Fighting Spam at the Frontline

Using DNS, Log Files and Other Tools in the Fight Against Spam

Aaron Poffenberger

akp@hypernote.com

2018-06-09T18:30:00Z

Thanks

- BSDCan
- BSDCan Sponsors
- BSDCan Volunteers

Introduction

Aaron Poffenberger

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

Origin of Talk

- 20+ years running mail servers
- 20+ years receiving spam
- Ineffectual client-side spam filtering
- Ineffectual server-side spam filtering
- Promise and pain of greylisting
- pf(4)
- Experience developing BSDCan tutorial “OpenSMTPD for the Real World”

The Goal

Reduce spam:

- Block at or before the MTA
- Avoid content analysis
- Allow “legitimate” senders to connect
- Prevent “illegitimate” senders from connecting
- Impose little or no delay on legitimate senders
- Minimal-resource usage
- **Mostly** automatable

Definitions

Let's start with some definitions.

What is Spam Email?

A common definition (with my additions):

> *Unsolicited [commercial] email, typically of an advertising nature, that recipients cannot [easily] unsubcribe from, and is* ***typically*** *sent from non-canonical senders (compromised systems).*

What is *Not* Spam Email?

For the purposes of (mostly) automated blocking at the MTA, the following are not spam:

- Newsletters your users signed-up for in 1997 and never unsubscribed from
- Opt-out marketing mail your users get when joining a website
- Political junk from uncle Joe

In short, email is not spam just because the recipient doesn't like it or finds it annoying.

X-Listing: White, Black and Grey

Blacklisting	block delivery from specific IPs
Greylisting	temporarily fail delivery from unknown IPs until some criteria met, then [temporarily] whitelist
Whitelisting	always allow delivery from specific IPs

In the Beginning was the Blacklist

Not really

- In the beginning we accepted email from anyone . . . and gladly forwarded on their behalf.

  That didn't last long.

- Rise of real-time blacklists (SpamHaus, et al.)

  Admins began blocking bad actors, then sharing the lists.

- Problem: False positives

  Q. How did my IP get on this list?

  A. You were reported for spamming.

  Q. How do I get off?

  A. Die spamming scumbag!

Let There Be Whitelists

- Blacklists inevitably beget whitelists
- Great idea, in theory:

  I'll add the IPs of must-receive senders to my firewall rules.
- Problem: Not scaleable

  What do you do when your "must-receive senders" change their IP address and you don't notice?

  How do you manage updates for dozens or hundreds of these "must-receive senders"?

Greylisting

- Whitepaper by Evan Harris in 2003
- Based on observation that spammers typically don't retry failed delivery
- Implemented in OpenBSD spamd(8) in 3.5

  Available in one form another for almost every MTA and platform
- Problem:
  - Delayed delivery can be costly.
  - Some MTAs have (in the past) treated temporary failure as permanent.
  - Some large senders (ahem, Google) send from more than one IP address. Most do not practice IP affinity, *i.e.,* they don't resend from the same IP address that recently failed.

What’s Changed to Make X-listing Viable?

Not much:

- Well, there is SPF
- Immense consolidation in the number of domains handling their own email

And:

- Spammers still spam from compromised IP addresses
- And do a lot of other bad things from those same IP addresses

From these three facts we can begin blocking more spam at the MTA.

Sender Policy Framework (SPF)

Publish DNS TXT records to specify authorized outbound mailers by IP, IP Range, host, and/or include SPF records of other domains

```
$ dig example.com txt

;; ANSWER SECTION:
example.com. 300 IN TXT "v=spf1 ip4:192.0.2.1 -all"
```

Why Does Specifying the Sending IPs or Hosts Matter?

20 years ago MX records usually indicated send and receive. Today none of the major mail domains use MX hosts to send email.

- SPF records are like MX records but for outbound mail
- If I have a canonical record specifying which IPs or hosts will be sending email for a give domain, I can build a better whitelist
- and automate the process

Are there Tools to Walk SPF Records?

Yes, yes indeed.

- `spf_fetch`
- `spfwalk`
- `smtpctl spf walk`

`spf_fetch`

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

- Initial release during 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)

`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`

Example

```
$ spfwalk 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`

How Do I Put All this Together to Prevent Spam?

- 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:

- It's a timey-wimey, I know one when I see one, kind of thing
- 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.

You went to Peter Hessler's bgp tutorial, right?

If not, the basics are pretty easy to learn.

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 If I Don't Run OpenBSD?

- OpenBSD is not a requirement:
  - Need a firewall that supports rules, ideally rules driven by tables that can be dynamically updated
- **spamd(8)** is not a requirement either:
  - Plugins or milters available for many MTAs that will greylist
  - Rspamd has a greylist module
- `smtpctl spf walk`, `spf_fetch`, and `spfwalk` work on many platforms:
  - `spf_fetch` is pure shell and should work almost anywhere
  - OpenSMTPD has been ported to many platforms
  - `spfwalk` hasn't been tested on many, but is clean c code

    Send complaints to akp@hypernote.com

Other Interesting Options

- Postscreen:

  > *[H]andles multiple inbound SMTP connections, and decides which clients may talk to a Postfix SMTP server process.*

  - Accepts connections on port 25 and decides whether to forward the connection on to Postfix
  - Postfix specific :(
- Post-MTA frameworks like Amavisd-new and Rspamd offer many of the same feature

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 the SPF and DKIM checks fail
- No more useful than SPF alone for deteting spam *before* the MTA receives the connection

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.

Analytics

- I don't have data . . . yet.
- Techniques developed over time with no concern about tracking results other than "Hey, my Inbox has less junk in it!"
- Testing some ideas for counting denied connections vs actual connections
- "Stand back! This may require code." But maybe not.

Where's the Code?

Most of it is in `spf_fetch` (see Links slide).

A few bits left to commit and push.

Credits

- Gilles Chehade
- Pitr Hansteen (pf, mail rants)
- OpenBSD developers
- Students in “OpenSMTPD for the Real World” tutorial
- And viewers like you!

Questions

You have questions. I may have answers.

Contact

- Aaron Poffenberger
- akp@hypernote.com
- http://akpoff.com
- @akpoff
- This presentation, look for blog post or on github
- KG5DQJ

Links

- bgp-spamd
- Greylisting Whitepaper
- Postscreen
- ‘spf_fetch’
- Gilles Chehade Blog Post Announcing `spfwalk`
- Standalone `spfwalk`