

Handling unwanted email

Almost entirely based on a presentation by
Brian Candler

What are the main sources of junk email?

- Spam
 - Unsolicited, bulk email
 - Often fraudulent – 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
 - These are called “collateral spam” or “Joe-jobs”
 - Junk mail is sent with forged MAIL FROM
 - Accepted by some intermediate MTA, but later it bounces
 - Bounces go to innocent third party

What are the costs?

- Important messages can be accidentally discarded
The more junk, the higher the risk
- Wasted time
Deleting junk
Setting up and maintaining filters
Checking discarded mail for false positives
- Wasted bandwidth and disk space
Especially for users on modems
Viruses and spam attachments can be large
- Annoyance, offence, even fraud

There are no easy answers!

Where can you filter?

- At the end-user hosts
 - ✓ Each client has full control and customization
 - ✓ Distributes the processing cost
 - ✗ Client must still download each message
- On the ISP's mail server
 - ✓ Easier for users
 - ✓ Sometimes can be rejected before receiving the body
 - ✓ Saves disk space on the server
 - ✗ Hard to make flexible for users to customize

The Joe-job problem

- Don't accept a message and then bounce it later
If its sender is forged, we are creating a Joe-job
- Much better to reject at RCPT TO or DATA stages
A real MTA sender will create a bounce
Spamware will ignore the rejection
- For content filtering, we have to reject at DATA time
If there are multiple recipients, that rejects it for all
This makes individual opt-in/opt-out difficult
- What about accepting and just discarding junk?
Risky because of false positives
If a real message is rejected by mistake, nobody knows

Legal problems with filtering

- Some customers may be upset that you are making value judgements on their mail, or looking at the contents
- Make sure your customer contract 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 are not liable for cases where the filter makes the wrong decision

Viruses in email

- The volume of virus mail is now huge
 - It is amazing how innocent some users are
 - Cambridge University rejects 80% of the email it is offered
 - See <http://canvas.csi.cam.ac.uk/stats/ppsw/index.html>
 - That excludes spam, which is tagged, not rejected
- Like spam, current viruses have forged senders and headers
- Naive implementation blocks all executable extensions
 - Can block some legitimate messages
 - Some viruses come in **.zip** files
- The only sure test is to use a virus scanner
 - Commercial solutions are expensive, may charge per-user
 - Free solutions such as *clamav* are pretty good
- New viruses are being written all the time
 - Frequent updating of the signatures is important
 - clamav* can do this automatically

Spam: identifying by source IP address

- As soon as the sending host connects, you know the IP address
- You can check the IP address against “blacklists” in real time
 - Blacklists of IP ranges assigned to known spammers
 - Blacklists of IP addresses of open relays or open proxies
 - Blacklists of IP addresses that have sent spam recently
- There are some public blacklists in the DNS

Using DNS black lists

- Advantages
 - ✓ Easy to configure
 - ✓ DNS lookups are relatively quick and cheap
 - ✓ Somebody else maintains the list
 - ✓ Mail is rejected before the body has been sent
- Disadvantages
 - ✗ Will not catch all spam
 - ✗ Not effective against viruses or collateral spam
 - ✗ The lists come and go (legal threats from spammers)

Which blacklists to use?

- Some are not free
e.g. mail-abuse.org
- Some are not good
Policies are too draconian; you lose mail you want
Someone else's policy may not be good for you
- Try these:
sbl-xbl.spamhaus.org (known spammers and infected hosts)
relays.ordb.org (open relays)

Spam: identifying by content

- Spammers are sad and predictable
- A human can recognize spam very easily
But it's harder to do it automatically
- Look for phrases that typically occur in spam
- Look for phrases type typically *do not* occur in spam
This helps reduce false positives
- The ratio of the two indicates the likelihood of spam
... and how sure we are

Disadvantages of content filtering

- Spammers use many tricks to disguise their spam
MIME base64 encoding, HTML mails, breaking up words, misspelling, etc...
- It is an arms race
As filters evolve, spammers change what they do
- Computationally expensive
- Liable to false positives
Unless rules are customized for each user, but this is hard to do for a server-side solution

Whitelists

- Accept mail only from people we already know
Effective at blocking spam and some viruses
Start-up problem (see next slide)
- Actually, spammers could forge messages so that they appear to come from people we already know
- For now, they don't seem to be collecting information about who we associate with
- But viruses and Trojans often use local address books

Handling mail from people not on the whitelist

- By password: e.g. a magic word in the **Subject:** header
- By content filter: e.g. a low spam score
- By challenge-reponse system
 - Put mail in a hold queue and send back a message
 - If the sender responds, they are whitelisted
- Challenge-response systems are not recommended
 - ✗ Adds to the collateral spam problem
 - ✗ Interacts badly with mailing lists
 - ✗ Some people get very annoyed
 - ✗ Difficult to deploy in a scalable way

Disadvantages of whitelists

- Difficult/annoying for people to contact for the first time
- Difficult for a server-side solution
 - Each user needs a separate list and a way to edit it
 - Automatically whitelisting addresses we send TO isn't easy
- Filtering at the MAIL FROM stage is getting harder
 - Envelope sender may differ from **From:** in headers
 - It could even be different for every message someone sends
- Whitelists do not help with collateral spam (joe-jobs)

Handling unwanted bounce messages

- All bounces have an empty envelope sender
`MAIL FROM:<>`
Not any use for filtering
- Joe-job bounces are genuine MTA bounce messages
...but for messages that we did not send
Content filtering to identify a bounce does not help
- Discarding all bounces is not an option
Many users mistype email addresses
Mailboxes are often down or over quota
The bounce is the only way the user learns of a problem
- Sites the block all bounces are broken
There is a DNS black list that records them
dsn.rfc-ignorant.org

Associating bounces with messages we sent

- Bounce messages are not standardised in a way that allows this
- Only thing you can rely on is that bounces go to MAIL FROM
- One solution is to rewrite MAIL FROM
`MAIL FROM:<user=ac7546dc@example.com>`
- Change the magic value every day or so
- Check that incoming bounces quote a recent value
- If spammers collect the address, it is not valid for long
- Or use a cryptographic “cookie” (very hard to guess)
Work is being done to refine these ideas
- This is not a spam solution; it’s a Joe-job solution
...though it does kill spam sent with MAIL FROM:<>

Disadvantages of rewriting MAIL FROM

- Interacts badly with mailing list software and whitelists
...if they look at MAIL FROM rather than the *From:* header
- This could be lessened if there were an agreed standard
Several proposals are being discussed
- Your users *must* send outgoing email through your MTA
Otherwise the rewriting won't happen and bounces will be lost
- Generates long local parts
RFC2821 requires only 64 characters
- Another possibility is to rewrite the domain instead
This gives you up to 255 characters
But now there are DNS implications

People are trying to find solutions

- BATV (Bounce Address Tag Validation)
A scheme for adding tags to local parts
- CSA (Client SMTP Authorization)
DNS lists which machines are permitted to send email
- SPF (Sender Policy Framework)
Sender-ID is Microsoft's version of SPF
DNS lists which hosts may use which envelope senders
Completely breaks email forwarding
Claims that it will kill all spam are exaggerations
- SRS (Sender Rewriting Scheme)
An attempt to patch up SPF, but not problem-free
- Domainkeys (Yahoo!) or Identified Internet Mail (Cisco)
Digitally sign messages with a per-domain private key
The signature is placed in a header

No proposal is problem free

- Some of these solutions have licence/patent issues
- Some SPF discussion references:
 - <http://david.woodhou.se/why-not-spf.html>
 - http://bradknowles.typepad.com/considered_harmful/2004/05/spf.html
 - <http://homepages.tesco.net/~J.deBoynePollard/FGA/sntp-spf-is-harmful.html>
- And for a bit of light relief:
 - <http://www.rhyolite.com/anti-spam/you-might-be.html>

Many options: what should you do?

- Use DNS blacklists
 - Surprisingly effective
 - Very easy to implement
 - Low maintenance
- Consider implementing virus scanning and content filtering
 - Opt-in users agree to let you do this
 - Just tagging spam lets the user decide what to do
- Think about the resource costs
 - These services are expensive to scale and manage
 - Opt-in users pay extra?
- Advise users about client-side spam filters
 - Bayesian filters and whitelists are more easily handled there
 - Find those that work well with your client's software

Consider outsourcing

- There are companies that will handle the whole thing
Example: **www.message-labs.com**
- Point your MX at their servers
They filter for spam and viruses
They forward only clean mail to your servers
- No investment in hardware, software, management, or maintenance
- May be more cost-effective for small organizations