

# Setting up Redhat 9 as a failover mail server with the ability to read queued mail using the Chainmail Milter

Author: Stuart Colville  
Version 1.3  
Last updated 14<sup>th</sup> October 2004

**WARNING: The information within this guide is provided as is; acting upon this information is carried out at your own risk.**

This guide is intended to give an insight into how straight forward it is to set-up a Linux box as a back-up to your existing mail solution. When the primary mail server goes down mail cannot be received. Most ISPs allow for this by spooling mail for you up on their own servers. This is great but the drawback is if your server is down for any length of time you will need to be able to get to read the mail or your business could miss some important information.

This guide provides a solution to this problem and shows how you can set-up a simple linux box running sendmail to enable your business to read spooled mails and then send mail onto your primary server when the primary mail server is back up and running.

The previous version of this document referred to utilizing mqcontrol <http://sourceforge.net/projects/mqcontrol/> as a method for reading spooled mail from your sendmail server. In practice I found this to not work particularly well as the program seemed to stop responding when the queue built up.

I thought up several ways to make it possible to read the queued mail. One was a web-based script in PHP that allowed you to read the mail by parsing the queue files directly. Whilst this worked I thought there had to be a better way.

I then made a shell script that processed the queue files in such a way that the mail could be re-directed to local accounts on the server.

This method worked and it respected sendmail's own file locking using qtool.pl to move the queue files around.

The drawback that this queue manipulation relied on being scheduled via cron and this meant that you unless you ran it frequently you wouldn't be able to get mail redirected as close to real time.

The next step was to look at all of the inbuilt sendmail tools and functionality and see if there was an even simpler method that would get mail to local accounts in real time.

I then found the Chainmail Milter. <http://chainmail.sourceforge.net/> This Mail filter (Milter) would redirect mail based on a simple rule set.

This document covers setting up a redhat server, configuring sendmail and the Chainmail Milter to redirect incoming mail to local accounts.

## Overview

1. Install Redhat
2. Add MX records to your DNS
3. Installing the Chainmail Militer
  - a) Obtaining the source code for chainmail
  - b) Checking sendmail-devel is installed
  - c) Compiling and installing chainmail.
4. Modify the command search paths for convenience.
5. Modifying the host file
6. Modify the sendmail config
7. Configuring and starting Chainmail

### **1. Installing Redhat**

For the purposes of the failover sendmail server the server hardware needn't be the latest spec. For this example I used a P2 350Mhz desktop with 128MB Ram.

Install Redhat with the minimum amount of options. I would recommend adding in any of the text based tools that you are most comfortable with. I prefer "Pico" as my default text editor and as we are going to need to send mails to test our configuration it would be useful to install "Pine" (a text based mail client).

The basic requirements are:

Sendmail

Sendmail-devel

Programming tools - to be able to compile from source code

Text tools

Etc...

Only install a graphical UI if you are going to need to use graphical tools. For this set-up you only need a command line but if you wish to see what's going on then by all means install the desktop of your choice. You can always turn it off later by altering the default run level in /etc/inittab to:

```
id:3:initdefault
```

Use the command telinit to change the run-level on the fly

```
telinit 3
```

Alternatively you can restart your machine to bring it up in text mode.

## 2. Editing the MX records

Now that Redhat is up and running it's a good idea to add an MX record to your existing ones so that if your primary server goes down mail is sent to this Linux box.

Say your existing Mx records are

10 mail.yourdomain.com

you need to add a second record e.g:

20 backup.yourdomain.com

This will give you

10 mail.yourdomain.com

20 backup.yourdomain.com

You will also need to add DNS entries that correspond with the new server so that backup.yourdomain.com points to the IP address of your back-up mail server.

Once this is complete (and once changed these changes can take up to 48 hours to propagate). When the mail server at the IP address relating to mail.yourdomain.com is not available mail will be sent to backup.yourdomain.com

## 3. Installing the Chainmail Milter

To be able to run the Chainmail Milter you will need to check that you have a version of sendmail compiled with Milter support. By default the version of sendmail with RH9 has this support already.

To check Milter support (for other distros) type:

```
# sendmail -d0.1 -bt < /dev/null
```

You should see something like this:

```
Version 8.12.11
Compiled with: DNSMAP LOG MAP_REGEX MATCHGECOS MILTER MIME7TO8 MIME8TO7
              NAMED_BIND NETINET NETUNIX NEWDB PIPELINING SCANF TCPWRAPPERS
              USERDB XDEBUG

===== SYSTEM IDENTITY (after readcf) =====
(short domain name) $w = myhost
(canonical domain name) $j = myhost.mydomain.com
(subdomain name) $m = mydomain.com
(node name) $k = myhost.mydomain.com
=====
```

Providing you have MILTER listed as shown above in red. You can continue. If not you will need to google for help on how to install sendmail which has milter support. This is beyond the scope of this article.

Next step is to download and install Chainmail and it's dependencies.

First we need to download the sourcecode the following explains how to do this from the command line.

### **a) Obtaining the source code for chainmail**

```
# cd /  
# wget http://heanet.dl.sourceforge.net/sourceforge/chainmail/chainmail-0.3.tar.gz
```

You will then see the following output

```
--13:14:32-- http://heanet.dl.sourceforge.net/sourceforge/chainmail/chainmail-0.3.tar.gz  
=> `chainmail-0.3.tar.gz'  
Resolving heanet.dl.sourceforge.net... done.  
Connecting to heanet.dl.sourceforge.net[193.1.219.87]:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 118,967 [application/octet-stream]  
  
100%[=====>] 118,967      29.58K/s  
ETA 00:00  
  
13:14:36 (29.58 KB/s) - `chainmail-0.3.tar.gz' saved [118967/118967]
```

No get the milteroo sourcecode:

```
# wget http://heanet.dl.sourceforge.net/sourceforge/chainmail/milteroo-0.1.tar.gz
```

You will then see the following output

```
--13:15:16-- http://heanet.dl.sourceforge.net/sourceforge/chainmail/milteroo-0.1.tar.gz  
=> `milteroo-0.1.tar.gz'  
Resolving heanet.dl.sourceforge.net... done.  
Connecting to heanet.dl.sourceforge.net[193.1.219.87]:80... connected.  
HTTP request sent, awaiting response... 200 OK  
Length: 157,181 [application/octet-stream]  
  
100%[=====>] 157,181      29.68K/s  
ETA 00:00  
  
13:15:21 (29.68 KB/s) - `milteroo-0.1.tar.gz' saved [157181/157181]
```

This will download the two packages to /

Next you will need to untar the source files.

```
# tar -xzf chainmail-0.3.tar.gz
```

```
# tar -xzf milteroo-0.1.tar.gz
```

This will have unpacked the source code into two directories: chainmail-0.3 and milteroo-0.1

## **b) Checking sendmail-devel is installed**

Now you can compile and build the sourcecode. Before you can do this you need to check that you have the sendmail-devel package installed. Which you should have done if you have installed redhat from scratch. However if you haven't it's not hard to rectify..

To find out if you have it run the following command.

```
# rpm -q sendmail-devel
```

If you get a message saying "package sendmail-devel is not installed" you will need to download and install the version correct for your version of sendmail. To find out which version you need type the following:

```
# rpm -q sendmail
```

My Redhat test server responded the following because I have upgraded sendmail. If this happens make a note of the most recent version. If you only have one version number in the response then use that.

```
sendmail-8.12.8-4  
sendmail-8.12.8-9.90
```

In my case I need to get sendmail-devel-8.12.8-9.90.i386.rpm

```
# wget ftp://updates.redhat.com/9/en/os/i386/sendmail-8.12.8-9.90.i386.rpm
```

Should you have trouble using wget try the passive switch e.g.

```
# wget --passive-ftp <sourceurl>  
# wget --passive-ftp ftp://updates.redhat.com/9/en/os/i386/sendmail-devel-8.12.8-9.90.i386.rpm
```

You should see something like this:

```
--13:48:17-- ftp://updates.redhat.com/9/en/os/i386/sendmail-devel-8.12.8-9.90.i386.rpm  
=> `sendmail-devel-8.12.8-9.90.i386.rpm'  
Resolving updates.redhat.com... done.  
Connecting to updates.redhat.com[209.132.176.40]:21... connected.  
Logging in as anonymous ... Logged in!  
==> SYST ... done. ==> PWD ... done.  
==> TYPE I ... done. ==> CWD /9/en/os/i386 ... done.  
==> PASV ... done. ==> RETR sendmail-devel-8.12.8-9.90.i386.rpm ... done.  
Length: 96,132 (unauthoritative)
```

```
100%[=====>] 96,132      24.06K/s
ETA 00:00
```

```
13:48:23 (24.06 KB/s) - `sendmail-devel-8.12.8-9.90.i386.rpm' saved [96132]
```

The next step is to install sendmail-devel

```
# rpm -ivh sendmail-devel-8.12.8-9.90.i386.rpm
```

You should see the following:

```
Preparing...
##### [100%]
 1:sendmail-devel
##### [100%]
```

If there are any error messages about missing dependencies the you will need to locate and install those dependencies using the rpm command. (Type "man rpm" for more info)

### **c) Compiling and installing chainmail.**

Now we can finally get round to compiling chainmail and the milteroo package.

As milteroo is a dependency of chainmail we must carry that out first:

```
# cd milteroo-0.1
# ./configure
# make
# make install
```

Next install chainmail

```
# cd ../chainmail-0.3
# ./configure
# make
# make install
```

Once this is completed without any errors Chainmail has been installed successfully.

To make sure chainmail starts when the server boots do the following. This assumes you are still in the chainmail-0.3 directory.

```
# cp chainmail.init /etc/rc.d/init.d/chainmail.init
```

Before you test chainmail there's a few tasks we need to carry out to complete the sendmail configuration.

## **4. Customizing the command search path**

That's that done. Now lets make out lives easier by adding /usr/local/sbin to the BASH command search paths.

To run chainmail we would need to type:

```
# /usr/local/sbin/chainmail
```

This is a bit long winded, additionally the sendmail command is found in /usr/sbin and the service command is in /sbin so it's a good idea to add both to facilitate controlling and restarting sendmail!

To fix this you need to add /usr/sbin /usr/local/sbin and /sbin to the command search path:

To view your existing paths type:

```
# cd /  
# env
```

Look at the line PATH and you will see all of the command paths separated by ":"

To append a path to the end to the path we need to type:

```
# PATH=$PATH:/usr/sbin:/usr/local/sbin:/sbin
```

The \$PATH variable covers the existing path structure if we left that out we would overwrite all of the other paths with /usr/sbin:/usr/local/sbin:/sbin

Type env again to check the path has been amended.

## **5. Modifying the host file**

Great the queue works but when mail arrives in the queue when the primary server is down we need to make sure that it will get to the primary server when it comes back online.

First we need to tell the linux box the ip address of the main server.

Type:

```
# cd /  
# pico /etc/ hosts
```

Then add another line to the hosts file with the internal IP address the hostname and an alias for your mail server.

This would be for example

```
192.168.10.25 mail.yourdomain.com mail
```

## 6. Modify the sendmail config

That was easy! Now we just need to sort out the sendmail config so it knows to send mail directed to yourdomain.com internally rather than to try and send it externally.

This will be done by using mailertable.

Type:

```
# pico /etc/mail/mailertable
```

and then add the following line (changed to match your settings)

```
yourdomain.com smtp: [mail.yourdomain.com]
```

Next we need to add this to the mailertable database.

Type:

```
# makemap hash /etc/mail/mailertable.db < /etc/mail/mailertable
```

Finally it's necessary to add the domain you are acting as a backup for into the relay configuration.

In this case I have used the Access Database.

Type:

```
# cd /etc/mail  
# pico access
```

And then append the following lines to the bottom of your access file.

```
To:yourdomain.com RELAY
```

Now you should have something like this:

```
# by default we allow relaying from localhost...  
localhost.localdomain RELAY  
localhost RELAY  
127.0.0.1 RELAY  
To:yourdomain.com RELAY
```

Now as you can see it would have been simpler just to put:

```
yourdomain.com RELAY
```

However, using To: and Connect only allows mail to be relayed *to* yourdomain.com not *from* it. Maybe this is over the top but I prefer the door to be firmly shut rather than slightly ajar!

Now we just need to put these changes into the access.db file.

```
# makemap hash access < access
```

Now there's one last task to perform. Sendmail by default only listens on Port 25 for connections from localhost. Now it won't be much good if you couldn't receive mail from elsewhere would it!

To make this possible you need to delve into the sendmail mc file. Now I recommend backing up the sendmail.cf file incase you need to restore the original. Let's be safe OK!

Type:

```
# cd /etc/mail  
# cp sendmail.cf sendmail.cf.old
```

Now we need to edit the sendmail.mc file (the file that sendmail.cf is generated from) to allow mail to come in from hosts other than localhost!

Type:

```
# pico sendmail.mc
```

Then comment out the line that looks like this:

```
DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
```

So it looks like this (careful that's a lowercase "L" not a number "1" after "dn"!!!!)

```
dnl # DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')dnl
```

At this stage we also need to add a line to sendmail.mc so that sendmail knows to use the chainmail milter.

```
INPUT_MAIL_FILTER(`chainmail', `S=unix:/var/run/chainmail.sock, F=T')dnl
```

As far as where the above line should be I put mine underneath the line that reads:

```
EXPOSED_USER(`root')dnl
```

Save the changes.

Now that's done you need to create a new sendmail.cf file (make sure you're still in /etc/mail)

```
# m4 sendmail.mc > sendmail.cf
```

Now restart the sendmail service to make the changes complete

Type:

```
# service sendmail restart
```

## 7. Configuring and starting Chainmail

Before chainmail can be started it is necessary to set-up the configuration file:

The default config is read from /etc/chainmail.conf so you will need to copy the installed one to there for it to be read.

```
# cp /usr/local/etc/chainmail.conf /etc/chainmail.conf
```

Next open the config file with your favorite text editor for editing.

```
# pico /etc/chainmail.conf
```

Now add the following configuration for **every user**. If you have a lot of users try setting up a shell script to generate this config file.

```
Rule "user1"  
If Rcpt "user1@yourdomain.com"  
Action AddRcpt " user1@server.yourdomain.co.uk"  
  
Rule "user2"  
If Rcpt "user2@yourdomain.com"  
Action AddRcpt " user2@server.yourdomain.co.uk"
```

When mail comes in for [user1@yourdomain.com](mailto:user1@yourdomain.com) the chainmail milter will add a recipient [user1@server.yourdomain.co.uk](mailto:user1@server.yourdomain.co.uk) This means that a copy of all mail sent is copied to the local user with the same name.

### **IMPORTANT:**

For this to work you will need to add users to the server for every email address/mailbox you wish to be able to view. Type "man useradd" if you need to know more about how to do this. Again a shell script could come in handy if you have lots of users.

Before chainmail will do anything you will need to start it up.

To run chainmail just type

```
# chainmail
```

If the following error comes up.....

```
chainmail: error while loading shared libraries: libmilter.so.0: cannot open shared object file: No such file or directory
```

.....You will need to run the following command:

```
# ln -s /usr/local/lib/libmilter.so.0 /usr/lib/libmilter.so.0
```

Now try running chainmail again:

```
# chainmail
```

Now set-up an email client such as the excellent Mozilla Thunderbird to point to your back-up mail server. Personally I use IMAP connections for this but POP will do.

*Note: you will have to have set-up a local account on the server for this user before you do this.*

Once you have configured your mail client. Try using TELNET to connect to your back-up mailserver to test the configuration.

```
TELNET server.yourdomain.com 25
mailfrom:test@gmail.com
rcptto : user1@yourdomain.com
data
.
```

You should find if everything is working that a message arrives in the inbox of the local machine and your main email server.

The next test you need to do is to make sure that mail is redirected when the main server is down.

When no-one needs your main email server shutdown the SMTP service.

Send a mail from an external webmail account to [user1@yourdomain.com](mailto:user1@yourdomain.com)

You should see a message appear in your back-up mail account via your back-up mx. If you navigate to `/var/spool/mqueue` and type `dir` you should see some queue files sitting there.

```
# cd /var/spool/mqueue
# dir
```

You should see some output like this:

```
dfi2I1NPN0017668 qfi2I1NPN0017668
```

Now start your main email server's SMTP service again.

On you back-up mx server login via ssh as root and type the following command.

```
# sendmail -q
```

This will flush the mail queue immediately and send the mail through to your main email server. Open up your usual mail client and you should see the same mail you saw on your back-up mail client sitting there.

If all is working then it looks like you got there!

## **REFERENCES**

[Sendmail.org website](http://www.sendmail.org/)

<http://www.sendmail.org/>

[System Administration: Linux as a Backup E-mail Server](http://www.linuxjournal.com/article.php?sid=2469) - August 01, 1998 by John Blair

<http://www.linuxjournal.com/article.php?sid=2469>

[Using 'mailertable' in Sendmail](http://freebsd.peon.net/tutorials/16/) - 24 August, 2001 - written by David van Geyn

<http://freebsd.peon.net/tutorials/16/>

[Chainmail at sourceforge](http://chainmail.sourceforge.net)

<http://chainmail.sourceforge.net>