MailScanner
5.1.3-1

Milter Configuration Guide
1.4
November 23rd, 2018

Shawn Iverson
shawniverson@efa-project.org

Edition 1.1. This guide is based on the implementation of MailScanner as of Version 5.1.3.

I would like to gratefully acknowledge all the support and assistance provided by the following organizations:

This guide is dedicated to the continuing persistence and dedication of all people who fight bad email day in and day out worldwide.

Introduction

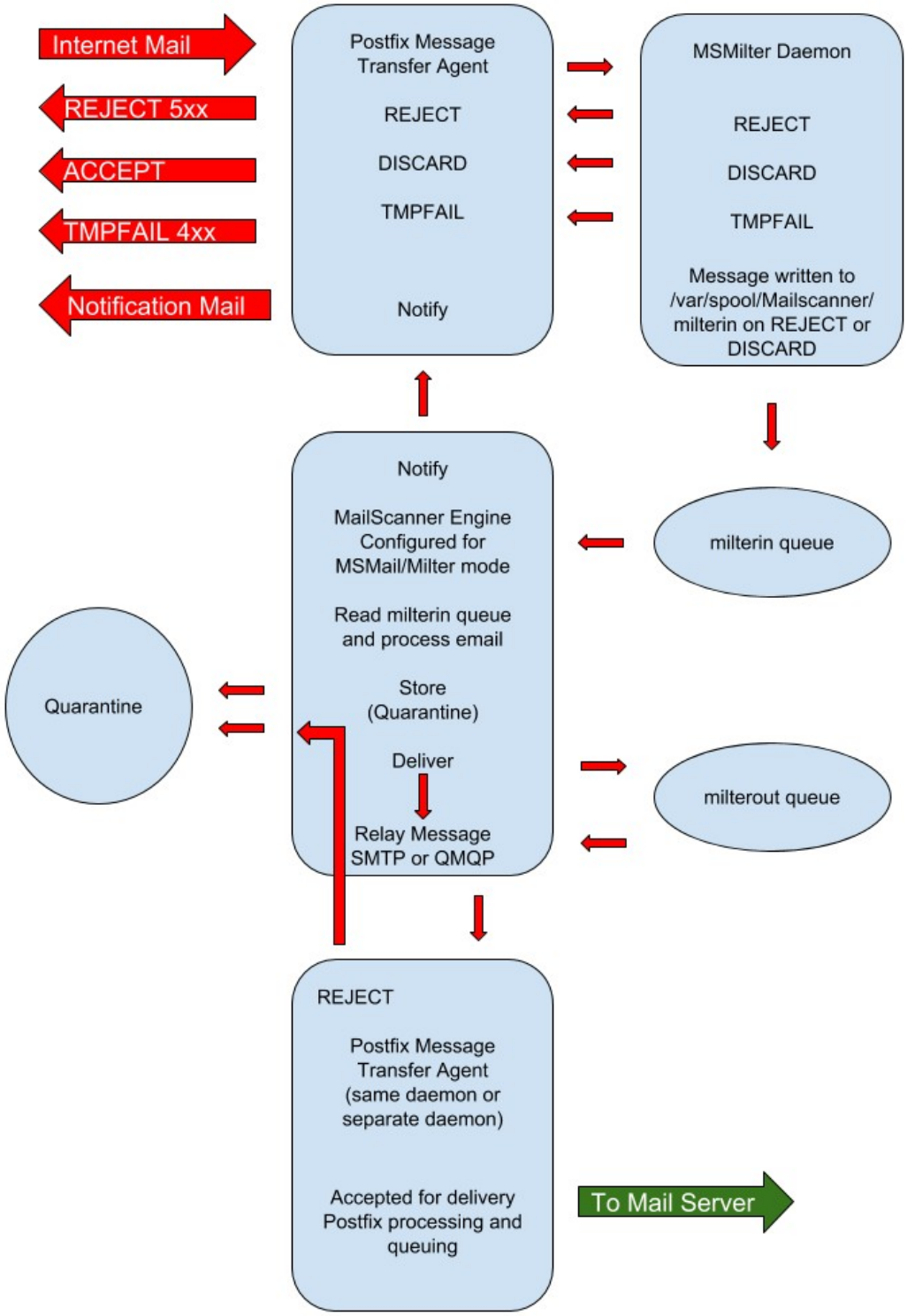
How the MailScanner Milter Works

Unlike previous versions of MailScanner, MailScanner 5.1.3+ introduces a milter daemon for postfix.

MailScanner still functions normally for all Message Transfer Agents (MTAs) and can run as it always has prior to this version. This optional functionality for MailScanner provides a Postfix compatible interface with MailScanner to process and scan email, and it will likely evolve with additional functionality in the future. Use of the milter decouples MailScanner from Postfix, so the Postfix queues are untouched, and MailScanner can operate independently with its own milter queues.

This is MailScanner Milter version 1.3, fourth release. Use of the MailScanner Milter is currently in beta, as bugs may be discovered after this release, and additional features will likely be added. This documentation will be updated to reflect the current status of the MailScanner Milter. As always, test it out before putting it into production.

On the next page you will find a diagram that demonstrates the mail flow in the MailScanner Milter in conjunction with MailScanner itself.



MailScanner Milter Process Diagram Fig. 1.

The milter interfaces with Postfix to capture the incoming email conversation and either REJECT, DISCARD, TMPFAIL, or ACCEPT. A REJECT can occur if the Milter Scanner is enabled and an email is blacklisted. TMPFAIL may occur if the milter cannot write to the disk (such as out of disk space or an access denied). ACCEPT is fired upon a localhost connection relaying email. A DISCARD, the most common response, tells Postfix to ACCEPT but silently drop the message. At the same time, the milter has written the message to */var/spool/MailScanner/milterin.*

The DISCARD technique supports large volumes of email for bulk scanning, but it neglects to send any sort of REJECT back along the pipeline as well as TMPFAIL in the event that the internal destination is unavailable. A future version of the milter may support “Full Milter Scanner” mode in which traditional MailScanner is turned off and the Milter does all scanning, returning REJECTS and TMPFAILS at the expense of sacrificing bulk scanning for those who need this functionality and have lighter workloads.

Installation and Configuration

Installing MailScanner

To obtain MailScanner Milter support, ensure your system meets the minimum system requirements and that the following is true:

MailScanner >= version 5.1.3-1
Postfix >= 2.3
Sendmail::PMilter perl module (compiled automatically from CPAN if not installed)

Initial Configuration

If you just installed MailScanner, you need to configure from scratch. Here's what you need to set to get basic MailScanner functional on your system.

/etc/MailScanner/defaults

```
run_mailscanner=1
```

/etc/MailScanner/MailScanner.conf

```
%org-name% = myorgname
%org-long-name% = my org name
Run As User = postfix | mail
Run As Group = postfix | mail
```

In additon to the configs, set the proper permisssions on the spool directories (use Run As User and Run As Group set above).

```
# chown -R postfix:postfix /var/spool/postfix
# chown -R postfix:mtagroup /var/spool/MailScanner
```

Configuring MailScanner for Milter Mode

To enable milter mode, edit */etc/MailScanner/MailScanner.conf* and change the following:

```
Incoming Queue Dir = /var/spool/MailScanner/milterin
Outgoing Queue Dir = /var/spool/MailScanner/milterout
MTA = msmail
MSMail Queue Type = short | long
Milter Scanner = yes | no
```

Use the queue type that matches postfix and choose whether the Milter Scanner is enabled (REJECT blacklisted emails).Ensure that the */var/spool/MailScanner/milterin* and */var/spool/MailScanner/milterout* directories are present and are owned by postfix (MailScanner Run As user). On debian-based systems, the user may be mail instead of postfix.

```
# mkdir -p /var/spool/MailScanner/milterin
# mkdir -p /var/spool/MailScanner/milterout
# chown postfix:mtagroup /var/spool/MailScanner/milterin
# chown postfix:mtagroup /var/spool/MailScanner/milterout
```

Using SMTP Delivery Method

To use SMTP as the delivery method after scanning mail, make sure the following settings are in effect in */etc/MailScanner/MailScanner.conf:*

```
MSMail Delivery Method = SMTP
MSMail Relay Port = 25
MSMail Relay Address = 127.0.0.1
```

You can also deliver mail to a MTA on another system. Adjust the address accordingly. If you use a loopback address, this line must be set to yes:

```
Milter Ignore Loopback = yes
```

Using QMQP Delivery Method

QMQP has distinct advantages over SMTP. Ignoring the loopback in the milter is not required when everything is on a single host, allowing messages submitted locally via loopback to also pass through MailScanner. QMQP is also faster since mail is immediately queued and bypasses most SMTP checks that have already been performed prior to scanning the mail.

Using QMQP With a Unix Socket

Using a Unix socket is ideal for a single node installation. To enable QMQP via a Unix socket, configure postfix as follows.

In */etc/postfix/master.cf,* add or edit the following line:

```
qmqp      unix  n       -       n       -       -       qmqpd
```

In */etc/postfix/main.cf,* add the following line:

```
qmqpd_authorized_clients = 127.0.0.1
```

In */etc/MailScanner/MailScanner.conf:*

```
MSMail Delivery Method = QMQP
MSMail Socket Type = unix
MSMail Socket Dir = /var/spool/postfix/public/qmqp
Milter Ignore Loopback = no
```

Using QMQP With an Inet Socket

Alternatively, you can use QMQP via an ip address and port. This is ideal for clusters where email is submitted to a common and dedicated QMQP relay.

In */etc/postfix/master.cf,* add or edit the following line:

```
628     inet  n       -       n       -       -       qmqpd
```

In */etc/postfix/main.cf,* add the following line:

```
qmqpd_authorized_clients = 127.0.0.1
```

When using a centralized QMQP relay, adjust the ip addresses accordingly for your MailScanner relays.

In */etc/MailScanner/MailScanner.conf:*

```
MSMail Delivery Method = QMQP
MSMail Socket Type = inet
MSMail Relay Port = 628
MSMail Relay Address = 127.0.0.1
Milter Ignore Loopback = no
```

Again, if you are using a centralized QMQP relay, adjust the relay address accordingly.

Configuring Postfix for Milter Mode

If /etc/postfix/header_checks contains the following line, remove it:

```
/^Received: / HOLD
```

Add the following to */etc/postfix/header_checks* to remove the localhost received header that postfix adds during local SMTP relay (optional). Replace <fqdn> with your host (without < >) as it appears in the Received header after delivery. This is important to ensure that you do not accidentally remove an external Received header.

```
/^Received:\ from\ <fqdn>\ \(localhost\ \[127.0.0.1/ IGNORE
/^Received:\ from\ <fqdn>\ \(localhost\ \[::1/ IGNORE
```

Add the following to /etc/postfix/main.cf:

```
smtpd_milters = inet:127.0.0.1:33333
```

(note, if you have multiple smtp milters, make sure this is the last milter listed as it will interrupt other milters)

Enable and restart everything.

(systemd)

```
# systemctl enable postfix
# systemctl restart postfix
# systemctl enable mailscanner
# systemctl restart mailscanner
# systemctl enable msmilter
# systemctl restart msmilter
```

(init)

```
# chkconfig postfix on
# service postfix restart
# chkconfig mailscanner on
# service mailscanner restart
# chkconfig msmilter on
# service msmilter restart
```

Troubleshooting

Services are running, but incoming mail receives a TMPFAIL.

Observe the maillog for incoming mail to see why the TMPFAIL is occurring. Most likely causes are the following:

- */var/spool/MailScanner/milterin* is not writable
- selinux or apparmor is enabled and not configured properly to allow access to milterin
- Missing configuration parameters in MailScanner.conf
- Postfix misconfiguration

Services won't start.

Examine the logs to determine the cause of failure. Most common causes include:

- Missing required perl modules
- Typo or misconfiguration in MailScanner.conf
- */etc/MailScanner/defaults* not configured and run_mailscanner=1 not set

/var/spool/MailScanner/milterin is filling up, MailScanner not processing the email.

Common causes:

- MailScanner is not running or misconfigured
  - Using "postfix" instead of "msmail" for the MTA
- selinux or apparmor aren't allowing MailScanner to read files

- Partial messages are being written to milterin (examine a message file for completeness of headers and body)
  - MailScanner will report INVALID messages found if this is occurring in the maillog.
  - Firewall or IDS upstream is ending the email conversation prematurely and/or mangling the messages

<u>/var/spool/MailScanner/milterout</u> is filling up but is not being delivered

Again, examine the maillog for cause of failure. Ensure that postfix is running, listening, and accepting messages (default port is 25 on localhost). Ensure that MailScanner can read and access the milterout directory.

Debugging the Milter

To enable debug mode, edit */etc/MailScanner/MailScanner.conf:*

```
Debug = yes
```

Restart msmilter for debug mode to turn on and received detailed milter logging to the mail log. Remember to set Debug = no when done and restart msmilter again.

Debugging the MSMail MailScanner Processor

After enabling debug mode, stop mailscanner, and run */usr/sbin/MailScanner* as root. MailScanner will process a message and exit, writing detailed logging to the mail log.

Remember to set Debug = no before starting the mailscanner daemon again.