

Internet mail infrastructure

- How email gets around
- Popular SMTP servers
- Postfix
- Popular IMAP/POP3 servers

History of email

- First RFC describing modern Internet email published in 1980.
- It was designed around the culture of login servers
- SMTP is a “push” protocol.
- Typical mail server did 4 things
 1. Accept mail from local programs and users
 2. Accept mail from other servers
 3. Deliver mail to local users
 4. Deliver mail to other servers

Result: SMTP does the following

- Accepts mail from other servers
- Sends mail to other servers

Result: Most SMTP servers also ...

- Accept mail from local programs/users
- Deliver mail to local mailboxes

Interesting historical description:

<http://www.coruscant.demon.co.uk/mike/sendmail/history.html>

MUA and MTA

- MTA (Message Transfer Agent) is basically the mail server. It basically has the chore of accepting messages and ensuring they get to the proper destination.
- MUA (Mail user agent) Provides a way for a user to interact with mail. It allows the user to submit new messages and view the messages that have been delivered to them.

POP3 and IMAP

- POP3 allows a client application to fetch messages from a server in a pull fashion. Messages are normally removed from the server and stored on the client.
- IMAP allows a client application to access and manage the messages on a server in a client/server fashion, messages are not normally removed from the server.

The modern MUA, a hodgepodge of parts

- “Traditional” Unix MUAs pass mail onto a local mail server (MTA) (ssmtp) for processing, and access mail directly from a mailbox file or Maildir.
- Most modern MUAs have ½ of an SMTP server (the sending half) but only speak POP3 and IMAP.

A tale of 3 MTAs

- Sendmail: One Big Program handles All four tasks. Sendmail has been around since about 1979.
- qmail: Different programs handle different tasks, includes POP3 server, and introduces a better way to store mailboxes (Maildir). No security problems found since 1.03 release in 1998
- Postfix: Greatly inspired by qmail, yet adds a number of features required by a modern mail system into the core system.

Sendmail

- One big program handles four mail functions: receiving from local program, receiving from remote systems, sending to remote systems, and storing in mailboxes
- Configuration file is about the most complex thing I've ever seen
- Long history of security problems

“Bernsteinisms”

General belief that software bugs are caused by lazy or incompetent programmers. DJB guarantees (with cash) that certain software (qmail and tinyDNS) are free of security flaws. He believes that when code is properly organized, it doesn't have bugs.

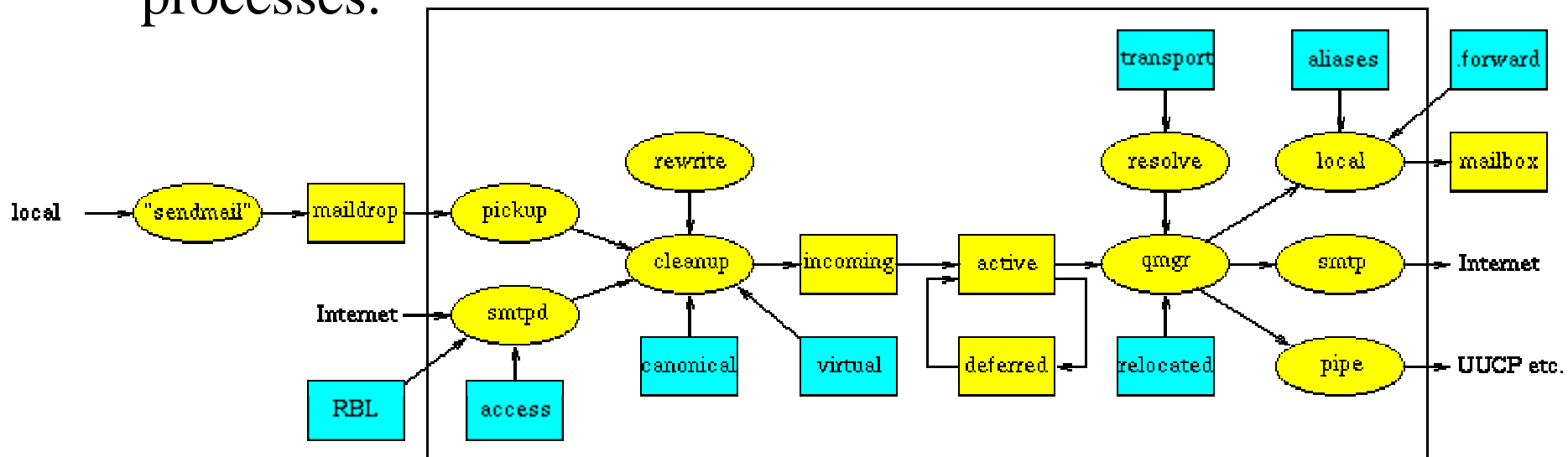
He accuses sendmail and bind of being the antithesis of this approach, and claims that both sendmail and bind will continue to have security problems with each new release until they properly reorganize their codebase.

qmail

- Many small programs, each with a specific part to play (The Unix Way)
- Many programs run as a special user that has only enough system rights to perform its designated function.
- This approach allows features (virus scanning, etc) to be added to the processing chain rather easily
- Configuration contained in /var/qmail, one file for each config parameter, pretty straight-forward
- Initial setup/install can be rather overwhelming
- Just about EVERYTHING requires an add-on program

Postfix

- Similar configuration to qmail that it consists of several programs working together.
- These programs are controlled by master.cf (usually in /etc/postfix or /usr/local/etc/postfix)
- master.cf allows you to easily do things like configure processes to be chrooted, or replace standard Postfix processes.



main.cf

- Usually in `/etc/postfix` or `/usr/local/etc/postfix`
- Controls many facets of how mail is actually processed
- Processing controls (i.e. how to deliver mail destined for local users)
- Anti-spam
- aliases

Processing controls

```
mydestination = $myhostname, localhost.$mydomain, $mydomain,  
mail.$mydomain, www.$mydomain, ftp.$mydomain
```

```
local_recipient_maps = unix:passwd.byname $alias_maps
```

```
alias_maps = hash:/etc/aliases
```

```
#home_mailbox = Mailbox
```

```
#home_mailbox = Maildir/
```

```
smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)
```

Anti-spam

```
smtpd_recipient_restrictions = reject_unauth_destination,  
reject_unknown_hostname,  
reject_non_fqdn_sender,  
check_client_access hash:/usr/local/etc/postfix/maps/access_sender,  
reject_invalid_hostname,  
reject_non_fqdn_hostname,  
reject_rbl_client bl.spamcop.net,  
reject_rbl_client relays.ordb.org,  
reject_rbl_client sbl.spamhaus.org
```

Example access list

virus-infected.com	REJECT
friend.spammer.com	OK
spammer.com	REJECT Stop spamming

Effectiveness of spam controls

- In a single day, 293 of 612 attempts to send mail were blocked as spam.
- Approximately 10 junk emails got through.
- 39 were blocked by the RBL lists.
- Approximately 83% of spam can be blocked by refusing to accept mail from misconfigured servers.
- In two weeks of operation 0 false positives.

POP3/IMAP

- POP3 for “downloading” mail from a server
- IMAP for managing mail on the server
- qmail's pop3d (part of the qmail program suite)
- qpopper (<http://www.eudora.com/qpopper/>)
- Cyrus (IMAP+POP3)
(<http://asg.web.cmu.edu/cyrus/imapd/>)
- dovecot (IMAP+POP3)
(<http://dovecot.procontrol.fi/>)

dovecot

- Simplified IMAP/POP3 system
- Supports just about everything you'd want in an IMAP/POP3 system, without a lot of complexity.
- Author is a strong believer in the “bug-free” software concept

Additional Features

- SMTP now supports password authentication, to allow “road warriors” to relay mail.
- SMTP, POP3, and IMAP also allow TLS to encrypt traffic, allowing safe use over untrusted connections.

Example setting for secure access

