Global Information Assurance Certification Paper

The Spam Battle 2002: A Tactical Update

Karl A. Krueger
SANS GSEC Practical, v1.4
September 2002

"He who sends a message by the hand of a fool
Cuts off his own feet and drinks violence."
-- Proverbs 26:6

Abstract

The past two years have been a watershed in the fight against spam, with many changes in the tactics used both by spammers seeking to abuse networks and by administrators seeking to protect them. Many of these changes have notable policy implications. As the cost of spam has increased for ISPs, businesses, and end users alike, keeping up with these methods has become increasingly essential to protect the usefulness of email.

This paper presents an overview of the present state of the spam situation, with focus on the new fronts and tactics of the past two years. It briefly addresses the history of spam fighting by way of providing background, then presents policy and technical tools for the security-minded administrator to face these new fronts.

Contents

Introduction: The Cost of Spam

From the point of view of the average Internet user, email spam is one of the Net's chief problems, alongside malware and fraud. Every day spammers bombard millions of addresses with unsolicited and unwanted advertisements, hawking dubious services ranging from herbal medicines to sexual services to spamming itself.

According to one study, just over one-third of all spam sent is "get rich quick" scams, and one-quarter is ads for pornography, which often include pornographic content in the advertisement. (Brightmail) Yet as objectionable as this content may be -- especially when pornographic ads are sent indiscriminately to children -- the economic cost of spam is the same regardless of content. Spam represents a drain on the economy of the Internet, and this cost is increasing.

A report before the European Commission in February 2001 (Gauthronet et al.) estimated the cost of spam at 10 billion euro (US$ 8 billion) per year. This cost is the combination of costs to ISPs and businesses which must handle the traffic and disk storage requirements of spam sent to their users, and the cost in time and in utility to users whose email inboxes are filled with spam.

History: The Story So Far

The term "spam" was coined in the mid-'80s to refer to various sorts of behavior we now term "flooding". Later it came to refer to the sending of excessive multiple posts on Usenet, and finally to unsolicited bulk email. (Templeton) Originally an isolated problem associated with a few malicious sites, spamming grew rapidly to become a widespread social problem by the mid-'90s. Denied from sending mail directly to their unwilling target audience by ISP acceptable-use policies, spammers exploited insecure mail servers ("open relays").

While there had previously been security problems with email software such as Sendmail, the advent of widespread email spamming in the mid-1990s was the first illustration that the email system as it stood could not reject systematic abuse. By 1997, an "anti-spam movement" had formed on USENET -- a panoply of system administrators and concerned users organizing a variety of efforts to improve the system and combat spam. These efforts had three major aims:

- **Education of ISPs and Users.**

  Early on, many ISPs did not understand the danger of open mail relays, the costs of spam, and many other issues. Spam-fighters created informative Web pages, FAQs, and other documentation to bring the issue to the attention of other administrators.

- **Criminalization of Spam.**

  Groups such as CAUCE (the Coalition Against Unsolicited Commercial Email) have lobbied for laws banning spam in various jurisdictions. To this date, several U.S. states and a few European nations have banned spam in one way or another. (Sorkin) More recently, a set of E.U. directives has called on E.U. member countries to do the same. (EuroCAUCE)

  However, enforcement is spotty at best. To date, this remains one of the chief differences between spam and other sorts of online crimes: despite the rising cost of spam, it is extremely rarely prosecuted or litigated. When enforcement of spam laws makes the news, it is as an exception, not the routine.

- **Improvement of Mail Software.**

  Early mail software was not designed with security from abuse in mind, since there were few systematic abusers before the advent of spam. Spam-fighting contributors added various filtering features to Sendmail and other open-source mail software. Soon, these features were cloned in proprietary mail software as well. Since then, numerous other mechanisms for rejecting abuse have been created, several of which are detailed below.

For a general overview of the spam situation as of 2000, see the past SANS GSEC papers by Mauro and Granato. For an overview of legal and policy matters, see Gauthronet et al.

Spam Sources in 2002

In the past, spammers primarily used open SMTP relays to cloak the origin of spam messages. An open SMTP relay, or "open relay", is simply a mail server which accepts messages regardless of their source and destination addresses, and can be used to forward messages to arbitrary addresses both within and outside of its actual domain. Older mail software often took open-relay behavior as the default, an attitude dating to more permissive (and noncommercial) eras of the Internet.

The anti-spam movement has been remarkably successful in closing down open relays in the U.S. and Western Europe, through a combination of educational efforts and "encouragement" via DNSBL listing of offending ISPs. However, the spam-fighting message has been slower to reach the increasingly well-connected Asian mainland and Eastern Europe. At the same time, spammers have learned to exploit other sorts of services besides SMTP servers.

Spammers' search for services to abuse can be considered an unfortunate instantiation of the old and usually positive dictum that "the Internet interprets censorship as system failure and routes around it." As the email

administration community has cracked down on domestic open relay and direct-to-MX spam, spammers have "routed around" these restrictions by seeking out other ways to deliver their illicit messages.

- **Korean and Chinese Relays**

  In 2001, South Korea began an ambitious project to supply Internet access to its schools. Unfortunately, many of the systems installed in this push harbored open SMTP relays or open Web proxies, and were quickly discovered and exploited by spammers. (Hunter) China likewise has experienced an Internet boom recently, and similarly has installed a large number of exploitable mail and Web systems.

  Some Western sites have responded by blocking email from China and Korea entirely. (J. Ha) This may seem an extreme maneuver, but for a site which does no business with China yet receives thousands of spam messages from ".cn" domains every week, it is particularly attractive for its simplicity.

  In March 2002, the Korean relay problem made the news: Campaigners for Bill Jones, a candidate for governor in California, exploited a Korean primary school's open mail relay to send campaign ads. (Delio)

  Later in the year, there have been some signs that Korea and its ISPs have begun to take spam seriously. China has been a harder nut to crack, but some provinces have started to fight spam in earnest.

- **Open HTTP CONNECT Proxies**

  Web proxies supporting the HTTP CONNECT method (Fielding et al.) can be used to tunnel arbitrary TCP services, including SMTP, over an HTTP or HTTPS connection. These proxies can be easily found by portscanning, as they commonly run on one of a few TCP ports (3128, 8000, 8080).

  Sites with slow Internet connections often use proxies to cache commonly accessed pages. Web content filters, such as those used by businesses and public-access sites to censor pornography, are also often implemented as HTTP proxies. Activists interested in protecting privacy have also set up deliberately open proxies to provide the public with anonymous Web access.

  Although the HTTP proxy protocol permits the same sorts of authentication that HTTP servers do, many of these proxies are set up with no authentication whatsoever. Spammers portscan for these services, then exploit them to connect to the mail servers they intend to spam. Proxies have the advantage over mail relays that they do not add the client's IP address to mail messages sent through them -- providing the spammer with added anonymity.

According to DShield.org, scanning for these ports is constant and ongoing, albeit at a much lower level than root-exploitable service scanning. (DShield)

- **Open SOCKS Proxies**

  The SOCKS protocols are an older, and more versatile, form of proxy system. (Leech et al.) Like HTTP-based proxies, they can be set up with or without authentication, and usually run on a standard port (1080). SOCKS proxies are sometimes referred to as "Wingates", after the name of a popular software product which implements SOCKS.

  SOCKS and HTTP CONNECT proxies have not only been a mechanism for email spam, but also for other sorts of net abuse. Attackers, USENET flooders, and chat system abusers have been known to use open proxies to conceal their traces. Administrators of IRC (Internet Relay Chat) networks sometimes configure chat servers to portscan client systems for open proxies, so as to cut down on proxy-gated abuse of the IRC network. (Mystical)

- **Insecure Mail CGI Scripts**

  Many Web sites provide forms CGI-based forms through which a reader may send email to the Webmaster or other site staff. One of the most popular programs to drive these forms, "FormMail.pl" by Matt Wright (Wright), has a history of security vulnerabilities allowing the user to send mail through the Web site to arbitrary addresses at other sites as well.

  In one way, FormMail.pl is more useful than an open mail relay to a spammer, since the SMTP headers of the spam resulting will point to the Web site operator, not to an IP address connected to the spammer in any way.

  Spammers seeking to abuse FormMail.pl scan Web sites' CGI directories for copies of this script, under this name and several variants, such as "FormMail.cgi" and "formmail.pl.cgi".

- **Spamware**

  The past few years have seen an explosion in the number of software products known as "spamware" -- commercial software designed to make spamming easier. Spamware includes address harvesting software -- which trawls the Web, Usenet, or mailing lists for email addresses -- as well as stealth mass-mailing software.

  Stealth mailing software is the spammer equivalent of the script kiddie's "exploit scripts" and "rootkits" -- it simply automates the techniques of abuse. Current stealth mailing programs include features to forge mail headers, exploit multiple open relays and proxies at the same time, and randomize message parts to evade

content filters. Some spamware now generates obfuscated HTML messages, wrapped in multiple MIME encodings -- these are easy for an HTML-aware mail program such as Outlook or Communicator to display, but hard for offended users to manually decode and track down.

Seven U.S. states have passed laws making it illegal to "sell, give, or otherwise distribute" spamming software. Nevertheless, spamware is itself one of the classes of products more commonly advertised in spam.

- **Spam Gangs and Fake ISPs**

  A small number of hard-line spam operations, or "spam gangs", are responsible for a disproportionate amount of spam. Spam-fighter Steve Linford has collected a list of eighty-plus such spamming operations which have been thrown off at least three distinct ISPs for spamming. (Linford 1)

  These operations use all of the usual techniques, and additionally have been known to create so-called "fake ISPs". A fake ISP is a network connected to a Tier-1 ISP under the pretense of reselling connectivity, but which is actually used solely for spamming. With a large enough IP address allocation from the Tier-1 provider, the spammer can move his mail server from IP to IP in an to get around address-based filters. (Livingston)

Getting Your House In Order

Before taking steps to protect your own users from spam from elsewhere, it makes sense to be sure that your own site is not an offender. This is an issue of both policy and practice. Every Internet site should have policies forbidding its employees and/or clients from spamming. In addition, every site should ensure that any automated system that sends or handles email -- including Web forms, Webmail systems, mailing list managers, and other programs as well as mail transfer agents (mail servers) -- does so in a way which is secure from outside abuse.

Spam Policy

Both ISPs and other organizations doing business on the Internet should include provisions against spam in their site policies. For ISPs, the chief issue is prohibiting customers from using the service to spam, and backing up that prohibition with penalties such as suspension of service and clean-up fees. For other businesses, as well as for charities and other nonprofits, the issue is making it clear to sales personnel and other employees that spamming is not an acceptable way to advertise goods and services. For all sites, a spam policy should be integrated with site security policy regarding network services offered, so as to disallow the operating of open relays, proxies, and exploitable forms.

The most important reason to have a strong spam policy is to avoid contributing to the Internet's spam problem. ISPs and businesses who need a more obviously self-interested reason should consider DNSBLs: Sites which permit or support spamming soon get listed on these public lists of IP addresses, some of which are used by thousands of other Internet sites as a means of filtering incoming email. In order to keep the privilege of being able to send email to the rest of the Internet, a site must not be seen as a spam source or spam supporter.

Thus, a spam policy combines aspects of security policy, site acceptable use policy (AUP), and principles of Internet cooperation.

Spam Policy for ISPs

To be effective -- and to keep the ISP off of DNSBLs -- an ISP spam policy must go beyond simply disallowing users from spamming through the ISP's own mail servers. It must disallow spamming through relays, proxies, or other methods. It must also forbid users from hosting a Web site or other service on the ISP, and spamming ads for that site or service through another provider. (The reason for this last policy is that spammers can continue to profit from a "spamvertised" Web site if it remains active even after their spamming accounts on other ISPs are closed.)

Some DNSBLs, such as the MAPS RBL, also list sites for "spam support services". These include the selling of spamware, the hosting of banner ads or images which are listed in spam, and the operation of affiliate marketing programs which permit or encourage affiliates to spam. Thus, if one hopes to avoid being listed, it helps to have a policy which disallows spam support services as well as direct spamming. In many cases these policies have been phrased in terms of "facilitating policy violation". (Linford 2)

It is likewise important that in the event that a customer does spam, the ISP's staff be available and responsive to the issue. Best practices, as documented in RFC 2142, require that a site's email and abuse staff accept email to the addresses <*abuse@domain*> and <*postmaster@domain*>. The purpose of these addresses is to give administrators and users of other sites a known-good contact address for the site, to facilitate professional cooperation in resolving abuse and other problems.

It is not acceptable, under RFC 2142 or established anti-spam practice, to require victims of spam to jump through hoops in order to report spam -- for instance, to require abuse reports to be made via Web forms, or in a particular format. The *postmaster* and *abuse* boxes need to be read by a responsible individual empowere to take steps to cut off abusive customer behavior. Failure to handle abuse reports is a good way to get listed on DNSBLs. Indeed, many of the larger spam-source DNSBLs (such as SPEWS) make failure to handle reports sent to *abuse@domain* their primary criterion for listing.

Spam Policy for Businesses

A spam policy for a non-ISP business need not be as formal as that for an ISP. It may not need to extend beyond a memorandum circulated in the marketing and sales departments. However, an understanding that spamming is unacceptable must be present. There have been several cases of businesses (and nonprofits) wherein one sales representative took it upon himself to spam or to hire a spammer, without authorization, and thereby damaged the company's reputation.

However, the operation of spammable services -- such as mail servers and Web proxies -- should be covered in site security policy. Just as a site should minimize exposed network services to cut down on potential avenues of attack, it should minimize exposed SMTP-capable resources in order to cut down on potential avenues of exploitation by spammers.

Closing Mail Relays

Not long ago, many sites saw a need to operate open mail relays, to allow employees or customers away from the home network to send mail through the site mail server. Today, there are several alternatives; when combined with the risk of third-party spam relaying, these make open relay an unacceptable solution today. Alternatives include:

- **POP-before-SMTP.** An early, primitive alternative to open relay for remote mail users, this involves configuring the user's mail client to authenticate against a POP server before sending mail. Once the client has authenticated, the SMTP server allows relaying from his IP address for a short time. POP-before-SMTP is today considered a kludge.
- **SMTP AUTH.** Recent mail servers and clients support an extension to SMTP which allows clients to authenticate with a username and password or a CRAM-MD5 challenge/response. By requiring remote users (or all users) to authenticate before sending mail, third-party relay can be entirely avoided. While password authentication is subject to sniffing in the absence of TLS or other encryption, CRAM-MD5 appears sufficiently robust for the purpose.
- **Webmail.** Rather than permitting remote clients to send mail using SMTP, a site may offer a Web-based mail application. While some users find Webmail systems limiting and slow (the present author is no fan of most Webmail interfaces) a significant number of sites find that they reduce support costs by providing a common mail interface for all users.
- **Virtual Private Network (VPN).** When the remote mail user's host joins a VPN and is assigned a virtual address within the local network, it is no longer "remote" for the mail server's purposes. A VPN also has substantial security advantages for remote users. For layered security, VPN can be combined with SMTP AUTH.

Closing Open Proxies

Organizations operate proxies for many reasons -- the most common being caching of frequently used Web pages, filtering of malware or inappropriate material, and monitoring of Web activity. Certain NAT setups also lend themselves to proxy or transparent proxy (transproxy) configurations.

Open proxies pose many more risks besides the conveyance of spam. An open HTTP CONNECT proxy operated inside the organizational firewall, which accepts connections from outside it, serves as a complete hole through the firewall: a *virtual public network* as it were. Proxies can also be abused by attackers to forward attacks against a third party on the outside Internet, masking the attack's origin. The use of open proxies to hide the origins of abusive behavior on the Internet Relay Chat (IRC) system has become so prevalent that many IRC networks now portscan clients for open proxies before permitting them to use the service. (Mystical)

Lists of hundreds of open proxies are trivially findable on the Web (Mikhed), and portscans for common proxy ports (1080, 3128, and 8080) are common. It is thus exceedingly likely that any available open proxy will become widely known in both the spammer and crackers black markets. Aside from public lists of proxies, there are also lists for sale on IRC and the Web, which are presumably purchased both by crackers and spammers.

The method to secure a particular proxy server is system-dependent, and with the myriad different pieces of proxy software will not be covered here.

Choosing and Using a DNSBL

A DNSBL, or DNS-based Blackhole List, is a specialized DNS zone which lists IP addresses linked to spam sources. Most mail servers can be configured to query a DNSBL when a remote site attempts to inject mail, and reject the attempt if the remote site is listed on the DNSBL. Various DNSBLs are maintained as commercial offerings and as public services, enabling mail server operators to choose those whose policies meet their needs.

A few years ago, there were only a few DNSBLs, with the MAPS and ORBS lists being the best-known. Paul Vixie's MAPS RBL was the first DNSBL, with clear formal policies as to how sites could be listed and delisted. Alan Brown's ORBS, MAPS's first competitor, used a more controversial policy of automatically testing mail servers for open relay. Competition between the two was fierce and frequently acrimonious, but yielded some kind of equilibrium: sites who preferred to play it safe chose MAPS, while those seeking to block more spam at greater risk of blocking legitimate email as well chose ORBS.

Yet ORBS shut down in June 2001 under legal threats, and MAPS went

commercial the very next month, leaving the public DNSBL scene without a clear leader. The number of public DNSBLs exploded in a matter of months; today, one lookup service lists thirty-six different ones (OpenRBL) while another lists over 200! (Moensted) How is a site new to spamfighting to choose between these?

First, understand that different DNSBLs work in vastly different ways, and your choice of DNSBL must be in service to your site's policies and needs. DIfferent DNSBLs list IP addresses for different reasons, and it is up to you to select the list or lists that best serve your policy. Some of the criteria DNSBLs use include:

- **Tested Open Relay/Proxy.** An address is listed because it has been tested and confirmed to be an open SMTP relay or open proxy. The list operator does not assert that it is actually being used to send spam, merely that it is insecure.
- **Spam Source.** An address is listed because it has actually been used to transit spam. Some DNSBLs use SMTP honeypots or dummy addresses to catch samples of spam, and list the IP addresses of the sending systems. Others rely on manual reports by DNSBL users.
- **Spammer Organization.** An address is listed because it belongs to a company or organization in the business of spamming. These are sometimes known as "spam gangs". One DNSBL, the Spamhaus ROKSO list, lists spam gangs exclusively. (Linford 1)
- **Spammer Hosting ISP.** An address is listed because it belongs to an ISP which knowingly hosts spammers or spam supporters.

Any given DNSBL may use a single one of these criteria, or some combination of these and/or other criteria. DNSBLs exist which list only single companies (such as the "Flowgo Away" list, which lists IP addresses belonging to Flowgo, a portal site company which spams) as well as ones which list entire nations (such as South Korea; see above for why).

Some, such as MAPS RBL, VISI RSL, ORDB, and Spamhaus SBL, have clearly documented policies as to how IP addresses get on or off their lists. Others, such as Dorkslayers and XBL, operate much less formally; they list addresses which spam or irritate the list's operators.

Example: Writing DNSBL Usage Into Your Site's Email Policy

> PURPOSE: To reduce the costs imposed on ExampleInc. network, mail servers, and employees by unsolicited bulk email ("spam").
>
> POLICY: ExampleInc. mail servers shall make use of DNS-based Blackhole Lists (DNSBLs), publicly accessible lists of IP addresses known as spam sources. IP addresses listed on the DNSBLs we use will be disallowed from sending email into the

ExampleInc. network, with the exception of the "postmaster" and "abuse" administrative accounts.

ExampleInc. mail administrators shall select DNSBLs on these qualifications:

- All DNSBLs selected must have a clear policy as to why addresses are listed, and how they can be removed.
- One DNSBL shall be selected which solely lists "open relays" and "open proxies" -- addresses with insecure services exploited by spammers to forward spam. As of August 2002, the DNSBL selected for this purpose is the Open Relay Database, ORDB.org.
- One DNSBL shall be selected which solely lists "known spam operations" -- networks operated by organizations whose business is to send unsolicited commercial email. As of August 2002, the DNSBL selected for this purpose is the Spamhaus Blackhole List (SBL), sbl.spamhaus.org.

Because DNSBLs are volunteer services, their policies may change. Our choice of DNSBLs, as well as the possibility of using other techniques to reject spam, shall be reviewed quarterly by the systems administration staff and the Internet Services Group manager.

Using DNSBLs in Sendmail and Postfix

In Sendmail 8.10 and later, you can filter mail with a DNSBL by adding the "dnsbl" feature to the "sendmail.mc" file, as such:

```
FEATURE(`dnsbl', `dnsbl.example.net', `550 Reject Message')dnl
```

Replace "dnsbl.example.net" with the actual domain of the DNSBL you wish to use, and "Reject Message" with the error message that should be sent to listed sites which attempt to send you mail.

Sendmail 8.9 uses identical syntax, but the feature name is 'rbl'.

In Postfix, you can use multiple DNSBLs by defining the "maps_rbl_domains" parameter in the "main.cf" file, and adding "reject_maps_rbl" to one of the "smtpd_*_restrictions" parameters:

```
maps_rbl_domains = relays.example.net, proxies.example.net
smtpd_client_restrictions = reject_maps_rbl
```

Most popular mail servers can use DNSBLs; see the ORDB FAQ for configuration details. It is also possible to use DNSBLs as input to hashing filters or heuristic filters; see below for more about these.

SPEWS

It is impossible to speak of DNSBLs today without discussing SPEWS, the Spam Prevention Early Warning System. Widely regarded among spam-fighters as one of the most effective lists, it has also been the focus of controversy in the media and in the network administration community. SPEWS is, simply, an anonymously operated DNSBL with a policy of listing netblocks belonging to ISPs that host spammers. It is hosted in Irkutsk, Russia to avoid SLAPP lawsuits from Western spammers of the sort which downed ORBS and hampered MAPS.

Some regard this sort of operation as irresponsible. Others regard it as entirely necessary given the history of spam to date. Originally, the purpose of DNSBLs such as MAPS RBL was *not* to minimize spam by blocking email, but to educate spammy sites by shunning them until they cleaned up. Over time, it became apparent to some that this purpose was not being served, as ISPs could move their spammers around from address to address to evade listings. By listing netblocks, SPEWS takes this strategy away from the spam-friendly site: a site must choose whether it wishes to host spammers or to communicate with sites that do not want spam.

The choice of whether or not to use SPEWS is both practical and somewhat political. Some regard SPEWS as a source of too many "false positives" to use in more than an advisory capacity. However, the number of false positives will depend on what kind of site one is operating, and what kinds of email one's clients wish to receive; in the author's experience at a research institution, virtually no non-spam mail is received from SPEWS-listed sites.

RHSBLs: Domain-Based Blackhole Zones

Conventional DNSBLs, such as SPEWS and ORDB, use a lookup protocol devised by Paul Vixie for MAPS RBL. Whenever an SMTP client connects to an SMTP server using a DNSBL, the server does a modified reverse-DNS lookup of the client's IP address in the DNSBL's DNS zone.

For instance, to query the ORDB DNSBL for the IP address 192.168.42.23, the SMTP server does a DNS lookup for *23.42.168.192.relays.ordb.org*. (Note the reversal of the IP address bytes, as in reverse DNS.) This gets resolved by the ORDB nameserver; a positive response indicates that the address is listed, and mail should be rejected.

An RHSBL -- short for *Right-Hand-Side Blackhole List* -- is a variant flavor of DNSBL, which applies to domain names rather than IP addresses. The term "right-hand side" refers to the side of an email address to the right of the `@` sign. Specifically, RHSBLs do their checks on the SMTP envelope domain -- the right-hand side of the return address specified in the SMTP `MAIL FROM:` command.

Thus, to look up the domain *spampants.com* in the "RFC-Ignorant" RHSBL (a list of domains which violate RFC 2142 by failing to operate abuse or postmaster addresses), the SMTP server looks up the address *spampants.com.dsn.rfc-ignorant.org*. Again, as with DNSBLs, a positive response indicates that the domain is listed, and that mail should be rejected.

RHSBLs are a new kind of DNSBL, currently supported only in a few mail server systems. Sendmail's support is the most comprehensive, with an "rhsbl" FEATURE directive equivalent to the one for DNSBLs described in the previous section.

The intention of RHSBLs is not to supplant DNSBLs, but to supplement them. Since a domain and the IP netblock which hosts it are often under different administration -- a Web site and its colo ISP, for instance -- many regard it as only fair to block traffic from the more closely-fitting of the two. This is a philosophy rather different from that used by (for instance) SPEWS, but it is yet another useful tool in abuse rejection.

Hashing Spam: DCC and Vipul's Razor

The strength of DNSBLs and their ilk is that they facilitate sharing of information among sites for the purpose of blocking spam. Like the DShield.org block lists, they provide an automatically (or manually) collated list of sources of abuse. DNSBLs are ultimately rather coarse-grained: they filter solely on the basis of IP addresses. If an IP address is the source of some spam and some legitimate mail -- as with many larger mail servers -- a site may wish not to block it. A shared filtering system with a finer resolution -- a resolution of individual messages -- is an obvious thing to want.

The security-minded reader will already see a trust problem here: whom am I going to trust to tell me what messages I should reject -- what, in essence, my site's users should not be allowed to read? There is also a privacy problem: if I am to query another site to determine whether to accept a message or not, how much of that message must be revealed in my query?

The two per-message filtering systems I discuss below have addressed these concerns in different ways.

Vernon Schryver's DCC: Measuring Bulkiness

DCC, short for *Distributed Checksum Clearinghouse,* is a client/server system for the detection of bulk mail. (Schryver) A DCC client is usually an SMTP server, though it may also be a mail user agent (MUA -- a mail client). Whenever it receives a message, it calculates several checksums of that message, and transmits them to a server, which returns *the number of times it has seen each of those checksums.* If a message has been seen many times by DCC clients, these numbers will be high, indicating

that the message is likely bulk mail. DCC servers can also exchange checksums with one another, forming a redundant server-network similar in structure to that of IRC.

As the above description should make clear, DCC does not attempt to judge whether a message is spam. Vernon Schryver, the system's creator, believes that it is not feasible for an unintelligent system to accurately discern whether a particular message is spam. What DCC judges is the *"bulkiness"* of the message -- how many copies of it have been transmitted. As a result, clients which reject mail on this basis must also maintain a *whitelist* of non-spam bulk mail senders, such as legitimate mailing lists. This imposes some overhead on DCC users, but presumably not as much as maintaining a local blacklist of every spam source.

The checksums that DCC uses are not the same kind of checksums used by cryptographic algorithms. A crypto checksum or message digest is designed to maximize the output change caused by a small input change. Since spammers usually add changing elements such as tracking numbers to spam messages, such a checksum would not work for spam. Instead, the DCC checksums are *fuzzy checksums* under which such small input changes do not change the output. These work by checksumming not the bits of the message, but the arrangement of meaningful elements such as letters and URLs.

Vipul Ved Prakash's Razor: Who Says It's Spam?

Vipul's Razor (Prakash) is another checksum-based system -- but one which, unlike DCC, does claim to address the spamminess of mail. It does this not by attempting to discern how spammy a message looks from its content, but by relying on mail users to manually report spam they receive. This is done using a filter program in Unix, or a comparable script on other systems, which contacts a server and transmits a message checksum similar to DCC's.

A mail system using Razor to filter mail queries a server for every message, and receives back an indication of whether Razor reporters have flagged the message as spam. It may then reject the message on this basis, or use it as an element of a decision to reject or accept.

The latest versions of Razor add to this model a means for evaluating the trustworthiness of reports from particular reporters. Under the Razor 2.0 protocol, each reporter generates a key, which is used to sign reports. The server maintains indices of mutually confirming reports: the more in agreement with other users a reporter is as to what is spam, the higher trust that reporter receives.

Heuristic Filtering with SpamAssassin

An increasingly popular method of spam filtering for individual clients is to filter on the basis of message content. However, with a few exceptions

such as virus filtering, this has proven difficult to maintain at the server level, due to the wide variety of messages sent as spam and the undesirability of accidentally dropping legitimate mail. A few techniques have emerged which attempt to avoid this; one of the most successful is SpamAssassin, a heuristic filtering system which uses weighted rules to describe spammy email. (Mason et al.)

The configuration of SpamAssassin includes over 400 of these rules, most of which are Perl regular expressions, which represent various patterns that may be present in spam -- such as, say, the words "Herbal Viagra" or a URL containing "unsubscribe.pl". Each rule has a numerical weight, a real number which indicates how strongly the system should take that rule's matching into account. When a message is received, the rules are matched against its headers and body, and the total weight of matching rules is added up. If the sum exceeds a particular value, the message is flagged as spam. The weights are fully adjustable.

SpamAssassin is not restricted to content filtration. Its ruleset includes a number of Perl routines that implement a number of other spam detection systems, including DNSBL checks, DCC, and Vipul's Razor. In effect, SpamAssassin provides a *language for expressing descriptions of "spammy" email.* It is fully possible, using the SpamAssassin system, to do no content-based checking, and say "Flag a message as spam if it has passed through mail servers on three different DNSBLs," or "Flag a message as spam if it is both bulk mail according to DCC and has passed through a mail server in China or Korea," simply by appropriately setting the weights on rules for DCC and the appropriate DNSBLs.

SpamAssassin is a powerful tool, whether it is used to filter messages or merely to mark them as suspicious to alert users. Unfortunately it is a bit resource-intensive since it runs as a byte-compiled Perl script and its default ruleset includes a few hundred regular expression matches. Nonetheless, for systems with the capacity to run it, it can be quite useful.

Closing Comments

I hope that this report will be of use to security-minded mail system administrators in preparing to defend their systems against the theft of resources which spam represents. I would also like to extend an invitation to members of the security community interested in the fight against spam to join the anti-spam community on the Usenet newsgroup *news.admin.net-abuse.email*.

Bibliography

Anonymous. "SPEWS: Spam Prevention Early Warning System". Web site and DNSBL. URL: http://spews.org/

Anonymous. "openrbl.org: DNSBL Lookup". Web application. URL: http://openrbl.org/

Brightmail Inc. "The Spam Problem and Brightmail's Solution." White paper. February 2002. URL: http://www.brightlight.com/pdfs/Spam_Problem_Whitepaper.pdf

Crocker, David H. "Mailbox Names for Common Services, Roles, and Functions". RFC 2142, Internet Mail Consortium. May 1997. URL: ftp://ftp.isi.edu/in-notes/rfc2142.txt

Delio, Michelle. "Candidate: Spam in Every Pot". Wired News article, March 2002. URL: http://www.wired.com/news/politics/0,1283,50761,00.html

European Coalition Against Unsolicited Commercial Email. "Data Protection and Privacy in Electronic Commerce Sector". Legislative timeline. URL: http://www.euro.cauce.org/en/timeline1.html

Fielding, R., et al. "Hypertext Transfer Protocol -- HTTP/1.1" RFC 2616. June 1999. URL: ftp://ftp.isi.edu/in-notes/rfc2616.txt

Gauthronet, Serge; and Etienne Drouard. "Unsolicited Commercial Communications and Data Protection". Report to the European Commission, February 2001. URL: http://europa.eu.int/comm/internal_market/en/dataprot/studies/spamstudyen.pdf

Granato, John T. "You Got Mail - I Mean Spam!" SANS GSEC practical, March 2001. URL: http://rr.sans.org/email/spam.php

Ha, Jong-dae. "Chinese Netizen, 'Mail is Blocked'". Dong-A Ilbo, English edition, 5 March 2002. URL: http://english.donga.com/srv/service.php3?biid=2002030594218

Hunter, Jean. "Korean school relay spam: a blot on Korea's IT reputation." Letter, Asia Business News, 19 March 2002. URL: http://www.emailclub.net/bbs_read.html?db=asia1&no=34

Jensen, Thomas. "Open Relay Database - FAQ". Web article. URL: http://ordb.org/faq/

Leech, M.; M. Ganis; et al. "SOCKS Protocol Version 5". RFC 1928, Network Working Group. March 1995. URL: http://www.ietf.org/rfc/rfc1928.txt

Linford, Steve, aka Stiff Linefeed. "Registry of Known Spam Operations". Web site. URL: http://rokso.org/

Linford, Steve. "Responsible ISP AUPs". URL: http://www.spamhaus.org/aups.html

Livingston, Brian. "Inside the spammer's world". C|Net News.com, 29 June 2001. URL: http://news.com.com/2010-1080-281499.html?legacy=cnet&tag=bt_pr

Mason, Justin, et al. "SpamAssassin". Heuristic spam recognition software and Web site. URL: http://spamassassin.taint.org/

Mauro, Edward A. "Unsolicited Bulk Email - The Problem and Some Hope". SANS GSEC practical, January 2001. URL: http://rr.sans.org/email/bulk.php

Mikhed (alias?). "Free proxies list". List of 3000+ open proxies. 29 August 2002. URL: http://mikhed.narod.ru/download/lists/goodproxy.txt

Moensted, Christian, aka Joergen Mash. "dr. Joergen Mash's: DNSBL database check". Web application. URL: http://moensted.dk/spam/

Mystical IRC Network. "Open Proxy/Wingate". Web page. URL: http://www.mystical.net/proxy.html

Prakash, Vipul V., et al. "Vipul's Razor." Software. June 2002. URL: http://razor.sourceforge.net/

SANS. "Distributed Intrustion Detection System." Web-based information service. URL: http://www.dshield.org/, URL: http://www.incidents.org/

Schryver, Vernon. "Distributed Checksum Clearinghouses". Software system. September 12, 2002. URL: http://www.rhyolite.com/anti-spam/dcc/

Sorkin, David. "Spam Laws". Web site. URL: http://spamlaws.com

Templeton, Brad. "Origin of the term 'spam' to mean net abuse". Web article. URL: http://www.templetons.com/brad/spamterm.html

Wright, Matt. "FormMail". Perl software and Web site. URL: http://worldwidemart.com/scripts/formmail.shtml

Last Updated: October 18th, 2017

Upcoming Training

SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201710,	Oct 23, 2017 - Nov 29, 2017	vLive
SANS Seattle 2017	Seattle, WA	Oct 30, 2017 - Nov 04, 2017	Live Event
SANS San Diego 2017	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	Live Event
San Diego Fall 2017 - SEC401: Security Essentials Bootcamp Style	San Diego, CA	Oct 30, 2017 - Nov 04, 2017	vLive
SANS Gulf Region 2017	Dubai, United Arab Emirates	Nov 04, 2017 - Nov 16, 2017	Live Event
SANS Miami 2017	Miami, FL	Nov 06, 2017 - Nov 11, 2017	Live Event
Community SANS Vancouver SEC401^	Vancouver, BC	Nov 06, 2017 - Nov 11, 2017	Community SANS
Community SANS Colorado Springs SEC401~	Colorado Springs, CO	Nov 06, 2017 - Nov 11, 2017	Community SANS
SANS Sydney 2017	Sydney, Australia	Nov 13, 2017 - Nov 25, 2017	Live Event
SANS Paris November 2017	Paris, France	Nov 13, 2017 - Nov 18, 2017	Live Event
SANS London November 2017	London, United Kingdom	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS Portland SEC401	Portland, OR	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS San Francisco Winter 2017	San Francisco, CA	Nov 27, 2017 - Dec 02, 2017	Live Event
Community SANS St. Louis SEC401	St Louis, MO	Nov 27, 2017 - Dec 02, 2017	Community SANS
SANS Khobar 2017	Khobar, Saudi Arabia	Dec 02, 2017 - Dec 07, 2017	Live Event
Community SANS Ottawa SEC401	Ottawa, ON	Dec 04, 2017 - Dec 09, 2017	Community SANS
SANS Austin Winter 2017	Austin, TX	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS Munich December 2017	Munich, Germany	Dec 04, 2017 - Dec 09, 2017	Live Event
SANS vLive - SEC401: Security Essentials Bootcamp Style	SEC401 - 201712,	Dec 11, 2017 - Jan 24, 2018	vLive
SANS Bangalore 2017	Bangalore, India	Dec 11, 2017 - Dec 16, 2017	Live Event
SANS Cyber Defense Initiative 2017	Washington, DC	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS Cyber Defense Initiative 2017 - SEC401: Security Essentials Bootcamp Style	Washington, DC	Dec 14, 2017 - Dec 19, 2017	vLive
SANS Security East 2018	New Orleans, LA	Jan 08, 2018 - Jan 13, 2018	Live Event
Community SANS Nashville SEC401^	Nashville, TN	Jan 08, 2018 - Jan 13, 2018	Community SANS
Community SANS Hawaii SEC401	Honolulu, HI	Jan 08, 2018 - Jan 13, 2018	Community SANS
Mentor Session - SEC401	Memphis, TN	Jan 09, 2018 - Mar 13, 2018	Mentor
SANS Amsterdam January 2018	Amsterdam, Netherlands	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VA	Jan 15, 2018 - Jan 20, 2018	Live Event
Mentor Session - SEC401	Minneapolis, MN	Jan 16, 2018 - Feb 27, 2018	Mentor
Las Vegas 2018 - SEC401: Security Essentials Bootcamp Style	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	vLive
SANS Las Vegas 2018	Las Vegas, NV	Jan 28, 2018 - Feb 02, 2018	Live Event