Filtering unwanted E-mails

Brian Candler

What are the main 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
- Malicious bounces ("Joe-jobs")
  - Spam or viruses sent with forged MAIL FROM
  - Any bounces go to innocent third party

What are the costs?

- Important E-mail messages can be accidentally discarded in a sea of junk
- Wasted time
  - Deleting junk
  - Setting up and maintaining filters
  - Scanning discarded messages looking for false positives
- Wasted bandwidth and disk space
  - Especially for users on modems
  - Viruses and spam attachments can be large
- Annoyance, offence, or even fraud

Where can you filter?

- At the end-user machines
  - ✔ each client has full control and customisation
    - Especially good for Bayesian filtering
  - ✔ distributes the processing cost
  - ✘ client must still download each message even if it's junk
- On the ISP's mail server
  - ✔ easier for users
  - ✔ in some cases mail can be rejected before transmission of the body
  - ✔ saves disk space on the server
  - ✘ hard to make flexible for users to configure or for them to browse rejected mail

Legal problems with filtering

- Some customers may be upset that you are making value judgements on their mail, or looking in the contents
- So make sure your contract with the customer allows you to do this
- Or allow individual customers to opt-in or opt-out of filtering
- Filtering is never 100% correct so make sure you're not liable for cases where filters make the wrong decision

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
- Not effective against viruses or joe-jobs

Choosing which blacklists to use

- Many are free, some are not
  - e.g. mail-abuse.org
- Some are not good
  - Policies are too draconian; you end up losing connectivity to people you want
  - Someone else's policy may not be good for you (e.g. a list which blocks all Nigerian address space is not useful for an African ISP)
- Try these:
  - sbl.spamhaus.org (known spammers)
  - relays.ordb.org (open relays)
  - bl.spamcop.net (dynamic spam sources)

Configuring blacklists in Exim

- Easy: uncomment two lines in the configure file and customise to your chosen lists

```
deny    message    = rejected because $sender_host_address is in a black list \
                     at $dnslist_domain\n$dnslist_text
        dnslists   = sbl.spamhaus.org : relays.ordb.org : bl.spamcop.net
```

- If your users are in a database, it's possible with some configuration work to use different dnslists for each user (opt-in, opt-out, choice of policies)

Testing blacklists with exim -bh

- exim -bh x.x.x.x sets up a pretend SMTP session as if it were from address x.x.x.x
- Many lists have test IP addresses which will definitely reject - e.g. 127.0.0.2

```
# /usr/exim/bin/exim -bh 127.0.0.2

**** SMTP testing session as if from host 127.0.0.2
**** This is not for real!
220 noc.t1.ws.afnog.org ESMTP Exim 4.34 Wed, 19 May 2004 10:26:40
mail from:<>
250 OK
rcpt to:<inst@noc.t1.ws.afnog.org>
550-rejected because 127.0.0.2 is in a black list at sbl.spamhaus.org
550 http://www.spamhaus.org/SBL/sbl.lasso?query=SBL233
quit
221 noc.t1.ws.afnog.org closing connection
```


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
  - Unless rules are customised for each user, but then it's more difficult to build a good server-side solution

Content filtering in Exim

- Apply the exiscan-acl patch *before* building exim
  - http://duncanthrax.net/exiscan-acl/
- Install spamassassin and run spamd
  - http://www.spamassassin.org/
- Set up an ACL to check the *body* of the mail and either reject or add a warning header
- Update spamassassin rules regularly
- Not trivial to implement

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 (cont.)

- 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 (see before)
- New viruses are written all the time, signatures need updating very frequently

"Joe-jobs"

- A spammer or virus sends out mail with forged envelope sender

MAIL FROM:<innocent-user@example.com>
RCPT TO:<target@target-domain.com>

- The message is accepted by some intermediate mailer, and later bounces (e.g. non-existent recipient, user over quota, virus detected)
- The bounce goes to <innocent-user> who had nothing to do with it

Difficulties with blocking joe-job bounces

- All bounces have empty envelope sender, MAIL FROM:<>
  - Not any use for filtering
- Joe-job bounces are genuine MTA bounces - just not to messages that *we* sent
  - content filtering to identify a bounce doesn't help
- Discarding all bounces is definitely not an option
  - Many users mistype E-mail address
  - Often mailboxes are down or over-quota
  - The bounce is the only way the user knows that something bad happened

We need to associate bounces with messages *we* sent

- Unfortunately, bounce messages are not standardised in a way which allows this
- The only thing we can rely on is that the bounce goes to the MAIL FROM address
- So, one solution is to rewrite the MAIL FROM address to a secret value which changes every day or so: known as Variable Envelope Return Path (VERP)

MAIL FROM:<username=**ac7933dc**@example.com>

Advantages of VERP

- Good bounces are kept, bad bounces discarded
- A cryptographic "cookie" is very difficult for spammers to guess
- Hard for spammers to collect envelope senders
  - They might appear in Return-Path: headers on mailing list archives
  - If widely adopted, mailing lists will strip this header
  - Even if they do collect them, valid for a few days only

Disadvantages of VERP

- Could interact badly with mailing lists and other people's whitelists (if they look at MAIL FROM rather than the From: header)
- Interoperability problems could be minimised if there was an agreed standard for the address format, but there isn't
  - One is called "SRS" (Sender Rewriting Scheme) but there are others
- Must force your users to send outgoing mail through *your* mailserver
  - Otherwise the cookie won't be added and they will lose bounces

Disadvantages of VERP (2)

- Generates long left-hand sides on E-mail addresses; RFC2821 only requires mail servers to accept up to 64 characters
- Doesn't stop any spam, except spam sent with a null envelope sender MAIL FROM:<>

Exim implementation of SRS

- http://www.infradead.org/rpr.html
- Requires a "shared secret" on all your mail servers
  - On the outgoing servers: to add a valid cookie
  - On the incoming servers: to check the cookie for bounces, and discard bounces which do not have a valid cookie
- Stay out of heated discussions on related issues like SPF!

Minimising the joe-jobs we relay

- We don't want to accept a mail and then bounce it later; that means we're sending the joe-job to some unfortunate victim
- We prefer to reject messages at the RCPT TO or DATA stage of the SMTP session - it is then the sender's job to bounce, not ours
  - Exim: reject in the ACL
- For content filtering we have to reject at DATA, but if the mail has multiple recipients, that bounces it for *all* of them (makes separate opt-in/opt-out difficult)

We could just accept the message and discard it silently

- If a message is rejected because it's spam or a virus, don't send a bounce
- Risky strategy for false positives: if a rejected mail is actually good, then neither the sender nor the recipient will have any notification that delivery did not occur
- Which is worse: lots of joe-job bounces or occasional false positives?
  - joe-jobs annoy random third-parties, but false positives affect our own customers and the people they communicate with

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

Consider outsourcing

- There are companies which will handle the whole thing for you
  - Example: www.messagelabs.co.uk
- Point your MX records at their servers; they filter for spam and viruses, and forward the cleaned mail to your servers
- No investment in hardware, software, ongoing management and maintenance
- Maybe more cost-effective for smaller organisations