Spam Tools That Work

Barbara Dijker
Netrack, Inc.
barb@netrack.net

The Problem - Volume

- So-called spam now accounts for 25-50% of all email traffic on the backbones, depending on whom you believe.
- For some users, it can be 50-90% of their mailbox.
- Efforts at the source seem to have not been effective, or the spammers are more effective at working around them.

The Problem - Techniques

- Spam & Run: messages are usually not sent from the same source more than once.
- Spammers are aware of common filters techniques and work around them.
- They're finding news ways to obscure source information (e.g., web proxies).

Solutions at the Destination

- If ISPs can't stop spam from being sent, we have to manage it better on the receiving end.
- Existing tools include:
  - Blackhole Lists
  - Checksum & Signature Databases
  - Content Analyzers
  - Composite Analyzers

Terminology Identifying Spam

- False negative – a message that is identified not to be spam, but really is
- False positive – a message that is identified to be spam, but really is not

Blackhole Lists

- A Blackhole List is a list of IP addresses
- What IP addresses are on the list depends on the purpose of the list and the list policies
- Lists can be populated by scans or nominations or a combination of both
- Some lists verify entries with tests
- Some automatically remove hosts
- Can be used by a mailer to refuse connections from that host
- Lookups usually implemented in DNS

Blackhole Lists

- Pros
  - Blocks spam from known confirmed sources, relays, or improbable hosts
  - Fast
  - Can be used to block before accepting message in smtp session
- Cons
  - Blocks all mail from a site, not just spam
  - Can't be used to block until spam has been sent/relayed from that source and reported
  - Easy to defeat with spam & run

RBL+ at mail-abuse.org

- RBL+ includes RBL, RSS, and DUL
  - Realtime Blackhole List – spam source hosts
  - Relay Spam Stopper – open mail relays
  - Dialup User List – dialup host address blocks
- Costs money now for anyone except individuals
- Well-managed, only confirmed hosts
- Hard to get on, easy to get off
- Low false positives relative to other lists

Other Black lists

- Each list is different
- Mostly list open mail relay hosts because they can be pro-actively sought out and tested via automated tools
- List accuracy is only as good as the testing tools, the list maintainer and his/her policies, or lack thereof
- At least 30 or more exist
- Go to relays.osirusoft.com and search on an IP address for a listing of all the blacklists

Checksum & Signature Databases

- They contain a unique value for a mail message that depends on the msg content
- If the content is different, so is the value
- What content is used and how much it has to change for the computed value to change depends on the database
- Used to compare the computed value of a new message against the value of known spam messages to identify spam
- The spam determination is usually made by counting reports for the same value

Checksum & Signature Databases

- Pros
  - False positive rate can be very low
  - Fast
- Cons
  - Only as good as the algorithm and verification as spam
  - Can't be used to block spam until it has been received by those reporting and verifying
  - May be easy to defeat with tools that customize or serialize content per recipient – algorithm has to evolve

Vipul's Razor

- Razor.sourceforge.net
- "Vipul's Razor is a distributed, collaborative, spam detection and filtering network. Through user contribution, Razor establishes a distributed and constantly updating catalogue of spam in propagation that is consulted by email clients to filter out known spam. Detection is done with statistical and randomized signatures that efficiently spot mutating spam content. User input is validated through reputation assignments based on consensus on report and revoke assertions which in turn is used for computing confidence values associated with individual signatures."
- Not a checksum but a statistical similarity

The DCC

- Distributed Checksum Clearinghouse
- www.rhyolite.com/anti-spam/dcc/
- Fuzzy and evolving checksum algorithm
- Locally grown

Content Analyzers

- A filter that analyzes at the content (and possibly the headers) of mail messages to determine if it is spam
- Analysis methods varies widely
- Often require training
- Training provides the tool with good vs. bad data to build a comparison base

Content Analyzers

- Pros
  - Can identify possible spam before it is reported to a database
  - Slower than simple lookup tools like Blackhole Lists and Checksum Databases
- Cons
  - Required training can be extensive
  - Effectiveness is related to degree to which spam content differs from legitimate content
  - That difference can vary significantly from user to user

Bogofilter

- Bogofilter.sourceforge.net
- Status based on statistical analysis of "good" and "bad" words in content
- Requires significant training (1000 or more messages)
- New and hotly debated, evolving, considered "promising"
- Others: www.paulgraham.com/filters.html

Composite Analyzers

- Each method described so far has its shortcomings
- Why not employ a combination of mechanisms to identify spam?

Composite Analyzers

- Pros
  - No single test alone will identify a message as spam – accumulated score is compared to a threshold
  - Results can be optimized by combining approaches
  - Most effective in most cases
- Cons
  - Slower, more resources required
  - Higher false positive rate than checksums alone

Spam Assassin

- www.spamassassin.org
- Includes:
  - Complex rule-based content analyzer
  - RBL lookups (subscription required)
  - Support for Razor, DCC, & Pyzor checking
- No training required
- False positives can be mitigated by adjusting threshold and whitelists
- Can be installed site-wide and customized per user

How to use Spam Assassin

- It's a perl module that includes a simple filter program and client/server pair
- Assigns a score based on rules that match
- The score is compared against a set threshold to decide if a message is spam
- Headers are modified to include information about the results of the tests
- Use the headers to filter your messages

Installing Spam Assassin - user

- Download and unpack tar ball
  - perl Makefile.PL PREFIX=~
  - make
  - make install
- You get
  - ~/bin/spamassassin
- Edit $DEF_RULES_DIR and “use lib” path
- Put your preferences in ~/.spamassassin/user_prefs

Using Spam Assassin - user

- Filter mail through spamassassin
- Examples:
  - .forward (not recommended)

    ~/bin/spamassassin >/var/mail/barb
  - .procmailrc

    PATH = ~/bin

    :0fw

    * < 256000

    | spamassassin

Example spam message headers

```
Subject: UCE: Rates at record lows
X-Spam-Status: Yes, hits=10.3 required=5.0
       tests=AWL,INVALID_MSGID,LOCAL_HTTP,LOCAL_IMG,
            MIME_HTML_NO_CHARSET,SPAM_PHRASE_03_05,WEB_BUGS
       version=2.41
X-Spam-Flag: YES
X-Spam-Level: **********
X-Spam-Checker-Version: SpamAssassin 2.41 (1.115.2.8-2002-09-05-exp)
X-Spam-Report:   10.30 hits, 5 required;
  *  1.2 -- Message-Id is not valid, according to RFC 2822
  *  2.0 -- BODY: Image tag with an ID code to identify you
  *  0.1 -- BODY: Spam phrases score is 03 to 05 (medium)
           [score: 3]
  *  3.0 -- BODY: Body has img tag
  *  0.2 -- BODY: Body has http://
  *  1.4 -- RAW: Message text in HTML without specified charset
  *  2.4 -- AWL: Auto-whitelist adjustment
```

SA Preferences

- ~/.spamassassin/user_prefs

  required_hits 5
  rewrite_subject 1
  subject_tag UCE:
  use_terse_report 1
  report_header 1
  spam_level_stars 1
  defang_mime 0

Installing Spam Assassin - site

- Latest version has PERL prerequisites
  - File-Spec, PodParser, HTML-Parser
- For PERL wonks
  - perl –MCPAN –e shell
  - o conf prerequisites_policy ask
  - install Mail::SpamAssassin
- Otherwise
  - Unzip and untar modules needed
  - perl Makefile.PL
  - make install

Installing Spam Assassin - site

- What you get is:
  - Mail::SpamAssassin perl module in perl lib
  - /usr/bin/spamassassin
  - /usr/bin/spamd and /usr/bin/spamc
  - /usr/share/spamassasin rule set
- Rule set is updated regularly and tested against a large batch of spam – so you'll want to keep your install up-to-date
- Distribution includes statistical results of rule set tests in the rules dir

Installing Spam Assassin - site

- Don't change the rules in /usr/share/spamassassin
- Create site preferences and rules in /etc/mail/spamassassin/*.cf
- Create user preferences and rules in ~/.spamassassin/user_prefs
- See /usr/share/spamassassin/STATISTICS.txt for information on hit rate results from the test set of messages

Installing Spam Assassin - site

- Now you need to hook it into your mailer
- Two different ways:
  - Have mailer filter all messages before local delivery (e.g., sendmail milter)
  - Have mailer use procmail for local delivery and users decide to invoke SA
- If the mailer filters the message, the mailer user (root, smmsp) is running spamassassin (or spamc/spamd) – so per-user prefs and auto whitelists DO NOT WORK
- Hooks also exist for Exim and Qmail

Spamass Milter vs. Procmail

- Spamass-Milter
  - New code, not too stable, known bugs like crashes dealing with simultaneous messages
  - Maintainer not settled
  - Usually requires recompile of sendmail
  - Fewer processes because it is a socket to spamc rather than a process
  - Per-user SA features don't work because spamassassin doesn't know who the user is
- Procmail
  - Proven code, been around and in use forever, stable
  - Usually already installed and easy to add to mailer
  - Requires particular permissions on home directories

Spamass Milter

- A sendmail mail filter (milter) that invokes spamassassin
- savannah.gnu.org/projects/spamass-milt
- Need to
  - Recompile sendmail
  - Build spamass-milter
  - Add spamass-milter to sendmail.cf
  - Start spamass-milter
  - Start spamd
  - Restart sendmail
  - Modify startup scripts

Installing Spamass Milter

- Recompile sendmail (8.12+) w/milters
  - In devtools/Site/site.config.m4 add

    APPENDDEF(`conf_sendmail_ENVDEF'), `-DMILTER')
  - Run `Build –c install' not just Build
- Build and install spamass-milter
  - ./configure; make install
  - Requires libmilter.a in sendmail src for link
- Add to sendmail mc (example)

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

Sendmail Milters – outgoing mail

- Caution
- Sendmail milters are applied to all messages coming in via SMTP
- Outgoing email often comes into the mail server via SMTP from client workstations
- So spam filters via spamass-milter are then applied to outgoing email too

Spamd and Spamc

- For site-wide use to save resources, run spamd and have users (or mailer) run spamc instead of spamassassin
- Spamass-milter uses spamc
- Spamd options you'll want
  - -a Use auto-whitelists
  - -d Run as a daemon
  - -u user Run as that user rather than root
- You might need –A to tell spamd what its own IP address is

Procmail to spamd - site

- This is the preferred method:
  - Allows per-user prefs and auto-whitelists
  - Spamd does setuid to the user who called it
- Make sure procmail is installed
- Rebuild sendmail.cf with
  - Mailer(procmail) instead of Mailer(local)
- Provide a template .procmailrc file in /etc/skel from Spamassassin distribution
- Gives full user control to disable or customize
- Spamd MUST run as root to setuid

Spam Assassin Local Configuration

- Site-wide local configs go in /etc/mail/spamassassin
- Recommended options same as for user

  required_hits 5

  rewrite_subject 1

  subject_tag UCE:

  use_terse_report 1

  report_header 1

  spam_level_stars 1

  defang_mime 0

Example spam message headers

```
Subject: UCE: Rates at record lows
X-Spam-Status: Yes, hits=10.3 required=5.0
       tests=AWL,INVALID_MSGID,LOCAL_HTTP,LOCAL_IMG,
            MIME_HTML_NO_CHARSET,SPAM_PHRASE_03_05,WEB_BUGS
       version=2.41
X-Spam-Flag: YES
X-Spam-Level: **********
X-Spam-Checker-Version: SpamAssassin 2.41 (1.115.2.8-2002-09-05-exp)
X-Spam-Report:   10.30 hits, 5 required;
  *  1.2 -- Message-Id is not valid, according to RFC 2822
  *  2.0 -- BODY: Image tag with an ID code to identify you
  *  0.1 -- BODY: Spam phrases score is 03 to 05 (medium)
           [score: 3]
  *  3.0 -- BODY: Body has img tag
  *  0.2 -- BODY: Body has http://
  *  1.4 -- RAW: Message text in HTML without specified charset
  *  2.4 -- AWL: Auto-whitelist adjustment
```

Example clean message headers

- If the message hits are below the threshold, only status and level are included

```
X-Spam-Status: No, hits=0.6 required=5.0
       tests=AWL,NO_REAL_NAME,SPAM_PHRASE_02_03
       version=2.41
X-Spam-Level:
```

What Next?

- Use SpamAssassin headers to filter your mail.
- Filtering depends on your mailer.
- If local delivery is procmail, then procmail can refile or delete, e.g.,

```
#Delete if score is 12 or higher
:0:
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\
/dev/null
#Refile if score is 8 or higher
:0:
* ^X-Spam-Level: \*\*\*\*\*\*\*\*\
Spam-folder
```

Adding white and black lists

- In local.cf or user_prefs
- Examples

```
whitelist_from support@netrack.net
whitelist_from *@security-focus.com
whitelist_to whiny-user@netrack.net
blacklist_from *@hotmail.com
blacklist_to *friend*
```

Customizing SA rules

- All files in /etc/mail/spamassassin are read in alpha order
- Can override or add rules
- Be extremely careful and fully test new local rulesets

Examples rules

```
#
spamphrase 500 risk free
spamphrase 500 save big
#
score NO_REAL_NAME 1.5
#
body L_OFFER  /(?:special|awesome|cool|limited|one).{1,30}offer/i
describe L_OFFER        You can’t miss this offer
score L_OFFER           2.5
#
body MORTGAGE_RATES         /mortgage.{1,76}(?:loan|rate)/i
```

Adding Checksum Tests

- Need to download and install software for the individual checksum system
- SpamAssassin will use it if it is installed
- Supported in SpamAssassin:
  - Razor
  - DCC
  - Pyzor
- Adjust weight for them by overriding score, e.g.,
  - score DCC_CHECK 5.0

SpamAssassin False Positives

- Real mail will get tagged: false positive
- Rarely will real mail score >15
- False positives are usually mailing lists that contain disclaimers and remove instructions and contain spam-like content, e.g.,
  - HTML mail
  - Marketing or pornography lists
- Use white_list feature both site-wide and per user to mitigate

SpamAssassin False Negatives

- SA is 97% effective out of the tarball
- Adding checksums increases effectiveness
- Still 95% effective with 0.3% false positives at threshold of 7
- Hard to get:
  - Messages containing one img tag where the image is the content
  - Extended and in-direct prose

Statistics

- SpamAssassin tests with over 232K msgs
- Threshold 5 – FP 0.55%, FN 2.87%
- Threshold 7 – FP 0.30%, FN 4.16%
- Threshold 10 – FP 0.15% FN 7.41%
- Threshold 15 – FP 0.05% FN 13.11%

Cases

- My Mailbox
- Before
  - >50% spam, over 100 messages/day
- After
  - 54.7% of incoming mail deleted by procmail as spam (threshold 6)
  - Deleted messages are logged by procmail no FP
  - Fewer than 3 spam messages/day to Inbox
- That's with no checksum checks no whitelists – just some rule massaging

Cases

- Commercial ISP mail server, ~250 boxes
- Threshold 6
- Average spam score = 17.19
- Average clean score = -0.85
- 25.1% of all mail tagged as spam

- Note that some users may not like the privacy implications

Issues

- SpamAssassin does take considerable resources
- Each message takes an extra 1-5 seconds for delivery
- Spammers are always trying to outsmart the anti-spam tools, all of them
- Rules, algorithms, and methods need to be regularly updated or re-trained