How to Un-Spam Yourself

By Aurobindo Sundaram

UNSOLICITED commercial messages ("spam") constitute between 15% and 40% of all e-mail today, and the figure continues to grow rapidly. Spam results in enormous costs to enterprises, in terms of bandwidth usage, the risk of embedded viruses, user productivity impact, and potential litigation due to inappropriate material being stored on enterprise servers. There are several approaches to reduce spam in an organization. Some of these are technology-based, while others are best practices. By combining these, spam can be reduced significantly in the workplace.

HISTORY

Spam[1] is generally defined as unwanted electronic garbage or junk postings. It was named after SPAM, the luncheon-meat-in-a-can, and the Monty Python skit in which the word was repeated incessantly. (The proper spelling is "spam" for junk e-mail and "SPAM" for the meat product).

Spam first entered the computer world as a verb meaning "to break a Multi-User Dungeon" or MUD—a real-time multi-user game such as Dungeons and Dragons—by overrunning its buffers and sending it too much data. Spam then entered the Usenet dialect and took on a much closer meaning to its current one: to send one post to many recipients (for example, posting unwanted messages to hundreds of newsgroups at the same time). Finally, the general public came to use the word to refer to unsolicited commercial e-mail (e-mail advertising services or products that users did not request).

By early 2003, it was estimated that between 17% and 38% of all Internet e-mail was spam, with percentages rising. Research firms say businesses spent approximately $100M on spam prevention in 2002, a number that was expected to double in 2003. The resulting productivity loss totaled $8.9B in 2002[2], and rose to $20B in 2003.

While there does not appear to be a single, effective way to stop spam, we will describe some effective technical and administrative methods to reduce it.

THE ROLE OF MAIL TRANSFER AGENTS

Most large enterprises used to have dozens (sometimes hundreds) of mail servers exposed to the Internet. This configuration makes it difficult to implement consistent policies, and particularly difficult to control the influx of spam. Any anti-spam configuration needs to be deployed and controlled at each of these servers, and the management of this complex system can get out of control quickly. The use of central Mail Transfer Agents (MTAs) that control the flow of all incoming mail (and possibly outgoing mail too) is one effective mechanism to control spam. MTAs allow the enterprise to implement spam control rules at one single logical point—this brings economies of scale when applying server-based spam blocking tools, a Realtime Blackhole List (RBL[3]) configuration, and data mining tools. It is important to note that MTAs do not reduce spam per se, they simply provide a single place to implement the technical mechanisms discussed in the next section.

It is interesting to note that most MTAs now come with hooks to add spam control systems, or with spam control systems built-in (e.g. CipherTrust Ironmail[4], sendmail, etc.). Rather than being an add-on or enhancement, spam and anti-virus controls are being integrated into mail transfer agents. This is a welcome development, as it moves the onus of integration testing on the vendors, rather than the implementers. Implementers are encouraged to make this a requirement when they evaluate mail transfer agents.

TECHNICAL MECHANISMS

Realtime Blackhole Lists (RBL) contain the IP addresses of e-mail servers that have been known to send spam. Most of these servers are configured inadvertently to allow anyone to send e-mail through them; they are called "open relay mail servers" for that reason.

RBL is provided as a service which integrates easily with most mail servers. An enterprise purchases the service, and configures their mail server(s) to lookup the RBL on every mail delivery attempt. If the

sending server is in the RBL, the mail is rejected because it is likely to be spam.

Users often blame MTAs for not blocking all spam, even though spam prevention was touted as a key reason to install them. New open relays are discovered and exploited by spammers all the time, existing relays change their IP addresses and the amount of spam increases exponentially; therefore, the RBL service response is simply not quick enough to keep up. Nevertheless, RBL can be extremely useful in the context of an Internet community effort: as soon as a mail server is reported as delinquent, all organizations that subscribe to the service immediately start blocking mail from that server. It also motivates server administrators to tighten their mail server configurations (lest they be reported as a delinquent server to others). RBL is a low-cost, efficient mechanism to reduce spam—however, it is clear that this solution alone will not suffice.

Spam control services work differently from RBL in several ways. RBL only looks at the address of the sending mail server to decide whether to reject a message, it never looks at the content of the message itself. In contrast, spam control services (such as BrightMail[5]) analyze the entire context of the message, including its contents.

Spam control services claim to reduce spam by doing the following:

- Creating millions of decoy e-mail accounts, all over the Internet, for the sole purpose of attracting spam
- Analyzing e-mail messages sent to these accounts, and creating intelligent filters for them (e.g. matching not only "Viagra," but also "V1agra," "V!agra" and "V I A G R A")
- Pushing these filters out in real-time to enterprise subscribers

This approach is interesting, because it sidesteps the issue of whether a mail server is malicious or not. The vast probes of decoy mail accounts are expected to capture (and allow to be filtered) almost all true spam. Spam control services promise false positive rates (legitimate mail that is incorrectly identified as spam) of only one in a million messages. Their false negative rate (spam mail that is incorrectly classified as legitimate, and therefore will get through) is much higher, typically 10%.

Client filters are typically programs, resident on the user's computer, that filter messages. They operate very similarly to spam control services, but are local rather than global or offsite. The advantage of client filters is that they allow users to personalize them: one user may blacklist the entire "hotmail.com" domain, while another user may not. The issue with using this solution in an enterprise is manageability—deploying and administering the client filters, especially after they have been customized, is not likely to be easy. In addition, several of these products need to be extensively "trained" by their users, particularly when the products use artificial intelligence or Bayesian mathematics instead of explicit rules. This means that the user must provide feedback to the system, over a period of time, leading to improvements in the filtering.

Client filters can be as effective as spam control services, and in some cases, more effective. However, due to the expertise required to deploy and manage them, we do not recommend that most enterprises use them.

Collaborative filtering[6] is a new type of spam control mechanism that works by enrolling in the effort the very people who receive spam. When users running this software receive spam, they simply forward it to a central server, which immediately pushes an alert out to all other users. In the best case, as soon as one individual receives and reports spam, all other recipients (thousands, possibly millions) are immediately protected.

In practice, this is not as easy as it seems. The system depends on the users who receive spam to diligently report it, without getting tired of the process and simply pressing the "delete" key. In addition, the effectiveness of the system depends on the number of subscribers, and whether they receive spam or not. One improvement to this method would be to use "decoy" e-mail accounts that also submit reports automatically to this service. We are now starting to see a convergence of client filters and collaborative filtering, which is more efficient than either by themselves. Again, due to the limited effectiveness of this method, it is not recommended.

New standards[7] for verification of senders and domains in the SMTP protocol are being developed. Some of these are Microsoft's Caller-ID for e-mail, Sender Policy Framework (SPF), and Submitter Optimization. All of these proposals essentially modify the SMTP (mail) protocol by explicitly verifying the sender of the email in some way. Since almost all spam has fake headers/senders, this is a simple way to eliminate spam. However, all these proposals are in the draft technical standards phase with the IEEE. Implementers are urged to ensure that their mail system will support these standards when they are made official.

NON-TECHNICAL MECHANISMS

A significant amount of spam occurs because of configuration errors and risky behaviors of users that cannot simply be fixed with a tool[8]. Enterprises should address these issues by educating users about spam and training them to avoid the following practices:

- Configuring their Internet browsers with their real e-mail address. This can be dangerous because any web site can retrieve this information with JavaScript code and use it to spam the user.
- Similarly, enterprise users should post to newsgroups using free addresses to avoid receiving spam at work.
- Replying to spam or clicking on the link that reads, "Click here to unsubscribe." In general, doing this only confirms to the spammer that the user's address exists and is read by a human. Users should never respond directly to spam, as it may only exacerbate the problem.
- Clicking on links on web sites (especially friends' web sites) that say, "Click here to add an entry to my guest book." Spammers have tools that "crawl" web pages and harvest all e-mail addresses from them.
- Configuring their e-mail clients to show graphics in e-mail. This is a problem because of "web bugs"—customized images with the e-mail address of the user embedded in the URL; when the user follows the link, the e-mail address is immediately confirmed. Users who can configure their e-mail clients not to display images in HTML e-mail should do so. Users should consider creating a separate free e-mail account purely for registration on web sites. If users must post to mailing lists or Usenet lists using their real address, some form of obfuscation is recommended (e.g. writing the address as "userATcompanyDOTcom" instead of "user@company.com"). Research has shown that obfuscated e-mail addresses are almost never targeted for spam.

Legislation[9] to curb spam has been introduced in several countries. In the United States, it has been introduced/passed in a number of states as well as at the federal level. Most of the laws being considered require commercial e-mail to be labeled as such (e.g. with "ADV:" in the subject line), to have a valid return address and to offer a real and easy way to "opt out."

While these are admirable goals, there are several major issues that could impede the usefulness of such laws:

- E-mail knows no boundaries; spammers will move (or outsource) their operations to countries without anti-spam laws, where they will not be subject to prosecution under the laws voted on in their victims' countries.
- It would take significant effort to prosecute spammers, with little financial upside for consumers. However, we are now seeing welcome developments in some states—a spammer was sentenced to hard-time (9 years in jail) for sending thousands of messages through AOL servers in Virginia[10].
- The risk-reward proposition of spam is far too attractive to deter spammers today, although if the trend in the previous paragraph continues, many spammers could be discouraged from risking jail time.

For all these reasons, users should not expect spam to disappear magically after legislation is passed.

CONCLUSION

Clearly, stopping spam is not easy, and there is no panacea. Considering the rapid growth of spam, its impact on enterprise networks and what can be legitimately expected of end-users, we advocate a "defense in depth" system:

- Controls at the mail server level (RBL, spam control services) offer the most manageable and scalable characteristics, and should be applied together, rather than separately.
- User education is critical in defending against spam—users should be taught that their email addresses are personal information, not to be used and distributed without care. Simple steps as documented above will go a long way towards improving the user experience.

In the future, anomaly detection engines (using Bayesian mathematics, for example) will be applied to this problem. Just as with virus detection and intrusion detection, signature-based spam detection is unlikely to be a solution in the long term.

APPENDIX A: SAMPLE REQUIREMENTS FOR ANTI-SPAM IMPLEMENTATIONS

Enterprises may wish to use the requirements below in their own evaluation of potential solutions. We caution the reader that these are not comprehensive. We strongly recommend a central server based approach, due to its ease of implementation and lower cost of management.

- Central management of rule-sets.
- Ability to block email by keyword, regular expression pattern, originating IP address, sender domain, and sender name/email.
- Ability to block email by message type (e.g with a specific web-bug URL in them, with particular attachments, etc.)—this should integrate with the anti-virus implementation.
- Integration with RBL systems.
- Ability to perform statistical analysis on messages (e.g. Bayesian mathematics) to heuristically detect anomalous messages.
- Simple user rule specification (e.g. "John Smith wants NO hotmail messages").
- Simple reporting options, such as spam by domain, sender, and receiver.
- Integration with common mail clients such as Outlook/Notes (e.g. by moving suspicious messages to a "Junk Mail" folder).
- User control over blocked messages (e.g. with a quarantine folder that allows users to release messages that are not actually spam).
- Ability for users to easily report false-positives and false-negatives to the system.
- Ability to detect anomalous traffic patterns (e.g. a flood of email connections) and choke off/block incoming email.
- Commitment from the vendor to support the official anti-spam standard released by the IEEE (most vendors support one or the other draft standard).

NaSPA member Aurobindo 'Robin' Sundaram, CISSP/CISM, is the Director of Network Security at Choicepoint INC. Robin has worked in the Information Security Business for over 7 years in various capacities to include: Research Scientist, Information Security Officer, The Americas, Schlumberger Ltd, Engineering Program Manager for SchlumbergerSema (where he created the technical architecture for information security at the 2004 summer Olympic games in Athens, Greece)

[1] http://www.templetons.com/brad/spamterm.html
[2] http://www.usatoday.com/usatonline/20030508/5139591s.htm
[3] http://mail-abuse.org/rbl/
[4] http://www.ciphertrust.com/products/spam_and_fraud_protection/
[5] http://www.brightmail.com
[6] http://www.cloudmark.com/
[7] http://spf.pobox.com/faq.html (Sender Policy Framework)
http://www.microsoft.com/mscorp/twc/privacy/spam/senderid/overview.mspx (Sender ID Framework)
[8] http://www.cdt.org/speech/spam/030319spamreport.shtml
[9] http://www.spamlaws.com
[10] http://www.computerworld.com/printthis/2004/0,4814,97229,00.html