Chapter 1. Integrating amavisd-new in Postfix

2. Basic Postfix and amavisd-new configuration

There are several moments at which Postfix can hand over messages over to amavisd-new (before it accepts a message from a client or after) and there are different filter approaches (globally, per recipient (domain), per network interface, etc.) that can trigger Postfix to transport a message to amavisd-new.

The transport methods - transporting a message from Postfix to amavisd-new and backwards - however always remain the same. They will be described in this section first. The section that follows will deal with different filter approaches.

Integration procedure

The following examples have been structured to cause minimum trouble on an online mail system. The order of steps ensures that filtering will be enabled at the very last moment. Several tests will have been conducted to verify the delivery chain works before the filter is enabled. Once enabled the complete system should work at once.

2.1. Configuring amavisd-new for Postfix

Configuring amavisd-new to work with Postfix answers the following two questions:

1. Which port should the amavisd-new daemon listen to for incoming connections from Postfix?
2. Which IP-address and port should the amavisd-new SMTP client use to (re)inject filtered messages (and notifications about message statuses) into the Postfix SMTP delivery system?

2.1.1. Configuring amavisd-new for incoming connections

The *$inet_socket_port* in `/etc/amavisd.conf` parameter sets the port number amavisd-new will listen for incoming (E)SMTP connections. The following example explicitly configures amavisd-new to bind to port `10024` (default setting `undef`):

```
$inet_socket_port = 10024;
```

2.1.2. Configuring the reinjection path

Two parameters, *$forward_method* and *$notify_method*, need to be configured (usually identically) to reinject messages into the Postfix mail system.

The first parameter, *$forward_method*, specifies where amavisd-new should transport scanned messages to, while the second parameter, *$notify_method*, specifies where notifications about scanned messages should be transported to.

By default amavisd uses `127.0.0.1` on port `10025` to contact a SMTP server for reinjection of filtered messages. Unless a different IP address or port should be used, no modifications must be applied and this section can be skipped.

In case a different IP address or port should be used, the parameters *$notify_method* and *$forward_method* need to be adjusted to reflect these requirements. The following example edits these parameters in `/etc/amavisd.conf` and uses `192.0.2.1` as IP address and port `20025`:

```
$notify_method  = 'smtp:[192.0.2.1]:20025';
$forward_method = 'smtp:[192.0.2.1]:20025';
```

2.2. Configuring the transport from Postfix to amavisd-new

Both, amavisd-new and Postfix, are able to use either SMTP- or LMTP-communication to transport a message from Postfix to amavisd-new. Both

variants will be described in this section.

Why configure a dedicated service?

Theoretically it's possible to transport messages from Postfix to amavisd-new using the existing smtp-, lmtp, or even the relay-service in `/etc/postfix/master.cf`.

In practice transporting messages to amavisd-new requires imposing transport limits on the transporting service. Imposing such limits on a globally available service would impose these limits on the complete Postfix mail system - it would slow down the system significantly and should be avoided.

Note

The number of Postfix clients that may connect simultaneously to amavisd-new instances must be limited to the maximum number of daemon child processes amavisd-new starts.

If the Postfix transport client was allowed to open more connections amavisd-new can handle, amavisd-new would start to queue incoming Postfix connections. Postfix in turn would interpret such behaviour as “unresponsive remote MTA” and would itself begin to queue mail that should be filtered. All this would possibly throttle down the complete system and all further filtering attempts would suffer.

2.2.1. Configuring a dedicated lmtp-client

The following example creates a new, dedicated lmtp-transport named `amavisfeed` in `/etc/postfix/master.cf`. Its configuration details are explained following the listing:

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================

...

amavisfeed unix    -       -       n        -      2     lmtp
    -o lmtp_data_done_timeout=1200
    -o lmtp_send_xforward_command=yes
    -o disable_dns_lookups=yes
    -o max_use=20
```

Important

A noteworthy quote from the Postfix documentation: “...do not specify whitespace around the ‘’. In parameter values, either avoid whitespace altogether, ...”. Further details on `master.cf` configuration syntax can be found in `master.cf` or master(5).

Here's a quick rundown on the settings that differ from other services defaults:

`maxproc`
The maximum number of concurrent Postfix amavis-service processes has been limited to `2` (default: *`default_process_limit`* = `100`). This value reflects the default of `2` amavisd-daemon children processes and is a good setting to start from. The value may be raised later, when the system works stable and still can take a higher load. It should not exceed the number of simultaneous amavisd child processes.
`lmtp_data_done_timeout`
Setting *`lmtp_data_done_timeout`* to `1200` (seconds) doubles the default time span a regular Postfix client waits after message delivery for the server to reply `DONE` to claim successful delivery. It must be larger than amavisd setting *`$child_timeout`* (default `8*60` seconds) and should add a sufficient safety margin, for example to cater for periods of automatic database maintenance (e.g. bayes database on non-SQL database types) which can take a long time in some cases.

If the server does not reply within the configured time span, the Postfix client will quit the connection, put the message into the deferred queue, log a delivery failure and retry later to transport the message to amavisd-new.

Note

Raising this value serves a trick amavisd uses to avoid message loss in case of power outage etc. The trick consists in keeping the incoming

connection as long open as it takes to filter the message and take appropriate action (reinjection, notification, quarantine, etc.).

Only when the message (or notifications etc.) has been reinjected amavisd will send `DONE` to the client and the client will close the connection. This way Postfix will always keep the message in its own mail queue, where it can be reactivated after a system failure.

`lmtp_send_xforward_command`

Enabling *`lmtp_send_xforward_command`* configures the Postfix lmtp-client to forward the original clients HELO name and IP address to amavisd-new. amavisd-new in turn can use these informations for

- logging and notifications (macro `%a`)
- switching policy banks (`MYNETS`, *`@mynetworks_maps`*)
- pen pals functionality
- p0f fingerprinting

`disable_dns_lookups`

The transport route from Postfix to amavisd-new, it will be configured later in Section 3, “Message filtering examples”, will probably never change. It will - probably - only change when the whole mail system is being reconfigured. The target host may therefore be specified as IP address instead of using a DNS hostname. This saves “expensive” DNS-request (3 lookups) and improves performance.

`max_use`

By default Postfix reuses a service instance 100 times (*`max_use`* = `100`), before the instance terminates. The master daemon will reinvoke such a service if required. There's no need for the amavisfeed-service to have such a long life-span. Best practice has it to set *`max_use`* to `20`.

2.2.2. Configuring a dedicated smtp-client

Configuring a dedicated smtp-client is almost identical to configuring a dedicated lmtp-client. The syntax differences in detail are that the names of parameters start with *`smtp_`* instead of *`lmtp_`* and that the command at the end of the service invokes the smtp- and not lmtp-client. The same reasons given for differing lmtp client options apply to the dedicated smtp client configuration.

Here's an example of a dedicated smtp client given the service name *`amavisfeed`*:

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================

...

amavisfeed 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
```

2.3. Configuring a dedicated SMTP-server for message reinjection

The second service that needs to be added to the Postfix mail system is a dedicated SMTP-server. It will exist only to accept filtered messages and notifications from amavisd-new to transported them closer to their final destination.

This dedicated smtpd server will differ in many aspects from the default smtpd daemon. The most important difference is that it configures an empty *`content_filter`* parameter, thus overriding any global external content filtering settings in Postfix.

Note

Delegating messages to an external content filter in Postfix is done using the *`content_filter`* parameter. If the dedicated smtpd-daemon would not override any global *`content_filter`* settings, the reinjected message would be sent of to the external content filter again - the mail would end in an endless loop.

The following Postfix example uses amavisd-new default settings taken from the *`$forward_method`* and *`$notify_method`* parameters. These settings configure amavisd-new to forward filtered messages and notifications to `127.0.0.1` on port `10025`; the Postfix smtpd daemon will be configured to

bind to that IP address and listen on the specified port for incoming connections:

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================

...

127.0.0.1:10025 inet n    -       n       -       -     smtpd
    -o content_filter=
    -o smtpd_delay_reject=no
    -o smtpd_client_restrictions=permit_mynetworks,reject
    -o smtpd_helo_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject
    -o smtpd_data_restrictions=reject_unauth_pipelining
    -o smtpd_end_of_data_restrictions=
    -o smtpd_restriction_classes=
    -o mynetworks=127.0.0.0/8
    -o smtpd_error_sleep_time=0
    -o smtpd_soft_error_limit=1001
    -o smtpd_hard_error_limit=1000
    -o smtpd_client_connection_count_limit=0
    -o smtpd_client_connection_rate_limit=0
    -o receive_override_options=no_header_body_checks,no_unknown_recipient_checks,no_milters
    -o local_header_rewrite_clients=
```

Here's a quick rundown on the settings that differ from smtpd defaults:

content_filter
The empty *content_filter* overrides other, globally set *content_filter* delegations.
..._maps
Empty *..._maps* override any other globally set map lookups. Procedures to enforce settings specified in such maps have already taken place when Postfix accepted the message from the external client. Doing them again will not produce new results but only waste resources.
..._restrictions_...
There's no need to apply any already enforced *..._restrictions_...* another time. It would also only waste resources.
mynetworks
To avoid abuse from remote hosts, the dedicated smtpd-daemon will only allow clients from `127.0.0.0/8` to relay messages.
local_header_rewrite_clients
By default this option would “rewrite message header addresses in mail from these clients and update incomplete addresses with the domain name”. If such action has already been taken by Postfix before the message went off to amavis, it should not be done a second time when it reenters the Postfix mail system. Leaving this option empty disables local header rewrites and saves resources.
remaining options
All remaining options either configure the dedicated smtpd-daemon to be more failure tolerant or exist to avoid unnecessary use of resources.

Running the postfix reload will activate the new transports (Postfix will not yet send regular mail to amavisd). Combined with the tail command problems can easily be detected:

```
# postfix reload && tail -f /var/log/maillog
```

If there are no problems reported, basic configuration can be tested.

2.4. Testing basic configuration

Testing basic configuration consists of three separate tests, starting at the end of the new delivery chain and working to it's beginning. Their goal is to answer the following questions:

1. Will amavisd-new accept connections at the specified IP address and port?
2. Will the new dedicated smtpd-daemon accept connections at the specified IP address and port?
3. Will a test message, injected into amavisd-new, be filtered, sent to Postfix and delivered into a mailbox?

2.4.1. Testing amavisd's host and port

A test, using the telnet command, serves to verify that amavisd listens on the specified IP address and port. A successful connection looks like this:

```
$ telnet localhost 10024
220 [127.0.0.1] ESMTP amavisd-new service ready
EHLO localhost
250-[127.0.0.1]
250-VRFY
250-PIPELINING
250-SIZE
250-ENHANCEDSTATUSCODES
250-8BITMIME
250-DSN
250 XFORWARD NAME ADDR PROTO HELO
QUIT
221 2.0.0 [127.0.0.1] amavisd-new closing transmission channel
```

If the test fails, the following questions may help to debug the problem:

- Is the amavisd-new daemon running?
- Does amavisd-new write an error to the log?
- Do the IP address and port number specified in the amavisd-new configuration match the values used during the test?
- Does a firewall intercept connections?

2.4.2. Testing the dedicated Postfix smtpd-daemon

When Postfix was reloaded, the new, dedicated smtpd-daemon (127.0.0.1:10025) should have been activated. A successful connection looks like this:

```
$ telnet 127.0.0.1 10025
220 mail.example.com ESMTP Postfix (2.3.2)
EHLO localhost
250-mail.example.com
250-PIPELINING
250-SIZE 40960000
250-ETRN
250-STARTTLS
250-AUTH PLAIN CRAM-MD5 LOGIN DIGEST-MD5
250-AUTH=PLAIN CRAM-MD5 LOGIN DIGEST-MD5
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
QUIT
221 2.0.0 Bye
```

If the test fails, the following questions may help to debug the problem:

- Is the Postfix master daemon running?
- Does Postfix write an error to the log?
- Do the IP address and port number specified in the new services configuration match the values used during the test?
- Does a firewall intercept connections?

2.4.3. Testing the new transport chain

This test proves amavisd accepts e-mail as specified in Section 2.1, “Configuring amavisd-new for Postfix”, filters it and finally hands it over to Postfix' dedicated smtpd-daemon as specified in Section 2.3, “Configuring a dedicated SMTP-server for message reinjection”.

The following example uses the content of `test-messages/sample-nonspam.txt` from the amavisd test-messages to send an e-mail:

```
$ telnet localhost 10024
220 [127.0.0.1] ESMTP amavisd-new service ready
HELO localhost
250 [127.0.0.1]
MAIL FROM: <>
250 2.1.0 Sender  OK
RCPT TO: <postmaster>
250 2.1.5 Recipient postmaster OK
DATA
354 End data with <CR><LF>.<CR><LF>
From: virus-tester
To: undisclosed-recipients:;
Subject: amavisd test - simple - no spam test pattern

This is a simple test message from the amavisd-new test-messages.
.
250 2.6.0 Ok, id=30897-02, from MTA([127.0.0.1]:10025): 250 2.0.0 Ok: queued as 079474CE44
QUIT
221 2.0.0 [127.0.0.1] amavisd-new closing transmission channel
```

The maillog shows the delivery path. Here's an excerpt from a successful delivery process:

```
Nov  1 11:28:10 mail postfix/smtpd[30986]: connect from localhost[127.0.0.1] ❶
Nov  1 11:28:10 mail postfix/smtpd[30986]: 079474CE44: client=localhost[127.0.0.1]
Nov  1 11:28:10 mail postfix/cleanup[30980]: 079474CE44: message-id=<20061101102810.079474CE44@mail.example.com>
Nov  1 11:28:10 mail postfix/qmgr[20432]: 079474CE44: from=<>, size=822, nrcpt=1 (queue active)
Nov  1 11:28:10 mail amavis[30897]: (30897-02) Passed BAD-HEADER, <> -> <postmaster>, quarantine: badh-le5gjszxowBk, mail_id: le5gjszxowBk, Hits: -1.76, queue
Nov  1 11:28:10 mail postfix/smtpd[30986]: disconnect from localhost[127.0.0.1]
Nov  1 11:28:10 mail postfix/local[30987]: 079474CE44: to=<postmaster@example.com>, relay=local, delay=0.27, delays=0.14/0.05/0/0.08, dsn=2.0.0, status=sent
Nov  1 11:28:10 mail postfix/qmgr[20432]: 079474CE44: removed
```

amavisd connects with Postfix dedicated smtpd-daemon and hands over the e-mail that had been sent during the telnet session. smtpd gives a queue-id of 079474CE44 that can be tracked throughout the maillog.

amavisd notices it has checked and sent an e-mail to <postmaster>.

Postfix' local-service logs it successfully delivered an e-mail with queue-id 079474CE44 to the mailbox of postmaster.

If the test fails, the following questions may help to debug the problem:

- Does amavisd-new log errors?
- Does running amavisd-new in debug-mode report errors?