

SECTION 12 Configure a Mail Server

The standard mail server in SUSE Linux Enterprise Server 10 is Postfix.

In this section you will learn some backgrounds about the SMTP Protocol. Tools to detect spam mails and viruses are also introduced.

Objectives

1. Understand SMTP Communication
2. Manage Spam
3. Use a Virus Scanner for Email

Objective 1 Understand SMTP Communication

In the following we want to have a closer look at:

- The SMTP Commands
- Command Syntax
- SMTP Reply Codes
- Minimal SMTP Command Implementation
- An Example for Sending Mail with Telnet

The SMTP Commands

The commands are:

- **HELO** (Hello) is used by the SMTP sender to open a connection to an SMTP recipient (a greeting) and to introduce itself with its full host name. This host name is passed as an argument attached to the **HELO** command.

The SMTP recipient answers this greeting with its full host name. This completes the initialization of communication between the SMTP sender and the recipient.
- **MAIL FROM** initializes transmission of an email message. In the simplest case, the **MAIL FROM** command takes the sender's email address (reverse-path) as an argument. The argument can be optionally extended to include a list of hosts in front of the recipient's address via which a reply should be routed (source route relaying), e.g.:
 - @venus.example.com
 - @mars.example.com:Sales@example.com
- **RCPT TO** (Recipient) sets the recipient's address (forward-path) for an item of mail. If the mail should be sent to several recipients at the same time, this command is repeated that number of times.

The command may also include a list of hosts (e.g., @mars.example.com,@venus.example.com:info@example.com) via which the mail should be routed (source route relaying).

If the argument contains source route relaying through a list of hosts, the mail will be routed directly to the first host given. This host removes itself from the beginning of the list and forwards the mail to the next host in the list.

- **DATA** tells the SMTP mailer that anything that follows is the content of the mail. The end of the mail content is indicated by Enter.

When the SMTP recipient recognizes the end of the mail transmission, it begins to process the information received. This involves interpreting and possibly updating the forward and reverse paths. If this data processing results in no failures, the SMTP sender is sent an **OK**. Otherwise, an error message is sent.

A time stamp is inserted at the beginning of the mail as soon as the SMTP recipient receives a mail to forward to a further SMTP recipient or for direct delivery to the recipient. This time stamp contains information about the identity of the SMTP sender, the SMTP recipient, and the time when the mail reached the SMTP recipient.

Mails that are routed over several SMTP relays contain a corresponding number of these time stamps. If the SMTP recipient is responsible for the ultimate delivery of the mail (if, for example, the addressee's mailbox is located on the SMTP recipient), the current contents of the return path (i.e., the contents of the **MAIL FROM** argument after being modified through the previous SMTP relays) are inserted at the top of the mail.

- **VERFY** (Verify) verifies a user ID. This causes the SMTP recipient to check the validity of the addressee given as an argument. If the SMTP reader knows the addressee, the address is expanded into a full address (including domain). The **VERFY** command has no influence on the **MAIL FROM**, **RCPT TO**, and **DATA** instructions.
- **EXPN** (Expand) instructs the SMTP recipient to treat the argument as a mailing list. As a result, the members of the list are transmitted to the SMTP sender.
- **RSET** (Reset) resets all the information previously stored about the SMTP recipient. The forward path, reverse path, and mail contents are lost.
- **HELP** takes as its argument any other SMTP command. A page of tips about how to use the corresponding instruction will be displayed.
- **NOOP** (No Operation) causes the SMTP recipient to answer with an **OK**. Apart from that, no changes are made to the mail content or the forward and return paths.
- **QUIT** ends the connection between the SMTP sender and recipient.

Command Syntax

SMTP commands provide direct communication between the sender and recipient. The most commonly used SMTP commands are described in the following table.

Table 12-1

| Command Syntax | Description of the Arguments |
|--|--|
| HELO <i>hostname</i> | <i>hostname</i> contains the complete host name of the SMTP sender. |
| MAIL FROM: <i>reverse-path</i> | <i>reverse-path</i> contains the relay path for source route relaying and the sender's address. |
| RCPT TO: <i>forward-path</i> | <i>forward-path</i> contains the relay path for source route relaying and the recipient's address. |
| DATA <i>data</i> | <i>data</i> constitutes the contents of the mail (the actual message). |
| RSET | ./. |
| VERFY <i>string</i> | <i>string</i> contains a user or mailbox name. |
| EXPN <i>string</i> | <i>string</i> contains a mailing list identifier. |
| HELP [<i>string</i>] | <i>string</i> contains any SMTP command. |
| NOOP | ./. |
| QUIT | ./. |

Commands can be written in lower case or upper case; SMTP commands are not case sensitive. For example, the SMTP recipient will treat the following commands in the same way: **MAIL FROM**, **Mail From**, **mail from**, **Mail FrOm**.

In contrast, the arguments in *forward-path* and *reverse-path* may be case-sensitive. The interpretation of these arguments depends on the SMTP recipient's operating system and the structure of the user database, which is why lowercase and uppercase characters may be treated differently.

SMTP Reply Codes

During communication between the SMTP sender and SMTP recipient, the sender transmits various commands to the recipient and controls the course of the communication as a whole. The recipient acknowledges the message's receipt and gives the status of command processing with a corresponding reply code. Only when the sender has received a receipt for the previous command can it start transmitting the next command.

The most commonly found SMTP reply codes are listed in the table below:

Table 12-2

| Reply Code | Description |
|-------------------|---|
| 211 | System status or system help reply. |
| 214 | Displays the help message after entering HELP . |
| 220 | SMTP server ready. |
| 221 | SMTP server has closed connection. |
| 250 | Specified command has been carried out. |
| 251 | User not local; will forward to <i>forward-path</i> . |
| 354 | Start of the DATA entry. |
| 421 | SMTP service not available. |
| 450 | Requested action not taken (mailbox is already in use). |

Table 12-2 (continued)

| Reply Code | Description |
|-------------------|--|
| 451 | Requested action aborted. |
| 452 | Requested action not taken; insufficient storage space in system. |
| 500 | Syntax error, command unrecognized. |
| 501 | Syntax error in parameters or arguments. |
| 502 | Command not implemented. |
| 503 | Bad sequence of commands. |
| 504 | Command not implemented for that parameter. |
| 550 | Requested action not taken; file unavailable (e.g., mailbox not found, no access). |
| 551 | User does not exist on mail server; try <i>forward-path</i> . |
| 552 | Requested mail action aborted: storage allocation exceeded. |
| 553 | Requested action not taken: mailbox name not allowed (e.g., mailbox syntax incorrect). |
| 554 | Transaction failed. |

Minimal SMTP Command Implementation

Not all commands introduced up to now are implemented by every server. Certain SMTP commands are intentionally not included in some server implementations to increase the security of the server. One such security-critical command is **HELP**. The Postfix SMTP server by Wietse Venema has not fully implemented this command and simply responds to a **HELP** command with reply code 502 ("Command not implemented").

The following list shows the SMTP commands that an SMTP server absolutely must implement to provide SMTP communication:

- **HELO**
- **MAIL FROM:**
- **RCPT TO:**
- **DATA**
- **RSET**
- **NOOP**
- **QUIT**

An Example for Sending Mail with Telnet

The text below shows a typical example of the log output that can easily be reconstructed using a telnet session on port 25 (standard mail server port):

```
da53:~ # telnet da51.digitalairlines.com 25
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
220 da51.digitalairlines.com ESMTP Postfix
HELO da53.digitalairlines.com
250 da51.digitalairlines.com
MAIL FROM: jgoldman@digitalairlines.com
250 Ok
RCPT TO: geeko@digitalairlines.com
250 Ok
DATA
354 End data with <CR><LF>.<CR><LF>
Subject: Lorem ipsum dolor sit amet

Lorem ipsum dolor sit amet, consectetur adipiscing elit,
sed diem nonummy nibh euismod tincidunt ut laoreet dolore
magna aliquam erat volutpat. Ut wisis enim ad minim veniam,
quis nostrud exerci tution ullamcorper suscipit lobortis
isl ut aliquip ex ea commodo consequat.
.
250 Ok: queued as 45B5116A9D
QUIT
221 Bye
Connection closed by foreign host.
da53:~ #
```

Objective 2 Manage Spam

This objective explains the following:

- Use SpamAssassin
- Test SpamAssassin

Use SpamAssassin

spamassassin (and **spamc**) expect their input from STDIN.

The simplest way to use SpamAssassin is to pipe the mailbox into the command **spamassassin**:

```
da51:~ # cat /var/spool/mail/root | spamassassin
```

If you want to use SpamAssassin with your Postfix configuration, the easiest way is to use Procmail. You need to edit only three files:

- Create the file `/etc/procmailrc` with this content:

```
#LOGFILE=/tmp/procmail.log
#VERBOSE=yes

# Until now, mail is untagged, you may add rules for
# mail that must not be tagged

:0 hbfw
| /usr/bin/spamc
```

The mail is piped in the first filter rule to the `spamc`. The flags of the filter rule are explained below:

- **h.** Feed the header to the pipe, file, or mail destination (default).
- **b.** Feed the body to the pipe, file, or mail destination (default).

- **f.** Consider the pipe as a filter.
- **w.** Wait for the filter or program to finish and check its exitcode (normally ignored); if the filter is unsuccessful, then the text will not have been filtered.
- Make sure that you activate Procmail in `/etc/postfix/main.cf`:

```
mailbox_command = /usr/bin/procmail
```

- Make sure that your smtpd definition in the file `/etc/postfix/master.cf` is set to default

```
smtp      inet  n       -       n       -       -       smtpd
```

After this, start spamd by entering

rcspamd start

Spam can be filtered from the mail client or in a further Procmail configuration (`~/procmailrc`) by adding lines similar to this:

```
:0
* ^X-Spam-Status: Yes
$MAILDIR/Spam
```

Test SpamAssassin

You can use **telnet** to test your configuration. You also can send a spam email directly using the **sendmail** command:

```
cat sample-spam.txt | /usr/sbin/sendmail geeko@digitalairlins.com
```

Objective 3 Use a Virus Scanner for Email

Communication via email is very important for companies and individuals today, but email can be infected by virus software.

Most of the viruses attack Windows clients, but the mail server of a company is often a Linux or UNIX machine. To avoid damage we recommend to search for viruses before the infected mail arrives at the user's client.

SUSE Linux Enterprise Server 10 provides tools to detect viruses in email on your mail server.

In this objective, the following tools are introduced:

- AVMailGate
- AMaViSd-new

AVMailGate

AVMailGate is the abbreviation for AntiVir MailGate, an antivirus mail filter from H+BEDV Datentechnik GmbH (<http://www.hbedv.com>).

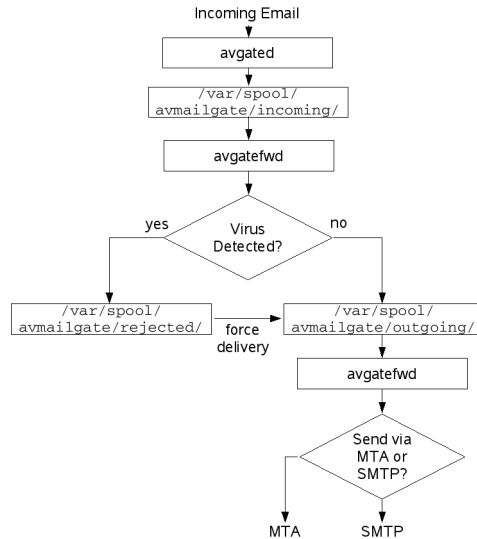
AVMailGate updates the virus definition file and the engine itself.

This topic describes the following:

- AVMailGate Architecture
- Install AVMailGate
- Configure AVMailGate
- Understand the AVMailGate Configuration Files
- Update AVMailGate

AVMailGate Architecture

Figure 12-1



AVMailGate is composed of three queues and two kinds of processes:

- **Queues** (located in /var/spool/avmailgate/)
 - **incoming/**. The input queue for all incoming email.
 - **rejected/**. The queue where possible infected email are stored.
 - **outgoing/**. The output queue for not infected email.
- **Processes**
 - **avgated**. The smtpd receiver that stores incoming email in the input queue. (daemon)
 - **avgatefwd**. Virus-scanning function and SMTP forwarder and sendmail invoker. (daemon)

Both processes can be configured by editing the file **/etc/avmailgate.conf**.

This file includes comments that describe the possible settings.



For a description of all options, see
</usr/share/packages/avmailgate/doc/MANUAL>.

Install AVMailGate

To install AVMailGate, select the package **avmailgate**.

A startscript is available in **rcavgate** to start and stop AVMailGate.

To start AVMailGate at the system's start, enter **insserv avgate**.

Configure AVMailGate

There are two possibilities to use AVMailGate with Postfix:

- AVMailGate Waits for Mails on Port 25 and Forwards Them to Postfix
- AVMailGate Is Used from Postfix as Content Filter

AVMailGate Waits for Mails on Port 25 and Forwards Them to Postfix

To use AVMailGate as mail proxy that waits for mails on port 25 you simply have to avoid that Postfix listens on this port. Therefore mark the following line in `/etc/postfix/master.cf`: as command by adding a hash (“#”) in front

```
#smtp      inet  n       -       n       -       -       smtpd
```

Then restart Postfix.

AVMailGate Is Used from Postfix as Content Filter

The best way to use AVMailGate with Postfix is to use the Full Content Filter API (see `FILTER_README` in the Postfix package for details).

It waits for SMTP connections on port 10024 and sends scanned email messages or virus warnings out per SMTP on port 10025.

To integrate AVMailGate as a filter in Postfix, do the following:

1. Edit `/etc/postfix/master.cf` and uncomment –if applicable– or add the following line:

```
localhost:10025 inet n - y - - smtpd -o
content_filter=
```

2. Edit `/etc/postfix/main.cf` and add the following line:

```
content_filter = smtp:127.0.0.1:10024
```

3. Since AVMailGate listens on port 10024, not port 25, edit `/etc/avmailgate.conf` and set

```
ListenAddress    localhost    port 10024
```

4. You have to tell AVMailGate it should send email back to Postfix via SMTP on host localhost via port 10025.

Edit `/etc/avmailgate.conf` and set

```
ForwardTo    SMTP: localhost port 10025
```

5. Since AVMailGate sends out notification messages as AVMailGate, set an alias in `/etc/aliases`:

```
vmailgate:    root
```

You must run **newaliases** afterward.

6. After these changes, enter

rcpostfix reload

and

rcavgate start

Your system is now ready to scan email.

Understand the AVMailGate Configuration Files

As mentioned before, the configuration file of AVMailGate is `/etc/avmailgate.conf`. The options of this file can be grouped:

- General Parameters
- Scanning of Files in an Archive
- Handling Envelope Recipient Addresses
- Adding a Notification in the Body of Transmitted Mails
- Other Configuration Files

General Parameters

- **User** and **Group**. `avgated` and `avgatefwd` run with the permissions of this user and group.
- **Postmaster**. Errors and alert messages are sent to this user.
- **MyHostName**. Hostname of the computer. If not set, it is retrieved by `gethostname`.
- **SpoolDir**. The directory where the queues are stored.
- **AntiVirDir**. Directory where the virus signatures are stored.
- **TemporaryDir**. Temporary directory where email messages will be extracted and scanned.

- **PidDir.** The location of the PID files.
- **LogFile.** Location of the log file.
- **SyslogFacility.** Facility argument for the syslog daemon (default: **mail**).
- **MaxIncomingConnections.** Maximum number of simultaneous connections.
- **SMTPTimeout.** Number of seconds until an SMTP timeout occurs.

More detailed timeouts can be configured by using the following options:

- **SMTPGreetingTimeout**
- **SMTPHeloTimeout**
- **SMTPMailFromTimeout**
- **SMTPRcptTimeout**
- **SMTPDataTimeout**
- **SMTPDataBlockTimeout**
- **SMTPDataPeriodTimeout**
- **MaxMessageSize.** Maximum size of a message in bytes.
- **MaxRecipientsPerMessage.** Maximum number of recipients.
- **MinFreeBlocks.** Number of free file system blocks. If the limit is reached, no more incoming email is accepted. (0=disable feature)
- **MaxForwarders.** Maximum number of forward processes of `avgatefwd`. The number depends on the quality of the network connection (low network quality > higher value).
- **BlockSuspiciousMime.** If set to **YES**, suspicious MIME email will be blocked.
- **BlockExtensions.** Filename extensions that should be blocked (separated by a semicolon).

- **ExposeRecipientsAlerts, ExposeSenderAlerts, ExposePostmasterAlerts.** Specifies if alerts will be sent to the recipient, sender, or postmaster. The possible values are
 - **NO.** No alerts will be sent.
 - **LOCAL** (not available for **ExposePostmasterAlerts**). Alerts will be sent if the recipient/sender is a local user.
 - **YES.** Alerts will be sent.
- **AlertsUser.** Name of sender of alerts. (Syntax: *username* or *username@domain*)
- **ListenAddress.** Interface and port the SMTP daemon listens on. The default interface 0.0.0.0 means all interfaces.
Syntax: **ListenAddress interface port port**
- **ForwardTo.** Type of mail forwarding.
 - By piping: **ForwardTo path_to_sendmail**
 - By SMTP: **ForwardTo SMTP: host port port**Default: **ForwardTo /usr/sbin/sendmail -oem -oi**
- **RefuseEmptyMailFrom.** If set to **YES**, mails containing a blank sender address will be blocked.
- **PollPeriod.** Interval (in seconds) of queue scanning done by *avgatefwd*.
- **QueueLifetime.** Maximum time a message can stay in the queue before it will be bounced. (0 = disable feature)
- **ForwarderRetryDelay.** Maximum time between retrying to send a queued message.
- **ThrottleMessageCount.** Number of messages that will be reprocessed in a given time. (Only needed for large queues.)
After reprocessing, the *avgatefwd* will sleep for **ThrottleDelay** seconds.
- **BounceMessageUser.** Name or mail address of the sender of error messages (e.g., if an email could not be delivered).

Scanning of Files in an Archive

- **ArchiveMaxRecursion.** Maximum of recursion depth of unpacking and scanning archives. (0 = unlimited depth)
- **ArchiveMaxSize.** Maximum file size (in bytes) of an archive that will be scanned. (0 = unlimited size)
- **ArchiveMaxRatio.** Maximum compression ratio of an archive that will be scanned.
- **BlockSuspiciousArchive.** If set to **YES**, email that reach the limits **ArchiveMaxRecursion**, **ArchiveMaxSize** or **ArchiveMaxRatio** are blocked.
- **BlockEncryptedArchive.** If set to **YES**, email with encrypted archives are blocked.
- **BlockUnsupportedArchive.** If set to **YES**, email with archives that cannot be scanned are blocked.
- **BlockOnError.** If set to **YES**, mail that cannot be scanned due to scan timeout or process error are blocked.

Handling Envelope Recipient Addresses

Source routing is a technique whereby the sender of a packet can specify the route that a packet should take through the network.

The following options concern the source routing used by avgated. For a detailed description, read the AVMailGate documentation.

- **AllowSourceRouting**
- **InEnvelopeAddressesBangIs**
- **InEnvelopeAddressesPercentIs**
- **AcceptLooseDomainName**

Adding a Notification in the Body of Transmitted Mails

- **AddStatusInBody.** If set to **YES**, a default status text is inserted in the email's body.

If you want to insert your own text, you can specify a file name here. For no status text, enter **NO** (default).

- **MaxMessageSizeStatus.** Specify a message size up to where the status text is added to the message.

Syntax: **MaxMessageSizeStatus Xm | k | b**

- **ForwardAllEmailAsMIME.** If set to **YES**, all incoming non-MIME mails are converted to MIME.
- **AddPrecedenceHeader.** If set to **YES**, each notice mail is marked with "Precedence:" in the header. You also can enter a custom text.
- **AddressFilter.** If set to **YES**, each sender and/or recipient address will be matched against the tables `/etc/avmailgate.scan` and `/etc/avmailgate.ignore`.

The order in which the tables are scanned can be specified in **FilterTableOrder**. (first hit matches)

- **UseProxy.** You can optimize the scans by using the proxy feature in an AVMailGate pool.

The number of anti-virus scanners in the pool can be specified with the **ProxyScanners** option.

ProxyConnections specifies the number of simultaneous allowed connections.

- **AddHeaderToNotice.** If set to **YES**, a mail header is added to postmaster notice mails.
- **BounceMessageSizeBody, BounceMessageSizeHeader.** Limit the size (in bytes) of bounced mails.
- **AddXHeaderInfo.** If set to **YES**, the information about the scanning status is added to the header of the checked mail.

- **AddReceivedByHeader.** If set to **YES**, a “Received:” is added to the mail’s header.
- **MaxHopCount.** If there are more than **MaxHopCount** "Received:" lines in the header, the mail will not be accepted.
Prevent mail loops.

Other Configuration Files

There are four more configuration files for AVMailGate. These files contain a couple of regular expressions. We will only give a short overview here:

- **/etc/avmailgate.acl.** Defines which hosts are considered local and for which the server is allowed to relay emails.
- **/etc/avmailgate.ignore.** Defines mail addresses that should not be scanned.
- **/etc/avmailgate.scan.** Defines mail addresses that are always scanned.
- **/etc/avmailgate.warn.** In this file one can specify who receives a mail in case of an alert.

Update AVMailGate

The file that includes the virus signatures is
`/usr/lib/AntiVir/antivir[0123].vdf`.

To update these files, enter

```
/usr/lib/AntiVir/antivir --update
```

The output looks like the following:

```
AntiVir / Linux Version 2.1.5-24 +gui
Copyright (c) 1994-2005 by H+BEDV Datentechnik GmbH.
All rights reserved.

Warning: the file "antivir.vdf" is more than 14 days old
checking for updates

06.32.00.60 = 06.32.00.60 [vdf database (part 0), on-disk]
06.32.18.16 < 06.34.00.105 [vdf database (part 1), on-disk]
06.32.18.17 < 06.34.00.106 [vdf database (part 2), on-disk]
06.33.00.07 < 06.34.00.124 [vdf database (part 3), on-disk]
06.33.00.11 < 06.34.00.14 [scan engine, running]
06.33.00.11 < 06.34.00.14 [scan engine, on-disk]
antivir1.vdf 100% |*****| 1630 KB 1.59 MB/s 0:00
ETA
antivir2.vdf 100% |*****| 1 KB 0.00 KB/s --:--
ETA
antivir3.vdf 100% |*****| 35 KB 0.00 KB/s --:--
ETA
antivir 100% |*****| 695 KB 0.00 KB/s --:--
ETA
06.32.00.60 = 06.32.00.60 [vdf database (part 0), on-disk]
06.34.00.105 = 06.34.00.105 [vdf database (part 1), on-disk]
06.34.00.106 = 06.34.00.106 [vdf database (part 2), on-disk]
06.34.00.124 = 06.34.00.124 [vdf database (part 3), on-disk]
06.34.00.14 = 06.34.00.14 [scan engine, on-disk]

scan engine 06.33.00.11 --> 06.34.00.14 (/usr/lib/AntiVir/antivir)
vdf database 06.33.00.07 --> 06.34.00.124 (/usr/lib/AntiVir/antivir1.vdf,
/usr/lib/AntiVir/antivir2.vdf, /usr/lib/AntiVir/antivir3.vdf)

AntiVir updated successfully
```

If you only want to check whether new updates are available without updating the files, enter

`/usr/lib/AntiVir/antivir --update --check`

The output looks like the following:

```
AntiVir / Linux Version 2.1.5-24 +gui
Copyright (c) 1994-2005 by H+BEDV Datentechnik GmbH.
All rights reserved.

checking for updates

06.32.00.60 = 06.32.00.60 [vdf database (part 0), on-disk]
06.32.18.16 < 06.34.00.105 [vdf database (part 1), on-disk]
06.32.18.17 < 06.34.00.106 [vdf database (part 2), on-disk]
06.33.00.07 < 06.34.00.124 [vdf database (part 3), on-disk]
06.33.00.11 < 06.34.00.14 [scan engine, running]
06.33.00.11 < 06.34.00.14 [scan engine, on-disk]

an update for the scan engine is available (/usr/lib/AntiVir/antivir)
an update for the VDF database is available
(/usr/lib/AntiVir/antivir1.vdf, /usr/lib/AntiVir/antivir2.vdf,
/usr/lib/AntiVir/antivir3.vdf)
```

Exercise 12-1 Use AVMailGate as a Virus Scanner for Email

In this exercise, you install and configure AVMailGate as a virus scanner for mails. Finally, you update the AVMailGate virus signatures.

Do the following:

- Part I - Install AVMailGate
- Part II - Configure Postfix to Use AVMailGate as Content Filter
- Part III - Configure the Ports for AVMailGate to Use
- Part IV - Check Configuration Using a Virus File from CD
- Part V - Update Your Virus Signatures

Part I - Install AVMailGate

1. From the main menu, start **YaST**.
2. Enter the root password (**novell**) and select **OK**.
3. From the YaST Control Center, select **Software > Software Management**.
4. From the filter drop-down menu, select **Search**.
5. In the Search field, enter **avmailgate**; then select **Search**.
6. On the right, select the **avmailgate** package.
7. Select **Accept**; then insert the *SUSE Linux Enterprise Server 10* DVD.
8. Select **Continue** to resolve dependencies.
9. When installation is complete, remove the DVD and close the YaST Control Center.

Part II - Configure Postfix to Use AVMailGate as Content Filter

1. Open the file `/etc/postfix/master.cf` in a text editor.
2. Uncomment the following line (on one line):
localhost:10025 inet n - n - - smtpd -o content_filter=
3. Add the following line in `/etc/postfix/main.cf`:
content_filter = smtp:127.0.0.1:10024
4. Save the file.
5. Enter **postfix reload**.

Part III - Configure the Ports for AVMailGate to Use

1. To ensure that AVMailGate listens on port 10024 and not on port 25, edit `/etc/avmailgate.conf`:
ListenAddress 127.0.0.1 port 10024
2. To ensure that AvMailGate sends mails back to Postfix via SMTP on host localhost via port 10025, edit `/etc/avmailgate.conf`:
ForwardTo SMTP: localhost port 10025
3. Because AvMailGate sends out notification messages as AvMailGate, set an alias in `/etc/aliases`:
avmailgate: root
4. Enter **newaliases**.
5. Enter **rcavgate start**.

Part IV - Check Configuration Using a Virus File from CD

1. Log in as user `geeko`.
2. Send an infected mail to user `root` by entering
**mail root -s "Virus Test" -a
/media/cdrecorder/section_4/sample-virus-executable.txt**

3. Enter some text for the email message
This is an infected mail.
.
4. Log in as user root.
5. Check whether the mail queue is empty by entering
mailq
6. Check whether the infected mail arrived by entering
mail
7. Check whether the infected mail was detected by entering
ls /var/spool/avmailgate/rejected

Part V - Update Your Virus Signatures

1. To check for a new version of the virus signatures, enter
/usr/lib/AntiVir/antivir --update --check
2. To download the virus signatures, enter
/usr/lib/AntiVir/antivir --update

(End of Exercise)

AMaViSd-new

AMaViSd-new (*A Mail Virus Scanner*) consists of the daemon and some optional helper programs, which are only needed during setup between the message transfer agent (MTA) and one or more content checkers (virus scanners or SpamAssassin).

The mail server sends all incoming and outgoing mails to AMaViSd. The email will be extracted and tested by AMaViSd-new.

AMaViSd recognizes four kinds of unwanted mails. They are mails that have:

1. Invalid headers
2. Banned file types
3. Viruses
4. Spam

AMaViSd tests incoming mails for these four types in the order listed.

If nothing bad is found, AMaViSd-new will send the email back to the mail server, ready for delivery.

By default, AMaViSd-new listens at port 10024 for incoming email from the mail server.

AMaViSd-new sends clean mails to the mail server via port 10025. It is normally positioned at or near a central mail server, not necessarily where users' mailboxes and final delivery takes place.

This topic describes the following:

- Install AMaViSd-new
- Configure AMaViSd-new

Install AMaViSd-new

To install AMaViSd-new, select the package **amavisd-new**.



AMaViSd-new will not work if no virus scanner is installed on your system. If you want to use AMaViSd-new only for spam filtering, search for **@bypass_virus_checks_acl** in `/etc/amavisd.conf` and remove the comment sign (“#”) at the beginning of the line.

The compressing tools `gzip`, `bzip2`, `arc`, `lha`, `unrar`, `zoo`, `cpio`, and `lzop` should be installed, too.

During the SUSE Linux Enterprise Server 10 installation, a user `vscan` is created. It is used for AMaViSd.

```
da51:~ # grep vscan /etc/passwd
vscan:x:65:104:Vscan account:/var/spool/amavis:/bin/false
```

After installing AMaViSd-new, you can start the daemon with **rcamavis start**.

The daemon should listen on port 10024.

```
da51:~ # telnet 127.0.0.1 10024
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
```

Using SMTP commands you can write an email now.

```
da51:~ # telnet 127.0.0.1 10024
Trying 127.0.0.1...
Connected to 127.0.0.1.
Escape character is '^]'.
220 [127.0.0.1] ESMTP amavisd-new service ready
mail from: jgoldman@digitalairlines.com
250 2.1.0 Sender jgoldman@digitalairlines.com OK
rcpt to: geeko@digitalairlines.com
250 2.1.5 Recipient geeko@digitalairlines.com OK
data
354 End data with <CR><LF>.<CR><LF>
Subject: Test

Test
.
250 2.6.0 Ok, id=09232-01, from MTA([127.0.0.1]:10025): 250
Ok: queued as 5747B16A68
quit
221 2.0.0 [127.0.0.1] amavisd-new closing transmission
channel
Connection closed by foreign host.
da51:~ #
```

The email header should contain a line for amavisd.

```
geeko@da51:~> mail
mailx version nail 11.25 7/29/05.  Type ? for help.
"/var/spool/mail/geeko": 1 message 1 new
>N 1 jgoldman@digitalai Tue May 16 15:30 20/829 Test
? 1
Message 1:
From jgoldman@digitalairlines.com Tue May 16 15:30:58 2006
X-Original-To: geeko@digitalairlines.com
Delivered-To: geeko@digitalairlines.com
Subject: Test
X-Virus-Scanned: amavisd-new at example.com
Date: Tue, 16 May 2006 15:30:58 -0400 (EDT)
From: jgoldman@digitalairlines.com
To: undisclosed-recipients:;

Test

?
```

Configure AMaViSd-new

On SUSE Linux Enterprise Server 10 you can configure AMaViSd-new by editing one of the following files:

- `/etc/sysconfig/amavis`
- `/etc/amavisd.conf`

`/etc/sysconfig/amavis`

The configuration file `/etc/sysconfig/amavis` is only available on SUSE Linux products.

In this file there are only two parameters:

- **USE_AMAVIS**. If set to **yes**, Sendmail or Postfix are prepared to use AMaViSd-new.

- **AMAVIS_SENDMAIL_MILTER.** If set to **yes**, the milter (*Mail Filter*) interface of Sendmail will be started.

Using the milter interface, it is possible to connect mail filter applications to Sendmail in a standardized form.

After changing the file `/etc/sysconfig/amavis`, you have to run the command **SuSEconfig**.

Setting `USE_AMAVIS` to `yes` makes three changes in `/etc/postfix/master.cf`:

- The `smtp` protocol

```
smtp      inet  n       -       n       -       2       smtpd
  -o content_filter=smtp:[127.0.0.1]:10024
```

- The `smtps` protocol (still marked as comment)

```
#smtps    inet  n       -       n       -       2       smtpd
  -o smtpd_tls_wrappermode=yes -o content_filter=smtp:[127.0.0.1]:10024
```

- Port 10025

```
localhost:10025 inet  n       -       n       -       -       smtpd
  -o content_filter=
```

In this file you also have to add a process for `amavis`:

```
smtp-amavis unix  -       -       n       -       2       smtp
  -o smtp_data_done_timeout=1200
  -o smtp_send_xforward_command=yes
  -o disable_dns_lookups=yes
  -o max_use=20
```

In the Postfix configuration file `/etc/postfix/main.cf` you have to add the following line:

```
content_filter = smtp-amavis:[127.0.0.1]:10024
```

Restart Postfix by entering **`rpostfix reload`**.

`/etc/amavisd.conf`

AMaViSd is configured by editing the file `/etc/amavisd.conf`. The syntax of this configuration file is plain Perl code. Because of this, each line ends in a semicolon.

In Perl strings in double quotation marks can include variables that start with “\$” or “@”. To include characters “@” and “\$” in double-quoted strings, they must be preceded by a backslash.

In single-quoted strings, the “\$” and “@” lose their special meaning, so it is usually easier to use single quoted strings.

In both cases, the backslash needs to be doubled.

In the documentation directory of AMaViSd-new (`/usr/share/doc/packages/amavisd-new/`), two more variants of the configuration file are available:

- **`amavisd.conf-default`**. Includes all possible parameters with their defaults.
- **`amavisd.conf-sample`**. A more structured file with a lot of explanations and examples.

In this section we want to discuss some of the most important parameters in order of occurrence in the `/etc/amavisd.conf` file.

- **`@bypass_virus_checks_maps`**. This array includes a list of virus lookup tables. You can disable virus checking by setting this array to “1” (uncomment the line).

- **@bypass_spam_checks_maps.** This array includes a list of spam lookup tables. You can disable spam checking by setting this array to “1” (uncomment the line).



Next to @bypass_virus_checks_maps and

@bypass_spam_checks_maps, there are two more arrays of the same kind available:

@bypass_banned_checks_maps enables checking for banned names or file types.

@bypass_header_checks_maps enables checking for invalid headers.

- **\$max_servers.** Number of pre-forked children.

Should match the number of your MTA pipe, e.g., the **maxproc** field in /etc/postfix/master.cf.

```
#
=====
# service type private unpriv chroot wakeup maxproc command + args
#          (yes)   (yes)   (yes)   (never) (100)
#
=====
smtp      inet  n       -       n       -       2       smtpd -o
content_filter=smtp:[127.0.0.1]:10024
```

- **\$daemon_user** and **\$daemon_group.** User and group to which the daemon will change.
- **\$mydomain.** The domain name.
- **\$MYHOME.** Location where the AMaViSd-new files are stored.
- **\$TEMPBASE.** The path of a temporary directory that can be used by AMaViSd-new. This directory must exist or needs to be created manually.

In the following line an environment variable **\$TMPDIR** is defined with the content of the **\$TEMPBASE** variable.

- **\$QUARANTINEDIR.** The path to where infected mails can be put. This can be
 - **a file.** Path does not end with backslash.
 - **a directory.** Path ends with backslash.
 - **disabled.** Leave it empty.
- **\$quarantine_subdir_levels.** If set to “1” subdirectories in \$QUARANTINEDIR are created to disperse quarantine.
- **\$daemon_chroot_dir.** Run the daemon in the specified chroot jail. If you do not want chroot, leave it empty.
- **\$db_home.** Path of the databases. Default: \$MYHOME/db.
- **\$helpers_home.** Sets the environment variable \$HOME. The value is passed to other SpamAssassin modules.
- **\$pid_file.** Path of the PID file.
- **\$lock_file.** Path of the lock file.
- **@local_domains_maps.** List of lookup tables that can be used to decide whether a recipient is local or not, i.e., if the message is outgoing or not.
- **@mynetworks.** List of IP ranges which determines if the original SMTP client IP address belongs to the internal networks, i.e. if mail is coming from inside.
- **\$log_level.** Log level.
 - **0.** Startup/exit/failure messages, viruses detected
 - **1.** Args passed from client, some more interesting messages
 - **2.** Virus scanner output, timing
 - **3.** Server, client
 - **4.** Decompose parts
 - **5.** More debug details

- **\$log_recip_tmpl.** Template for log file entries.
A list of the available macros can be found in `/usr/share/doc/packages/amavisd-new/README_FILES/README.customize`.
- **\$DO_SYSLOG.** If set to “1”, the syslog daemon is used for loggings.
- **\$SYSLOG_LEVEL.** Level of syslog loggings.
- **\$enable_db.** Enable use of BerkeleyDB/libdb. If it is enabled, you can also enable the libdb cache using **\$enable_global_cache**.
- **\$inet_socket_port.** Port number that the AMaViSd-new should listen to.
- **\$unix_socketname.** If you are using Sendmail milter, you have to enter the socket name here.
- **\$sa_tag_level_deflt.** Spam info headers are added to the mail if the spam level is at or above the given number.
- **\$sa_tag2_level_deflt.** Spam detected headers are added to the mail if the spam level is at or above the given number.
- **\$sa_kill_level_deflt.** Spam evasive actions (bounce/reject/drop) are triggered if the spam level is at or above the given number.
- **\$sa_dsn_cutoff_level.** Spam with a spam level beyond this number is not sent.
- **\$sa_quarantine_cutoff_level.** Spam with a spam level beyond this number is not quarantined.
- **\$sa_mail_body_size_limit.** Email messages larger than the given number is not passed to SpamAssassin. (Less than 1% of spam is larger than 64 KB.)
- **\$sa_local_tests_only.** If set to “1”, no SpamAssassin tests requiring Internet access are performed.

- **\$sa_auto_whitelist.** If set to “1”, AWL (auto-whitelist) in SpamAssassin 2.63 or older is turned on (irrelevant for SpamAssassin 3.0; on SUSE Linux Enterprise Server 10, version 3.1.0 is available)



AWL tracks scores for your regular correspondents in a small on-disk database. Since version 3.0, it is enabled by default.

- **@lookup_sql_dsn.** Array with information about where to find SQL server(s) and database to support SQL lookups. One item includes a triple of data: source name, user, and password.
- **\$virus_admin.** Fully qualified address of the antivirus administrator.
- **\$mailfrom_notify_admin.** Fully qualified address of the sender of admin notifications.
- **\$mailfrom_notify_recip.** Fully qualified address of the sender of virus notifications.
- **\$mailfrom_notify_spamadmin.** Fully qualified address of the sender of spam notifications.
- **\$mailfrom_to_quarantine.** Whom quarantined messages appear to be sent from. If undefined, the original sender is used.
- **@addr_extension_virus_maps.** The specified string is added to the recipient’s address if a virus is detected.

The **@addr_extension_spam_maps**,
@addr_extension_banned_maps, and
@addr_extension_bad_header_maps strings work in the same way.

The string is separated from the recipient address by the string specified in **\$recipient_delimiter**.

Example:

geeko@digitalairlines.com > geeko+spam@digitalairlines.com

- **\$path**. The content of this variable is passed to the PATH environment variable.
- **\$dspam**. Activate the dspam content filter (<http://www.nuclearelephant.com/projects/dspam/>).
- **\$MAXLEVELS**. Maximum recursion level for extraction and decoding.
- **\$MAXFILES**. Maximum number of extracted files.
- **\$MIN_EXPANSION_QUOTA**,
\$MAX_EXPANSION_QUOTA. Minimum and maximum storage size (in bytes) that is available for mail extraction.
- **\$sa_spam_subject_tag**. String that is prepended to the subject header when message exceeds **\$sa_tag2_level_deflt** level.
- **@*_lovers_maps**. Email to the specified recipients is not examined and filtered.
- **@blacklist_sender_maps**. A message from a blacklisted envelope sender address is marked as spam.

A **@whitelist_sender_maps** is also available. Mail with sender addresses from this array are delivered although they are marked as spam.
- **@score_sender_maps**. Using this variable you can add or subtract a specified value to/from the spam value.

The next variables (beginning with **@viruses_that_fake_sender_maps**) contain regular expressions to filter mails.

Many regular expressions are predefined and you can enable them by removing the comment hash sign at the beginning of the line(s).

Of course you can modify the regular expressions if they do not fit your needs.

Some other important variables are:

- **\$final_virus_destiny**, **\$final_banned_destiny**, **\$final_spam_destiny**, **\$final_bad_header_destiny**. Defines what to do with email that has a virus, a banned sender, spam content, or incorrect headers.

You can use the following values:

- **D_PASS**. Mail will pass to recipients, regardless of bad contents.
- **D_DISCARD**. Mail will not be delivered to its recipients; sender will not be notified.
- **D_BOUNCE**. Mail will not be delivered to its recipients; a nondelivery notification (bounce) will be sent to the sender by AMaViSd-new.
- **D_REJECT**. Mail will not be delivered to its recipients; the sender should preferably get a rejection notification, (SMTP permanent reject response or nondelivery notification from the MTA).

If this is not possible, AMaViSd-new sends a bounce by itself (the same as **D_BOUNCE**).

- **\$notify_method**. Specify a host and port where the notifications are sent to.
- **\$notify_sender_tmpl**. Add this variable if you do not want the sender of an email notified.
- **\$notify_virus_sender_tmpl**. Specify a text file if you do not want the default notification sent to the sender of an email that contained a virus.
- **\$notify_virus_admin_tmpl**. Specify a text file if you do not want the administrator notified when a virus is detected.

- **\$notify_virus_recips_tmpl.** Specify a text file if you do not want to notify the recipients of an email that contained a virus.
- **\$notify_spam_sender_tmpl.** Specify a text file if you do not want to notify the sender of an email that contained spam.
- **\$notify_spam_admin_tmpl.** Specify a text file if you do not want to notify the administrator if a spam email is detected.

Exercise 12-2 Use AMaViSd as Virus Scanner for Email

In this exercise, you install and configure AMaViSd. Virus notifications should be sent to user root. You test your configuration by using telnet and by sending a test virus file by mail.

Do the following:

- Part I - Install AMaViSd
- Part II - Change /etc/sysconfig/amavis
- Part III - Change /etc/amavisd.conf
- Part IV - Test the Configuration
- Part V - Send a Virus Email

Part I - Install AMaViSd

1. From the main menu, start **YaST**.
2. Enter the root password (**novell**) and select **OK**.
3. From the YaST Control Center, select **Software > Software Management**.
4. From the filter drop-down menu, select **Search**.
5. In the Search field, enter **amavis**; then select **Search**.
6. On the right, select the **amavisd-new** package.
7. Select **Accept**; then insert the *SUSE Linux Enterprise Server 10* DVD.
8. Select **Continue** to resolve dependencies.
9. When installation is complete, remove the DVD and close the YaST Control Center.

Part II - Change /etc/sysconfig/amavis

1. Open the file /etc/sysconfig/amavis by entering
vi /etc/sysconfig/amavis
2. Change the line with the variable USE_AMAVIS to
USE_AMAVIS="yes"
3. Exit vi by entering **:wq**.
4. Enter **SuSEconfig**.
5. Look at the messages of the output. If the file /etc/postfix/master.cf is left untouched, overwrite this file by entering
mv /etc/postfix/master.cf.SuSEconfig /etc/postfix/master.cf
6. Open the file /etc/postfix/master.cf by entering
vi /etc/postfix/master.cf
7. Add the following lines to the file /etc/postfix/master.cf:
smtp-amavis unix - - n - 2 smtp
-o smtp_data_done_timeout=1200
-o smtp_send_xforward_command=yes
-o disable_dns_lookups=yes
-o max_use=20
8. Exit vi by entering **:wq**.
9. Open the file /etc/postfix/main.cf by entering
vi /etc/postfix/main.cf
10. To remove Procmail from the mailbox_command (entered in a previous exercise), enter:
mailbox_command =
11. Add the following line to /etc/postfix/main.cf:
content_filter = smtp-amavis:[127.0.0.1]:10024
12. Exit vi by entering **:wq**.
13. Restart Postfix by entering **rcpostfix reload**.

Part III - Change /etc/amavisd.conf

1. Open the file /etc/amavis.conf by entering
vi /etc/amavis.conf
2. Modify the \$mydomain variable to
\$mydomain = 'digitalairlines.com';
3. Change the mail address where virus notifications should be sent to root:
\$virus_admin = "root\@\$mydomain";
4. Exit vi by entering **:wq**.
5. Start the AMaViSd by entering **rcamavis start**.

Part IV - Test the Configuration

1. Check whether Postfix listens on port 10025 by entering
telnet 127.0.0.1 10025
2. Enter **quit**.
3. Check whether the AMaViSd listens on port 10024 by entering
telnet 127.0.0.1 10024
4. Enter **mail from: jgoldman@digitalairlines.com**
5. Enter **rcpt to: geeko@digitalairlines.com**
6. Enter **data**
7. Open the file
/usr/share/doc/packages/amavisd-new/test-messages/
sample-virus-simple.txt and copy the last line into the clipboard.
The line looks like this:

```
X5O!P% @AP[4PZX54(P^)7CC)7]$EICAR-STANDARD-  
ANTIVIRUS-TEST-FILE!$H+H*
```

8. Paste the content of the clipboard into the first terminal and do the following:
 - a. Press **Enter**.
 - b. Type **.** (dot).
 - c. Press **Enter**.
9. You should get a virus warning like the following one:
**250 2.7.1 Ok, discarded, id=14069-01-2 - VIRUS:
Eicar-Test-Signature**
10. Enter **quit**.

Part V - Send a Virus Email

1. Log in as user **jgoldman** by entering **su - jgoldman**.
2. Send a virus mail to user **tux** by entering
**mail geeko@digitalairlines.com <
/usr/share/doc/packages/amavisd-new/test-messages/sample
-virus-simple.txt**
3. Log out by entering **exit**.
4. As root, enter **mail** to look for new email.

There should be an email from **virusalert** in your mail folder.

(End of Exercise)

Summary

| Objective | Summary |
|----------------------------------|---|
| 1. Understand SMTP Communication | <p>The following SMTP commands must be implemented in an SMTP server to provide SMTP communication:</p> <ul style="list-style-type: none">■ HELO■ MAIL FROM:■ RCPT TO:■ DATA■ RSET■ NOOP■ QUIT |
| 2. Manage Spam | <p>spamassassin (and spamc) expect their input from STDIN.</p> <p>If you want to use SpamAssassin with your Postfix configuration, the easiest way is to use Procmail.</p> <p>Spam can be filtered from the mail client or in a further Procmail configuration (<code>~/procmailrc</code>).</p> <p>You can use telnet to test your configuration. You also can send a spam email directly using the sendmail command.</p> |

| Objective | Summary |
|---|---|
| 3. Use a Virus Scanner for Email | <p data-bbox="814 245 1197 337">AVMailGate is an antivirus email filter from H+BEDV Datentechnik GmbH.</p> <p data-bbox="814 354 1197 412">AVMailGate can update the virus definition file and the engine itself.</p> <p data-bbox="814 428 1197 487">AVMailGate is composed of two kinds of processes:</p> <ul data-bbox="814 503 1197 678" style="list-style-type: none"><li data-bbox="814 503 1197 578">■ avgated. The smtpd receiver that stores incoming email in the input queue.<li data-bbox="814 594 1197 678">■ avgatefwd. Virus scanning function and SMTP forwarder and sendmail invoker. <p data-bbox="814 695 1197 781">Both processes can be configured by editing the file /etc/avmailgate.conf.</p> <p data-bbox="814 797 1197 826">AMaViSd-new consists of</p> <ul data-bbox="814 842 1197 1083" style="list-style-type: none"><li data-bbox="814 842 1197 872">■ The daemon.<li data-bbox="814 888 1197 980">■ Optional helper programs that are only needed in setup between the message transfer agent (MTA).<li data-bbox="814 997 1197 1083">■ One or more content checkers: virus scanners such as SpamAssassin. <p data-bbox="814 1099 1197 1185">The email server sends all incoming and outgoing email to AMaViSd.</p> <p data-bbox="814 1201 1197 1287">The emails will be extracted by AMaViSd-new and tested with a virus scanner.</p> |

| Objective | Summary |
|--|--|
| 3. Use a Virus Scanner for Email (continued) | If no viruses are found, AMaViSd-new sends the email back to the mail server ready for delivery. AMaViSd-new is configured in the file /etc/amavisd.conf . |
