

sendmail, postfix, and MeTA1

A Comparison of MTAs

Claus Aßmann

2008-06-13

Content

- ▶ History
- ▶ Documentation
- ▶ Configuration
- ▶ Security
- ▶ Features

History

Why implemented?

- ▶ sendmail: written in 1970 for UUCP, Bitnet, ARPAnet
General purpose internetwork mail router
- ▶ postfix: secure replacement for sendmail
- ▶ MeTA1: reliable, secure, efficient, extendible MTA

First Release:

S 1993 (sendmail 8)

P 2001

M 2005

Documentation

- S** INSTALL, 9 README files, op.ps plus cf/README: some information duplicated and spread out (binary / configuration file)
- P** 51 README files, html docs
- M** basically one README file (different formats), plus some files for developers

Program Structure

S one monolithic program, `fork()` new process for each SMTP connection.

P similar to MeTA1

- M**
1. `mcp`: starts all other modules, runs as root:
 2. `smtps`: the SMTP server receives e-mails.
 3. `smtpc`: the SMTP client sends e-mails.
 4. `smar`: the address resolver provides lookups in various maps including DNS for mail routing.
 5. `qmgr`: the queue manager controls the flow of e-mails through the SMTP servers and clients.

Configuration

- ▶ Preprocessing

- S mc file processed by m4 into cf file.

- P none necessary.

- M none necessary.

- ▶ Structure

- S strict layout: one option per line; no sections.

- P strict layout: one option per line; no sections.

- M free layout; sections and subsections.

- ▶ Syntax checks

- S no feedback on syntax errors in mc file.

- P complete syntax check.

- M complete syntax check.

Configuration (cont.)

S Options:

- ▶ 0 X=value
- ▶ 0 Option=value
- ▶ 0 Option=name=value, name=value, ...
0 DaemonPortOptions=Family=inet,
Address=0.0.0.0, Name=MTA,
InputMailFilters=f1;f2, Modifier=flags

P module_option = value

Problems:

- ▶ continuation lines
- ▶ TABs in master configuration (inherited from inetd?)

M structured, simple C-like syntax

Configuration Syntax for MeTA1

Syntax similar to C programs

```
smtps {  
    log { facility=mail; ident="smtps"; }  
    flags = {access} CDB_gid=353;  
    listen_socket { type=inet; port = 25; }  
    start_action =pass; pass_fd_socket = smtps/smtpsfd;  
    user= metals;  
    path = "/usr/local/meta1/libexec/smtps";  
    arguments = "smtps -f /etc/meta1/MeTA1.conf";  
}
```

sendmail 8 versus postfix versus MeTA1

- ▶ Security

- S** not initially designed for security;
uses direct manipulation of strings (`char` arrays).

- P** designed for security

- M** designed from the ground up for security;
uses abstract string type that can check for array boundaries.

- ▶ Runtime user

- S** runs as `root` by default.

- P** only one module runs as `root`.

- M** only one module runs as `root`.

Message Manipulation

- ▶ Header manipulation
 - S via rulesets (generic rewrite engine) and milter.
 - P regex and milter.
 - M via milter.
- ▶ Body manipulation
 - S via milter.
 - P via milter and others.
 - M via milter.

Delivery Modes and Resource Control

- ▶ Operation modes
(how to deal with incoming mail before delivery)
 - S background, queue, interactive.
 - P queue.
 - M queue.
- ▶ Resource control
 - S several options to control number of processes, but those work only for some cases.
 - P number of processes specified.
 - M number of processes fixed, most data structures have a limited size.

Scheduling and Queue Control

- ▶ Scheduling policies
 - S tied to a queue (group).
 - P some options to control qmgr.
 - M flexible by design, but not yet configurable beyond a few basic options (delay, number of connections, etc).
- ▶ Queue Control: Configuration
 - S queue runs at regular intervals: entire queue content is read and all transactions are tried (by default).
 - P options to control qmgr.
 - M entries in queue are scheduled based on *next time to try*, scheduling is done per recipient (not per transaction).

Misc

- ▶ Checks in SMTP server (envelope, headers, policy)
 - S via rulesets (generic rewrite engine) and milter.
 - P build in, milter.
 - M some build in, rest via milter.
- ▶ User interface
 - S `sendmail` command used for many purposes.
 - P postfix and several other commands.
 - M startup via `mcp`; other commands for other purposes.
- ▶ Map types
 - S Berkeley DB, LDAP, NIS, socketmap, regex, etc.
 - P Berkeley DB, LDAP, NIS, regex, etc.
 - M currently only Berkeley DB, CDB, socketmap

- ▶ Extensibility

- S milter; rewrite engine and ruleset hooks.

- P milter; policy daemon

- M milter; module system planned.

Milter APIs

MeTA1 milter API essentially the same as sendmail 8. postfix implements almost the full milter API.

Important differences:

1. MeTA1 can connect only to one milter.
Solution: milter multiplexor.
2. MeTA1 operates on the entire message, not header and body separately.
Library functions are available to offer header parsing.
3. MeTA1 streams message to milter while it is coming in.
sendmail 8 stores full message first then invokes milter(s).

Questions

Are MTAs still relevant?