LHN

HomePurchase PDFsForumsAbout

Quick HOWTO : Ch21 : Configuring Linux Mail Servers

From Linux Home Networking

Contents

Introduction

Email is an important part of any Web site you create. In a home environment, a free web based email service may be sufficient, but if you are running a business, then a d
probably be required.

This chapter will show you how to use sendmail to create a mail server that will relay your mail to a remote user's mailbox or incoming mail to a local mail box. You'll also
send mail via your mail server using a with mail client such as Outlook Express or Evolution.

Debian / Ubuntu Differences

This chapter focuses on Fedora / CentOS / RedHat for simplicity of explanation. Whenever there is a difference in the required commands for Debian / Ubuntu variations

The universal difference is that the commands shown are done by the Fedora / CentOS / RedHat root user. With Debian / Ubuntu you will either have to become root usin
or you can temporarily increase your privilege level to root using the "sudo <command>" command.

Here is an example of how to permanently become root:

```
user@ubuntu:~$ sudo su -
[sudo] password for peter:
root@ubuntu:~#
```

Here is an example of how to temporarily become root to run a specific command. The first attempt to get a directory listing fails due to insufficient privileges. The second
sudo keyword is inserted before the command.

```
user@ubuntu:~$  ls -l /var/lib/mysql/mysql
ls: cannot access /var/lib/mysql/mysql: Permission denied
user@ubuntu:~$ sudo ls -l /var/lib/mysql/mysql
[sudo] password for peter:
total 964
-rw-rw---- 1 mysql mysql   8820 2010-12-19 23:09 columns_priv.frm
-rw-rw---- 1 mysql mysql      0 2010-12-19 23:09 columns_priv.MYD
-rw-rw---- 1 mysql mysql   4096 2010-12-19 23:09 columns_priv.MYI
-rw-rw---- 1 mysql mysql   9582 2010-12-19 23:09 db.frm
...
...
...
user@ubuntu:~$
```

Now that you have got this straight, let’s continue with the discussion.

Configuring Sendmail

One of the tasks in setting up DNS for your domain (my-site.com) is to use the MX record in the configuration zone file to state the hostname of the server that will handle
most popular Unix mail transport agent is sendmail, but others, such as postfix and qmail, are also gaining popularity with Linux. The steps used to convert a Linux box in
be explained here.

How Sendmail Works

As stated before, sendmail can handle both incoming and outgoing mail for your domain. Take a closer look.

Incoming Mail

Usually each user in your home has a regular Linux account on your mail server. Mail sent to each of these users (username@my-site.com) eventually arrives at your mail
processes it and deposits it in the mailbox file of the user's Linux account.

Mail isn't actually sent directly to the user's PC. Users retrieve their mail from the mail server using client software, such as Microsoft's Outlook or Outlook Express, that su
IMAP mail retrieval protocols.

Linux users logged into the mail server can read their mail directly using a text-based client, such as mail, or a GUI client, such as Evolution. Linux workstation users can u
access their mail remotely.

Outgoing Mail

The process is different when sending mail via the mail server. PC and Linux workstation users configure their e-mail software to make the mail server their outbound SM

If the mail is destined for a local user in the mysite.com domain, then sendmail places the message in that person's mailbox so that they can retrieve it using one of the meth

If the mail is being sent to another domain, sendmail first uses DNS to get the MX record for the other domain. It then attempts to relay the mail to the appropriate destinati Simple Mail Transport Protocol (SMTP). One of the main advantages of mail relaying is that when a PC user A sends mail to user B on the Internet, the PC of user A can processing to the mail server.

Note: If mail relaying is not configured properly, then your mail server could be commandeered to relay spam. Simple sendmail security will be covered later.

Sendmail Macros

When mail passes through a sendmail server the mail routing information in its header is analyzed, and sometimes modified, according to the desires of the systems admini highly complicated regular expressions listed in the /etc/mail/sendmail.cf file, sendmail inspects this header and then acts accordingly.

In recognition of the complexity of the /etc/mail/sendmail.cf file, a much simpler file named /etc/sendmail.mc was created, and it contains more understandable instructions use. These are then interpreted by a number of macro routines to create the sendmail.cf file. After editing sendmail.mc, you must always run the macros and restart sendma effect.

Each sendmail.mc directive starts with a keyword, such as DOMAIN, FEATURE, or OSTYPE, followed by a subdirective and in some cases arguments. A typical exam

As stated before, sendmail can handle both incoming and outgoing mail for your domain. Take a closer look.

```
FEATURE(`virtusertable',`hash -o /etc/mail/virtusertable.db')dnl
```

The keywords usually define a subdirectory of /usr/share/sendmail-cf in which the macro may be found and the subdirective is usually the name of the macro file itself. So name is /usr/share/sendmail-cf/feature/virtusertable.m4, and the instruction `\ hash -o /etc/mail/virtusertable.db' is being passed to it.

Notice that sendmail is sensitive to the quotation marks used in the m4 macro directives. They open with a grave mark and end with a single quote.

```
FEATURE(`masquerade_envelope')dnl
```

Some keywords, such as define for the definition of certain sendmail variables and MASQUERADE_DOMAIN, have no corresponding directories with matching macro /usr/share/sendmail-cf/m4 directory deal with these.

Once you finish editing the sendmail.mc file, you can then execute the make command while in the /etc/mail directory to regenerate the new sendmail.cf file.

```
[root@bigboy tmp]# cd /etc/mail
[root@bigboy mail]# make
```

If there have been no changes to the files in /etc/mail since the last time make was run, then you'll get an error like this:

```
[root@bigboy mail]# make
make: Nothing to be done for `all'.
[root@bigboy mail]#
```

The make command actually generates the sendmail.cf file using the m4 command. The m4 usage is simple, you just specify the name of the macro file as the argument, in redirect the output, which would normally go to the screen, to the sendmail.cf file with the ">" redirector symbol.

```
 [root@bigboy tmp]# m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

I'll discuss many of the features of the sendmail.mc file later in the chapter.

Installing Sendmail

Most RedHat and Fedora Linux software product packages are available in the RPM format, whereas Debian and Ubuntu Linux use DEB format installation files. When s remember that the filename usually starts with the software package name and is followed by a version number, as in sendmail-8.12.10-1.1.1.i386.rpm. (For help on down required packages, see Chapter 6, Installing Linux Software).

Note: You will need to make sure that the sendmail, sendmail-cf, and m4 packages are installed.

Managing the sendmail Server

Managing the sendmail daemon is easy to do, but the procedure differs between Linux distributions. Here are some things to keep in mind.

1. Firstly, different Linux distributions use different daemon management systems. Each system has its own set of commands to do similar operations. The most comm management systems are SysV and Systemd.
2. Secondly, the daemon name needs to be known. In this case the name of the daemon is **sendmail**.

Armed with this information you can know how to:

1. Start your daemons automatically on booting
2. Stop, start and restart them later on during troubleshooting or when a configuration file change needs to be applied.

For more details on this, please take a look at the "Managing Daemons" section of Chapter 6 "Installing Linux Software"

Note: Remember to configure your daemon to start automatically upon your next reboot.

How To Restart Sendmail After Editing Your Configuration Files

In this chapter, you'll see that sendmail uses a variety of configuration files that require different treatments for their commands to take effect. This little activate-sendmail.sh required post configuration steps.

```
#
# Script: /usr/local/bin/activate-sendmail.sh
#

#!/bin/bash
cd /etc/mail
/usr/bin/make
/usr/bin/newaliases
systemctl restart sendmail.service
systemctl restart spamassassin.service
```

It first runs the make command, which creates a new sendmail.cf file from the sendmail.mc file and compiles supporting configuration files in the /etc/mail directory accord file /etc/mail/Makefile. It then generates new e-mail aliases with the newaliases command, (this will be covered later), and then restarts sendmail.

The script also restarts spamassassin, a package that will be discussed later.

Use this command to make the script executable.

```
[root@bigboy tmp]# chmod 700 /usr/local/bin/activate-sendmail.sh
```

You'll need to run the script each time you change any of the sendmail configuration files described in the sections to follow.

```
[root@bigboy tmp]# /usr/local/bin/activate-sendmail.sh
```

In a production system you may want to be more selective and only restart the specific applications on which you are working. I included all of them in the script so you do

The /etc/mail/sendmail.mc File

You can define most of sendmail's configuration parameters in the /etc/mail/sendmail.mc file, which is then used by the m4 macros to create the /etc/mail/sendmail.cf file. C sendmail.mc file is much simpler than configuration of sendmail.cf, but it is still often viewed as an intimidating task with its series of structured directive statements that ge in most cases you won't have to edit this file very often.

How to Put Comments in sendmal.mc

In most Linux configuration files a # symbol is used at the beginning of a line convert it into a comment line or to deactivate any commands that may reside on that line.

The sendmail.mc file doesn't use this character for commenting, but instead uses the string "dnl". Here are some valid examples of comments used with the sendmail.mc co

- These statements are disabled by dnl commenting.

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

- This statement is incorrectly disabled:

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

- This statement is active:

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

Note: Remember to run the activate-sendmail.sh script to activate any configuration changes.

Configuring DNS for sendmail

Remember that you will never receive mail unless you have configured DNS for your domain to make your new Linux box mail server the target of the DNS domain's M 18, "Configuring DNS", or Chapter 19, "Dynamic DNS", for details on how to do this.

Configure Your Mail Server's Name In DNS

You first need to make sure that your mail server's name resolves in DNS correctly. For example, if your mail server's name is bigboy and it you intend for it to mostly han site.com, then bigboy.my-site.com must correctly resolve to the IP address of one of the mail server's interfaces. You can test this using the host command:

```
[root@smallfry tmp]# host bigboy.my-site.com
bigboy.my-site.com has address 192.168.1.100
[root@smallfry tmp]#
```

You will need to fix your DNS server's entries if the resolution isn't correct.

Configure The /etc/resolv.conf File

The sendmail program expects DNS to be configured correctly on the DNS server. The MX record for your domain must point to the IP address of the mail server.

The program also expects the files used by the mail server's DNS client to be configured correctly. The first one is the /etc/resolv.conf file in which there must be a domain
the domains the mail server is expected to handle mail for.

Finally, sendmail expects a nameserver directive that points to the IP address of the DNS server the mail server should use to get its DNS information.

For example, if the mail server is handling mail for my-site.com and the IP address of the DNS server is 192.168.1.100, there must be directives that look like this:

```
domain my-site.com
nameserver 192.168.1.100
```

An incorrectly configured resolv.conf file can lead to errors when running the m4 command to process the information in your sendmail.mc file.

```
WARNING: local host name (smallfry) is not qualified; fix $j in config file
```

The /etc/hosts File

The /etc/hosts file also is used by DNS clients and also needs to be correctly configured. Here is a brief example of the first line you should expect to see in it:

```
127.0.0.1 bigboy.my-site.com localhost.localdomain localhost bigboy
```

The entry for 127.0.0.1 must always be followed by the fully qualified domain name (FQDN) of the server. In the case above it would be bigboy.my-site.com. Then you r
localhost and localhost.localdomain. Linux does not function properly if the 127.0.0.1 entry in /etc/hosts doesn't also include localhost and localhost.localdomain. Finally y
your host may have to the end of the line.

How To Configure Linux Sendmail Clients

All Linux mail clients in your home or company need to know which server is the mail server. This is configured in the sendmail.mc file by setting the SMART_HOST st
server. In the example below, the mail server has been set to mail.my-site.com, the mail server for the my-site.com domain.

```
define(`SMART_HOST',`mail.my-site.com')
```

If you don't have a mail server on your network, you can either create one, or use the one offered by your ISP.

Once this is done, you need to process the sendmail.mc file and restart sendmail. To do this, run the restarting script we from earlier in the chapter.

If the sendmail server is a Linux server, then the /etc/hosts file will also have to be correctly configured too.

Note: Remember to run the activate-sendmail.sh script shown at the beginning of the chapter to activate any configuration changes.

Converting From a Mail Client to a Mail Server

All Linux systems have a virtual loopback interface that lives only in memory with an IP address of 127.0.0.1. As mail must be sent to a target IP address even when there
sendmail therefore uses the loopback address to send mail between users on the same Linux server. To become a mail server, and not a mail client, sendmail needs to be co
messages on NIC interfaces as well.

1) Determine which NICs sendmail is running on. You can see the interfaces on which sendmail is listening with the netstat command. Because sendmail listens on TCP p
grep for 25 to see a default configuration listening only on IP address 127.0.0.1 (loopback):

```
[root@bigboy tmp]# netstat -an | grep :25 | grep tcp
tcp 0 0 127.0.0.1:25 0.0.0.0:* LISTEN
[root@bigboy tmp]#
```

2) Edit sendmail.mc to make sendmail listen on all interfaces. If sendmail is listening on the loopback interface only, you should comment out the daemon_options line in t
with dnl statements. It is also good practice to take precautions against spam by not accepting mail from domains that don't exist by commenting out the accept_unresolvab
the fourth and next to last lines in the example.

```
dnl
dnl This changes sendmail to only listen on the loopback
dnl device 127.0.0.1 and not on any other network
dnl devices. Comment this out if you want
dnl to accept email over the network.
dnl DAEMON_OPTIONS(`Port=smtp,Addr=127.0.0.1, Name=MTA')
dnl
...
...
...
dnl
dnl We strongly recommend to comment this one out if you want
dnl to protect yourself from spam. However, the laptop and
dnl users on computers that do
dnl not have 24x7 DNS do need this.
dnl FEATURE(`accept_unresolvable_domains')dnl
dnl FEATURE(`relay_based_on_MX')dnl
dnl
```

Note: You need to be careful with the accept_unresolvable_names feature. In the sample network, bigboy the mail server does not accept e-mail relayed from any of the ot
they are not in DNS. Chapter 18, "Configuring DNS", shows how to create your own internal domain just for this purpose.

Note: If your server has multiple NICs and you want it to listen to one of them, then you can uncomment the localhost DAEMON_OPTIONS entry and add another one f
on which to wish to accept SMTP traffic.

3) Comment out the SMART_HOST Entry in sendmal.mc. The mail server doesn't need a SMART_HOST entry in its sendmail.mc file. Comment this out with a dnl at th

```
dnl define(`SMART_HOST',`mail.my-site.com')
```

4) Regenerate the sendmail.cf file, and restart sendmail. Again, you can do this with the activate-sendmail.sh script from the beginning of the chapter.

5) Make sure sendmail is listening on all interfaces (0.0.0.0).

```
[root@bigboy tmp]# netstat -an | grep :25 | grep tcp
tcp 0 0 0.0.0.0:25 0.0.0.0:* LISTEN
[root@bigboy tmp]#
```

You have now completed the first phase of converting your Linux server into a sendmail server by enabling it to listen to SMTP traffic on its interfaces. The following sect
define what type of mail it should handle and the various ways this mail can be processed.

A General Guide To Using The sendmail.mc File

The sendmail.mc file can seem jumbled. To make it less cluttered I usually create two easily identifiable sections in it with all the custom commands I've ever added.

The first section is near the top where the FEATURE statements usually are, and the second section is at the very bottom.

Sometimes sendmail will archive this file when you do a version upgrade. Having easily identifiable modifications in the file will make post upgrade reconfiguration much

```
dnl ***** Customised section 1 start *****
dnl
dnl
FEATURE(delay_checks)dnl
FEATURE(masquerade_envelope)dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_entire_domain)dnl
dnl
dnl
dnl ***** Customised section 1 end *****
```

The /etc/mail/relay-domains File

The /etc/mail/relay-domains file is used to determine domains from which it will relay mail. The contents of the relay-domains file should be limited to those domains that c
spam. By default, this file does not exist in a standard RedHat / Fedora install. In this case, all mail sent from my-super-duper-site.com and not destined for this mail server

```
my-super-duper-site.com
```

One disadvantage of this file is that controls mail based on the source domain only, and source domains can be spoofed by spam e-mail servers. The /etc/mail/access file ha
restricting relaying by IP address or network range and is more commonly used. If you delete /etc/mail/relay-domains, then relay access is fully determined by the /etc/mail/

Note: Be sure to run activate-sendmail.sh script from the beginning of the chapter for these changes to take effect.

The /etc/mail/access File

You can make sure that only trusted PCs on your network have the ability to relay mail via your mail server by using the /etc/mail/access file. That is to say, the mail server
those PCs on your network that have their e-mail clients configured to use the mail server as their outgoing SMTP mail server. (In Outlook Express, you set this using:
Tools>Accounts>Properties>Servers)

If you don't take the precaution of using this feature, you may find your server being used to relay mail for spam e-mail sites. Configuring the /etc/mail/access file will not s
only spam flowing through you.

The /etc/mail/access file has two columns. The first lists IP addresses and domains from which the mail is coming or going. The second lists the type of action to be taken w
or destinations is received. Keywords include RELAY, REJECT, OK (not ACCEPT), and DISCARD. There is no third column to state whether the IP address or domain
of the mail, sendmail assumes it could be either and tries to match both. All other attempted relayed mail that doesn't match any of the entries in the /etc/mail/access file, sen
this, my experience has been that control on a per e-mail address basis is much more intuitive via the /etc/mail/virtusertable file.

The sample file that follows allows relaying for only the server itself (127.0.0.1, localhost), two client PCs on your home 192.168.1.X network, everyone on your 192.168
passing e-mail through the mail server from servers belonging to my-site.com. Remember that a server will be considered a part of my-site.com only if its IP address can be
zone file:

```
localhost.localdomain           RELAY
localhost                       RELAY
127.0.0.1                       RELAY
192.168.1.16                    RELAY
192.168.1.17                    RELAY
192.168.2                       RELAY
my-site.com                     RELAY
```

Note: You'll now have to convert this text file into a sendmail readable database file named /etc/mail/access.db. The activate-sendmail.sh script we configured at the begin
for you too.

Remember that the relay security features of this file may not work if you don't have a correctly configured /etc/hosts file.

The /etc/mail/local-host-names File

When sendmail receives mail, it needs a way of determining whether it is responsible for the mail it receives. It uses the /etc/mail/local-host-names file to do this. This file h domains for which sendmail accepts responsibility. For example, if this mail server was to accept mail for the domains my-site.com and another-site then the file would loo

```
my-site.com
another-site.com
```

In this case, remember to modify the MX record of the another-site.com DNS zonefile point to my-site.com. Here is an example (Remember each "." is important):

```
; Primary Mail Exchanger for another-site.com

another-site.com. MX 10 mail.my-site.com.
```

Note: Be sure to run the activate-sendmail.sh script from the beginning of the chapter for these changes to take effect.

Which User Should Really Receive The Mail?

After checking the contents of the virtusertable, sendmail checks the aliases files to determine the ultimate recipient of mail.

The /etc/mail/virtusertable file

The /etc/mail/virtusertable file contains a set of simple instructions on what to do with received mail. The first column lists the target email address and the second column l a remote email address, or a mailing list entry in the /etc/aliases file to which the email should be forwarded.

If there is no match in the virtusertable file, sendmail checks for the full email address in the /etc/aliases file.

```
webmaster@another-site.com   webmasters
@another-site.com            marc
sales@my-site.com            sales@another-site.com
paul@my-site.com             paul
finance@my-site.com          paul
@my-site.com                 error:nouser User unknown
```

In this example, mail sent to:

- webmaster@another-site.com will go to local user (or mailing list) webmasters, all other mail to another-site.com will go to local user marc.
- sales at my-site.com will go to the sales department at my-othersite.com.
- paul and finance at my-site.com goes to local user (or mailing list) paul

All other users at my-site.com receive a bounce back message stating "User unknown".

Note: Be sure to run the activate-sendmail.sh script from the beginning of the chapter for these changes to take effect.

The /etc/aliases File

You can think of the /etc/aliases file as a mailing list file. The first column has the mailing list name (sometimes called a virtual mailbox), and the second column has the me separated by commas.

To start, sendmail searches the first column of the file for a match. If there is no match, then sendmail assumes the recipient is a regular user on the local server and deposits

If it finds a match in the first column, sendmail notes the nickname entry in the second column. It then searches for the nickname again in the first column to see if the recip mailing list.

If sendmail doesn't find a duplicate, it assumes the recipient is a regular user on the local server and deposits the mail in their mailbox.

If the recipient is a mailing list, then sendmail goes through the process all over again to determine if any of the members is on yet another list, and when it is all finished, th mail message.

In the example that follows, you can see that mail sent to users bin, daemon, lp, shutdown, apache, named, and so on by system processes will all be sent to user (or mailin is actually an alias for a mailing list consisting of user marc and webmaster@my-site.com.

```
# Basic system aliases -- these MUST be present.
mailer-daemon:        postmaster
postmaster:           root

# General redirections for pseudo accounts.
bin:                  root
daemon:               root
...
 ...
abuse:                root
# trap decode to catch security attacks
decode:               root

# Person who should get root's mail
root:                 marc,webmaster@my-site.com
```

Notice that there are no spaces between the mailing list entries for root: You will get errors if you add spaces.

Note: The default /etc/aliases file installed with RedHat / Fedora has the last line of this sample commented out with a #, you may want to delete the comment and change

Also after editing this file, you'll have to convert it into a sendmail readable database file named /etc/aliases.db. Here is the command to do that:

```
[root@bigboy tmp]# newaliases
```

In this simple mailing list example, mail sent to root actually goes to user account marc and webmaster@my-site.com. Because aliases can be very useful, here are a few m
/etc/aliases file.

- Mail to "directors@my-site.com" goes to users "peter", "paul" and "mary".

```
# Directors of my SOHO company
directors:      peter,paul,mary
```

- Mail sent to "family@my-site.com" goes to users "grandma", "brother" and "sister"

```
# My family
family:        grandma,brother,sister
```

- Mail sent to admin-list gets sent to all the users listed in the file /home/mailings/admin-list.

```
# My mailing list file
admin-list:     ":include:/home/mailings/admin-list"
```

The advantage of using mailing list files is that the admin-list file can be a file that trusted users can edit, user root is only needed to update the aliases file. Despite this, ther
mail reflectors. One is that bounce messages from failed attempts to broadcast go to all users. Another is that all subscriptions and unsubscriptions have to be done manuall
administrator. If either of these are a problem for you, then consider using a mailing list manager, such as majordomo.

One important note about the /etc/aliases file: By default your system uses sendmail to mail system messages to local user root. When sendmail sends e-mail to a local user,
mail header. If you then use a mail client with a spam mail filtering rule to reject mail with no To: in the header, such as Outlook Express or Evolution, you may find yours

To get around this, try making root have an alias for a user with a fully qualified domain name, this forces sendmail to insert the correct fields in the header; for example:

```
# Person who should get root's mail
root:                webmaster@my-site.com
```

Note: Be sure to run the newaliases command for these changes to take effect.

Sendmail Masquerading Explained

If you want your mail to appear to come from user@mysite.com and not user@bigboy.mysite.com, then you have two choices:

- Configure your email client, such as Outlook Express, to set your email address to user@mysite.com. (I'll explain this in the "Configuring Your POP Mail Server" s
- Set up masquerading to modify the domain name of all traffic originating from and passing trough your mail server.

Configuring masquerading

In the DNS configuration, you made bigboy the mail server for the domain my-site.com. You now have to tell bigboy in the sendmail configuration file sendmail.mc that a
on bigboy should appear to be coming from my-site.com; if not, based on our settings in the /etc/hosts file, mail will appear to come from mail.my-site.com. This isn't terrib
your Web site to be remembered with the word "mail" in front of it. In other words you may want your mail server to handle all email by assigning a consistent return addr
matter which server originated the email.

You can solve this by editing your sendmail.mc configuration file and adding some masquerading commands and directives:

```
FEATURE(always_add_domain)dnl
FEATURE(`masquerade_entire_domain')dnl
FEATURE(`masquerade_envelope')dnl
FEATURE(`allmasquerade')dnl
MASQUERADE_AS(`my-site.com')dnl
MASQUERADE_DOMAIN(`my-site.com.')dnl
MASQUERADE_DOMAIN(localhost)dnl
MASQUERADE_DOMAIN(localhost.localdomain)dnl
```

The result is that:

- The MASQUERADE_AS directive makes all mail originating on bigboy appear to come from a server within the domain my-site.com by rewriting the email heade
- The MASQUERADE_DOMAIN directive makes mail relayed via bigboy from all machines in the another-site.com and localdomain domains appear to come from
  domain of my-site.com. Using DNS, sendmail checks the domain name associated with the IP address of the mail relay client sending the mail to help it determine w
  masquerading or not.
- FEATURE masquerade_entire_domain makes sendmail masquerade servers named *my-site.com, and *another-site.com as my-site.com. In other words, mail from
  masqueraded as my-site.com. If this wasn't selected, then only servers named my-site.com and my-othersite.com would be masqueraded. Use this with caution wher
  necessary authority to do this.
- FEATURE allmasquerade makes sendmail rewrite both recipient addresses and sender addresses relative to the local machine. If you cc: yourself on an outgoing ma
  cc: to an address he knows instead of one on localhost.localdomain.

  Note: Use FEATURE allmasquerade with caution if your mail server handles email for many different domains and the mailboxes for the users in these domains res
  allmasquerade statement causes all mail destined for these mailboxes to appear to be destined for users in the domain defined in the MASQUERADE_AS statement
  MASQUERADE_AS is my-site.com and you use allmasquerade, then mail for peter@another-site.com enters the correct mailbox but sendmail rewrites the To:, ma
  sent to peter@my-ste.com originally.

- FEATURE always_add_domain always masquerades email addresses, even if the mail is sent from a user on the mail server to another user on the same mail server
- FEATURE masquerade_envelope rewrites the email envelope just as MASQUERADE_AS rewrote the header.

Masquerading is an important part of any mail server configuration as it enables systems administrators to use multiple outbound mail servers, each providing only the glob
company and not the fully qualified domain name of the server itself. All email correspondence then has a uniform email address format that complies with the company's l

Note: E-mail clients, such as Outlook Express, consider the To: and From: statements as the e-mail header. When you choose Reply or Reply All in Outlook Express, the
the To: and From: in the header. It is easy to fake the header, as spammers often do; it is detrimental to e-mail delivery, however, to fake the envelope.

The e-mail envelope contains the To: and From: used by mailservers for protocol negotiation. It is the envelope's From: that is used when e-mail rejection messages are sen

Note: Be sure to run the activate-sendmail.sh script from the beginning of the chapter for these changes to take effect.

Testing Masquerading

The best way of testing masquerading from the Linux command line is to use the "mail -v username" command. I have noticed that "sendmail -v username" ignores masqu
should also tail the /var/log/maillog file to verify that the masquerading is operating correctly and check the envelope and header of test email received by test email accoun

Other Masquerading Notes

By default, user "root" will not be masqueraded. To remove this restriction use:

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

command in /etc/mail/sendmail.mc. You can comment this out if you like with a "dnl" at the beginning of the line and running the sendmail start script.

Using Sendmail to Change the Sender's Email Address

Sometimes masquerading isn't enough. At times you may need to change not only the domain of the sender but also the username portion of the sender's e-mail address. Fo
bought a program for your SOHO office that sends out notifications to your staff, but the program inserts its own address as sender's address, not that of the IT person.

Web-based CGI scripts tend to run as user apache and, therefore, send mail as user apache too. Often you won't want this, not only because apache's e-mail address may n
because some anti-spam programs check to ensure that the From:, or source e-mail address, actually exists as a real user. If your virtusertable file allows e-mail to only pred
about the apache user will fail, and your valid e-mail may be classified as being spam.

With sendmail, you can change both the domain and username on a case-by-case basis using the genericstable feature:

1) Add these statements to your /etc/mail/sendmail.mc file to activate the feature:

```
FEATURE(`genericstable',`hash -o /etc/mail/genericstable.db')dnl
GENERICS_DOMAIN_FILE(`/etc/mail/generics-domains')dnl
```

2) Create a /etc/mail/generics-domains file that is just a list of all the domains that should be inspected. Make sure the file includes your server's canonical domain name, wh
command:

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

Here is a sample /etc/mail/generics-domains file:

```
my-site.com
another-site.com
bigboy.my-site.com
```

3) Create your /etc/mail/genericstable file. First sendmail searches the /etc/mail/generics-domains file for a list of domains to reverse map. It then looks at the /etc/mail/gener
email address from a matching domain. The format of the file is

```
linux-username       username@new-domain.com
```

Your e-mails from linux-username should now appear to come from username@new-domain.com.

Here are some other examples:

```
alert          security-alert@my-site.com
peter          urgent-message@my-site.com
apache         mailer@my-site.com
```

Note: Be sure to run the activate-sendmail.sh script from the beginning of the chapter for these changes to take effect.

Troubleshooting Sendmail

There are a number of ways to test sendmail when it doesn't appear to work correctly. Here are a few methods you can use to fix some of the most common problems.

Testing TCP connectivity

The very first step is to determine whether your mail server is accessible on the sendmail SMTP TCP port 25. Lack of connectivity could be caused by a firewall with inco
forwarding rules to your mail server. Failure could also be caused by the sendmail process being stopped. It is best to test this from both inside your network and from the I

Chapter 4, "Simple Network Troubleshooting", covers troubleshooting with TELNET.

Further Testing of TCP connectivity

You can also mimic a full mail session using TELNET to make sure everything is working correctly. If you get a "500 Command not recognized" error message along the
typographical error. Follow these steps carefully.

1) Telnet to the mail server on port 25. You should get a response with a 220 status code.

```
[root@bigboy tmp]# telnet mail.my-site.com 25
Trying mail.my-site.com...
Connected to mail.my-site.com.
Escape character is '^]'.
220 mail.my-site.com ESMTP server ready
```

If this basic step fails, you probably have a connection problem that could be the result of typical network issues outlined in Chapter 4, "Simple Network Troubleshooting"
find yourself having problems related to basic connectivity.

2) Use the hello command to tell the mail server the domain you belong to. You should receive a message with a successful status 250 code at the beginning of the respons

```
helo another-web-site.org
250 mail.my-site.com Hello c-24-4-97-110.client.comcast.net [24.4.97.110], pleased to meet you.
```

3) Inform the mail server from which the test message is coming with the MAIL FROM: statement.

```
MAIL FROM:sender@another-web-site.org
250 2.1.0 sender@another-web-site.org... Sender ok
```

4) Tell the mail server to whom the test message is going with the " RCPT TO:" statement.

```
RCPT TO: user@my-site.com
250 2.1.5 user@my-site.com... Recipient ok
```

5) Prepare the mail server to receive data with the DATA statement

```
DATA
354 Enter mail, end with "." on a line by itself
```

6) Type the string "subject:" then type a subject. Type in your text message, ending it with a single period on the last line. For example.

```
Subject: Test Message
Testing sendmail interactively
.
250 2.0.0 iA75r9si017840 Message accepted for delivery
```

7) Use the QUIT command to end the session.

```
QUIT
221 2.0.0 mail.my-site.com closing connection
Connection closed by foreign host.
[root@bigboy tmp]#
```

Now verify that the intended recipient received the message, and check the system logs for any mail application errors.

The /var/log/maillog File

Because sendmail writes all its status messages in the /var/log/maillog file, always monitor this file whenever you are doing changes. Open two TELNET, SSH, or console
them and monitor the sendmail status output in the other using the command

```
[root@bigboy tmp]# tail -f /var/log/maillog
```

This tactic will make it much easier to troubleshoot any issues you may find in sendmail.

Common Errors Due To Incomplete RPM Installation

Both the newaliases and m4 commands require the sendmail-cf and m4 RPM packages. These must be installed. If they are not, you'll get errors when running various sen

- Sample Errors when running newaliases

```
[root@bigboy mail]# newaliases
Warning: .cf file is out of date: sendmail 8.12.5 supports version 10, .cf file is version 0
No local mailer defined
QueueDirectory (Q) option must be set
[root@bigboy mail]#
```

- Sample errors when processing the sendmail.mc file

```
[root@bigboy mail]# m4 /etc/mail/sendmail.mc > /etc/mail/sendmail.cf
```

```
/etc/mail/sendmail.mc:8: m4: Cannot open /usr/share/sendmail-cf/m4/cf.m4: No such file or directory
[root@bigboy mail]#
```

- Sample errors when restarting sendmail

```
[root@bigboy mail]# systemctl restart sendmail.service
Shutting down sendmail: [ OK ]
Shutting down sm-client: [FAILED]
Starting sendmail: 554 5.0.0 No local mailer defined
554 5.0.0 QueueDirectory (Q) option must be set
[FAILED]
Starting sm-client: [ OK ]
[root@bigboy mail]#
```

If these errors occur, make sure your m4, sendmail and senmail-cf RPM packages are installed correctly.

Incorrectly Configured /etc/hosts Files

By default, Fedora inserts the hostname of the server between the 127.0.0.1 and the localhost entries in /etc/hosts like this:

```
127.0.0.1     bigboy    localhost.localdomain    localhost
```

Unfortunately in this configuration, sendmail will think that the server's FQDN is bigboy, which it will identify as being invalid because there is no extension at the end, su
then default to sending e-mails in which the domain is localhost.localdomain.

The /etc/hosts file is also important for configuring mail relay. You can create problems if you fail to place the server name in the FDQN for 127.0.0.1 entry. Here sendmai
FDQN was my-site and that the domain was all of .com.

```
127.0.0.1   my-site.com  localhost.localdomain   localhost  # (Wrong!!!)
```

The server would therefore be open to relay all mail from any .com domain and would ignore the security features of the access and relay-domains files I'll describe later.

As mentioned, a poorly configured /etc/hosts file can make mail sent from your server to the outside world appear as if it came from users at localhost.localdomain and not

Use the sendmail program to send a sample e-mail to someone in verbose mode. Enter some text after issuing the command and end your message with a single period all l
example:

```
[root@bigboy tmp]# sendmail -v example@another-site.com
test text
test text
.
example@another-site.com... Connecting to mail.another-site.com. via esmtp...
220 ltmail.another-site.com LiteMail v3.02(BFLITEMAIL4A); Sat, 05 Oct 2002 06:48:44 -0400
>>> EHLO localhost.localdomain
250-mx.another-site.com Hello [67.120.221.106], pleased to meet you
250 HELP
>>> MAIL From:<root@localhost.localdomain>
250 <root@localhost.localdomain>... Sender Ok
>>> RCPT To:<example@another-site.com>
250 <example@another-site.com>... Recipient Ok
>>> DATA
354 Enter mail, end with "." on a line by itself
>>> .
250 Message accepted for delivery
example@another-site.com... Sent (Message accepted for delivery)
Closing connection to mail.another-site.com.
>>> QUIT
[root@bigboy tmp]#
```

localhost.localdomain is the domain that all computers use to refer to themselves, it is therefore an illegal Internet domain. Consider an example: Mail sent from computer P
from a user at localhost.localdomain on PC1 and is rejected. The rejected e-mail is returned to localhost.localdomain. PC2 sees that the mail originated from localhost.local
rejected e-mail should be sent to a user on PC2 that may not exist. You end up with an error in /var/log/maillog:

```
Oct 16 10:20:04 bigboy sendmail[2500]: g9GHK3iQ002500: SYSERR(root): savemail: cannot save rejected email anywhere
Oct 16 10:20:04 bigboy sendmail[2500]: g9GHK3iQ002500: Losing ./qfg9GHK3iQ002500: savemail panic
```

You may also get this error if you are using a spam prevention program, such as a script based on the PERL module Mail::Audit. An error in the script could cause this typ

Another set of tell tale errors caused by the same problem can be generated when trying to send mail to a user (the example uses root) or creating a new alias database file.
command later.)

```
[root@bigboy tmp]# sendmail -v  root
WARNING: local host name (bigboy) is not qualified; fix $j in config file
[root@bigboy tmp]# newaliases
WARNING: local host name (bigboy) is not qualified; fix $j in config file
[root@bigboy tmp]#
```

An accompanying error in /var/log/maillog log file looks like this:

```
Oct 16 10:23:58 bigboy sendmail[2582]: My unqualified host name (bigboy) unknown; sleeping for retry
```

When you have got sendmail finally working it will be time to focus your attention on fighting unwanted email, or SPAM. This will be covered next.

Fighting SPAM

Unsolicited Commercial Email (UCE or SPAM) can be annoying, time consuming to delete and in some cases dangerous when they contain viruses and worms. Fortunate use your mail server to combat SPAM.

Using Public SPAM Blacklists With Sendmail

There are many publicly available lists of known open mail relay servers and spam generating mail servers on the Internet. Some are maintained by volunteers, others are n
companies, but in all cases they rely heavily on complaints from spam victims. Some spam blacklists simply try to determine whether the e-mail is coming from a legitimate

The IP addresses of offenders usually remain on the list for six months to two years. In some cases, to provide additional pressure on the spammers, the blacklists include n
address but also the entire subnet or network block to which it belongs. This prevents the spammers from easily switching their servers' IP addresses to the next available o
if the spammer uses a public data center, it is possible that their activities could also cause the IP addresses of legitimate e-mailers to be black listed too. It is hoped that thes
pressure the data center's management to evict the spamming customer.

You can configure sendmail to use its dnsbl feature to both query these lists and reject the mail if a match is found. Here are some sample entries you can add to your /etc/s
all be on one line.

- RFC-Ignorant: A valid IP address checker.

```
FEATURE(`dnsbl', `ipwhois.rfc-ignorant.org',`"550 Mail from " $&{client_addr} " refused. Rejected for bad WHOIS info on IP of your SMTP server - see http://www.rfc-ignorant.org/"'
```

- Easynet: An open proxy list.

```
FEATURE(`dnsbl', `proxies.blackholes.easynet.nl', `"550 5.7.1 ACCESS DENIED to OPEN PROXY SERVER "$&{client_name}" by easynet.nl DNSBL  (http://proxies.blackholes.easynet.nl/error
```

- Spamcop: A spammer blacklist.

```
FEATURE(`dnsbl', `bl.spamcop.net', `"450 Mail from " $`'&{client_addr} " refused - see http://spamcop.net/bl.shtml"')
```

- Spamhaus: A spammer blacklist.

```
FEATURE(`dnsbl',`sbl.spamhaus.org',`Rejected - see http://spamhaus.org/')dnl
```

Note:

- Visit the URLs listed in each FEATURE command to learn more about the individual services.
- Be sure to run the activate-sendmail.sh script from the beginning of the chapter for these changes to take effect.

Spamassassin

Once sendmail receives an e-mail message, it hands the message over to procmail, which is the application that actually places the e-mail in user mailboxes on the mail serv
temporarily hand over control to another program, such as a spam filter. The most commonly used filter is spamassassin.

spamassassin doesn't delete spam, it merely adds the word "spam" to the beginning of the subject line of suspected spam e-mails. You can then configure the e-mail filter r
any other mail client to either delete the suspect message or store it in a special Spam folder.

Downloading And Installing Spamassassin

Most RedHat and Fedora Linux software product packages are available in the RPM format, whereas Debian and Ubuntu Linux use DEB format installation files. When s
remember that the filename usually starts with the software package name and is followed by a version number, as in spamassassin-2.60-2.i386.rpm. (For help downloadin
RPM Software").

Managing the spamassassin Server

Managing the spamassassin daemon is easy to do, but the procedure differs between Linux distributions. Here are some things to keep in mind.

1. Firstly, different Linux distributions use different daemon management systems. Each system has its own set of commands to do similar operations. The most comm
   management systems are SysV and Systemd.
2. Secondly, the daemon name needs to be known. In this case the name of the daemon is **spamassassin**.

Armed with this information you can know how to:

1. Start your daemons automatically on booting
2. Stop, start and restart them later on during troubleshooting or when a configuration file change needs to be applied.

For more details on this, please take a look at the "Managing Daemons" section of Chapter 6 "Installing Linux Software"

Note: Remember to configure your daemon to start automatically upon your next reboot.

Configuring procmail for spamassassin

The /etc/procmailrc file is used by procmail to determine the procmail helper programs that should be used to filter mail. This file isn't created by default.

spamassassin has a template you can use called /etc/mail/spamassassin/spamassassin-spamc.rc. Copy the template to the /etc directory.

```
[root@bigboy tmp]# cp /etc/mail/spamassassin/spamassassin-spamc.rc /etc/procmailrc
```

This will activate spamassassin for all your mail users.

Configuring Spamassassin

The spamassassin configuration file is named /etc/mail/spamassassin/local.cf. A full listing of all the options available in the local.cf file can be found in the Linux man pag
command:

```
[root@bigboy tmp]# man Mail::SpamAssassin::Conf
```

You can customize this fully commented sample configuration file to meet your needs.

```
###################################################################
# See 'perldoc Mail::SpamAssassin::Conf' for
# details of what can be adjusted.
###################################################################

#
# These values can be overridden by editing
# ~/.spamassassin/user_prefs.cf (see spamassassin(1) for details)
#

# How many hits before a message is considered spam. The lower the
# number the more sensitive it is.

required_hits           5.0


# Whether to change the subject of suspected spam (1=Yes, 0=No)
rewrite_subject         1


# Text to prepend to subject if rewrite_subject is used
subject_tag             *****SPAM*****


# Encapsulate spam in an attachment (1=Yes, 0=No)
report_safe             1


# Use terse version of the spam report (1=Yes, 0=No)
use_terse_report        0


# Enable the Bayes system (1=Yes, 0=No)
use_bayes               1


# Enable Bayes auto-learning (1=Yes, 0=No)
auto_learn              1


# Enable or disable network checks (1=Yes, 0=No)
skip_rbl_checks         0
use_razor2              1
use_dcc                 1
use_pyzor               1


# Mail using languages used in these country codes will not be marked
# as being possibly spam in a foreign language.
# - english

ok_languages            en


# Mail using locales used in these country codes will not be marked
# as being possibly spam in a foreign language.

ok_locales              en
```

Note: Be sure to run the activate-sendmail.sh script from the beginning of the chapter for these changes to take effect.

Testing spamassassin

You can test the validity of your local.cf file by using the spamassassin command with the --lint option. This will list any syntax problems that may exist. In this example tv
corrected before the command was run again.

```
[root@bigboy tmp]# spamassassin -d --lint
Created user preferences file: /root/.spamassassin/user_prefs
config: SpamAssassin failed to parse line, skipping: use_terse_report        0
config: SpamAssassin failed to parse line, skipping: auto_learn              1
lint: 2 issues detected.  please rerun with debug enabled for more information.
[root@bigboy tmp]# vi /etc/mail/spamassassin/local.cf
...
...
...
[root@bigboy tmp]# spamassassin -d --lint
[root@bigboy tmp]
```

Tuning spamassassin

You can tune the sensitivity of spamassassin to the type of spam you receive by adjusting the required_hits value in the local.cf file. This can be made easier by viewing th
a message in its header. In most GUI based email clients this can be done by looking at the email's properties. In this case, a Nigerian email scam spam was detected and gi
marked as spam.

```
X-Spam-Status: Yes, score=20.1 required=2.1 tests=DEAR_FRIEND,
        DNS_FROM_RFC_POST,FROM_ENDS_IN_NUMS,MSGID_FROM_MTA_HEADER,NA_DOLLARS,
        NIGERIAN_BODY1,NIGERIAN_BODY2,NIGERIAN_BODY3,NIGERIAN_BODY4,
        RCVD_IN_BL_SPAMCOP_NET,RCVD_IN_SBL,RISK_FREE,SARE_FRAUD_X3,
        SARE_FRAUD_X4,SARE_FRAUD_X5,US_DOLLARS_3 autolearn=failed
        version=3.0.4
X-Spam-Report:
        *  0.5 FROM_ENDS_IN_NUMS From: ends in numbers
        *  0.2 RISK_FREE BODY: Risk free.  Suuurreeee....
        *  0.4 US_DOLLARS_3 BODY: Mentions millions of $ ($NN,NNN,NNN.NN)
        *  0.8 DEAR_FRIEND BODY: Dear Friend? That's not very dear!
        *  2.2 NA_DOLLARS BODY: Talks about a million North American dollars
        *  1.8 RCVD_IN_BL_SPAMCOP_NET RBL: Received via a relay in bl.spamcop.net
        *      [Blocked - see <http://www.spamcop.net/bl.shtml?213.185.106.3>]
        *  1.1 RCVD_IN_SBL RBL: Received via a relay in Spamhaus SBL
        *      [213.185.106.3 listed in sbl-xbl.spamhaus.org]
        *  1.4 DNS_FROM_RFC_POST RBL: Envelope sender in postmaster.rfc-ignorant.org
        *  1.9 NIGERIAN_BODY3 Message body looks like a Nigerian spam message 3+
        *  2.9 NIGERIAN_BODY1 Message body looks like a Nigerian spam message 1+
        *  1.4 NIGERIAN_BODY4 Message body looks like a Nigerian spam message 4+
        *  1.7 SARE_FRAUD_X5 Matches 5+ phrases commonly used in fraud spam
        *  0.5 NIGERIAN_BODY2 Message body looks like a Nigerian spam message 2+
        *  1.7 SARE_FRAUD_X3 Matches 3+ phrases commonly used in fraud spam
        *  1.7 SARE_FRAUD_X4 Matches 4+ phrases commonly used in fraud spam
        *  0.0 MSGID_FROM_MTA_HEADER Message-Id was added by a relay
```

If SPAM slips through your spamassassin system, you can use this method to adjust your rules to reduce the risk in future.

Updating Spamassassin’s Built-in Rules

The spamassassin package comes with a file, /etc/cron.d/sa-update, which updates the rule files in the /etc/mail/spamassassin/ directory each day. This makes the administra
easier.

Limiting your spam fighting efforts to the required_hits value isn't usually adequate. You will probably need additional spamassassin tools to be more selective and accurat
covered next.

Using Greylisting

To maximize the effect of their efforts, spammers try to send email as quickly as possible. They take note of the emails that bounce, so that they know which addresses to r
make their next mailing more efficient.

When mail servers receive mail too rapidly for them to handle, they can ask the sender to try again later. Spammers often view resending emails to valid addresses as a was
could be used to send mail to brand new addresses that belong to faster mail servers. Emails that need to be resent are usually abandoned.

Some emails need reliable delivery to be effective and the senders of these types of messages are willing to resend. These include bank statement notifications, ecommerce
subscription newsletters.

In a previous section we saw where spamassassin always rejects emails from blacklisted sources. With greylisting, sources are just asked to resend. One of the most popula
products is the milter-greylist package which also works seamlessly with spamassassin. It is easy to use and I’ll discuss how can be configured on your mail server.

Downloading and Installing milter-greylist

Most RedHat and Fedora Linux software product packages are available in the RPM format, whereas Debian and Ubuntu Linux use DEB format installation files. When s
remember that the filename usually starts with the software package name and is followed by a version number, as in milter-greylist-4.2.6-1400.fc14.x86_64.rpm. (For hel
installing the required packages, see Chapter 6, Installing Linux Software).

Note: The milter-greylist package is a sendmail add-on and does not run as a daemon. You do have to restart sendmail for the settings to take effect.

Configuring milter-greylist

Configuring milter-greylist requires these four quick steps:

1. Add the milter-greylist statements listed in the README file to your /etc/mail/sendmail.mc file:

```
INPUT_MAIL_FILTER(`greylist',`S=local:/var/milter-greylist/milter-greylist.sock')
define(`confMILTER_MACROS_CONNECT', `j, {if_addr}')
define(`confMILTER_MACROS_HELO', `{verify}, {cert_subject}')
define(`confMILTER_MACROS_ENVFROM', `i, {auth_authen}')
define(`confMILTER_MACROS_ENVRCPT', `{greylist}')
```

2. The previous step referenced the file /var/milter-greylist/milter-greylist.sock which now has to be created and owned by the grmilter user. You can do this by first search
/etc/passwd, to double check that the user first exists and that the directory is owned by this user also. Next create the file and change its ownership. The method can be see

```
[root@bigboy tmp]# grep grey /etc/passwd
grmilter:x:495:494:Greylist-milter user:/var/lib/milter-greylist:/sbin/nologin
[root@bigboy tmp]# touch /var/lib/milter-greylist/milter-greylist.sock
[root@bigboy tmp]# chown grmilter:grmilter \
                        /var/lib/milter-greylist/milter-greylist.sock
[root@bigboy tmp]# ll /var/lib/milter-greylist/milter-greylist.sock
-rw-r--r-- 1 grmilter grmilter 0 Dec 12 00:26 /var/lib/milter-greylist/milter-greylist.sock
[root@bigboy tmp]#
```

3. Configure Greylist to start automatically on reboot. Fedora / CentOS / RedHat

```
[root@bigboy tmp]# chkconfig spamassassin on
```

Ubuntu / Debian

```
user@ubuntu:~$ sudo sysv-rc-conf spamassassin on
```

4. Edit the /etc/mail/greylist.conf configuration file. Here we set the “try again later” to five minutes and use the whitelist command to deactivate the timer for trusted netwo

immediately.

```
#
# File: /etc/mail/greylist.conf
#

# How long a client has to wait before we accept
# the messages it retries to send. Here, 1 hour.
#
greylist 5m

#
# Whitelist addresses within my own home/office network
#
acl whitelist addr 192.168.0.0/16
```

5. Run the activate-sendmail.sh script for the new settings to take effect.

Your new spam mitigation tool should now be fully functional. You are ready to go!

Configuring milter-greylist

Now that we have milter-greylist installed, we need to be able to do some basic troubleshooting. The /var/log/maillog file should be used to determine what is happening tc
samples of what to expect:

```
Dec 24 00:32:31 bigboy sendmail[28847]: jBO8WVnG028847: Milter: to=<spamvictim@my-web-site.org>,
reject=451 4.7.1 Greylisting in action, please come back in 00:05:00

Dec 23 20:40:21 bigboy milter-greylist: jBO4eF2m027418: addr 211.115.216.225 from
<slashdot@slashdot.org> rcpt <spamvictim@my-web-site.org>: autowhitelisted for 24:00:00
```

In the first entry, the email received is given a tag (jBO8WVnG028847) based on key characteristics in the mail header and a request is sent to the sender to resend the ema
that is received with the same calculated key within the autowhite period configured in the greylist.conf file will then be automatically accepted without delay. In the secon
resent and immediately accepted. Any other email from that source within the next 24 hours will be accepted without delay.

Note: Greylisting is very effective, but you will have to tne its operation to make sure critical emails are not delayed at all. One soluton is to set the autowhite period in /etc
more than 24 hours especially if you get mail from certain recipients, such as newsletters, on a daily basis. This makes them arrive without interruption.

A Simple PERL Script To Help Stop SPAM

Blacklists won't stop everything, but you can limit the amount of unsolicited spam you receive by writing a small script to intercept your mail before it is written to your ma

This is fairly simple to do, because sendmail always checks the .forward file in your home directory for the name of this script. The sendmail program then looks for the fil
/etc/smrsh and executes it.

By default, PERL doesn't come with modules that are able to check e-mail headers and envelopes so you have to download them from CPAN (www.cpan.org). The most

- MailTools
- IO-Stringy
- MIME-tools
- Mail-Audit

I have written a script called mail-filter.pl that effectively filters out spam e-mail for my home system. A few steps are required to make the script work:

1. Install PERL and the PERL modules you downloaded from CPAN.
2. Place an executable version of the script in your home directory and modify the script's $FILEPATH variable point to your home directory.
3. Update file mail-filter.accept, which specifies the subjects and e-mail addresses to accept, and file mail-filter.reject, which specifies those to reject.
4. Update your .forward file and place an entry in /etc/smrsh.

Mail-filter first rejects all e-mail based on the reject file and then accepts all mail found in the accept file. It then denies everything else.

For a simple script with instructions on how to install the PERL modules, see Appendix II, "Codes, Scripts, and Configurations".

Configuring Your Dovecot POP / IMAP Mail Server

Each user on your Linux box will get mail sent to their account's mail folder, but sendmail just handles mail sent to your my-site.com domain. If you want to retrieve the m
user account using a mail client such as Evolution, Microsoft Outlook or Outlook Express, then you have a few more steps. You'll also have to make your Linux box a PC

Linux comes with the easy to use dovecot IMAP/POP server package which requires very little configuration after installation.

Installing Dovecot

Most RedHat and Fedora Linux software product packages are available in the RPM format, whereas Debian and Ubuntu Linux use DEB format installation files. When s
remember that the filename usually starts with the software package name and is followed by a version number, as in dovecot-0.99.11-1.FC3.4.i386.rpm. (For help on dow
required packages, see Chapter 6, Installing Linux Software).

Starting Dovecot

The methodologies vary depending on the variant of Linux you are using as you’ll see next.

Fedora / CentOS / RedHat

With these flavors of Linux you can use the chkconfig command to get dovecot configured to start at boot:

```
[root@bigboy tmp]# chkconfig dovecot on
```

To start, stop, and restart dovecot after booting use the service command:

```
[root@bigboy tmp]# service dovecot start
[root@bigboy tmp]# service dovecot stop
[root@bigboy tmp]# service dovecot restart
```

To determine whether dovecot is running you can issue either of these two commands. The first will give a status message. The second will return the process ID numbers

```
[root@bigboy tmp]# service dovecot status
[root@bigboy tmp]# pgrep spam
```

Note: Remember to run the chkconfig command at least once to ensure dovecot starts automatically on your next reboot.

Ubuntu / Debian

With these flavors of Linux the commands are different. Try installing the sysv-rc-conf and sysvinit-utils DEB packages as they provide commands that simplify the proces
and installing the packages, see Chapter 6, Installing Linux Software)

You can use the sysv-rc-conf command to get dovecot configured to start at boot:

```
user@ubuntu:~$ sudo sysv-rc-conf dovecot on
```

To start, stop, and restart dovecot after booting the service command is the same:

```
user@ubuntu:~$ sudo service dovecot start
user@ubuntu:~$ sudo service dovecot stop
user@ubuntu:~$ sudo service dovecot restart
```

To determine whether dovecot is running you can issue either of these two commands. The first will give a status message. The second will return the process ID numbers

```
user@ubuntu:~$ sudo service dovecot status
user@ubuntu:~$ pgrep dovecot
```

Note: Remember to run the sysv-rc-conf command at least once to ensure dovecot starts automatically on your next reboot.

Dovecot Configuration Files

You can define most of Dovecot's configuration parameters in the dovecot.conf file which may be located in either the /etc or /etc/dovecot directory depending on your ver

Remember to restart Dovecot after you make any changes to your configuration files. This is the only way to activate the new settings.

Choice of Protocols

You can select one of two protocols in your Dovecot configuration: IMAP and POP3. With POP3 your mail is downloaded to your computer so that you can work with it
reply to POP3 mail from different computers it will be difficult to get a complete picture of some threads as the replies sent on one computer won't be visible on the other. '
always remains on your mail server which eliminates this problem. It also allows you to create folders for your email which makes it easy to organize your e-mail and acce:

Each of these protocols operate on a different TCP port as shown in Table 21-1.

Protocol	TCP Port
POP	110
POPS	995
IMAP	143
IMAPS	993

This information will be required for your configuration file as you will soon see. You should also make sure your firewall rules allow traffic to access your server on these

Version 1.x

In this version, Dovecot would by default act as a server for IMAP, secure encrypted IMAP (IMAPS), POP and secure encrypted POP (POPS). You could limit this list by
the /etc/dovecot.conf file and then restarting dovecot for the change to take effect.In the example below dovecot is configured to serve only POP3.

Note: Unfortunately the POP3 and IMAP protocols send your username and password unencrypted which exposes your users to attacks. Dovecot expects you to use the r
IMAPS methods and therefore disables the use of plain text passwords by default. To enable the acceptance of plain text authentication the disable_plaintext_auth comman
the example also shows.

```
#
# File /etc/dovecot.conf sample
#
```

```
# Protocols we want to be serving imap imaps pop3 pop3s
#protocols = imap imaps pop3 pop3s
protocols = pop3
disable_plaintext_auth = no
```

You should always try to use secure POP3S or IMAPS for better peace of mind. More details on how to do this with newer versions of Dovecot will be covered next.

Version 2.x and Newer

In more recent versions, the syntax of the dovecot.conf statements used to define protocols has changed.

Both POP3 and IMAP settings are configured in a service section and you can define the IP addresses each should use and the TCP ports on which they should listen.

In this example, we have disabled IMAPS and POP3 by setting their inet_listener ports to zero. POP3S is working on address 192.168.1.100 while IMAP works on the lo
Both POP3S and IMAP listen on their respective TCP ports.

```
# Required to make POPS / IMAPS to work with certificates
ssl = yes
```

```
service pop3-login {
  inet_listener pop3 {
    port = 0
  }
  inet_listener pop3s {
    port = 995
    address = 192.168.1.100
  }
}

service imap-login {
  inet_listener imap {
    address = 127.0.0.1
    port = 143
  }
  inet_listener imaps {
    port = 0
  }
}
```

IMAPS and POP3S commonly rely on the use of SSL certificates for encryption. You make Dovecot aware that you intend to use this method with the ssl command. This
example. It is an important step.

Note: Always remember to restart Dovecot in order for these settings to take effect.

Verifiying Whether Dovecot is Listening

You can then use the netstat command to do a simple preliminary test to make sure dovecot is listening on the correct ports. In this example we see that IMAP is listening c
listening on the NIC IP address of server bigboy. It proof that our configuration works.

```
[root@bigboy tmp]# netstat -ta | egrep -i 'pop|imap'
tcp        0      0 localhost:imap              *:*     LISTEN
tcp        0      0 bigboy:pop3s       *:*              LISTEN
[root@bigboy tmp]#
```

It is often insufficient to use this as your only test. Try using the telnet command from another location to verify that remote client can contact your mail server on the corre
may have a routing or firewall issue, or dovecot may not be running. In this example we are testing on the POPS port, 995.

```
[root@bigboy tmp]# telnet mail.my-site.com 995
Trying 192.168.1.100...
Connected to mail.simiya.com.
Escape character is '^]'.
^]
telnet> quit
Connection closed.
[root@bigboy tmp]#
```

Connection problems could also be the result of typical network issues outlined in Chapter 4, "Simple Network Troubleshooting". Review this chapter if you find yourself
basic connectivity.

Configuring SSL Certificates for POP3S and IMAPS

As mentioned previously, when configuring POP3S and IMAPS you need to let Dovecot know where your certificates are. By default the certificates are named dovecot.p
should be found in your dovecot.conf file or one of its daughter configuration files in the /etc/dovecot/conf.d directory.The configuration should look like this.

```
ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
ssl_key = </etc/pki/dovecot/private/dovecot.pem
```

You can verify these commands are listed in your Dovecot configuration file tree. This can be done with a simple recursive grep command which searches /etc/dovecot an
with the string dovecot.pem in them. In this case the statements are found in the 10-ssl.conf file in the /etc/dovecot/conf.d directory.

```
[root@bigboy tmp]# grep -ir dovecot.pem /etc/dovecot/
/etc/dovecot/conf.d/10-ssl.conf:ssl_cert = </etc/pki/dovecot/certs/dovecot.pem
/etc/dovecot/conf.d/10-ssl.conf:ssl_key = </etc/pki/dovecot/private/dovecot.pem
[root@bigboy tmp]#
```

After finding the references you should verify that the files exist. This can be done with the locate command. Here we see the file locations previously listed in the configu
actually reside in the filesystem.

```
[root@bigboy tmp]# locate dovecot.pem
/etc/pki/dovecot/certs/dovecot.pem
/etc/pki/dovecot/private/dovecot.pem
[root@bigboy tmp]#
```

What do you do if you don’t have these files? Don’t worry, you can easily create them and this will be covered next.

Configuring SSL Certificates for POP3S and IMAPS

What do you do if you don’t have these files? Don’t worry, you can easily create them and this will be covered next. The mkcert.sh file will generate your Dovecot certific
configured in the dovecot-openssl.cnf file. You can use the locate command to find both files.

```
[root@bigboy tmp]# locate mkcert.sh
/usr/libexec/dovecot/mkcert.sh
[root@bigboy tmp]# locate dovecot-openssl.cnf
/etc/pki/dovecot/dovecot-openssl.cnf
[root@bigboy tmp]#
```

Though the contents of the dovecot-openssl.cnf file will be sufficient to genterate the SSL certificates, you may want to customize it to meet the needs of your organization

```
#
# File: dovecot-openssl.cnf
#

[ req_dn ]
# country (2 letter code)
C=US

# State or Province Name (full name)
ST=California

# Locality Name (eg. city)
L=San Francisco

# Organization (eg. company)
O=My-Site Inc

# Organizational Unit Name (eg. section)
OU=My-Site IT Department

# Common Name (*.example.com is also possible)
CN=mail.my-site.com

# E-mail contact
emailAddress=postmaster@my-site.com
```

The next step is to tun the mkcert.sh script and make sure the keys are in the right location.

```
[root@bigboy tmp]# /usr/libexec/dovecot/mkcert.sh
Generating a 1024 bit RSA private key
..........++++++
.....................++++++
writing new private key to '/etc/pki/dovecot/private/dovecot.pem'
-----

subject= /OU=My-Site IT Department/CN=mail.my-site.com/emailAddress=postmaster@my-site.com
SHA1 Fingerprint=A0:F9:95:1B:90:21:B9:B2:45:5B:CC:DF:20:2C:9E:25:74:69:F1:DD
[root@bigboy tmp]#
```

Now that your certificates have been created you should be ready to start serving secure email to your users.

Dovecot uses its own certificates and the method described here shows you how to create your own. If you are part of an enterprise with its own domain, you should inves
certificates created by an official certificate authority like Verisign. All email clients recognize organizations like these and will operate using POPS and IMAPS without di
stating that the certificate comes from an untrusted source.

For additional security you can install a separate certificate on all the client computers and configure Dovecot to only interact with clients these known credentials. How do
this book, but should be investigated to reduce your security risk.

Dovecot Mailboxes

Though sendmail sends your email to a local user account, Linux may store the content of the mail in one of many formats. Two common methods are mbox and maildir.

Dovecot uses the mail_location directive to define the type of mail format and the location of its files. This directive may be found in either your dovecot.conf file or one of
files in the /etc/dovecot/conf.d directory. It may also be commented out.

Verify that these directives are listed in your Dovecot configuration file tree. This can be done with a simple recursive grep command which searches /etc/dovecot and its s
the string mail_location in them. In this case the statements are found in the 10-mail.conf file in the /etc/dovecot/conf.d directory.

```
[root@bigboy tmp]# grep -ir mail_location /etc/dovecot
/etc/dovecot/conf.d/10-mail.conf:#   mail_location = maildir:~/Maildir
/etc/dovecot/conf.d/10-mail.conf:#   mail_location = mbox:~/mail:INBOX=/var/mail/%u
/etc/dovecot/conf.d/10-mail.conf:#   mail_location = mbox:/var/mail/%d/%1n/%n:INDEX=/var/indexes/%d/%1n/%n
/etc/dovecot/conf.d/10-mail.conf:#mail_location =
/etc/dovecot/conf.d/10-mail.conf:#mail_location = mbox:~/mail:INBOX=/var/mail/%u
[root@bigboy tmp]#
```

If you look closely, you will notice that the references are all commented out. The following sections will show you how to determine which method to use. If you select th
you won’t be able to download your mail, because Dovecot will be looking for it in the wrong location!

Configuring Dovecot for mbox

Mbox mail is stored in the directory /var/mail. Each user is assigned a single file that contains all their mail and the filename is the same as Linux username. If there are files
you are most likely using the mbox method.

```
[root@bigboy tmp]# ls /var/mail/
user1 user2 user3 user4 user5 user6 user7 user8 user9
[root@bigboy tmp]#
```

The configuration for mbox requires the addition of this line to your dovecot.conf file, or as in our case, uncommenting a similar line from the 10-mail.conf file. Either metl

```
mail_location = mbox:~/mail:INBOX=/var/mail/%u
```

Note: Remember to restart Dovecot for this setting to be activated.

Now it is time to take a look at the maildir method.

Configuring Dovecot for maildir

Maildir mails are almost always stored in a ~/Maildir/ directory in the users' home directory. Unlike the mbox method, with maildir each mail is stored in a separate file.

To configure Dovecot for your maildir mail, use this directive:

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

Note: Remember to restart Dovecot for this setting to be activated.

You are done! That was easy.

Different distributions of Linux use differing methods of storing email. If neither mbox or maildir seems to be the method your system is using then check the Dovecot wel
further details.

Configuring Your Mail Clients

By default your POP / IMAP e-mail accounts will be the regular Linux user accounts in which sendmail has deposited mail. You can now configure your e-mail client to u
server quite easily. For example to configure POPS Mail, set your POPS mail server in the client program to be the IP address of your Linux mail server. Use your Linux u
when prompted.

If you are using a self signed SSL certificate, your mail client will give a warning an ask whether the certificate should be accepted. You will have to say "yes'.

Next, set your SMTP mail server to be the IP address/domain name of your Linux mail server.

How to handle overlapping email addresses.

If you have user overlap, such as John Smith (john@my-site.com) and John Brown (john@another-site.com), both users will get sent to the Linux user account john by de
for a solution:

- Make the user part of the email address different, john1@my-site.com and john2@another-site.com for example, and create Linux accounts john1 and john2. If the u
names, then you may need to modify your virtusertable file.
- Create the user accounts john1 and john2 and point virtusertable entries for john@my-site.com to account john1 and point john@another-site.com entries to account
configuration in Outlook Express for each user should retrieve their mail via POP using john1 and john2, respectively.

With this trick you'll be able to handle many users belonging to multiple domains without many address overlap problems.

Troubleshooting Dovecot Mail

The very first troubleshooting step is to determine whether your server is accessible on the correct TCP ports. For example, with POP use TCP port 110 or for POPS use p
connectivity could be caused by a firewall with incorrect permit, NAT, or port forwarding rules to your server. Test this from both inside your network and from the Intern
with TELNET is covered in Chapter 4, "Simple Network Troubleshooting")

Always Start with Logging

Whenever you are in doubt turn on Dovecot's debugging features to reveal more about what is happening. In more recent versions of Dovecot, the logging sections in dov
to a logging configuration file in the /etc/dovecot/conf.d directory. In this example the file is named 10-logging.conf.

```
[root@bigboy tmp]# ls /etc/dovecot/conf.d/*log*
/etc/dovecot/conf.d/10-logging.conf
[root@bigboy tmp]#
```

The file has many sections that allow you to turn on very verbose debugging level messages for authentication, SSL, and general messaging. It is an invaluable source of tı
Dovecot logs to the /var/log/maillog file. For details on setting up Linux logging refer to Chapter 5, "Troubleshooting with syslog." Here are some good examples:

- In this case the Maildir mail_location method was incorrectly chosen and the expected mail files were not found

```
Dec  5 20:49:47 bigboy dovecot: pop3(mail-user1): Debug: maildir: access(/home/users/mail-user1/Maildir, rwx): failed: No such file or directory
Dec  5 20:49:47 bigboy dovecot: pop3(mail-user1): Debug: maildir: couldn't find root dir
```

- In this case Dovecot's autodetection method failed to determine the correct mail_location. The directive had to be manually added.

```
Dec  5 09:10:26 bigboy dovecot: pop3(mail-user2): Error: user lhn-mail: Initialization failed: mail_location not set and autodetection failed: Mail storage autodetection failed wi
```

Whenever there is any doubt, look for the error message in the log file, try to understand what it means and what could be done to fix the problem. Remember, finding help
Internet will be much easier if you search for key parts of your log message.

Conclusion

E-mail is an important part of any Web site, and you need to plan its configuration carefully to make it a seamless part of the Web experience of your visitors. Without it, y
complete.

A fully functioning Web site is just the beginning. It needs to be maintained to reduce the risk of failure and monitored to help detect potential problems. Chapter 22, " Mor
Performance", discusses many Linux-based tools that you can be use to track the health of your Linux server.

Retrieved from "http://www.linuxhomenetworking.com/wiki/index.php?title=Quick_HOWTO_:_Ch21_:_Configuring_Linux_Mail_Servers&oldid=4331"