Evaluating and Implementing Anti-Spam Solutions

Michael Lamont

Senior Software Engineer
Process Software

Presentation agenda

- Definitions of spam
- Why spam is a problem - and why you should care
- Experiences of three sites
- Common anti-spam technologies
  - How they work
  - How effective they are
  - Strengths/weaknesses
  - How spammers try to sneak messages past them
- Anti-spam software evaluation techniques

Defining spam

- Every person, every site, and every anti-spam product has their own definition of spam
- Definite spam: pornographic, phishing, ...
- Not spam: work email, email from friends, nag-o-grams from your mother, ...
- Grey area: newsletters, mailing lists, unsolicited email from sites like Amazon
- Be sure your anti-spam software has the same definition of spam that you do (or can be configured to)

Spam by the numbers

- Spam being sent on average worldwide (IDC)
  - 4 million in 2001
  - 17 billion in 2004
- Half of all business emails are spam (Time Magazine)
- Productivity cost is $8.9 billion (Time Magazine)
- Revenue for vendors selling anti-spam products will reach approximately $130 million in 2003, and soar by 200 percent in 2004 to $360 million (Ferris)

HP WORLD 2004
Solutions and Technology Conference & Expo

Direct effects of spam

- Wastes network bandwidth
- Wastes CPU and disk on mail server
- Wastes CPU and disk on desktops
- Wastes end users’ time
- Wastes administrators’ time
- Pisses off everyone in general

Indirect effects of spam

- Legal exposure
- “Brand damage”
- Lost productivity over and above the time directly spent dealing with spam
- Increasing downward slide of modern society

How much is spam costing you?

- Lots of complex ROI formulas and whiz-bang web calculators out there - feel free to use them
- ROI factors to consider:
  - How much are you paying employees to deal with spam?
  - How much money are you losing because employees are dealing with spam instead of working?
  - How much is the additional hardware/bandwidth costing you?

Example site: Bio-pharm manufacturer

- Two sites, one on each coast. Two email servers at each site (one primary, one backup)
- 40,000 incoming email messages each day
- About 55% of all incoming mail was spam
- Anti-spam solution couldn’t filter out legitimate mail containing pharmaceutical marketing phrases
- Most email consisted of either technical or business content

Example site: University

- Large public university with 30,000 email accounts on four central mail servers
- 300,000 incoming messages on average weekday, 80,000 on weekends. Diverse content.
- Almost 70% of all incoming mail was spam (public email directory)
- Mail servers were already heavily loaded, so solution had to be lightweight
- Solution had to give students/faculty access to all filtered messages for censorship reasons

Example site: Government agency

- Government agency in Europe with nine mail servers in various locations.
- 75,000 incoming email messages each day
- 35% of incoming mail was spam
- Incoming mail could potentially have content in any major European language
- Solution had to conform to EU rules for exposing and deleting message content

Spam filtering technologies

- Heuristic (rules)
- Bayesian (statistical)
- Signature matching
- DNS blacklisting
- Challenge/response
- Legal
- Retribution

Heuristic: How it works

- Matches rules (usually regular expression based) against the headers and content of an email message
- Simplest heuristic filters just look for bad words
- More complex heuristic filters use hundreds and hundreds of rules to search for features of a message that indicate it is or isn't spam
- Each rule has a different weight, with a larger weight indicating a message is more likely to be spam

Heuristic: How it works

- The weights of every rule a message matches are added together
- If the total weight is greater than a specified threshold, the message is considered spam

Heuristic: Effectiveness

- One of the highest spam detection rate of current filtering methods (90% to 95%)
- Simple implementations tend to have a relatively high false positive rate (0.5%)
- More evolved implementations have an acceptable false positive rate

Heuristic: Strengths and weaknesses

- Excellent accuracy
- Easy to install and maintain
- If a spammer gets his hands on a copy of the software, it’s trivial to circumvent
- Rules have to be updated on a regular basis to catch new spam tricks

Heuristic: Circumvention

- If rules are public (freeware solutions), or even if they’re not, spammers can craft their messages to deliberately avoid detection
- Spammers tend to be lazy, so frequent rule updates discourage this
- Spammers can deliberately word their messages in an attempt to evade detection even without having the rules, but this usually happens at the expense of their message content

HP WORLD 2004
Solutions and Technology Conference & Expo

Bayesian: How it works

- The filter "learns" what you consider spam by looking at large bodies of spam and non-spam messages you present to it
- Basically, the filter uses the frequency of certain words appearing in spam messages to figure out the statistical probability that a message containing those words is spam
- Each word of an incoming message is examined to determine the probability that it indicates the message is spam

Bayesian: How it works

- If the sum of the probabilities of interesting words in the message is above a certain threshold, the message is treated as spam

Bayesian: Effectiveness

- *When trained properly*, a Bayesian filter has almost perfect accuracy
- When the training is done incorrectly, the results will not make people happy

Bayesian: Strengths and weaknesses

- Excellent accuracy (for most people)
- Snorts up CPU and memory like a junkie who needs a fix
- Most implementations aren't suitable for large-scale production use (accuracy suffers badly)
- Requires substantial user education on how to train it properly. Autotraining systems can help alleviate this issue.

Bayesian: Circumvention

- Only sure way to circumvent a Bayesian filter is to avoid the use of a lot of "spammy" words in a message.
- Kind of hard to sell viagra without using the word "viagra", though.
- No known circumvention technique to date has worked
- Spammers have tried sneaking messages by the filter by including large numbers of non-spam words in the message, but most Bayesian filters are smart enough to ignore that

HP WORLD 2004
Solutions and Technology Conference & Expo

Signature matching: How it works

- The anti-spam software vendor sets up a large set of test addresses and uses them as spam bait
- Whenever a test address receives a spam message, the vendor creates a signature for it
- The signatures are a hash of the message headers and body, and (at least in theory) are specific to that message
- The signatures for spam messages go into a giant database, which is pushed out to customer mail servers every few minutes

Signature matching: How it works

- When a message is received by a customer's mail server, the anti-spam software calculates its signature
- The message's signature is compared against the signatures in the database. If it matches, it's treated as spam

Signature matching: Effectiveness

- Very low false positive rate (not quite as low as the vendors advertise, but still very low)
- Low spam detection rate. Despite vendor claims of 99.9% accuracy, 50% to 70% is more accurate.
- Published numbers for the largest vendor of signature matching software give it only 70% accuracy (MIT Technology Review)

Signature matching: Strengths and weaknesses

- Significant numbers of false positives aren't likely to occur
- Relatively low system load
- Low spam detection accuracy
- Very easy to circumvent with modern spamware
- Requires very frequent database updates, with accuracy falling off significantly in a matter of hours if something prevents the updates from occurring

HP WORLD 2004
Solutions and Technology Conference & Expo

Signature matching: Circumvention

- Old signatures are removed from the database quickly, so “old” spam will sail right through
- Simple signature hashing algorithms are easy to beat by adding random text or words to each message
- Vendors come up with new signature generation algorithms all the time, but they’re all easy to beat with modern spamware that makes major modifications to each message

DNS blacklisting: How it works

- When a connection is established to your mail server, the mail server performs a DNS lookup of the remote site against a special DNS server
- The special DNS server is actually a giant database of IP addresses and domains that are known to send large quantities of spam
- Based on the return value from the DNS lookup, your mail server either accepts or rejects the connection

DNS blacklisting: Effectiveness

- Has a relatively low spam detection rate, around 40% for most sites
- Because it requires so little system resources, most sites use it as a first line of defense against spam

DNS blacklisting: Strengths and weaknesses

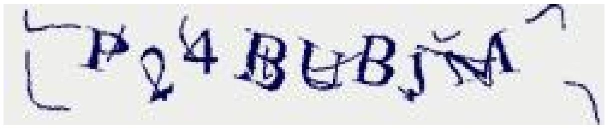
- Allows you to block messages from spam domains without having to even examine the message
- Requires very little system resources
- Has a very low spam detection rate, and can be easily avoided by a savvy spammer
- Good chance you’re going to lose legitimate mail because a legit site accidentally got blacklisted
- You have no control over what sites are blacklisted and which sites are not

DNS blacklisting: Circumvention

- Domains and IP addresses are cheap - easy for spammers to constantly hop around between domains
- Too easy to write a worm/virus that turns desktop systems into “spam zombies”, the sheer quantity of which makes it impossible to keep the database up to date
- If a spammer hacks/spoofs a site you *must* receive mail from, it’ll force you to turn off blacklisting for your domain

Challenge/response: How it works

- When a message is received by the C/R software, it holds the message and sends a challenge message back to the sender
- The challenge message directs the sender to a web site, where they have to pass some sort of test to prove that they're human (rather than automated spamware)
- Most common is distorted text image
- Sample:



Challenge/response: How it works

- If the sender passes the challenge, then the original message is delivered to the recipient
- If the sender doesn’t pass the challenge within a specified period of time, the message is dropped
- Some implementations whitelist a sender who passes the challenge, so future messages won’t require a re-test

HP WORLD 2004
Solutions and Technology Conference & Expo

Challenge/response: Effectiveness

- On paper, this method has a 100% spam catch rate and a 0% false positive rate
- That requires everybody to play by the rules, and since when have spammers done that?
- Reality is that spam catch rate can be 0% if a spammer is smart/lucky, with an unacceptably high false positive rate

Challenge/response: Strengths and weaknesses

- Major strength is that it looks good on paper
- Lots and lots of weaknesses:
  - Can’t deal with mailing lists and automated messages
  - Confuses a lot of senders
  - Easy to circumvent if whitelisting is enabled
  - Unacceptable mail delays
  - Honks off a lot of senders (including me), who won’t do the challenges

Challenge/response: Circumvention

- If you’re using whitelisting, a spammer just has to get lucky and guess an address you might have whitelisted (mailing list, Amazon, travel agency)
- Use porn fiends to solve the challenges (Simson Garfinkel)
- “Rent brains” in developing countries
- Odd twist: spammers are sending out bogus messages that look like challenges. They skate right by most anti-spam software, and either contain a marketing message or direct recipients to a web site that does

Legal: How it (doesn't) work

- "Legit" spammers have a powerful lobby, so most anti-spam legislation is chock-full-o-loopholes
- It's all but impossible to pursue a spammer over national borders...
- ...And there will always be one jurisdiction that welcomes spammer money with open arms
- Most spammers ignore anti-spam laws anyway
- Published numbers indicate less than 15% of sexually explicit spam obeys current FTC regulations (Vircom)

HP WORLD 2004
Solutions and Technology Conference & Expo

Retribution

- Filters that fight back (FFB)
  - Crawl all URLs listed in message, bringing down spamvertized web site, driving up spammer bandwidth costs
- Tar pitting
  - Email server deliberately slows down SMTP transaction, slowing down spammer as well
- Neither one works particularly well, and both have the potential to get IT staff fired

Filtering technology wrap-up

- Heuristic, Bayesian, and DNS blacklisting work
- Signature matching and challenge/response don't
- Anti-spam laws mostly force the quasi-legit mailers to cross over to the dark side
- Retribution, while fun, isn't terribly constructive
- Any one filtering method can be circumvented by a spammer with sufficient time and resources
- An anti-spam solution with multiple filtering methods is the way to go

Evaluating anti-spam software

- Filter evaluation criteria
- User interface evaluation criteria
- Non-production testing methods
- Production testing methods
- Evaluation fallacies
- Soliciting user feedback

Filter evaluation criteria

- Accuracy
- Configurability
- Information
- Filtering methods
- Performance
- Security
- Time required to implement and maintain

Accuracy

- Two key measures of accuracy: spam detection rate and false positive rate
- A lot of poorly written spam filters have high spam detection rates and high false positive rates, and vice versa
- A good solution strikes a balance with a high spam detection rate and a low false positive rate
- Messages identified as spam shouldn’t be immediately discarded - even the best spam filters make mistakes from time to time

Configurability

- Software should be extensively configurable to work with your site, but it should also be effective out-of-the-box so you don't have to spend hours getting it to work
- System administrators should be able to add, delete, and modify filtering rules
- Users should be able to personalize their spam filtering options, if the administrator chooses to allow them to
- Users should not have to install software on their desktops to perform configuration tasks

Information

- Both administrators and users should be able to quickly tell why a message was classified as spam
- Anti-spam software should provide succinct but useful log files with at least one entry for every message examined by the software
- At least basic statistics (number of incoming messages, number of messages filtered, etc) should be provided

Filtering methods

- Anti-spam software that provides only one filtering method should be avoided
- Anti-spam products should use filter methods that provide a rich feature set while balancing accuracy and system resource consumption
- Avoid methods that are easy to circumvent or confuse users, such as signature checking and challenge/response

Performance

- Email is an application that’s highly visible to both internal and external users
- Message processing delays will quickly be noticed, so an anti-spam product should not become a bottleneck
- Anti-spam software should be scaleable, so it can grow as your site grows

Security

- Your site's email messages are private communications that could do serious harm if lost, made public, or given to a competitor
- Messages with sensitive content should not be sent off-site for filtering
- The administrator should have the ability to approve or reject new spam filtering rules before they are put into place
- Anti-spam software shouldn't send any information whatsoever offsite without the administrator's specific permission

Time required to implement/maintain

- Solution shouldn't require more time to manage than the problem
- The administrator should have the option to shove as much administration as possible off to the end users:
  - Filtering thresholds
  - Quarantine preview and release
  - Whitelists and blacklists

User interface evaluation criteria

- Simple and natural dialog
- Natural language support
- Minimize user memory load
- Consistency
- Feedback
- Clearly marked exits
- Good error messages
- Help and documentation

Simple and natural dialog

- Instructions and labels in the interface should be written in a conversational tone
- Jargon or acronyms that would be unfamiliar to end users should be avoided

Natural language support

- The interface has to be able to speak the same language as the users for it to be useful
- Most of the world’s population is at least somewhat functional in English, but the abbreviated language used in some parts of user interfaces may be confusing
- You can’t expect anti-spam software to “speak” all of the world’s languages out-of-the-box, but it should be easy for the system administrator or a translator to rewrite all instructions and labels in the user’s native language

Minimize user memory load

- End users shouldn’t have to remember information specific to the interface between usage sessions
- Interface should be clear and intuitive
- Help should be easily obtainable

Consistency

- The interface should have a consistent layout, color scheme, and text
- Changes between different parts of the interface can disorient and confuse users
- A consistent layout reduces the amount of time new users need to become comfortable using the interface

Feedback

- The interface should provide clear feedback about actions it's taking on the user's behalf
- Example: if the user chooses to release a quarantined message, the interface should clearly state that the message has been released
- Just returning the user to the page they started from might leave them in doubt as to whether or not the message really was released

Clearly marked exits

- Users should be able to exit the interface (logout) from anywhere it makes sense to do so
- User should also be able to return to their main page from anywhere in the interface
- The interface should warn the user about unsaved changes before allowing them to exit

Good error messages

- If an error occurs, the error message should be informative
- A sad face will effectively convey the fact that an error occurred, but it won’t be much help in fixing what’s wrong
- First tier helpdesk staff should be able to tell if a serious error requiring system administrator intervention has occurred
- What caused the error (and what needs to be done to fix it) should be obvious from the error text

Help and documentation

- Help for the user interface should be contained in the user interface
- The average user isn't going to read documentation on how to use anti-spam software, regardless of how pretty the pictures are
- They'd much rather bombard the system administrator with the same question over and over again, which wastes valuable admin time

Non-production testing methods

- Non-production testing has no effect on your live mailstream or production mail server
- Corpus testing: large blocks of known spam and non-spam messages are run through anti-spam software on a test system
- Forking user mail: production mail server forks a copy of incoming messages for select users off to a test system running the anti-spam software

Production testing

- Production testing involves running your live mailstream through an anti-spam product
- Production testing will almost always be visible to end users, so be sure to plan it carefully
- Make sure you choose a good cross-section of your organization to participate in the testing
- Give the test users plenty of warning before the test period starts and before it ends
- Create a mailing list for the test users to post questions/issues to. Have IT staff monitor it.

HP WORLD 2004
Solutions and Technology Conference & Expo

Production testing methods

- Log monitoring: incoming messages are not modified in any way, but the anti-spam software logs whether or not a message would have been considered spam
- Header insertion: insert informational headers in messages it processes
- Subject modification: prepend a token ([SPAM]) to the subject line of messages that are identified as spam
- Full testing: enable the anti-spam software’s full range of spam handling techniques, including quarantining and discarding

Evaluation fallacies

- Using a small group of testers - you need enough to be statistically significant
- Using only testers from one department or workgroup - won’t give a true idea of accuracy or user response
- Fowarding spam - strips off important headers
- Using raw spam from public repositories
- Using homogenous message blocks to test Bayesian filters

User feedback

- Soliciting user feedback at the end of an evaluation is important
- If you don’t ask for your users’ opinions then, you’re going to get them later anyways
- In addition to the obvious questions regarding accuracy, true/false perception questions can be useful in the decision making process:
  - Using *product* would improve my email workflow
  - *Product* would reduce the amount of time I spend dealing with junk email
  - Learning how to use *product* was easy for me

HP WORLD 2004
Solutions and Technology Conference & Expo

HP WORLD 2004
Solutions and Technology Conference & Expo

Co-produced by:

interex
shared knowledge • shared power

encompass
AN HP USER GROUP