OpenSMTPD for the Real World

BSDCan - Mail Server Tutorial

Aaron Poffenberger

akp@hypernote.com

2019-05-15T13:00:00Z

Introduction

Aaron Poffenberger

- Software developer
- OpenBSD user since ~3.2
- Mail Server admin 20+ years

Introduction – You

Enough about me, let's talk about you...

- Who runs:
  - OpenBSD
  - FreeBSD
  - NetBSD
  - DragonFly BSD
  - HardenedBSD
  - MidnightBSD
  - Other

Tutorial Goals

- Configure smtpd as a Mail Transfer Agent (MTA) for single and multi-domain use
- Install a certificate and configure smtpd to provide or require TLS
- Accept or reject mail based on criteria like recipient, source, sender and domain
- Tag mail
- Configure SpamAssassin
- Configure ClamAV
- Configure smtpd to work with Spam Assassin, ClamAV and Local-Mail-Transfer-Protocol (LMTP) services (in series or individually)
- Sign Outbound Email with DKIMProxy
- Deliver mail to Dovecot

Tutorial Technologies

- One of the supported OSs
- Amavisd (amavisd-new)
- ClamAV
- DKIMProxy
- Dovecot
- OpenSMTPD
- SpamAssasin
- Spamd
- Optional but Recommended:
  - OpenBGPD

OpenSMTPD - Overview

- Current version: 6.4.1, released 2018-12-16
  - *N.B.*: OpenBSD -current has 6.5.0
- History: The smtpd program first appeared in OpenBSD 4.6.
  - man smtpd(8):
- Started by Gilles Chehade in late 2008
- ISC license
- Primarily developed by:
  - Gilles Chehade, Eric Faurot, Charles Longeau and Sunil Nimmagadda
  - And cast of others
- Portable version begun by Charles Longeau
- Found on:
  - NetBSD, FreeBSD, DragonFlyBSD, Mac OS X, Linux

OpenSMTPD Overview - Notes

- OpenSMDTP scrupulously follows RFCs
  - It's a good idea
  - Spammers don't ... extra layer of protection
- If you're trying a manual process (*e.g.*, sending mail manually via telnet) and it fails, you're probably doing it incorrectly. *E.g.:*
  - Correct: rcpt to:<user@domain>
  - Incorrect: rcpt to:user@domain
  - Incorrect: rcpt to: <user@domain>
  - Incorrect: rcpt to: user@domain

N.B.: Often easier to use `mail -s`

Important Keywords

- *Macros*
- `include`
- `table`
- `pki`
- `listen`
- `tag`
- ~~`accept`~~|~~`reject`~~
- ~~`deliver`~~
- `action`
- `match` + `reject`

See file: etc/mail/smtpd.conf

Important Keywords - Macros

- Expanded in context
- Names must start with a letter, digit, or underscore, and may contain any of those characters
- May not be reserved words

N.B.: Macros are not expanded within quotation marks!

Important Keywords - `table`

- Use to provide additional configuration information:
  - Lists or key-value mappings
- Types: db or file
- May also be inlined with comma-separated values (list or key=value)

Important Keywords - `pki`

- Several uses for this keyword:
- Specify a given hostname with a specific cert file
- Specify a key file for a specific certfile
- Specify which certificate to present in a **listen** directive
- Specify which certificate to present in a **relay** directive

N.B.: Grammar change `certificate` -> `cert`

Important Keywords - `listen`

- Specify a socket or an interface for incoming connections
- Multiple listeners may be specified
- May specify:
  - hostname to present in the greeting banner
    - default is server name or name in ${ETC}/mail/mailname
  - port
  - tls/ssl requirements
  - address family (inet4 | inet6)
  - directives to tag traffic

N.B.: Grammar change `mask-source` -> `mask-src`

Important Keywords - `action`

- Represent shift from one rule to match and and take action, to two-rule approach defining actions and matching criteria
- Specify name, method, and options
- The name will be referenced in subsequent `match` rules
- The method is the delivery method such as maildir, mbox, lmtp, relay, and others.
- Local delivery options (*i.e.*, non-relay options) allow specifying alias, virtual, and userbase lookup tables, as well as `mda`s.
- Relay delivery options include options for configuration as a backup mx or delivery to other hosts.

N.B.: Grammar change `deliver` removed and made part of `action`

Negate [!]

- ! negates the sense of the rule
- Shown as [!] in man pages and tutorial where optional

N.B.: Grammar change [!] moved to proceed most conditions

Important Keywords - `match` + `reject`

- Accept or reject messages based on SMTP session info
- Criteria for evaluation:
  - [!] for any: mail for any destination (not typical)
  - [!] for local: default and typically omitted
  - [!] tag tag-name: tag applied as part of client session from any: match all
  - [!] from local: match only local connections (default, not necessary)
  - [!] from src <table>: match connections from clients whose address is declared in the specified table
  - [!] mail-from <sender>: specify that transactions's MAIL FROM should match the string or list table sender
  - Numerous others. See `man smtpd.conf`

N.B.:

- Grammar change `tagged` -> `tag`

TLS with `acme-client`

- Let's Encrypt client https://letsencrypt.org/
- Works with OpenSMTPD, Dovecot, `httpd(8)`, and other tls-compatible clients

See file: etc/acme-client

OpenSMTPD Configuration

- man smtpd.conf(5)
- Simple, English-like syntax very similar to PF and other OpenBSD-projects
- Typical ${ETC}/mail/smtpd.conf
- May reference other configuration files
- include “/etc/mail/smtpd.conf.local”

See file: etc/mail/smtpd.conf

Dovecot - Overview

- Dovecot is an open source IMAP and POP3 email server for Linux/UNIX-like systems, written with security primarily in mind.
- High performance
- Standards compliant
- Plugin architecture

Dovecot - Configuration

- Very easy to configure
- Edit file ${ETC}/dovecot/local.conf to override default settings
- Virtual user password algorithm option identical to OpenSMTPD

See file: - etc/dovecot/local.conf - etc/dovecot/passwd

PF - Overview

- man pf(4)
- History: The pf packet filtering mechanism first appeared in OpenBSD 3.0.
- Packet Filter (PF) is OpenBSD's system for filtering TCP/IP traffic and doing Network Address Translation. PF is also capable of normalizing and conditioning TCP/IP traffic, as well as providing bandwidth control and packet prioritization.
- Originally written by Daniel Hartmeier and is now maintained and developed by the entire OpenBSD team.
- Ported to:
  - FreeBSD (somewhat incompatible now), NetBSD, Mac OS X
  - Oracle (seriously . . . and doing really good work on SMP!)

PF Configuration

- Need to add a few rules

See file: etc/pf.conf

`spf_fetch`

Collection of utilities to recursively look-up SPF records and manage whitelists:

- Initial release at BSDCan 2016 as part of the "OpenSMTPD for the Real World Tutorial"

Features: - `spf_fetch` to recursively look-up SPF records and resolve all entries to IP addresses (IPv4 and IPv6) - `spf_update_pf` to manage adding and removing IPs from pf tables - `spf_mta_capture` to watch log files for outbound sent mail, run spf_fetch on the domain, and add to whitelist - List of common domains (Google.com, Outlook.com, etc)

See: - git repo: `spf_fetch` - file: var/cron/tabs/root

`smtpctl spf walk` and `spfwalk`

- Rewrite of spf_fetch in c in collaboration with Gilles Chehade. (Let's be honest here, Gilles did most of the coding. Thanks, Gilles!)
- Imported into spmtpctl(8) as `smtpctl spf walk`. (I maintain the standalone version. See Links slide.)
- Same core features as `spf_fetch` (walking `spf` records)

`spfwalk` Example

```
$ smtpctl spf walk google.com

172.217.0.0/19
172.217.32.0/20
172.217.128.0/19
<snip>
2800:3f0:4000::/36
2a00:1450:4000::/36
2c0f:fb50:4000::/36
```

Or:

- `echo google.com | smtpctl spf walk`
- `spf_fetch google.com`

BGP-Spamd - Overview

- Distribution of whitelist and blacklist IPs using bgp protocol.
- BGP: purpose is to exchange information concerning "network reachability" with other BGP systems.
- Project run by Peter Hessler (phessler@ OpenBSD)
- Receives blacklist and whitelist data from several sources, including Peter Hansteen
- Works with pf and spamd
- pf by modifying a specific white-list table
- spamd by creating file create by a cron job

BGP-Spamd - Configuration

- ${ETC}/bgpd.conf
- Select a private AS id (autonomous system): default == 65001
- Un-comment a neighbor close to you
- Add pass rule to ${ETC}/pf.conf

See file: etc/pf.conf

BGP-Spamd - Configuration

- Add source to ${ETC}/mail/spamd.conf

See files: - bgp-spamd/bgp-spamd.black.sh - etc/mail/spamd.conf

BGP-Spamd Cron Job

- Add line to root crontab

See file: var/cron/tabs/root

Preventing Spam

- Two ways, not exclusive:
  - Email analysis
  - Black and white listing

Email Analysis

Scanning email when it arrives, scoring email based on several attributes:o

- DKIM Signing
- Origin
- Sender
- Receiver
- Content

Options

- DKIMProxy
- OpenSMTPD Filters
- Amavisd (new)
- Rspamd
- ClamAV
- SpamAssassin

DKIMProxy - Overview

- Perl module that implements the new Domain Keys Identified Mail (DKIM) standard
- Can sign and verify emails using digital signatures and DNS records
- Also an SMTP-proxy that signs and/or verifies emails
  - Designed for Postfix but works with any MTA that supports LMTP

DKIMProxy_in - Verify

- Option to handle in Amavisd or Rspamd:
  - Enable DKIMProxy_in when not using `amavisd` or `rspamd`

DKIMProxy_out - Sign

- Generate TLS signing key
  - openssl genrsa -out private/dkim$_{\text{exmple}}$.org.key 4096
  - chmod 400 private/dkim_exmple.org.key
  - Ensure daemon user for dkimproxy_out has access to private key
- Get public key from private
  - `openssl rsa -in private/dkim_example.org.key`
- Update DNS records to add public key

See file: var/nsd/zones/master/example.org.zone

OpenSMTPD Filters

- In early testing only
- New keyword `filter`

Example:

```
filter fcrdns builtin connect check-fcrdns reject \
    "550 go away you punk"

listen on all filter fcrdns
```

See blog post: More on OpenSMTPD Filters

Amavisd - Overview

- `amavisd-new` is a high-performance interface between mailer (MTA) and content checkers: virus scanners, and/or SpamAssassin
- Runs as a service
- Manages sending email through multiple services like ClamAV and SpamAssassin
- Supports LMTP or (E)SMTP
- LMTP is a queue-less counterpart to SMTP
- Which raises the question:

  "How do I ensure my mail isn't lost while it's in the LMTP receiver?"

- Several solutions possible. Amavisd does not return success to the caller until it has delivered the mail to

Amavisd - Configuration

- Amavisd configuration file is a dog's breakfast
- Fortunately, just a few changes are **necessary**
- Lots of optional dials and knobs

See file: etc/amavisd.conf

Rspamd vs Amavisd

- Rspamd is written in `c`, has a nice configuration wizard `rspamadm configwizard`, web UI, integrates with `dovecot` and other mail tools, has Lua scripting
- As noted above, the amavisd config file is a mess but, there are not that many you typically have to touch
- Rspamd seems to have the most momentum behind it
- Amavisd just had a major new release in 2018, and is still written in Perl for "easy" updating

The real question is: "Why use content-based filtering at all?"

ClamAV - Overview

- ClamAV is an open source antivirus engine for detecting trojans, viruses, malware & other malicious threats.
- Runs as a service or can be called ad-hoc
- Includes freshclam service to download up-to-date definitions

ClamAV - Configuration

- Just a few changes are **necessary**
  - Lots of optional dials and knobs

See files: - etc/clamd.conf - etc/freshclam.conf

N.B.: Must comment out word “Example” near top of file!

SpamAssassin - Overview

- Open Source anti-spam platform giving system administrators a filter to classify email and block spam (unsolicited bulk email)
- It uses a robust scoring framework and plug-ins to integrate a wide range of advanced heuristic and statistical analysis tests on email headers and body text including text analysis, Bayesian filtering, DNS blocklists, and collaborative filtering databases

SpamAssassin - Configuration

- Not many **necessary** changes
- Lots of optional dials and knobs
- Most notably, the option: *trusted_networks*
- Not critical, but will exclude mail from listed addresses from checking

Black and White Listing

- Configure firewall
- Build Whitelists
- Build Blacklists
- Test
- Automate
- Share lists among related servers

Step 1: Configure Firewall

- As noted, whitelists should always win

  Prevents missed email from our known-good list

- Expire blacklisted IPs regularly

Be conservative about blacklisting. 24 hours is usually long enough, especially when mining from logs. You don't want to blacklist forever an IP that might move to another host later.

Step 2: Whitelist Known Good Mailers

Who are the known good mailers?

For our purposes, mailers who play by the rules: - Mostly large, well-established players like Gmail, Microsoft, Fastmail, and yes, Yahoo

Step 2a: Common Domains

Currently ~140 domains `spf_fetch common_domains` list

- Add other domains we don't want to miss emails from
- Whitelists from other sources:
  - bpg-spamd

Step 2b: Watch for Outbound Mail and Add spfwalk'd Domains to Whitelist

- `spf_mta_capture` from my `spf_fetch` project is a good example script for monitoring `/var/log/maillog` for domains users send to
- Domains that appear frequently could be added to the list of "Known Good Mailers"

Step 3: Blacklist Known Bad Actors

Who are the bad actors?

- For our purposes, mailer who don't play by the rules:
  - Blacklisted by trustworthy sources
  - Send email to spamtrap addresses
  - Attempt to access resources that don't exist and they probably shouldn't be accessing anyway

Step 3a: Trusted Blacklists

Find your favorite, but note, you probably shouldn't have to pay for a good list:

- NixSpam
- bgp-spamd

  Very cool use of bgp protocol to distribute known good and bad IP addresses

  Run by Peter Hessler with input from trusted sources like Pitr Hansteen
- Numerous others

Step 3b: Log File Mining for Bad Actors

Theory

Spammers don't just spam from a given address. They also use compromised machines for other activity like finding ssh daemons that allow root logins, and for hunting for admin sites for common web apps.

Mine log files for IP addresses engaged in other bad behavior:

- httpd logs
- ssh logs
- Other logs

Step 3c: httpd logs

Scan logs looking failed requests:

- Broad-scope:
  - 403 and 404 errors
- Narrow scope (examples):
  - phpMyAdmin
  - mysql-admin
  - wp-admin
  - tmUnblock.cgi (Cisco/Linksys routers)

Should anyone outside your firewall be requesting these urls? Probably not.

Remember: If your firewall rules are setup correctly, whitelisted senders will still be able to send.

See script: `scan_logs_bruteforce`

Step 3d: sshd logs

- Broad-scope:
  - Any failed login attempt
  - Better: usernames not listed in `AllowUsers` or `AllowGroups` directive in `sshd_config`
    (You do use one of the Allow directives in `sshd_config`, right?)
- Narrow-scope:
  - root

Again, should anyone outside your firewall be trying to login as root via ssh?

See script: `scan_logs_bruteforce`

Step 4: Test

- Whitelists are mostly safe since they grant immediate access to the MTA
- Blacklists can cause trouble, but again, if the firewall is configured so that whitelisted IPs always pass, little trouble to be expected
- Greylisting is mostly safe, but ...
- Be sure to monitor logs:
  - Use cron to mail list of whitelisted IPs to mail admin
  - Use cron to mail list of blacklisted IPs to mail admin
  - `pfctl -t <table> -T show` is your friend

Step 5: Automate with **cron(8)**

- Add scripts to crontab(1)
- Most of the scripts work on a 24-hour rolling window

Step 6: Share X-Lists Among Servers (as Necessary)

`bgp-spamd` is an excellent example of how to distribute lists among your own hosts.

Isn't there a Risk of Blocking Legitimate Senders?

Not really.

Are Google, Microsoft, FastMail, or Hypernote (my domain) likely to connect to your sshd instance trying to login as root, or anyone for that matter? Or will anyone from their sending IPs try to view /wp-admin on your mail server?

Still, firewall rules should be configured to always allow whitelisted senders through.

What About DKIM and DMARC?

DKIM is the "Domain Keys Identified Mail" system:

- Useful for verifying that an email wasn't spoofed even if it came from your domain or one of your servers
- Not very useful for detecting spam (some spammers publish DKIM keys), and certainly not *before* the MTA receives the connection

DMARC is the "Domain-based Message Authentication, Reporting and Conformance" system:

- System based on both SPF, DKIM or both to detect spoofed email
- Provides a mechanism for providing feedback to the domain owner
- Allows setting a policy for what to do if one or both of

Is It Effective?

Yes, very.

Hypernote.com:

- Registered in 1995
- My primary email, akp@hypernote.com, in use since 1995
- On every spam list known to spammer-kind
- My primary email, akp@hypernote.com, is the domain catchall
- Typically receive 0 to 3 spams per week, occassionally peaks at 5

That said, the plural of anecdote is not data.

Conclusion

- You have questions, I may have answers

Contact Details

- Aaron Poffenberger
- akp@hypernote.com
- http://akpoff.com
- [@akpoff](https://twiter.com/akpoff)
- This presentation, look for blog post on http://akpoff.com
- KG5DQJ

Credits

- OpenSMTPD Team
- Pitr Hansteen

Thanks

- BSDCan, Sponsors, and Volunteers

Support OpenBSD and the OpenSMTPD Team

- [http://www.openbsdfoundation.org/]

Resources

N.B.: Most of these are links

- Updates to OpenSMTPD Grammar
- Amavisd
- BGP-Spamd - Peter Hessler
- ClamAV
- DKIMProxy
  - DKIM Tutorial
- Dovecot
- OpenBSD PF FAQ
- Rspamd
- SpamAssassin

Blogs and Other Information

- Other Blogs and Posts about BSD Mail Serving
- Frozen Geek
- Hugo Osvaldo Barrera - Good discussion about using Sqlite for Virtual Users
- That Grump BSD Guy - Peter Hansteen
  - Spamd
  - BSDly - Blog about OpenBSD, PF and greytrapping