Mail Server Implementation

Timothy Vismor

November 2005

Abstract

Configuration of a full featured mail server using Postfix, Cyrus IMAP, Clam AV, Amavis-new, and Spam Assassin.

Note

This document is retained for archival purposes. It does not describe current practice.

Contents

1 Software and Local Network Environment **1**

2 Postfix Mail Server **1**
2.1 Verifying Recipients with Postfix 3
2.2 Checking Recipients with Cyrus IMAP 5
2.3 Postfix Configuration 5

3 Amivis Mail Filtering Proxy **8**
3.1 Amavis Before Queue Filter 8
3.2 Amavis Configuration 8

4 Concluding Remarks **10**

List of Figures

1 Inbound Mail Routing Diagram 2
2 Recipient Verification Using Postfix 4
3 Recipient Verification Using Cyrus IMAP 6
4 Mail routing showing Amavis implementation 9

1 Software and Local Network Environment

This document describes the configuration of mail services on a Fedora (FC3 or FC4) machine serving a small research network of Windows, Mac, and Linux boxes. The system is probably "over engineered" for its actual use case but may prove interesting in other environments. The main components include:

- Postfix (2.2.2) as the mail transfer agent.
- Amavisd-new (2.3.1) as a SMTP proxy for spam filtering and anti-virus scanning.
- SpamAssassin (3.0.3) as the spam filtering agent.
- Razor Agent (2.67) as a supplemental spam filter.
- ClamAV (0.85) as the anti-virus scanner.
- Cyrus IMAP (2.2.12) as the mail store.
- Cyrus SASL (2.1.20) acts as the mail store's authentication agent.

The IMAP mail store also resides on this machine. Port 25 is open to the internet for SMTP traffic. Inbound mail is routed as shown in Figure 1.

Network Environment. The local network is isolated behind a NAT router. It is assigned a private domain name (e.g. *internal.lan*). The mail host is physically located on the local network (e.g. *mail.internal.lan*). A public domain name (e.g. *external.com*) is mapped to the router's IP address by a commercial DNS registration service. All local network users are provided with virtual mail boxes in the public domain's name space (e.g. *user@external.com*). All virtual mail boxes are housed in the IMAP mail store. Deliveries to local accounts on the mail host are also routed to the IMAP store.

Software Availability. The system is built from off the shelf RPM's. An attempt is made to stay as "close to the source" as possible. Hence, the preferred repositories are Fedora and Fedora Extras. Other "reputable" sources are used as needed. Currently, the following packages are used:

- From FC4 / FC4 Updates : Postfix, SpamAssassin, and Cyrus SASL.
- From FC4 Extras : Cyrus IMAP, ClamAV, and Razor-Agent.
- From Dag Wier's repository : amvisd-new.

2 Postfix Mail Server

Describes the message flow and configuration of a mail server *ſſ* based on the Postfix mail transfer agent. The basic mail routing scheme is Postfix receives incom-

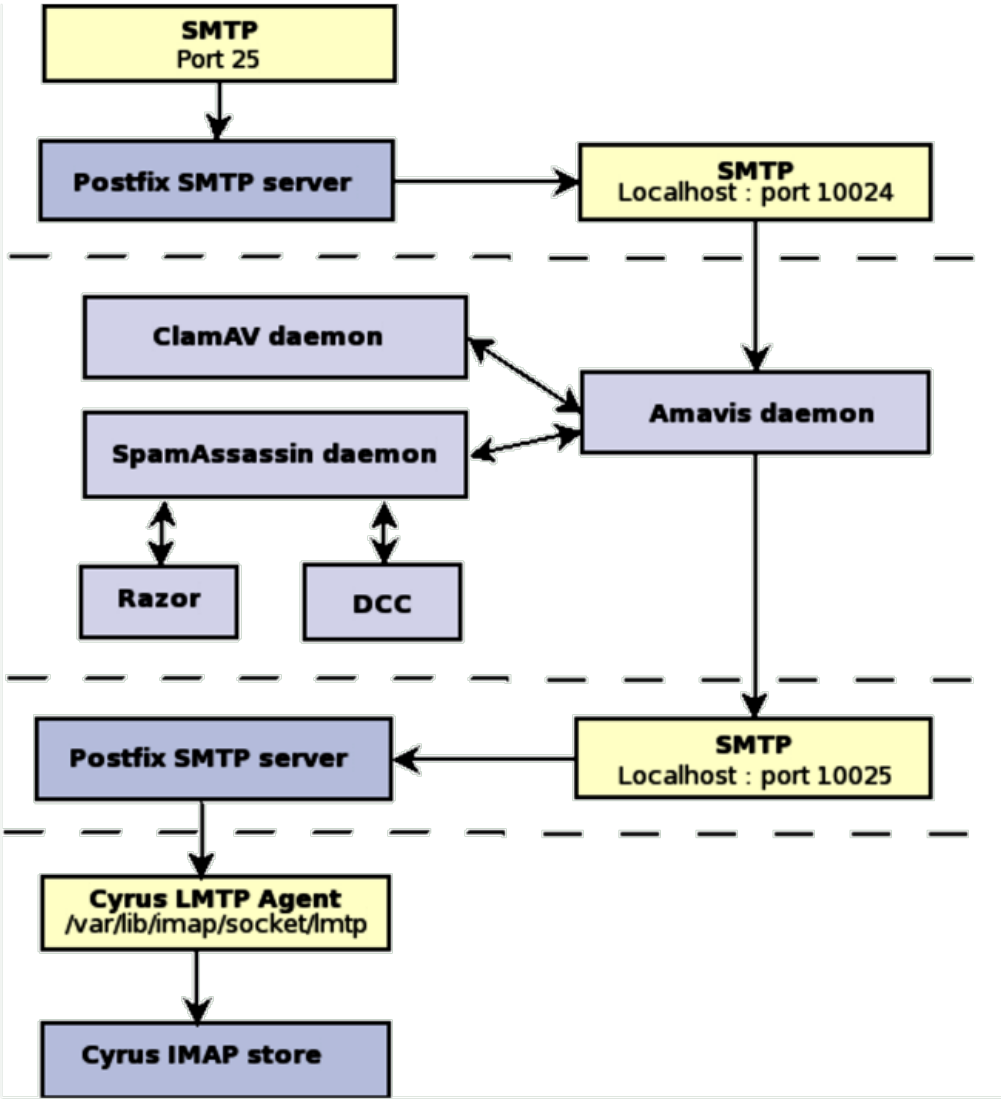

Figure 1: Inbound Mail Routing Diagram

ing messages and funnels them through a Amavis/Clam AV/Spam Assassin proxy where they are scanned for scanned for spam and malware. Clean messages are then passed along by Postfix to a Cyrus LMTP delivery agent and Cyrus IMAP mail store.

Two alternate recipient verification scenarios are examined. In the first, the Postfix server performs recipient verification prior to malware processing. In the second, Cyrus performs primary recipient verification just before mail is delivered to virtual recipients.

2.1 Verifying Recipients with Postfix

The Postfix server receives incoming SMTP transmissions on port 25, hands the message off to a pass-through filtering proxy (amavisd-new), then sends valid messages on to the Cyrus LMTP transport for delivery to the message store. The diagram in Figure 2 illustrates this process.

In this configuration, the amavisd-new mail scanner serves as a Postfix "before-queue" content filter. This means that spam and virus filtering occur before the message is added to the postfix mail queue for cleanup and distribution. This technique is suited for low volume sites (or high memory sites) since a separate amavis process is required for each concurrent connection to the server. High volume sites usually configure amavisd as an "after-queue" content filter.

The processing pipeline is as follows:

- The "before-filter" Postfix SMTP server receives mail from the Internet and performs relay access control and recipient verification. If the SMTP server detects one of these problems, the message is rejected outright. Otherwise, the unfiltered content is passed on to amavisd.

  Note: Postfix SASL authentication also occurs before the content filter is activated (although it is disabled on this host).

- Amavis examines the message content and either (a) reinjects it into the Postfix message stream or

  (b) returns an error code which causes Postfix to reject the message.

If the message is accepted for delivery, it is queued, cleaned up, and dispatched to the Cyrus LMTP delivery agent. If Cyrus is unable to locate the recipient's virtual mail box, the message is discarded and a bounce notification is passed along to the

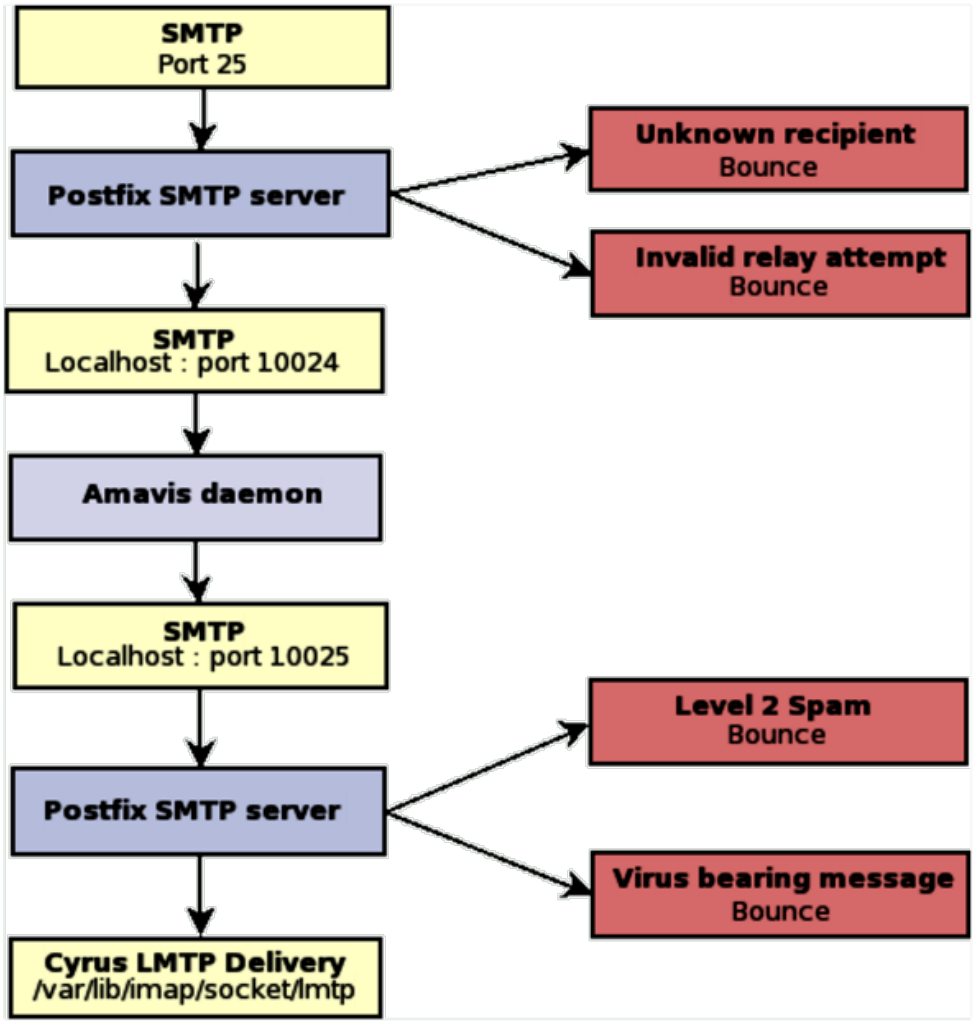

Figure 2: Recipient Verification Using Postfix

sender. This bounce should not occur unless the Postfix and Cyrus virtual mailbox lists are out of sync.

2.2 Checking Recipients with Cyrus IMAP

An alternate Postfix configuration eliminates the need for maintaining two lists of virtual users (one in a *vmailbox* file and a second in Cyrus IMAP). It is quite simple to implement: just remove the *virtual_mailbox_maps* definition from *main.cf*.

If the virtual mailbox list is undefined, the Postfix server does not reject any mail addressed to the virtual domain *(external.com)*. Mail addressed to virtual mail boxes passes though the normal pipeline (i.e. is scanned for viruses and spam) and is presented to the Cyrus LMTP transport for delivery. If Cyrus is unable to locate the recipient's virtual mail box, the message is discarded and it generates a bounce notice that is forwarded to the sender via the Postfix SMTP client. Figure 3 illustrates this process.

Note that under this configuration, bounces for unknown users in the public domain *(unknown@external.com)* don't occur until the mail has already been filtered and sent to the IMAP server for delivery. The actual Postfix server's recipient verification is fairly useless since it only bounces messages to the mail host *(unknown@mail.internal.lan)*. Since these addresses can only be reached while logged into the private network, the situation rarely arises.

Obviously, one drawback to this configuration is wasted time and processing power. Messages to invalid recipients are scanned for viruses and spam before they are bounced.

2.3 Postfix Configuration

Postfix is generally set to its default configuration. The exceptions are:

- Host/network data is changed to match the local site.
- Virtual mailbox deliveries for the public domain are routed to Cyrus IMAP through a unix socket.
- Local mailbox deliveries are routed to Cyrus IMAP through a unix socket.
- TLS support is enabled. The certificates are issued by a private certificate authority (us).

Customizations to the base configuration file (main.cf) follow.

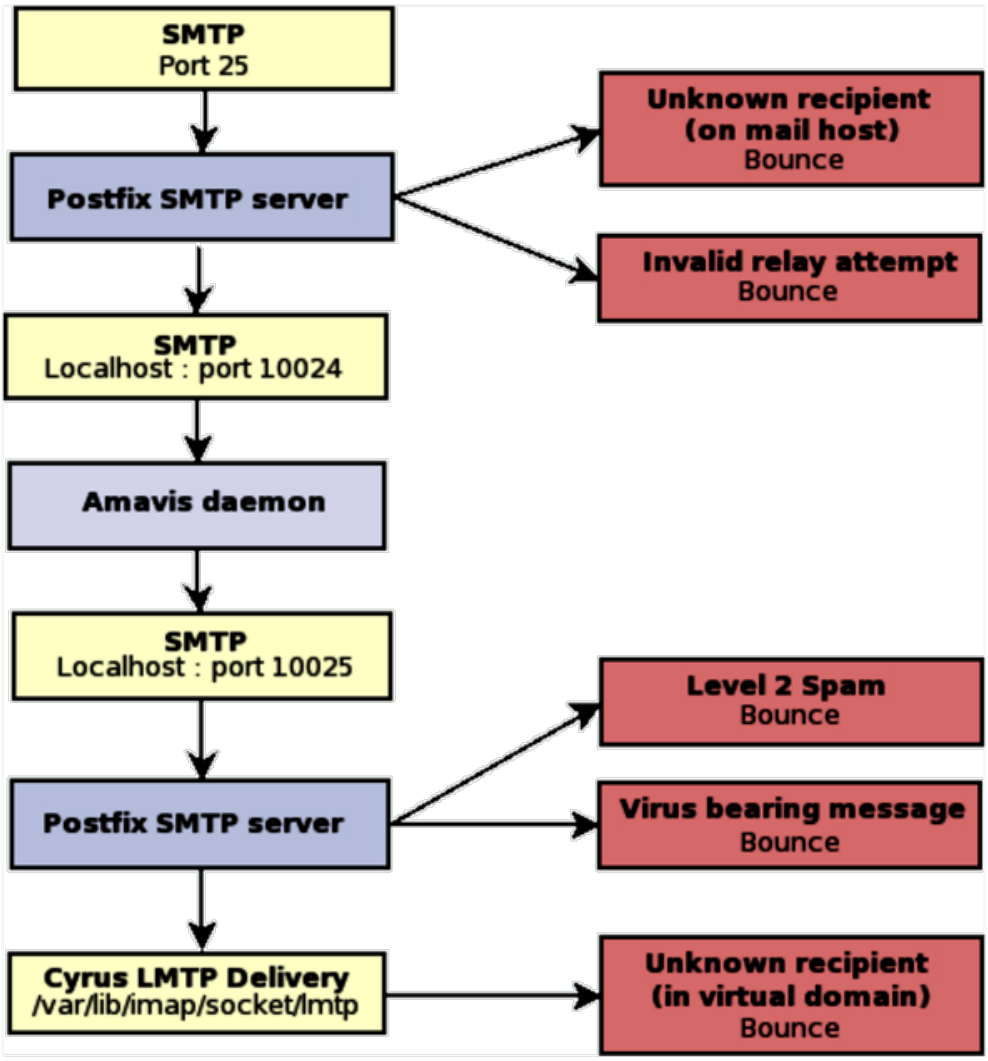

Figure 3: Recipient Verification Using Cyrus IMAP

```
# Define the Local network configuration.
myhostname = mail.internal.lan
mynetworks = 192.168.1.0/24, 127.0.0.0/8
# Define the public domain name.
virtual_mailbox_domains = vismor.com
# Define the list of virtual mail box addresses.
virtual_mailbox_maps = hash:/etc/postfix/vmailbox
# Use Cyrus Imap as the local delivery agent.
virtual_transport = lmtp:unix:/var/lib/imap/socket/lmtp
mailbox_transport = lmtp:unix:/var/lib/imap/socket/lmtp
# Enable the use of TLS.
smtpd_use_tls = yes
# Location of the server certificates.
smtpd_tls_cert_file = /etc/pki/tls/certs/smtp.crt
smtpd_tls_key_file = /etc/pki/tls/private/smtp.key
# Tighten rejection criteria slightly.
disable_vrfy_command = yes
smtpd_helo_required = yes
smtpd_data_restrictions =
   reject_unauth_pipelining,
   permit
```

The default Postfix server configuration file *(/etc/postfix/master.cf)* is modified to establish amavisd-new as a "before queue" content filter. Modifications to the default configuration follow.

```
# Before-filter SMTP server. Receive mail from the network and
# passes it on to the content filter (amavisd) on localhost
# port 10024.
# svc type  private unpriv  chroot  wakeup  maxproc cmd+args
smtp    inet  n       -       n       -       -    smtpd
  -o smtpd_proxy_filter=127.0.0.1:10024
  -o smtpd_client_connection_count_limit=4
# After-filter SMTP server. Receive mail from the content
# filter (amavisd) on localhost port 10026.
127.0.0.1:10025   inet  n     -      n      -      -    smtpd
  -o smtpd_authorized_xforward_hosts=127.0.0.0/8
  -o smtpd_client_restrictions=
```

```
  -o smtpd_helo_restrictions=
  -o smtpd_sender_restrictions=
  -o smtpd_recipient_restrictions=permit_mynetworks,reject
  -o smtpd_data_restrictions=
  -o receive_override_options=no_unknown_recipient_checks<
```

Valid virtual mailbox accounts are listed in the text file *etc/postfix/vmailbox*. Its contents are converted to a hash after each modification using the following command.

```
postmap /etc/postfix/vmailbox
```

3 Amivis Mail Filtering Proxy

The Amavis-new proxy coordinates all the malicious activity screening of the mail server. It routes incoming messages through the Clam Antivirus daemon and Spam Assassin. After a message is processed, it informs the Postfix server of the results of the evaluation. The use of external agents by SpamAssassin, such as DCC and Razor Agent, is supported.

3.1 Amavis Before Queue Filter

As mentioned in Section 2.1, amavisd-new is configured as a pass-through filtering proxy for the Postfix server. It serves as a "before-queue" content filter that scans messages for spam and viruses before they are added to the Postfix mail queue for cleanup and distribution. Amavisd-new provides a consistent infrastructure for screening incoming mail for undesirable content. Currently, amavis-new is set up to check for viruses with ClamAv and weed out spam with SpamAssassin. External spam filtering services, Razor and DCC, assist SpamAssassin in this task. Figure 4 illustrates the relationship between these components.

3.2 Amavis Configuration

Amavis is generally in its default configuration. The following exceptions are noted:

```
# number of pre-forked children (2-15 is common)
$max_servers = 1;
@local_domains_maps = ();
```

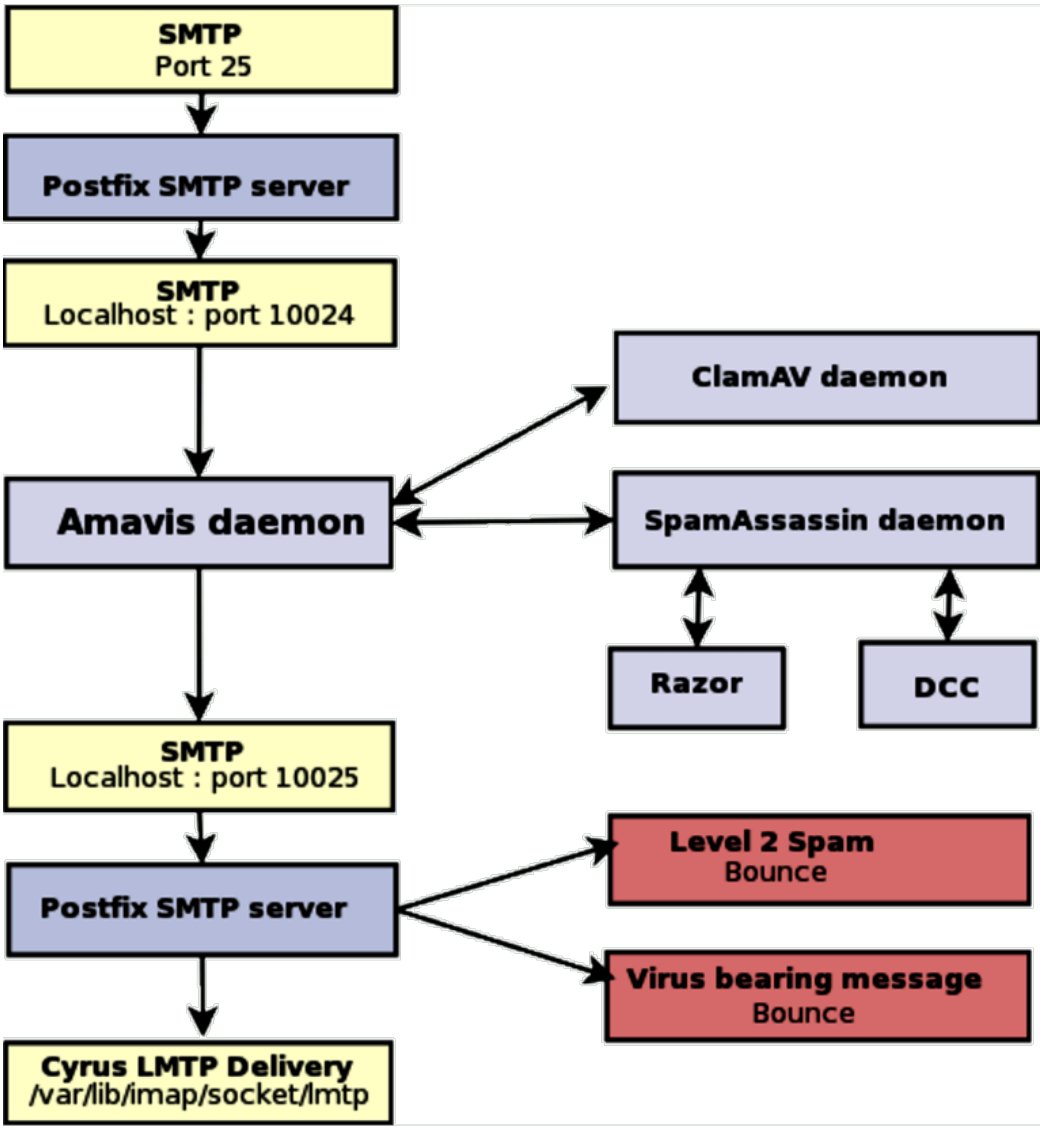

Figure 4: Mail routing showing Amavis implementation

```
#@local_domains_maps = ( [".$mydomain", "vismor.com"] );
# verbosity 0..5
$log_level = 2;
# spam level beyond which a DSN is not sent.
$sa_dsn_cutoff_level = 6.31;
# notifications recip.
$virus_admin = "root\@redbud.$mydomain";
$final_banned_destiny = D_REJECT;
[ 'ClamAV-clamd',
   &ask_daemon,
   ["CONTSCAN {}\n", "/var/run/clamd.amavis/clamd.sock"],
   qr/\bOK$/, qr/\bFOUND$/,
   qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

4 Concluding Remarks

This document was originally intended as internal documentation of the implementation of a Linux mail server on a small research network. The environment was constantly changing since the machine was following the Redhat/Fedora "Rawhide" distribution. Hence, formal documentation was important in keeping the mail server configuration steady in a constantly shifting environment. However, we have since outsourced our mail services to an external vendor, obviating the need for this document.

This is a long way of saying that the final two sections of this document will probably never be written. The original plan was to describe the implementation of the other two pieces of a mail solution: fighting spam and fighting malware (proving once again that if you stall long enough, most tasks become irrelevant).

In particular, the following tasks are incomplete:

- Documenting the details of *SpamAssassin* integration, including interconnection with external spam filters, and
- Documenting the details of *ClamAV* integration.

No internet facing mail service is complete (or practical) without these features.

Obviously, this documentation will remain "stuck in time" (during the FC4 timeframe) unless we start maintaining our own mail server again or decide to set up a local mail server for experimental purposes. Also, many Redhat-related links have become stale as the Fedora Project and its infrastructure have evolved over time.