Playing with Sendmail in Fedora

Created by Mark Dickinson, March 2009

Table of Contents

Introduction

This document is intended for those who run a network of **Fedora** Linux servers at home and want to be able to change from the default Mbox mail format to the Maildir format, and to forward mail using sendmail from their servers to one central place, so mail only needs to be checked on that one server.
It also covers the minimal install of Dovecot as a mail server so you can use Windows POP3 clients to retrieve the mail.

It does NOT cover setting up an email gateway to send emails to the internet servers, it is specifically for consolidating emails within an internal environment (including in the last section retrieving emails from ISP POP3 servers into your local IMAP server if you want to go that far).

It was created based upon what I needed to do to achieve this, and may not be best practises. Sendmail is used for mail forwarding as that is normally installed and running by default on Fedora servers anyway, albiet only accepting mail from localhost in the default configuration.

While it is intended for Fedora users it may be of use to users of other distributions also.

To work your way through this guide is is best to just start at the top and work through it if you want all the features discussed here, you may not want them all though.

- You do not need to use the Maildir format to forward mail to another server as you can do that with Mbox format quite happily, so you may skip the section on *Configuring Sendmail to use Maildir format using Procmail* if you don't want that.
- Likewise if you do not want to forward mail but only setup the Maildir format then you need only do the first step on the server(s) of choice to which is *Configuring Sendmail to use Maildir format using Procmail* to configure procmail and skip all the rest.
- And if you don't want to be able to retrieve your mail using a desktop POP3 client such as Thunderbird then you can completely skip everything in the Dovecot section.

And a brief explaination of Mbox and Maildir formats would be

- the Mbox format has been around for a long time and is the old standard. In this setup all emails for a user exist in a single file (well two, /var/spool/mail/userid for unread, normally ~/Mail for saved/read mail). The issue with this is that when you have more than a few hundred emails processing can geta little slow
- the Maildir format has a seperate file for each email, which allows for easier implementation of such things as email folders to categorise your emails

As the Mbox format has been around for such a long time it is still the default for many unix command line email readers, although most can also be asked to use Maildir mailboxes.

Disclaimer

This document is based on the steps I found through trial and error, and a lot of searches on google, to achieve the result I wanted. It is probably not a 'best practises' solution; but it works.
As everything found on the web, you use this document at your own risk and the author prvides no support.

Important: The environment discussed

This document discusses how I implemented email forwarding using sendmail in my personal environment which is comprised of two logically seperate networks

- 192.168.1.* - the normal internal network facing the DSL internet router, normally home networks will be on this range and be able to see each other **but I block my servers from seeing each other on this range** in the server firewall rules; you will not see this mentioned in any configuration examples
- 169.254.218.* - an internal network address range I use for my internal servers to communicate across, sometimes with dedicated cards and sometimes just ip aliases on shared cards; and to some even only one way traffic (I like to complicate things).

Thats why in the examples you will see I get things to listen on explicit addresses rather than all addresses, never listen on all available network interfaces.

Anyway for the purposes of this document on setting up mailforwarding in your own internal network.

- The examples use 169.254.218 ranges, adjust to suit your own range
- Your ISP provided router is not secure, do not allow sendmail to listen on any 192.168.1 address, if you do so you so so at your own risk
- I also do not allow ant SMPT (sendmail) traffic to the internat so I am safe to play, don't play unless you can guarantee the same thing

Remember, this is a walkthrough for home users to forward email internally, it should have no internet facing capability.

If you are a business that needs to listen for email from the internet this walkthrough is not for you.

The server examples used are

- osprey-private: 169.254.218.182. the server to be used to accept all the emails
- falcon-private: 196.254.218.183 a server used to send emails to the 'central' email server

At the start of this 'project' those names were the only entries in the hosts files, if you read on you will find the other entries I needed to add to /etc/hosts to get sendmail to forward emails.

All the changes needed are really minimal, but as I sepnt so long surfing the web for a solution on how to implement this simple solution I have put my solution into a single document for my later reference should I need it again.

Configuring Sendmail to use Maildir format, using Procmail

This is actually incredibly easy to do in a Fedora distribution as Sendmail is already configured to use Procmail (at least on all the FC8 and FC10 distributions I was playing with it was), just with no Procmail configuration file so no actions were being made by Procmail. So there are no sendmail changes themselves required; just the creation of the default Procmail configuration.

However, if you switch from Mbox to Maildir format you can expect any unix command line mail client you use to suddenly stop seeing your email, as by defauly most of them expect the Mbox format. Changing you command line client to use Maildir format will be a client specific configuration change and you should refer to the website of whatever client you use. In this document I **only** cover changing the mutt client as that is what I use so had to change.

Configuring Sendmail to use Procmail

Couldn't be simpler. As Sendmail is already configured with procmail support in Fedora distributions so you just need to create the global /etc/procmailrc file, and restart sendmail.

The commands below are all you need to achieve that on Fedora, after which your emails will be delivered to a new folder named Maildir in your home directory (~/Maildir).

```
The commands to be entered
      cat << EOF > /etc/procmailrc
      # Sendmail invokes procmail to use this
      MAILDIR=$HOME/Maildir/
      DEFAULT=$MAILDIR
      EOF
      service restart sendmail


TO TEST
      mail -s “test email” yourunixuserid
      enter lots of text
      .
      [ENTER]    if you get a cc field just enter again.

      A new file should appear in ~/Maildir/new, thats the email.
```

All done, just like that.

If it didn't work, perhaps your distribution wasn't setup with procmail enabled.
Tto check grep for “procmail” in file /etc/mail/sendmail.mc, there should be an entry beginning FEATURE(local_procmail …..

```
#grep -i procmail /etc/mail/sendmail.mc
sendmail.mc:FEATURE(local_procmail, `', `procmail -t -Y -a $h -d $u')dnl
sendmail.mc:MAILER(procmail)dnl
```

If it isn't there please google, I'm covering the setup on Fedora not on other distributions.

Changes to the the unix command line mutt mail client configuration

I only cover whats needed for the **mutt** client here, as that the one I use and this document is about configuring your mail environment, not all your clients.
As mutt is available with Fedora distributions you can test that you can read the previous test email with this anyway.

The **mutt** website solved my problem here, they have a page on the recommended changes needed for configuring mutt to usethe Maildir format mail.

They require that it be placed in the users .muttrc file; I'm not sure if there is an equivalent global file.

For the purposes of testing, and reading the test email send from above; just create a .muttrc file for yourself and run mutt to make sure it can retrieve/read the email.

The parameters here are from March 2009, you should check the mutt website for any updates.

```
cat << EOF > ~/.muttrc
set mbox_type=Maildir
set folder="~/Maildir"
set mask="!^\\.[^.]"
set mbox="~/Maildir"
set record="+.Sent"
set postponed="+.Drafts"
set spoolfile="~/Maildir"
mailboxes `echo -n "+ "; find ~/Maildir -maxdepth 1 -type d -name ".*" -printf "+'%f' "`
macro index c "<change-folder>?<toggle-mailboxes>" "open a different folder"
macro pager c "<change-folder>?<toggle-mailboxes>" "open a different folder"
macro index C "<copy-message>?<toggle-mailboxes>" "copy a message to a mailbox"
macro index M "<save-message>?<toggle-mailboxes>" "move a message to a mailbox"
macro compose A "<attach-message>?<toggle-mailboxes>" "attach message(s) to this
message"
EOF

mutt
```

Configuring Sendmail to allow mail forwarding

Changes needed in /etc/hosts on all servers for this to work

I'll cover /etc/hosts first as while the simplest it gave me the most trouble.
In my internal network I don't use DNS, just the hosts file, and to get sendmail to try to send mail I needed a few extra entries added, located from trawling through error messages so to save you the time... My internal network domain is still the default localdomain, which is what is used in these examples.

As I use /etc/hosts everywhere instead of a DNS, and sendmail ***requires*** a fully qualified network name on emails, the /etc/hosts files on all servers needed to be updated to include and extra name with the .localdomain appended... **however read the section on the sending server carefully**, each sending server needs two new names added.

These are my /etc/hosts file (well, the relevant bits). The server sending the the email is falcon-private, the server receiving it is osprey-private.

On falcon (the sending server)

```
169.254.218.183   falcon-private falcon-private.localdomain
169.254.218.182 osprey-private osprey-private.localdomain osprey-private.localdomain.
```

The key thing to notice here is that on the last line in addition to the normal default name I had to add an entry ***for both*** servername.localhost and servername.localhost. (that last has an extra **dot** at the end). I don't know if it is an issue with sendmail or just the Fedora setup but you do need both.

So you need to add both entries for the server you wish to send email to at the end of your /etc/hosts email target server name entry on every server you are likely to want to send email from to the target server; sendmail needs them both.

The first entry where the sender gets a localdomain suffix is required to manage mail bounceback if needed.

On osprey (the receiving server)

```
169.254.218.183 falcon-private falcon.localdomain
169.254.218.182 osprey-private osprey-private.localdomain
```

Needs an entry servername.localdomain so it knows that the mail sent to it (which will have localdomain added) is actually for it.

The entry for the sending server needs a special mention. You need one of these for every server you will be sending mail from. If the recieving server has probs with the email it will want to bounce it back to the sender and it needs a hosts entry for each sender with .localdomain on so it can find those servers and return the mail.
Yes it might be easier with DNS, but hosts files are so much simpler in a small network.

Reconfigure Sendmail to listen on the internal network on all servers

As noted somewhere much earlier, sendmail as shipped listens only on the localhost interface (127.0.0.1) as a very sensible way of avoiding becoming a spam relay.

Thats nice, but if we want to send emails about the internal network we need to listen on additional addresses.

The good news is thats easy to do. The bad news is that you must do this on every server to participate in your mail network.

The reason you must do it on every server of course is to handle the bounced back emails from the destination email server, so each sending server must also listen for anything coming back.

You do of course know the ip addresses you have assigned to each of your servers internal interfaces, if not do an ifconfig and get them now.

Then on every server edit the /etc/mail/sendmail.cf file. Search for a line like

```
      O DaemonPortOptions=Port=smtp,Addr=127.0.0.1, Name=MTA
```

and add a new line below it with the internal interface address, so you would end up with the below on osprey which is on the internal 182 address

```
      O DaemonPortOptions=Port=smtp,Addr=127.0.0.1, Name=MTA
      O DaemonPortOptions=Port=smtp,Addr=169.254.218.182, Name=MTA
```

But don't restart sendmail yet, you also need to do the below step.

Change the Sendmail Cw entry

Reguardless of the entries you added/fixed in the /etc/hosts file step above sendmail itself reports its email source as the value set in the Cw entry in the /etc/mail/sendmail.cf file. You must change that to match the entries added to the /etc/hosts files.

So '**vi**' /etc/mail/sendmail.cf (on every server) and search for **Cwlocalhost**, you need to change this to the hostname defined for your internal network address.

For example on my osprey it's now

```
[mark@osprey mail]$ grep Cw sendmail.cf
#Cwlocalhost
Cwosprey-private.localdomain
```

If you do not do the Cw change the relaying will fail as sendmail will get mail from the other servers reported as being sent from localhost, and know itself as localhost, and treat it as a loop condition and bounce back the email.

Once done you may do a 'service sendmail restart' now, and if you did the previous step netstat will show port 25 being listened to on both the loopback and newly added address, and you won't be stuck for hours trying to figure out why the /etc/hosts entries were ignored like I was.

The Cw entry must match the /etc/hosts entry for the ip address of the sender, thats important as it is somewhere in the spam checking logic of the mail environment.

Changes on the mail destination server needed for local relaying

By default as shipped Sendmail is setup to avoid mail relaying. This is a very good thing as you don't want to be a spam mail relay site as I keep mentioning.

But it does mean it will not accept any mail from your other servers. Even though you have configured it to listen on the internal network address now you will see lots of relaying denied messages in /var/log/maillog on the destination server if you try to send emails to the server (which as you have already done the above step of course will all be appearing in the sending servers user email box as bounced back mail).

Fortunately this is also easy to solve.

You need to update the file /etc/mail/access to allow sendmail to accept email from remote servers. As you can add a network range I just added my entire 169.254.218 internal network range. You may prefer to add specific server addresses, but as I keep plugging new toys in all the time the network range was easier for me.

At the end of /etc/mail/access I just added the line

```
      Connect:169.254.218                    RELAY
```

Restart sendmail with a 'service network restart'.

On Fedora a ls -ltr shows the access.db file (which is a berkley dayabase file) is refreshed automatically on the sendmail restart to pick up the change. If you are on another distribution and it doesn;t automatically refresh refer to the website documentation of your distribution.

Testing the changes

Completed all the changes then ?.

On any of the servers you intend to send email from just send a mail message to a user on the destination server

```
      mail -s "Test Message" userid @ destinationserver
      enter some text
      .
      [ENTER]
```

It should appear in the inbox on the destination server.
If it bounces back or gets blackholes see troubleshooting below.

Troubleshooting

On Fedora you will find any email errors in the log file /var/log/maillog. Check eny host names seen in error messages to ensure they match what is in your /etc/hosts entries and your sendmail.cf Cw entry on any affected server.

If there is nothing there check you have opened your ***internal*** firewall ports for traffic on port 25.

Dovecot IMAP and POP3 server

While it's nice to have all your emails forwarded to one place, you may not always want to log onto the unix server to read those emails.

I have installed Dovecot toprovide an IMAP/POP3 server interface so I can use Thunderbird from my desktop to retreive the emails off the server.

This covers the bare minimum installation, and requires a login with a valid unix level userid/password combintaion. I have no intention of investigating 'virtual' users at this point.

I also use only the POP3 interface of Dovecot to pull down the emails and delete them from the server (I use MailstoreHome to store/archive emails to CD so don't need to keep them on the server).

And it is sloooow...but I mention that just so you know, it hasn't crashed and it will get there.
And if you are one of those people who whine about slow free opensource products them go and purchase something from a nice commercial vendor that will charge you for the upgrade when the issue is fixed.

The assumption here is that your mail server is using Maildir format by now.

Installing

Ok, this is for Fedora users. It's in one of the repos.
All I did was 'yum install dovecot'
If you are on another repo, refer to your packaging software.

Configuring

After yum imstalls it it will create a file /etc/dovecot.conf.
Vi it and search for the lines containing mail_location (they will be in a comment block).
Uncomment the line

```
mail_location = maildir:~/Maildir
```

If you are still using the Mbox format you could try uncommenting
mail_location = mbox:~/mail:INBOX=/var/mail/%u
But I have never tried it and in Fedora its ~/Mail not ~/mail... so at your own risk of course.

Then back at the command line

```
chkconfig dovecot on
```

Do a 'service start dovecot' to start it now so you can test it before a reboot.

Testing the Dovecot server

Well that depends entirely on your desktop email client.

I used Thunderbird and created a new account for mark (unix userid) with the server being the ipaddr of osprey that I had started dovecot on.
It prompted me for my unix password, I enetered it, I downloaded mail. All OK for me, apart from being a bit slow, but it worked.

How you test it is up to what email client you use yourself I'm afraid.

However Dovecot also logs messages to /var/log/maillog (in Fedora) so you can look there if you experience problems.

Recomendatons

In my environment the purpose of forwarding all emails to one place was so I could check them all in one place.

That of course required you changing the entries in /etc/aliases so all the emails get sent to your userid. Over to you to decide if you want to do that, but I'd recommend it as for me that was the point of the exercise.

Next Steps – document getmail

Sorry, havn't got this far yet.

Getmail will retrieve the emails from an ISP POP3 account (like fetchmail, but according to all the forums less error prone; dunno havn't tried). But in theory all emails from all the ISP accounts will be avaiable in the one mailbox.

Need to investigate te spam plugins first, as there is a lot of junk in my ISP accounts, so to be continued, one day.

And you may not be interested, remember you can never hit the reply button (spam filters will see message was to xxx @ ispyyy but the reply came from yourserver @ somewhere and drop the email.