



metaparadigm

Open Source Mail Sanitation

Increasing the signal to noise ratio
for Internet mail

Jamie Clark <jamie@metaparadigm.com>

Presentation Overview

- ⇒ Background and Definitions
- ⇒ Potential Problems
- ⇒ Open Source Tools
- ⇒ Implementation Details
- ⇒ Results

Characteristics of spam

- ⇒ Definition: UBE or Unsolicited Bulk Email
- ⇒ Uses harvested addresses as well as dictionary-based 'attacks'
- ⇒ Uses random forged sender addresses
- ⇒ Largely originates from custom spammer toolkits
- ⇒ Bounces from filters are capable of being as big a problem as spam itself
- ⇒ It is tricky for a machine to reliably identify spam with the present lack of rigor in internet mail standards

Characteristics of email worms

- ⇒ Worms use forged, often valid, sender address.
- ⇒ All modern worms appear to use their own SMTP engine (to avoid being blocked by well-configured ISP mail relay)
- ⇒ Capable of dictionary attacks for recipients
- ⇒ Bounces from scanners increase the annoyance factor
- ⇒ Relatively easy to detect, with the notable exception of bounces generated by misconfigured virus filters

Challenges:

- ⇒ False positives. Machine-based scanners are not perfect. Very rare for virus scanners but likely for spam filters
- ⇒ Should you notify, flag, or 'blackhole' messages identified as virus or spam
- ⇒ If notification is desired; then who to notify
- ⇒ Implementation should follow policy

MUA filtering vs MTA filtering

- ⇒ MUA (Mail User Agent) can automatically mark, move, or delete emails but still requires user intervention to either delete marked messages or search junk folder for false positives. Auto-delete is dangerous
- ⇒ MTA (Mail Transfer Agent) can mark or quarantine bad emails similar to MUA. The MTA can also take measures during SMTP conversation (eg greylisting) not possible with MUA. Messages that can be rejected at SMTP time do not necessarily require notification.



Definition: Bounce vs Reject

Bounce

- ⇒ Mail server accepts the message at the initial SMTP conversation, then identifies message as bad, then creates and sends a notification to the (probably forged) sender
- ⇒ Common form of notification
- ⇒ Amplifies the problem, considered by some (on the receiving end of the bounce) to be misconfiguration



Definition: Bounce vs Reject

Reject

- ⇒ Mail server terminates the initial SMTP conversation without accepting the message
- ⇒ If the SMTP peer is a worm or spam engine then no further traffic is generated
- ⇒ Requires MTA with inline filtering interface
- ⇒ <http://spamlinks.openrbl.org/filter-bounce.htm>



Default delivery

- ⇒ Mail delivery based upon wildcards
- ⇒ * @ xyz.com -> foof @ xyz.com
- ⇒ Defeats sender verification
- ⇒ Will collect significantly more spam
- ⇒ Will be bombed by dictionary attacks
- ⇒ To be avoided if possible

Problem: Backup Mail Exchangers

- ⇒ Targeted by spammers, perhaps to sneak inside a whitelist, or to avoid greylisting or other SMTP conversation checks
- ⇒ Possibly outside your control
- ⇒ If used, then should have the same anti-virus and anti-spam measures in place
- ⇒ Preferably uses LDAP routing to avoid being a source of 'User Unknown' bounces



Some ideas for 'best practice'

- ⇒ Don't accept mail you can't or won't deliver. This will help you with the next point
- ⇒ Avoid sending bounce messages
- ⇒ Don't be tempted to block or blackhole DSN messages. You might mask a real problem. Take care when filtering these addresses
- ⇒ Do reject or silently quarantine mass mailing worms. Notifications usually only confuse or annoy recipients
- ⇒ Try to have your backup mail servers configured to reject invalid recipients with 'User Unknown'



Virus Detection: ClamAV

- ⇒ <http://www.clamav.net/>
- ⇒ GPL anti-virus toolkit for UNIX
- ⇒ On-access scanning (clamuko and dazuko)
- ⇒ Supports archive files (zip, tar, rar, etc), mail, OLE2
- ⇒ Milter interface
- ⇒ Can be configured to auto-update signatures from mirror sites. Virus signatures are signed by project coordinator's keys
- ⇒ Commercial support available



Spam Detection

- ⇒ Message checksum counting and matching
 - DCC
 - Razor
- ⇒ Greylisting
- ⇒ Sender validation
 - SPF
 - Milter-sender
- ⇒ Blackhole and dialup lists
- ⇒ Heuristics
 - SpamAssassin

Message counting / matching

DCC

- ⇒ <http://rhyolite.com/anti-spam/dcc/>
- ⇒ Distributed Checksum Clearinghouse
- ⇒ Breaks messages into components (header, body) and uses English dictionary plus magic algorithm to compute several digests (currently 3)

DCC (continued)

- ⇒ Digests are sent to nearest DCC server (based upon round-trip query time) where digest counts are accumulated and periodically flooded to other servers
- ⇒ Client receives a message count from server and uses this to determine if message is 'bulk'
- ⇒ Messages collected by spam traps are immediately flagged as bulk



DCC Pros and cons

⇒ Pros

- In theory, extremely unlikely to flag a one-to-one (or one-to-few) email as bulk
- Negligible false-positive rate for personal emails

⇒ Cons

- Can only determine the 'B' in UBE. Solicited bulk email (eg mailing lists) must be manually whitelisted
- Mutating spam can defeat digest matching
- Low hit rate



Razor

- ⇒ <http://razor.sourceforge.net/>
- ⇒ Computes message digests using magic algorithms
- ⇒ Queries Razor servers for spam status
- ⇒ Differs from DCC in that messages must be submitted by human to achieve spam status
- ⇒ Submitters must become trusted before their spam notifications carry any weight

Razor: Pros and cons

⇒ Pros

- In theory, will only ever flag real spam as spam

⇒ Cons

- Mutating spam content can defeat digest matching
- Low hit rate

Greylisting

- ⇒ General term defining a mechanism to defeat 'one-shot' spam engines
- ⇒ Local MTA records triplet of remote IP address, sender, and recipient during SMTP conversation. If the triplet is not found in greylist then local MTA returns temporary failure indication to remote MTA. Adds triplet to greylist.
- ⇒ When remote MTA resends the message after the configured retry interval - the message is accepted.
- ⇒ The greylist entry remains cached for a configurable period - often 1 day
- ⇒ Backup MX servers must be configured to share the greylist database



Greylisting: Pros and cons

⇒ Pros

- Does not rely on message content
- Greylist remains 'hot' for frequent correspondents, particularly if cache time is extended to 1 week.
- Effective. Zero false positive rate. Blocks a lot of spam.

⇒ Cons

- Delays some inbound email
- Delay depends on remote MTA configuration and can not be controlled
- Can be defeated if your backup MX does not also greylist



Sender Validation

- ⇒ Milter-sender. Milter plugin that uses standard SMTP protocol to validate sender.
- ⇒ SPF (Sender Policy Framework) uses supplemental DNS records
- ⇒ Microsoft Sender ID. Portions are patented. Patent requires license
- ⇒ Client certificates

Milter-sender

- ➔ <http://www.milter.info/milter-sender/index.shtml>
- ➔ Attempts to verify the standing of the sender's email address
- ➔ Goes through the motions of sending a DSN (bounce message), but does not actually send
- ➔ Rejects (permanently or temporarily) the inbound message, giving the sender's status back to the sender's MTA.



Milter-sender in action

- **Permanent rejection if sender's address is definitely invalid:**

```
220 moof.zeroth.org ESMTP Sendmail 8.13.1/8.13.1; Fri, 29 Oct 2004 10:55:18 +0800 (SGT)
HELO pearl
250 moof.zeroth.org Hello cm173.sigma193.maxonline.com.sg [218.212.193.173], pleased to meet you
MAIL FROM: <evilspammer@hotmail.com>
550 5.7.1 <evilspammer@hotmail.com>... MX 5 'mx1.hotmail.com.' [64.4.50.99] for
    <evilspammer@hotmail.com> rejected address saying "Requested action not taken: mailbox unavailable"
QUIT
221 2.0.0 moof.zeroth.org closing connection
```

- **Automatic greylisting if sender's address can not be completely verified:**

```
220 moof.zeroth.org ESMTP Sendmail 8.13.1/8.13.1; Fri, 29 Oct 2004 11:00:01 +0800 (SGT)
HELO pearl
250 moof.zeroth.org Hello cm173.sigma193.maxonline.com.sg [218.212.193.173], pleased to meet you
MAIL FROM: <imweasel@yahoo.com>
250 2.1.0 <imweasel@yahoo.com>... Sender ok
RCPT TO: <jamie@metaparadigm.com>
450 4.7.1 <jamie@metaparadigm.com>... from <imweasel@yahoo.com> via [218.212.193.173] to
    <jamie@metaparadigm.com> denied for 600 seconds
QUIT
221 2.0.0 moof.zeroth.org closing connection
```

Milter-sender: Pros and cons

⇒ Pros

- Not dependant on message content
- Very effective

⇒ Cons

- Will reject mail if sender's address is invalid
(perhaps you want this)
- Will reject mail from people with misconfigured mail servers



Blackhole / dialup lists

- ⇒ RBL, DUL, SORBS, SpamCop, NJABL, etc
- ⇒ Check if the sender MTA address is a known source of spam
- ⇒ Usually work on DNS lookup within special reverse-IP zones

RBL: Pros and cons

⇒ Pros

- Useful indicator for boosting confidence that message is spam

⇒ Cons

- Not effective as a standalone solution
- False positives may be likely
- Can be prone to abuse, cranky personalities

Heuristics: Spamassassin

- ⇒ <http://spamassassin.apache.org/>
- ⇒ Performs a wide range of tests
 - DCC, Razor, Pyzor
 - Various RBL lookups
 - Header and message body tests
 - Bayesian self-learning
- ⇒ Uses genetic algorithm (v2.x) or neural algorithm (v3.x)
- ⇒ If it looks like spam, smells like spam, tastes like spam, and it comes from a spam source - then it is probably spam

SpamAssassin: Pros and cons

⇒ Pros

- Runs just about every test under the sun
- Per-recipient rejection or flagging thresholds
- Extremely tunable

⇒ Cons

- Extremely tunable
- MUA forgery tests are prone to false-positives if rule base is not kept up-to-date
- Slow



Implementation

- ⇒ Sendmail + milter
- ⇒ LDAP routing
- ⇒ Milter-sender
- ⇒ ClamAV: clamd and clamav-milter
- ⇒ DCC: dccm
- ⇒ Spamassassin: spamd and spamass-milter

Sendmail

- ⇒ Use at least version 8.12.9 (currently 8.13.1)
 - Debian sarge has 8.13.1-15
- ⇒ If building from source then make sure milter, LDAP, SASL are enabled if you want to use them
 - Debian sarge package has the required options. Check your favourite distro
 - FreeBSD need to use sendmail-ldap port if you want LDAP
- ⇒ Configuration file is typically `/etc/mail/hostname.mc`

LDAP Routing

⇒ In sendmail.mc:

```
FEATURE(`ldap_routing', `ldap -l -T<TMPF> -v mailHost -k (&(objectClass=inetLocalMailRecipient)(mailLocalAddress=%0))',  
  `ldap -l -T<TMPF> -v mailRoutingAddress -k  
  (&(objectClass=inetLocalMailRecipient)(mailLocalAddress=%0))', `passthru', `strip')  
  
define(`confLDAP_DEFAULT_SPEC', ` -h gort.metaparadigm.com -b dc=metaparadigm,dc=com')  
  
LDAPROUTE_DOMAIN(`metaparadigm.com')  
  
define(`ALIAS_FILE', `ldap:-k (&(objectClass=mailForwardingAlias)(mailAlias=%0)) -v  
  mailForwardingAddress,/etc/mail/aliases')
```

⇒ Example LDAP entry

```
dn: uid=jclark,ou=people,dc=metaparadigm,dc=com  
mail: jclark@metaparadigm.com  
mailHost: gort.metaparadigm.com  
mailRoutingAddress: jclark@gort.metaparadigm.com  
mailLocalAddress: jclark@metaparadigm.com  
mailLocalAddress: jamie@metaparadigm.com  
mailLocalAddress: jamie@yeah.la
```

Clamd + clamav-milter

- ➔ Install standard ClamAV package. Example clamav-milter arguments:

```
/usr/sbin/clamav-milter --quiet --local --outgoing --max-children=50 /var/run/clamav/clmilter.sock
```

- ➔ Enable stream save to disk for archive scanning (clamav.conf):

```
StreamSaveToDisk
```

- ➔ Example sendmail.mc configuration:

```
INPUT_MAIL_FILTER(`clmilter',`S=local:/var/run/clamav/clmilter.sock, F=, T=S:4m;R:4m')
```



Milter-sender

- ⇒ <http://www.milter.info/milter-sender/index.shtml>
- ⇒ Edit `milter-sender.cf` to suit your policy and then run.
- ⇒ Sendmail.mc configuration:

```
define(
    `confMILTER_MACROS_CONNECT', confMILTER_MACROS_CONNECT`,
    p, {client_addr}, {client_name}, {client_port}, {client_resolve}'
)

define(
    `confMILTER_MACROS_HELO', confMILTER_MACROS_HELO`, {verify}'
)

INPUT_MAIL_FILTER(
    `milter-sender',
    `S=unix:/var/spool/milter-sender/socket, T=C:1m;S:30s;R:6m;E:1m'
)
```

DCC: dccm

- ⇒ Install dcc package
- ⇒ Disable dccd and run dccm (milter interface)

```
DCCD_ENABLE=off  
DCCM_ENABLE=on  
GREY_ENABLE=off
```

- ⇒ Set DCCM_ARGS to do nothing:

```
DCCM_ARGS="-SHELO -Smail_host -SSender -SList-ID -a IGNORE"
```

- ⇒ Sendmail.mc configuration:

```
FEATURE(dcc)
```

Spamassassin spamd + spamc

- ⇒ Install spamassassin package. Configure and run spamd daemon.
- ⇒ Spamd daemon does not have to run on same host as mail server.
- ⇒ Example arguments with per-user configuration store:

```
/usr/bin/spamd -a -u spamc -x --virtual-config-dir=/var/spamassassin/%u -c -d -A 192.168.10.10 -r /var/run/spamd.pid
```



Spamassassin: milter interface

- ⇒ On MTA machine install spamass-milter or milter-spamc package.
- ⇒ If Spamassassin is installed on other machine you may need local install also; for the spamc client
- ⇒ Example spamass-milter arguments:

```
/usr/sbin/spamass-milter -f -m -r 5 -u nobody -p /var/run/spamass-milter.sock -- -d 192.168.10.10
```

- ⇒ Sendmail.mc configuration:

```
INPUT_MAIL_FILTER(`spamassassin', `S=local:/var/run/spamass-milter.sock, F=, T=C:15m;S:4m;R:4m;E:10m')
```

Testing: GTUBE and EICAR

- ➔ Generic Test for Unsolicited Bulk Email
- ➔ <http://spamassassin.apache.org/gtube/>

XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X

- ➔ Eicar test virus - in case you don't have one handy
- ➔ <http://www.testvirus.org/>

Results: typical week day

⇒ Rejected

- 'User Unkown': ~1000
- Milter-sender permanent reject: ~150
- Milter-sender tempfail: ~200 (greylisted)
- Spamassassin: ~15
- ClamAV virus/worm reject: 1-5

⇒ Bounced

- None

⇒ Delivered

- Spam: ~5
- Ham: ~1300

