Filtering spam at the ISP

Patrick J Okui
Ayitey Bulley
(Liberal Borrowing from Brian Candler)

Sources of junk e-mail

- Spam
  - Unsolicited, bulk E-mail
  - Usually fraudulent - e.g. penis enlargement, lottery scams, close relatives of African presidents etc.
  - Low response rate -> high volume sent
- Viruses, Trojan Horses
  - Infected machine sends out mails without the owner's knowledge

How to filter:

- Accept all messages, then scan.
  - ✔ Simplest way for any MTA
  - ✔ Easy to have user customisation
  - ✘ Client must still download the spam **then** scan.
  - ✘ What do we do with the mail once it's classified as spam? (/dev/null? Bounce? Move to another mailbox?)
- Scan as messages arrive, reject at (or before)SMTP DATA time.
  - ✔ Users don't need to install anything.
  - ✔ Saves disk space
  - ✔ Reduces "collateral spam" (e.g. "you sent us a virus" messages)
  - ✘ Hard to have user customisation (white/black lists etc)

How to filter (2) 'hybrid' solutions

- All mails over a certain (high e.g. 25) "spam score" are discarded at SMTP time, other mails are tagged for customisable per-user filtering after that.

- Using the method outlined by Alan Flavell whereby you allow multiple reipients per SMTP session, but only if their scanning preferences match, and then scanning the e-mail based on the preferences. (I only know how to do this with exim)...

Views on filtering:

- Some customers may be upset that you are making value judgements on their mail, or looking in the contents
- In some cases, this could be seen as a value-added product for which customers may even be willing to pay.
- Best solution is to allow customers to be able to opt in or out for filtering. Also check that you are not liable in case the filters make errors.

Ways to identify spam: 1. By source IP address

- As soon as the sender connects, you know their IP address, which can't be forged
- You can check their IP address against 'blacklists' in real time
  - Blacklists of IP ranges assigned to known spammers
  - Blacklists of IP addresses of open relays / open proxies
  - Blacklists of IP addresses which have been seen sending spam recently
- Realtime Blocking Lists (RBLs) are queried via the DNS

Advantages of RBLs

- Easy to configure
- DNS lookups are relatively quick and cheap
- It's somebody else's job to maintain the lists
- Mail is rejected before the body has been sent, saving bandwidth

```
EHLO whitehouse.gov
250 OK Hello whitehouse.gov [192.0.2.1]
MAIL FROM:<president@whitehouse.gov>
250 OK
RCPT TO:<you@yourdomain.com>
550 rejected because 192.0.2.1 is in a black list at sbl.spamhaus.org
```

Disadvantages of RBLs

- RBLs are always under legal threats from spammers; they come and go
- Won't catch all spam
- Huge sections of Africa are in the RBL's – maybe even your own country!!!

Ways to identify spam: 2. By content

- Look for phrases which typically occur in spam
- Good systems also look for phrases which typically *don't* occur in spam to reduce false positives
- The balance between these two indicates whether it's spam (and how sure we are)

Advantages of content filtering

- Spammers are sad and predictable
- If you paid a human to delete spam, they could recognise it easily
- Doesn't matter where it came from: spam is spam

Disadvantages of content filtering

- Spammers use every trick in the book to disguise their wares
  - MIME base64 encoding, HTML mails, breaking up words with invisible tags in between ... etc
  - It's an arms race: as filters match particular patterns, spammers change their behaviour
- Computationally expensive
- Liable to false positives

Bayesian filtering

- Given a sample of messages which are known to be "spam" or "not spam", builds a map of which words occur more often in one than the other
- The "not spam" profile is different for everyone, and therefore much harder for spammers to guess
  - It's why many spams contain random words
- Filter is very effective, but needs ongoing "training" for mails which slip through

See **http://www.paulgraham.com/spam.html**

Ways to identify spam: 3. Whitelists

- Only accept mail from people we already know
- Actually, spammers could forge messages which appear to be from people we know
- But for now, they don't seem to be collecting information on who we associate with

Receiving mail from people not on our whitelist

- By password: e.g. if they include a magic word in the Subject: header
- By content filtering: e.g. if they pass spamassassin with a very low spam score
- Challenge-response systems put the mail in a hold queue and send back a message
  - If the person responds, they are assumed to be OK and are whitelisted.
  - One day soon, spammers will build robots to do this ☹

Advantages of whitelists

- Currently very effective at blocking spam and viruses
- Once we have established communication with someone, the probability of a future false positive is very low

Disadvantages of whitelists

- Makes it difficult or annoying for people we don't know to contact us for the first time
- On a server-side solution, each user needs a separate whitelist and a way to edit it
- Automatically whitelisting people we sent mail TO is tricky if done server-side
- Challenge-response systems are difficult to deploy in a scalable way
  - http://www.tmda.net/
  - http://www.paganini.net/ask/

Disadvantages of whitelists (2)

- If filtering at the MAIL FROM stage, beware that for many people the envelope sender is different to the From: address they put in their headers
  - MAIL FROM could even be different for every message they send (VERP: Variable Envelope Return Path)
- Challenge-response systems can interact badly with mailing lists
- Big risk of losing legitimate bounces
  - Bounces are an important part of the integrity of E-mail

BAD ways to identify spam

- Checking the domain of MAIL FROM:<...> or doing a callback to check the whole address
- Comparing the domain in MAIL FROM to the IP address the message came from (SPF)
- Checking whether the message is correctly formatted according to RFC rules, etc
- These rules might catch some spam, today (until the spammers adapt). But there are also plenty of badly-configured systems belonging to non-spammers. You WILL lose mail that you wanted to receive.

Identifying viruses

- Recent volume has increased massively
  - Users happily open and run attachments on mails from strangers!
- Like spam, current viruses have forged envelope sender and headers
- Naive implementation might block all attachments with executable extensions
  - Blocks too many legitimate uses of E-mail
  - Some viruses come in .zip files now

Identifying viruses (2)

- The only sure-fire way is content filtering: matching attachments against "signatures" (patterns) of known viruses
- Many solutions are commercial, expensive, cost increases with number of users
- Some are free, e.g. clamav
  - http://clamav.sourceforge.net/
  - Call it from exim using exiscan-acl (discussed later)
- New viruses are written all the time, signatures need updating very frequently

All those options: what should you do?

- Implement RBLs
  - surprisingly effective
  - very easy to do
  - low maintenance
- Consider implementing content filtering or virus scanning for a small proportion of your userbase
  - "Premium" users - pay extra?
  - These services are expensive to scale and to manage
  - For low spam scores, consider "tagging" the mail as spam instead of discarding it

What should you do? (2)

- Advise your customers to install client-side spam filters too
  - Bayesian filtering and whitelists are best handled here
  - Find ones which best suit the software which your customers tend to use
  - Find ones which best suit the software which your customers tend to use

Implementation for a few MTA's 1. Exim

- This is pretty easy, and most if not all of the ideas expressed above can be implemented using:
  - In built exim acl's for the blacklists by uncommenting sections of the configuration file (configure) e.g

    deny message = rejected becase $sender_host_address is in a blacklist at \
    $dnslist_domain\n$dnslist_text
    dnslists = sbl.spamhaus.org : relays.ordb.org: bl.spamcop.net
  - Exiscan – a patch to exim by Tom Kistner http://duncanthrax.net/exiscan-acl/ allows the exim acls to be extended to support both antivirus software like sophos and clamav as well as spamassasin and generic command line scanners. This is best suited for doing the virus scanning but since it has rather generic support for any scanner, it has limited reporting capabilities as far as scanning spam is concerned.

Implementation for a few MTA's 1. Exim (2)

- For more detailed control of spam scanning (only) you can use SA-Exim (http://marc.merlins.org/linux/exim/sa.html). In particular, SA-Exim lets you save rejected messages to a file, which might be handy if you arejittery about false positives.
- Advantage is both SA-Exim and Exiscan can co-exist peacefully, so best solution (IMHO) is use Exiscan for virus scanning and SA-Exim for spam scanning.
- Spam assasin in either case must be installed and ready to go (http://spamassasin.org). Other competing solutions include bogofilter and spamprobe.
- Supported virus software include Clam Antivirus (free), Sophos Antivirus (commercial ; can be daemonised or command-line), Kaspersky Antivirus (commercial), ScannerDaemon (free – from the OpenAntivirus project)

Implementation for a few MTA's
2. Postfix

- Lots of support exists on the postfix mailing lists and the web for all sorts of regular expressions to be matched, require a slight modification to your main.cf and/or master.cf configuration files.

- Virus and spam scanning is supported via amavis (and/or amavis-d) which is a perl interface to either sophos, clamav or other scanners (generic command line scanner support is also included). It requires one to also install the other components i.e SpamAssassin and/or Sophos etc.

Implementation for a few MTA's
3. Sendmail

- Consider changing MTAs 
- Scanning is possible, lots of rewrite rules can be played around with, and there is also support for passing mail to amavis to be scanned.
- However, writing these rules is by **no** means trivial, and is prone to error. Also very hard to customise for different user requirements.

Implementation for a few MTA's 4. etc

- Lots of other MTAs may not have inbuilt support for regex matching but probably will have patches written and actively supported by someone to do just that. Most plug into amavis which is a daemonised or comand line "middle-man" between the MTA and the horde of scanners available.

- In cases where this is not possible or feasible with the MTA being run, one could get pre-built scanners from vendors like Sophos to stand between the Internet and the MTA doing initial scanning... however this greatly increases the time it takes to get the message delivered and may not be affordable. Sometimes, it may be better to change MTAs.

Case studies/questions (from audience)

?