You had me at HELO

A case study in spam fighting

Randal L. Schwartz
Stonehenge Consulting Services, Inc.
mer(DELETETHIS)lyn AT stonehenge DOT com

What is spam?

- Unwanted mail
- ... as I defined “unwanted”
- Usually sales pitches
- Occasionally virus/worm payloads

First amendment

- I will defend your right to say what you want to say...
- ... but not at my expense
- “Freedom of the press” doesn’t mean everyone gets a free press
- ... and especially not you using my press to try to sell me something

My role

- Own a small business
- Therefore, the online presence manager
- Actually, more of a janitor
- Single machine, leased at a co-lo
- Biggest expense: web bandwidth
- ... in theory

B.S. (before spam)

- Could put an email address on a web page
- Or in a Usenet posting
- Or in a mailing list that got archived
- ... without worrying about it being spammed to death

A.S.S. (after spamming started)

- If you use an email address on a web page, or in Usenet posting...
- ... you can expect hundreds or thousands of messages per day
- Fewer (perhaps none) if you can entitize or use “illegal characters”
- But once it’s in the wild, it’s permanent

But it gets worse

- Apparently, 80 million addresses weren’t enough
- Dictionary attacks
- Address-book scraping worms

And worse

- “Joe jobbing” and friend-spam to get past human filters, using false “From” lines
- But this also triggers false replies from “helpful” (actually harmful!) anti-spam and anti-virus filters
- Important note: do not bounce spam. Ever. Ever! Just drop it. Please. Please!

The first step

- Mail for $user@stonehenge.com came in to my email via a procmail script
- In 2002, I added a SpamAssassin check, looking at content and RBLs
- And instantly, my unwanted mail went to about a third of what it was!
- Yeay!

Then came the MIRVs

- Multiple Independently targetable Reentry Vehicles
- Or in my case, spam addressed to multiple users within the stonehenge.com domain.
- One SMTP transaction, but many separate procmail deliveries

Why MIRVs hurt

- They came in clusters: my MTA delivered them all in a row quickly
- They launched separate Perl invocations for SpamAssassin
- Result: at first, load average spikes, then continuously bad load

The next step

- In mid 2003, I moved SpamAssassin into the MTA (Postfix), before the delivery queue
- Amavisd keeps code in preforked memory
- Also (most important) handles a MIRV before it bursts
- And, can reject spam during the SMTP handshake, so no need to bounce spam

A bit of relief

- For a while, life was good
- From about September 2003 to February 2004
- Spam slowly increased, but mangeably
- And then, “invasion of the worms”

Worms, ick!

- Windows infestations
- Very rapidly spread
- Many variants (antivirus people fighting to keep up)
- Included their own MTA, with its own rules about connections and delivery rates

Worm mail

- Fat payload
- Rapid fire (many connections at once)
- Ultimate purpose:
  - To spread
  - To create millions of spam relays controlled by private enterprise

My response

- Panic!
- Filter the worm payloads with procmail
- This kept my personal mail sorter from being invoked on worm payloads, saving me some CPU
- Tried to teach SpamAssassin about worm payloads, but failed

New problem: bandwidth

- Load average was marginal, but now my port 25 bandwidth far exceeded my web traffic
- Pushed me into overage charges
- Thank you sprocketdata.com for not actually charging me anything!

Fight back with filtering

- OpenBSD gives me pf: a very powerful packet filter and traffic shaper
- IP lists can be dynamically updated and efficiently scanned
- My goal: if someone is bad to me, then they only get one chance to be bad
- So I added a port 25 filter

Bad Boys, Bad Boys

- Block the people being bad to me
- Anyone sending mail to a known "spamtrap" mail address in stonehenge.com (mostly "thanks" to the evil henge.com)
- Anyone sending spam
- Anyone sending worm payloads

The details

- Perl process watching logs from Postfix, Amavisd, and procmail
- All first-hop IP addresses found there added to the port 25 block via pfctl
- Automatically removed after two hours
- Removal permits legit mail to flow again
- Most worms don’t retry once blocked

The results

- Email bandwidth reduced by half
- Now paralleled the web traffic
- Barely within purchased limits
- But at least this was livable
- At any given moment, 750 to 1500 IP addresses in the bad boy list

Why stop there?

- OpenBSD packet filter includes passive operating-system fingerprinting
- Based on p0f code
- Windows is evil
- So slow down all the windows email!
- Collective speed of all windows machines reduced to a dialup-modem speed!

Why slow and not stop?

- Because some of my friends actually run Windows and want to send me email
- Because some corporate mail gateways run Windows
- Net effect: most worms were slowed, but legit mail continued to trickle through
- All managed by pf: no work on my part

Who was I blocking?

- Packets being blocked are logged to pflog0
- Watch fingerprints with tcpdump
- Bad boys still attempting to reconnect about 30 to 90 times a minute
- Almost all connections are from Windows machines, probably wormed

More bad boys

- I realized that Postfix could block SMTP connections from unknown hosts
- Unknown hosts have no valid reverse DNS
- I added that block in mid-2004, and email was reduced a whopping additional 70%!
- Once again, SMTP traffic was dwarfed by web traffic! Load averages were normal!

The downside

- Reverse DNS should be set up for all corporate mail gateways
- But some Very Large Companies still aren’t paying attention
- So, I had to add a whitelist for the broken sites, and tools to watch the blocked list
- This extra workload is worth the result

High-MX spamtrap

- Legit mail should be delivered to low MX
- Some spam software delivers to higher MX values first (or only), violating RFCs
- Secondary machines rarely config'ed right
- This usually gets around spamblocks
- But this can also be a very solid spamsign

My implementation

- Added an alternate IP to my box
- Gave it a name, and a high MX value
- Taught Postfix to listen separately to the primary address and spamtrap address
- Postfix sends back a "450 Violation of RFC2821 Section 5 paragraph 8" for all connections to the spamtrap port 25

Caution about this spamtrap

- Must make sure that high MX wasn’t a legitimate sender rolling over because low MX was hosed (same host in my case)
- Return 450 code, so that a legit sender will retry, hopefully getting the non-spamtrap the next time
- Most spam senders will just go away

Results of High-MX spamtrap

- Implemented only recently
- Appears to be blocking 90 messages/hr
- No false positives seen!
- Also adding these to my bad-boy list, which foils spamsender from trying lower MX
- So, it's probably even better than it looks

Sharing the knowledge

- Publishing my bad-boys list experimentally as an RBL at rbl.stonehenge.com, updated every minute
- Please ask before using permanently
- Publishing list at http://www.stonehenge.com/pic/rbl.stonehenge.com.txt, updated every minute

To-do, to-don’t

- Tried and abandoned HELO-based domain verification blocks; too many machines are misconfigured
- Look at longer history of IPs that repeatedly end up in my bad-boys list, moving them from two hour to permanent
- But for now, this is working well enough

The cost of spamfighting

- Many hours of labor to keep things updated
- Bandwidth overages
- Excessive CPU load averages
- And this is just for a single domain! (Think of the cost at a large ISP.)
- False positives also cost me real business

My dream

- Make junk email illegal (like junk fax/cell)
- Culpability for Operating System Producers (read: "microsoft") for failure to adequately build/test security, based on actual damages
- Include "chain of responsibility", even if junk email comes from outside the US

More resources

- Magazine articles I’ve written (Google for “site:stonehenge.com spam”)
- Documentation for Postfix, AmavisD, SpamAssassin, OpenBSD’s pf
- Hire me to come give you the longer detailed talk and/or set up your systems (www.stonehenge.com for contact info)