail# su - hogwarts  
mail:hogwarts.jk {1} vadduser albus  
Enter the user's new password:  
Please type it again for verification:  
vadduser: user 'albus' successfully added  
mail:hogwarts.jk {2} exit
```

Done – user *albus@hogwarts.jk* is now ready to receive e-mail.

7.7 Setting qmail system aliases

Since we are going to provide a fully virtual access to all our users, we will most likely want all local mail (such as addressed to root or mailer-daemon) to be delivered to virtual users instead. This is done by setting a few aliases.

```
mail# cd /var/qmail/alias  
mail# echo '&root' > .qmail-toor  
mail# echo '&root' > .qmail-postmaster  
mail# echo '&root' > .qmail-admin  
mail# echo '&root' > .qmail-mailer-daemon  
mail# echo '&albus@hogwarts.jk' > .qmail-root
```

That should be enough. You may add as many as you need, making sure that all physical users' mail gets redirected to virtual servers (unless you know what you are doing).

Next, since an e-mail server is useless if users can't get to their e-mail, we will need to add two most popular protocols for accessing mailboxes – POP3 and IMAP4.

8 Configuring Courier-IMAP

Courier-IMAP will provide four services we need – imap-4, secure imap, pop-3, and secure pop. However, before we start on courier-imap, we need to take care of a minor annoyance.

8.1 Taking care of inetd

Many systems will have their own POP3 and IMAP handlers configured in inetd. This needs to be disabled, otherwise it will conflict with our installation. Edit your `/etc/inetd.conf` file, find a line starting with “pop3” and comment it out with a “#” sign. Do the same for the line beginning with “imap”, if you have one.

Frankly, there is no reason why inetd should be running at all on a mail-server. If you wish to disable it, check with your system’s manuals, but on most BSD systems it’s as easy as editing the `/etc/rc.conf` file and changing the setting to `inetd=NO`. Leaving it running will not impair you in any way, besides leaving holes open for any potential attacks. However, just make sure that it doesn’t try to serve pop3 and imap connections if you would like to leave it running for whatever purpose.

Restart inetd by running:

```
mail# ps ax | grep inetd | grep -v grep
22545 ??  Is      0:00.02 inetd
mail# kill -HUP 22545
```

Note that “22545” is only pertinent to my system, and yours will most likely return a different number.

8.2 Building and installing Courier

Get Courier-IMAP from <http://www.courier-mta.org/download.php>. Note that we don’t want the “*courier*” package, just “*courier-imap*”. The full package includes some other things we don’t need, like *squebmail*, which is not anywhere near as good as *squirrelmail*.

Courier-imap is tricky in a way that it requires you to build it as non-root (this is overall a very good idea, too). You may untar and configure it as a regular user, or you can just follow my example and override this:

```
mail# cd courier-imap-1.4.2
mail# ./configure --disable-root-check
```

Courier-imap will take a while to configure and it might seem like it’s going in a constant loop, but it’s not. When it’s done configuring, build and install it by running:

```
mail# make && make install
```

This will install courier-imap into `/usr/lib/courier-imap`.

Next step is creating a default configuration. Note, that for this you will need a “GNU make”, or **gmake**. If you don’t have gmake installed on your system, you should be able to obtain a package built for your flavor of BSD from the distributors. Install it by using **pkg_add**, or go to your ports tree and install gmake from there. When you have gmake, run:

```
mail# gmake install-configure
```

This will create some default configurations for Courier-IMAP that we will use in the next step.

8.3 Configuring Courier-IMAP

Follow these steps to enable courier-imap with vmailmgr. First, go back to your `vmailmgr-0.96-9` directory where you still have your vmailmgr distribution. Do this:

```
mail# cp authenticate/authvmailmgr \  
/usr/lib/courier-imap/libexec/authlib/
```

Authvmailmgr is needed so courier-imap knows how to authenticate your virtual users. Now we need to edit some settings in the config files for both imap and pop servers. The files are located in the `/usr/lib/courier-imap/etc` directory.

8.4 etc/imapd

Fire up your favorite editor and change the following settings in `imapd`:

MAXPERIP: The default setting is “4”, but since we are going to be using SquirrelMail, which accesses the server from the same IP address, we want to bump it up to something more sensible, like 10 or 20, or even 100, if you expect to have a LOT of users (make sure you adjust the *MAXDAEMONS* setting appropriately, though – it always needs to be higher than *MAXPERIP*). If we leave this setting at a default level, it is quite certain that some users will have problems connecting using SquirrelMail.

AUTHMODULES: There will be just one daemon on the line: *authdaemon*. Change this field so it is “*authvmailmgr authdaemon*”.

That’s it. If you feel confident enough, you may tweak other settings, but they are not vital to the functioning of the server (unless you break something).

8.5 etc/pop3d

You will only have to change one setting, since SquirrelMail is not using pop3 at all and we can safely leave the *MAXPERIP* setting at 4. Locate the *AUTHMODULES* setting and change it the same way as with the `imapd` file, so it reads “*authvmailmgr authdaemon*”. Save the file and you’re done.

8.6 Starting Courier-IMAP

Courier-imap developers were nice enough to provide a convenient startup script. To start courier-imap simply run:

```
mail# /usr/lib/courier-imap/libexec/imapd.rc start
mail# /usr/lib/courier-imap/libexec/pop3d.rc start
```

You will need to add the above lines to our ever-growing `/etc/rc.local` file if we want these services started during the system boot-up.

8.7 SSL pop and imap

SSL-imap and SSL-pop are very nifty services which run pop3 and imap over a secure tunnel, so all traffic is encrypted and none of our passwords get sniffed. We don't really care that much if the passwords get hijacked, since none of them are system passwords, but our users might wish to have this feature available, since a stolen password jeopardizes their privacy.

If you are serious about running SSL services, you will need to obtain valid SSL certificates for your mail server from Certification Authorities like *Thawte* or *VeriSign*. I will not give any instructions on how to go about getting the certificates in this guide, consider it a test of your Googling abilities. ;)

If you want to just try it out, then run the following commands:

```
mail# cd /usr/lib/courier-imap/share
mail# ./mkimapdcert
mail# ./mkpop3dcert
```

These will output a lot of dots and pluses, but at the end you will find two extra files in the directory: `imapd.pem` and `pop3d.pem`. Next, edit the configuration files for these services in the courier-imap's `etc` directory.

In `etc/imapd-ssl` change `IMAPDSSLSTART` to "YES", and in the `etc/pop3d-ssl` file change `POP3DSSLSTART` to "YES" as well. That's it.

To start the SSL services, execute:

```
mail# /usr/lib/courier-imap/libexec/imapd-ssl.rc start
mail# /usr/lib/courier-imap/libexec/pop3d-ssl.rc start
```

Add these lines to your `/etc/rc.local` to start these services during the boot time, but note that unless you get valid `.pem` files from a Certification Authority, your users will get "invalid certificate" warnings each time they try to connect securely.

8.8 Testing your setup

Let's see if everything is working as it should. Send an e-mail to `albus@hogwarts.jk` from some remote location (of course, substituting our little Harry-Potterish

examples with whatever you have set up instead), and then try checking your e-mail with some client.

Your login settings would be the following:

- Server: mail.hogwarts.jk
- Login: albus@hogwarts.jk
- Password: whatever

Try imap first, then try pop3. If it works, then pat yourself on the back – your system is almost ready. ;) If it doesn't, try troubleshooting the log files and re-checking your steps (especially MX and other DNS settings). Maybe you missed something?

8.9 Vmailmgr login handles

As you have already noticed, vmailmgr makes virtual users possible by requiring the full e-mail address as the login. Actually, it will accept any of the following:

- albus@hogwarts.jk
- albus:hogwarts.jk
- hogwarts-albus

However, heed my advice and don't use anything but the first one. Users tend to get very confused between colons and dashes, so just tell them to use their full e-mail address as username and everyone will be happy.

9 Setting Up Webmail

Now that our e-mail system is up and running, it's time to give our clients a nice web-mail interface so they can check their e-mail whenever they are away from their computers.

I am going to advise setting up a webserver on the same machine as where your mail server is running, meaning the box we just configured. A minimalistic Apache/PHP install doesn't require a lot of resources and will not weigh down your mailserv box by much, but will give you several important advantages.

First, if you decide to limit your webmail access to HTTPS/SSL, you will not have to worry about cleartext traffic going to IMAP server and back (Squirrel-Mail at the time of writing doesn't support IMAPS or any alternative encryption methods. You can work around this by using "stunnel" or similar forwarding services, however). By having both httpd and courier-imap server running on the same box, you can firewall your ports to only allow access to IMAP from the local machine and force everyone else to connect to SSL-imap and SSL-pop3.

Second, you will need a webserver on that machine anyway if you want to run vmailmgr webmail front-end scripts.

Thus, my advice is to set up webmail on the same machine as the mailserver. If you choose otherwise, you may still refer to the rest of this document for hints on how to best configure SquirrelMail and your Apache server. The rest of us – let’s see how we can install a webserver on this machine.

To do this, we will first need to compile and install the Apache httpd server.

9.1 Compiling and installing Apache

Most BSD systems come with Apache pre-installed, at least my OpenBSD-3.0 did. For whatever reason, it is installed not as package, but as part of the base system install. If the same is true for you, then you can skip this step as we can use the pre-installed version of apache to do what we need. Check if you already have apache by running “**which httpd**” – if you get “*httpd: Command not found*”, then you don’t have it and should build it from source. If your BSD system doesn’t come with apache already available, then download the tarball from <http://httpd.apache.org> and build it following these steps:

Untar the tar.gz file and cd into the `apache-1.3.23` directory. To configure and build Apache, run these commands:³

```
mail# cd apache_1.3.23
mail# ./configure --prefix=/usr/local/apache \
--enable-module=so
mail# make && make install
```

This will build and install apache into `/usr/local/apache` directory. Let’s go on directly to configuring and building PHP before we get dirty with `httpd.conf` adaptation.

10 PHP HTML Pre-processor

You can get a copy of PHP from <http://www.php.net/>. The latest version at the time of writing is *php-4.1.1*, but there are some annoying issues with it at this time, like various things being broken, so we are going to install a release that is actually known to work reliably – *php 4.0.6*. Get it from php.net, from the “older releases” section.

10.1 Compiling and installing PHP

Untar and cd into the `php-4.0.6` directory. Run these commands:

```
mail# ./configure --with-apxs=/usr/local/apache/bin/apxs
```

³**Note:** These instructions and configurations are only useful if you’re just using Apache/PHP to run SquirrelMail and nothing else. If you are planning to provide any additional hosting on the same server, your mileage will vary. Note that for security purposes you are not encouraged to share the same server to do both hosting and web-mail, although if you know what you are doing and can provide sufficient security, there is no reason not to go for it.

If you didn't build apache on your own, but used the one pre-installed on your system, then your configure line will look like so:⁴

```
mail# ./configure --with-apxs='which apxs'
```

You will need to make this substitution each time in the future when I'm talking about using the `--with-apxs` flag.

If you want SquirrelMail to talk in other languages, you will need to enable gettext support, too. Most BSD systems come with gettext already available and you just need to compile it into the PHP by adding `--with-gettext` to your configuration line. This is generally a good idea, anyway, since some of your users might speak in other languages, not just English. If you don't have gettext available on your BSD system, there should be a gettext package available from the distributors, or you can always build it from the ports tree.

With gettext support, the configure line will look like this (assuming that your gettext libraries are in `/usr/local`):

```
mail# ./configure --with-apxs=/usr/local/apache/bin/apxs \  
--with-gettext=/usr/local
```

After PHP configures itself, build it like you did all others:

```
root@mail:# make && make install
```

PHP will take forever to build, unless you're using a very fast box, but once it's done, you should have PHP configured on your system.

Don't fire up the server yet. It's far from ready.

11 Setting up SquirrelMail

Download it from <http://www.squirrelmail.org/>. The latest version at the time of writing is *squirrelmail-1.2.4*.

11.1 Untarring and setting up directories.

It is common practice to keep web-related documents in `/var/www`, so I will suggest the same approach. Create `/var/www` if it doesn't exist on your system and copy your `squirrelmail-1.2.4.tar.gz` there. Do:

```
mail# tar xzvf squirrelmail-1.2.4.tar.gz  
mail# rm squirrelmail-1.2.4.tar.gz  
mail# ln -s squirrelmail-1.2.4 squirrelmail  
mail# cd squirrelmail
```

⁴Note that these are not "apostrophes", but "backticks", which are located under the tilde on your keyboard.

The first order of business is setting up a secure data directory away from the document root. Do this:

```
mail# mkdir -p /var/lib/squirrelmail
mail# mkdir -p /var/cache/squirrelmail/attachments
mail# mv data /var/lib/squirrelmail/prefs
```

Now we need to find out the user and group your apache runs as. It used to be “*nobody:nogroup*”, but it varies greatly now. To find out, run this:

```
mail# egrep '^User|^Group' /usr/local/apache/conf/httpd.conf
```

If you are using a pre-installed version of apache, then your `httpd.conf` will be elsewhere, but most likely in `/var/www/conf`.

In the following command substitute the “*www:www*” with whatever the previous command returned, placing the username before the colon, and group after the colon:

```
mail# chown -R www:www /var/lib/squirrelmail/prefs
mail# chmod 0700 /var/lib/squirrelmail/prefs
mail# chown -R www:www /var/cache/squirrelmail/attachments
mail# chmod 0700 /var/cache/squirrelmail/attachments
```

11.2 Configuring SquirrelMail

Using a provided perl script, this is a no-brainer.

```
root@mail:# cd config
root@mail:# ./conf.pl
```

Settings in part 1 are up to you. In part 2 be sure to set:

- Your domain name (in our case *mail.hogwarts.jk*)
- Set it to use *SMTP*. You will need to make sure that we accept relays from 127.0.0.1 – run `edrelays.sh` and make sure we do.
- Set the server to “*courier*”.

There shouldn't be anything in need of changing in part 3. In part 4 you need to do two things:

- Change data directory from “*../data/*” to “*/var/lib/squirrelmail/prefs/*”.
- Change Attachment directory to “*/var/cache/squirrelmail/attachments/*”.

(If you're a non-English speaker and use charset other than iso-8859-1, change that setting, too).

That's it! You may play around with theme settings and such, if you want to. LDAP address books are not covered in this document (see SquirrelMail docs for info), and we will get to talk about plugins later.

12 Apache Configuration

Now it's up to the fun part – we get to dig apache's `httpd.conf`. There are plenty of different ways to configure it, but if you are going to use apache just to run Squirrelmail, a very minimalistic setup will suffice.

12.1 Minimalistic configuration

Open `httpd.conf` in your favorite editor. Don't be dismayed by the size of the file or by the abundance of comments – we will only need to change a few things as default settings will cover most anything. Let's go over the directives you need to change in the order in which they appear in our default `httpd.conf` installation.

- *Port*: The default can be 8080, but you will need to change it to 80
- *ServerAdmin*: `root@mail.hogwarts.jk`
- *ServerName*: Make sure this directive is commented out, since we will be accessing this machine as different domain each time.
- *DocumentRoot*: put `"/var/www/squirrelmail"`. You will also need to change the `<Directory '/usr/local/apache/htdocs'>` or `<Directory '/var/www/htdocs'>` to `<Directory '/var/www/squirrelmail'>` further down.
- *DirectoryIndex*: change it to be `"index.php index.html"`
- *UseCanonicalName*: change to `Off`
- Scroll a while and get to where it says `"AddType application/x-httpd-php3 .php"`. Somewhere around there should be declarations for PHP. Uncomment the one which is marked for PHP 4.x. You don't have to enable the `".phps"` setting, just the `.php`. The line should end up looking like this:
`AddType application/x-httpd-php .php.`

This completes the minimalistic setup of the apache server!
Now run:

```
mail# apachectl start
```

This should fire up your web server (you might have to specify the full path to `apachectl` when executing it). Go to `http://mail.hogwarts.jk/` in your browser and enjoy a nice webmail front end. To log in, use the same login and password you used before.

Since we would like to run Apache at boot time, let's add this line to our `rc.local`:

```
/usr/local/apache/bin/apachectl start
```

If you are using a pre-installed version of apache, then check the `/etc/rc.conf` and find the line that sets whether the web server is started at boot time or not. E.g. on OpenBSD you will need to set `httpd_flags=''` to have it started automatically.

12.2 Other possible configurations

Thick books have been written on the subject of Apache configuration. Naturally, most configurations are out of the scope of this guide. I can only give a few pointers.

If you are planning to use this Apache server for other stuff, you might want to bump all SquirrelMail-related stuff into VirtualHost directives so it doesn't get in the way. There are several ways to do this. For one, create a `_default_` VirtualHost which will handle all requests that don't have their own ServerName allocated. The DocumentRoot for this server would be `'/var/www/squirrelmail'` and therefore anybody accessing it as `mail.domain.tld` will be served this default setting.

You may also set up every `mail.domain.tld` VirtualHost separately if you can stand the hassle. If you choose to do this, then simply point their DocumentRoot's all to the same place where your SquirrelMail installation is.

If you are planning to host regular websites for these domains and want the users to access mail in a special `/mail` subfolder of the `www.domain.tld`, you might want to create a global alias `/mail/` pointing to `/var/www/squirrelmail`. This will save you a headache of creating symlinks in every documentroot.

Overall, there are many ways to configure SquirrelMail so there is only one document root for each virtual domain. You are encouraged to experiment on your own. ;)

12.3 SquirrelMail plugins

One of the greatest features of SquirrelMail is the ability to extend the basic application with additional snippets of code. Many plugins come bundled with SquirrelMail, and you are welcome to investigate them on your own. If you decide to activate SquirrelSpell, then make sure you have `aspell` available on your system.

Plugin installation is usually as easy as just untarring it in the plugins directory, and then running the SquirrelMail's `conf.pl` script, however some plugins will require you to manually edit config files or running installer scripts. Consult the plugin's documentation for more information and installation instructions.

13 Administration Front-end

If you are anything like me, you prefer to have a nice http front-end to ease such mundane tasks as adding and deleting users. Besides, if you are an ISP, you will probably want the domain owners to be able to administer their e-mail

users without involving you. Using the system we have just configured and a nice Vmailmgr Admin plugin I wrote, it is as easy as sneezing in February.

13.1 Installing Ucspi-unix

Very easy:

```
root@mail:# tar xzvf ucspi-unix-0.34.tar.gz
root@mail:# cd ucspi-unix-0.34
root@mail:# make && make install
```

13.2 Installing libmccrypt

There were some changes in *libmccrypt-2.4.19* and onwards, which make it incompatible with *php-4.0.6*, therefore we will be using *libmccrypt-2.4.18*. Get it from <http://mccrypt.hellug.gr/>, untar, compile, and install:

```
mail# tar xzvf libmccrypt-2.4.18.tar.gz
mail# cd libmccrypt-2.4.18
mail# ./configure --prefix=/usr/local
mail# make && make install
```

13.3 Running vmailmgrd

Now it's time to run vmailmgrd. This is done by executing the following command:

```
mail# /usr/bin/unixserver /tmp/.vmailmgrd \
/usr/local/sbin/vmailmgrd 2>&1 | /var/qmail/bin/splogger &
```

Of course, we will want to stick this into our `rc.local` file as well to run it at system startup:

```
/usr/bin/unixserver /tmp/.vmailmgrd /usr/local/sbin/vmailmgrd \
2>&1 | /var/qmail/bin/splogger &
```

That's it, vmailmgrd is up and running.

13.4 Vmailmgr Admin plugin for SquirrelMail

This plugin will allow you to administer your domains without leaving the comforts of your SquirrelMail interface. What's more important, it allows you to designate admins for each of your domains, so the owners of these domains can administer their users without involving you, the ever-busy administrator.

Since security is a very important issue here, we will make use of that libmccrypt library we have just installed. First, though, we will need to recompile PHP to enable the libmccrypt support.

13.4.1 Recompiling PHP4

Go back to the directory with your PHP4 source. Now run this:

```
mail# make distclean
mail# ./configure --with-apxs=/usr/local/apache/bin/apxs \
--with-gettext=/usr/local --with-mcrypt=/usr/local
mail# make && make install
mail# apachectl restart
```

This should activate the new build of PHP with libmcrypt support. Surf to “*mail.hogwarts.jk*” and make sure everything is working.

13.4.2 Getting and installing Vadmin

You can get the latest version of Vmailmgr Admin plugin for Squirrelmail here:

- <http://mricon.com/xmlparse/SM/>

The latest version is v1.0 at the time of writing. Download the plugin, then copy it into your squirrelmail/plugins directory:

```
mail# cp vadmin-v1.0.tar.gz /var/www/squirrelmail/plugins
mail# cd /var/www/squirrelmail/plugins
mail# tar xzvf vadmin-v1.0.tar.gz
mail# rm vadmin-v1.0.tar.gz
mail# cd vadmin
```

Vadmin provides a convenient installer to make the installation of the plugin effortless. To engage the installer, do this:

```
mail# cd scripts
mail# ./installer.sh
```

The `installer.sh` shell script is very verbose and will guide you through the installation process giving you very detailed instructions. It would be silly to repeat them here. When the installer asks you about who should be “elvis”, provide your login handle, which in our case is “*albus@hogwarts.jk*”.

After you have finished installing the plugin, you will need to restart your Apache server one more time so the changes can take effect:

```
mail# apachectl restart
```

Now activate the vadmin plugin by running SquirrelMail’s `conf.pl` script and choosing “plugins”. Activate vadmin and quit the configurator. The plugin is now activated.

After this, you should move the whole “`scripts`” directory somewhere where it’s easy to get to them. Do this by going back to vadmin plugin directory and running the following commands:


```
mail# mv scripts /root/vadmin-scripts
mail# cd /root/vadmin-scripts
```

Now we need to install a domain.

13.5 Adding domains to Vadmin

Let's add *hogwarts.jk* to the list of domains available under vadmin. To do so, make sure you're in your `vadmin-scripts` directory and run the `vaddomain.sh` script:

```
mail# ./vaddomain.sh hogwarts.jk
```

It will create the necessary directories and then will ask you “*would you like to add lowly admins to this domain?*” Answer “no”, since we're going to be the ones doing all administering for this domain. Answer “no” to the next question as well, since we don't want to edit cross-admins yet (we will deal with admins and cross-admins later).

Now go to `http://mail.hogwarts.jk/` and log into SquirrelMail. You will notice that an “Admin” option appeared in the menu line – vadmin has detected that you are one of the admins and gives you an option to log into the administrator interface.⁵ Go there.

You are given an option to log in. If you chose a “USER” option for cross-admin authentication (which is recommended for starters), then type in your mailbox password in the “Password” field, otherwise type in whichever is the password you chose when you were creating the *hogwarts.jk* domain using our `advirt.sh` script.

If you used your mailbox password when authenticating, the next screen will ask you the domain password for *hogwarts.jk*. This is the system password you chose when you were creating this domain. Type it in and click “Proceed”. The password will be saved on the server, but don't worry – it will be saved in an encrypted format and vadmin's security features are strong enough to make the job of hijacking the passwords rather hard.

The next screen will be the Vadmin main menu – click away to explore the convenience of this plugin. Everything should be pretty self-explanatory, plus lengthy descriptions are provided whenever something confusing comes up.

13.5.1 Using `domain_magic.sh` script

If you are adding vadmin to an existing system already running several domains configured with `vmailmgr` and `qmail`, you can run the `domain_magic.sh` script to make them automagically available for use with vadmin.

⁵One of the security features of vadmin is that an “Admin” link will only be shown when the login username matches the http domain name. In other words, if you login as *albus@hogwarts.jk*, you **must** log into `http://mail.hogwarts.jk`. If you login as *albus@hogwarts.jk* and you're accessing squirrelmail at `http://mail.theburrow.jk`, you will **not** see the “Admin” link.

13.6 Adding more domains

Now let's mainstream the process of adding domains. Let's say you want to add two virtual domains to your system. For the purposes of being descriptive, I will make them be *theburrow.jk* and *theministry.jk*.

First, let's edit our *addvirt.sh* script so it calls the vadmin's scripts after creating these domains within the system. Fire up your favorite editor, and close to the top uncomment the line setting up the "VADDDOMAIN" variable. Do the same for the *rmvirt.sh* script, except the variable you'll be looking for is going to be "VDELDDOMAIN".

Now let's add *theburrow.jk* domain:

```
mail# addvirt.sh theburrow.jk
```

When it gets to the question "Would you like to add lowly admins for this domain?" say "y" this time, because we would like "molly@theburrow.jk" to be the administrator of this domain. Answering "y" will bring up your editor (if you didn't set it up, this will bring up vi. Don't panic, just type "i", then type "molly@theburrow.jk", then hit "Esc", and type ":wq" + [Enter]).

The next question will be "Would you like to add cross-admins?". Answer "y". This will bring up the following screen:

```
Current cross-admins for 'theburrow.jk':
-----
--
Handle/killall/exit:
```

Type in *arthur@theburrow.jk* and hit enter. The screen will update and you will see "arthur@theburrow.jk" in the list of cross-admins. Typing "arthur@theburrow.jk" again will delete Mr. Weasley from the list of cross-admins.

Now that you have enabled this domain on your system and within vadmin, go to <http://mail.hogwarts.jk/>. Log in as *albus@hogwarts.jk* (the super-user), then click "Admin". When you see the login screen to our vadmin interface, erase "hogwarts.jk" where it says "Domain" and type in "theburrow.jk". Provide the domain password when it asks for it, and voila – you're in the administrative interface for *The Burrow*.

Now you need to add the "arthur" and "molly" users. Select "add new user", and follow instructions to create the user *arthur@theburrow.jk*, then repeat for *molly@theburrow.jk*. After this, make sure you have "mail.theburrow.jk" domain enabled in your DNS settings and everything is ready for "arthur@theburrow.jk" and "molly@theburrow.jk" to log into SquirrelMail, click on the "Admin" link, and start administering the domain.

Let's finish exploring Vadmin by adding yet another domain. Type:

```
mail# addvirt.sh theministry.jk
```

Answer "y" to the "lowly admins" question, and make *fudge@theministry.jk* your lowly admin. When you are asked about cross-admins, answer "y". This will bring up a window looking like this:

```
Current cross-admins for 'theministry.jk':
```

```
-----
```

```
--
```

```
Handle/killall/exit:
```

Since Arthur Weasley works at the Ministry of Magic, we will want him to be able to administer both *theburrow.jk* and *theministry.jk* domains. Type in: “*arthur@theburrow.jk*” to enable cross-admining of “*theministry.jk*” for Mr. Weasley.

Now go to *mail.hogwarts.jk*, log in, click “Admin”, erase “*hogwarts.jk*”, type in “*theministry.jk*”, provide password, create the “*fudge*” user, then log out, sign out, go to *mail.theburrow.jk*, log in as “*arthur@theburrow.jk*” and click on “Admin”. You will see that you have a choice of administering either “*theburrow.jk*” or “*theministry.jk*” domains.

If this all confuses you, play around with *vadmin* and *vadmin* scripts to familiarize yourself with them. Cross-admins are only useful if you have any clients owning several domains and they find it a hassle to have to log out of one mailbox and log into another one just to administer their users. If you don't have any clients who own several domains, you will only need to add “lowly admins” and not bother with cross-admins.

Vadmin is not capable of screwing up your system, so don't be afraid to play with it. The most it will do is make a domain operational or not operational from within the *vadmin* plugin, so try it away. To find out what other scripts in the *vadmin-scripts* directory do, run this command while in that directory:

```
mail# ./whatsthisdo.sh scriptname.sh
```

or just:

```
mail# ./whatsthisdo.sh
```

to see descriptions for them all.

(In case you are wondering why I didn't just make a web front-end interface to adding domains, admins, and cross-admins, then the answer is – there is no really secure way of doing it from the web: too many possible exploits are available if I was to do something like this, therefore I didn't.)

14 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 out there.

14.1 Why this is not recommended for large systems

The only reason this is not recommended for large systems is because Squirrel-Mail 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.

14.2 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 *icon@duke.edu* so I can make the correction or read your comments. You may also check my website at <http://mricon.com/SM/guide/> for the latest version of this document.

14.3 Thank you and good luck! ;)

If you found this Guide useful, please let me know by executing:

```
mail# uname -a | mail icon@duke.edu -s 'Thanks'
```

Sincerely, Konstantin Riabitsev, aka Mr.Icon.

A Scripts

The scripts presented in this section are all available for download on my website: <http://www.mricon.com/SM/guide/>. They have been tested on OpenBSD-3.0 and should work on most BSD systems with a few small changes, if any.

A.1 edrelays.sh

```
#!/bin/sh
#
# edrelays.sh
# -----
# This script makes editing selective relaying rules easy. It's
# distributed as a part of Qmail-VmailMgr-Courier-SquirrelMail
# Guide, http://www.mricon.com/SM/guide/.
#
# Copyright (c) 2001-2002 by Konstantin Riabitsev <icon@duke.edu>
# Licensed under the GNU GPL. For full terms see http://www.gnu.org/
#
# $Id: edrelays.sh.bsd,v 1.1.1.1 2002/03/13 00:29:41 graf25 Exp $
#
# @author Konstantin Riabitsev <icon@duke.edu> ($Author: graf25 $)
# @version $Date: 2002/03/13 00:29:41 $
#

# RULESPLAIN Is the variable that sets the location of the file with
#             relaying rules in a plain format.
# RULESTEMP  Is the variable setting the temporary file.
# RULESCDB   Is the location of the compiled cdb file.
# TCPRULES   Is the location of "tcprules" command.

RULESPLAIN="/var/qmail/control/relay.rules"
RULESTEMP="/var/qmail/control/relays.tmp"
RULESCDB="/var/qmail/control/relays.cdb"
TCPRULES="/usr/local/bin/tcprules"

# See if we have an EDITOR set up.
if test -z "$EDITOR"; then
    EDITOR="vi"
fi

$EDITOR $RULESPLAIN
$TCPRULES $RULESCDB $RULESTEMP < $RULESPLAIN
echo "Compiled rules placed in $RULESCDB."
```

A.2 advirt.sh

```
#!/bin/sh
#
# advirt.sh
# -----
# This script aims to greatly simplify the adding of new domains
# to a BSD system running qmail and vmailmgr. It's distributed
# as part of the Qmail-Vmailmgr-Courier-Squirrelmail guide,
# http://www.mricon.com/SM/guide/.
#
# Copyright (c) 2001-2002 by Konstantin Riabitsev <icon@duke.edu>
# Licensed under the GNU GPL. For full terms see http://www.gnu.org/
#
# $Id: advirt.sh.bsd,v 1.1.1.1 2002/03/13 00:29:40 graf25 Exp $
#
# @author Konstantin Riabitsev <icon@duke.edu> ($Author: graf25 $)
# @version $Date: 2002/03/13 00:29:40 $
#
#
# DGID:      Group name to use when creating new domains (dom)
# DHOME:    Base directory for all virtual domains (/home/dom)
# QHOME:    Your qmail installation directory (/var/qmail)
# VSETUP:   The location of Vmailmgr's "vsetup" command
#           (/usr/local/bin/vsetup)
# VADDDOMAIN: The location of vadmin's vaddomain.sh script.
#           If you don't use vadmin, comment the line out.
DGID="dom"
DHOME="/home/dom"
QHOME="/var/qmail"
VSETUP="/usr/local/bin/vsetup"
#VADDDOMAIN="/root/vadmin-scripts/vaddomain.sh"

# The original sh doesn't provide us with $UID.
if test ! $UID; then UID='id -u'; fi

# Check if we're root.
if test $UID != "0" ; then
    echo "Error: Must be root"
    exit 1
fi

# Check for omitted params
if test -z $1; then
    IAM='basename $0'
```

```
    echo "Usage: $IAM newdomain.com"
    exit 1
fi

# Check if we will need to use "useradd" or "pw useradd" for
# FreeBSD and Co.
if test -f /usr/sbin/useradd
    then USEPW="no"
    elif test -f /usr/sbin/pw;
then USEPW="yes"
    else
    # Something odd here.
    echo "Couldn't find either 'useradd' or 'pw'. Aborting."
    exit 1
fi

# Check if this domain already exists in virtualdomains.
if test -f $QHOME/control/virtualdomains; then
    if egrep -q "^$1:" $QHOME/control/virtualdomains; then
    echo "Error: Domain $1 exists"
    exit 1
    fi
fi

NEWDOM=$1

# truncate to 8 symbols
BASEUSER='echo $NEWDOM | sed 's/\./_/g; s/\((.....\)*/\1/'
echo -n "Base Username [$BASEUSER]: "
read REPLY
if test ! -z $REPLY; then BASEUSER=$REPLY; fi

# Check if we already have this baseuser. Loop if necessary.
while egrep -q "^$BASEUSER:" /etc/passwd; do
    echo "Base username $BASEUSER already exists. Please provide another."
    echo -n "Base Username [$BASEUSER]: "
    read REPLY
    if test $REPLY; then BASEUSER=$REPLY; fi
done

echo "Creating new domain '$NEWDOM'."
UFLAGS="-d $DHOME/$NEWDOM -g $DGID -m"
if test $USEPW = "yes"
    then pw useradd $BASEUSER $UFLAGS
    else useradd $UFLAGS $BASEUSER
```

```
fi
echo "Domain base user created in $DHOME/$NEWDOM."
echo "Please provide domain password for VmailMgr."
passwd $BASEUSER
echo -n "Adding domain to control/virtualdomains..."
echo "$NEWDOM:$BASEUSER" >> $QHOME/control/virtualdomains
echo "done"
echo -n "Adding records to control/rcpthosts..."
echo "$NEWDOM" >> $QHOME/control/rcpthosts
echo "done"
echo "Setting up the domain dir for vmailmgr..."
su $BASEUSER -c "$VSETUP"
echo "...done"
echo -n "Restarting Qmail..."
kill `ps ax | grep qmail-lspawn | grep -v grep | awk '{print $1}'`
$QHOME/rc &
echo "done"
echo
if test ! -z "$VADDDOMAIN"; then
    $VADDDOMAIN $NEWDOM
fi
echo "All done! Domain $NEWDOM created."
```


A.3 rmvirt.sh

```
#!/bin/sh
#
# rmvirt.sh
# -----
# This script aims to greatly simplify the removing of virtual domains
# from a BSD system running qmail and vmailmgr. It's distributed as
# part of the Qmail-Vmailmgr-Courier-Squirrelmail guide,
# http://www.mricon.com/SM/guide/.
#
# Copyright (c) 2001-2002 by Konstantin Riabitsev <icon@duke.edu>
# Licensed under the GNU GPL. For full terms see http://www.gnu.org/
#
# $Id: rmvirt.sh.bsd,v 1.1.1.1 2002/03/13 00:29:45 graf25 Exp $
#
# @author Konstantin Riabitsev <icon@duke.edu> ($Author: graf25 $)
# @version $Date: 2002/03/13 00:29:45 $ #
#
# QHOME:      Your qmail installation directory (/var/qmail)
# VDELDOMAIN: Location of vadmin's vdeldomain.sh script.
#             If you don't use vadmin, comment that line out.
QHOME="/var/qmail"
#VDELDOMAIN="/root/vadmin-scripts/vdeldomain.sh"

# The original Bourne sh doesn't provide us with $UID.
if test ! $UID; then UID='id -u'; fi

# Check if we are who we should be.
if test $UID != 0 ; then
    echo "Error: Must be root"
    exit 1
fi

# Check for omitted parameters.
if test -z "$1"; then
    IAM='basename $0'
    echo "Usage: $IAM domain.com"
    exit 1
fi

# Check if we will need to use "userdel" or "pw userdel" for
# FreeBSD and Co.
if test -f /usr/sbin/userdel
    then USEPW="no"
```

```
        elif test -f /usr/sbin/pw;
then USEPW="yes"
        else
        # Something odd here.
        echo "Couldn't find either 'userdel' or 'pw'. Aborting."
        exit 1
fi

# Check if this domain actually exists.
if ! egrep -q "^$1:" $QHOME/control/virtualdomains; then
    echo "Error: No such domain $1"
    exit 1
fi

# get the baseuser.
BASEUSER='egrep "^$1:" $QHOME/control/virtualdomains | sed 's/^.*://''

# Check if there are discrepancies between qmail lists and /etc/passwd.
if egrep -q "^$BASEUSER:" /etc/passwd; then
    echo "This will delete '$1' and all its users."
    echo -n "Proceed? y/[n]: "
    read YN
    if test -z "$YN"; then YN="n"; fi
    if test $YN != "y" ; then
echo "Aborted."
exit 1
    fi
else
    echo "Error: Cannot remove domain. Base User not found in system files"
    exit 1
fi

OLDDOM=$1
echo -n "Removing the domain with all user files..."
UFLAGS="-r"
if test $USEPW = "yes"
    then pw userdel $BASEUSER $UFLAGS
    else userdel $UFLAGS $BASEUSER
fi
echo "done"
echo -n "Removing domain from qmail's control files..."
cd $QHOME/control
grep -v $OLDDOM virtualdomains > virt.new
grep -v $OLDDOM rcpthosts > rcpt.new
mv virt.new virtualdomains
mv rcpt.new rcpthosts
```

```
echo "done"
echo -n "Restarting qmail..."
kill `ps aux | grep qmail-lspawn | grep -v grep | awk ' { print $2 } '`
$QHOME/rc &
echo "done"
echo
if test ! -z "$VDELDOMAIN"; then
    $VDELDOMAIN $OLDDOM
fi
echo "All done! Domain $OLDDOM deleted."
```

B GNU Free Documentation License

Version 1.1, March 2000

Copyright © 2000 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other written document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

B.1 Applicability and Definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The “Document”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “you”.

A “Modified Version” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “Secondary Section” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “Invariant Sections” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The “Cover Texts” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A “Transparent” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not “Transparent” is called “Opaque”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, \LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The “Title Page” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

B.2 Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

B.3 Copying in Quantity

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

B.4 Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together

with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).

- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- Include an unaltered copy of this License.
- Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties – for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

B.5 Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

B.6 Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection,

provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

B.7 Aggregation With Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an “aggregate”, and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document’s Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

B.8 Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

B.9 Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

B.10 Future Revisions of This License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be

similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST. A copy of the license is included in the section entitled "GNU Free Documentation License".

If you have no Invariant Sections, write "with no Invariant Sections" instead of saying which ones are invariant. If you have no Front-Cover Texts, write "no Front-Cover Texts" instead of "Front-Cover Texts being LIST"; likewise for Back-Cover Texts.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.