DAVE JOSEPHSEN

# homeless vikings

## SHORT-LIVED BGP SESSION HIJACKING—A NEW CHAPTER IN THE SPAM WARS

Dave Josephsen is author of the upcoming book *Building Monitoring Infrastructure with Nagios* (Addison-Wesley). He currently works as the senior systems administrator for a small Web hosting company and donates his spare time to the SourceMage GNU Linux project.

*dave-usenix@skeptech.org*

**THE FIRST UNSOLICITED, COMMER-** cially motivated bulk email was sent on ARPANET in 1978 by a DEC representative named Gary Thuerk [1]. A full 28 years later, spam has evolved into a 55-billion-messages-per-day [2] global epidemic that has affected areas of technology unimaginable by the ARPANET engineers of 1978. This article will chronicle the history of the spam wars, a war that has almost always been waged along two technological fronts: those of content filtering and delivery countermeasures. By examining the history of the arms race in the context of recent attacks with zombied PCs and short-lived BGP session hijacks, I conclude that one of these fronts may in fact be a dead end and worth abandoning altogether.

From 1978 to 1994, the business of spam remained a nonissue because email itself was in an infantile state. In the early 1990s, most spam was sent in the context of USENET newsgroups, and by a few identifiable individuals, such as Canter, Siegel, and Wolff [3]. In 1994 the Net witnessed its first real spam, sometimes referred to as the "spam heard round the world," when Canter and Siegel's "green card" message was sent to at least 6000 USENET groups [4].

In the early days, retribution was swift [5], but things degenerated quickly. In 1995, Floodgate, the first commercially available spamware, was available. By 1996, four more automated spam packages were available for sale, as were lists of millions of email addresses [6]. The spammers wasted no time legitimizing their so-called business model with various pro-spam trade groups such as the Freedom Knights and the IEMMC and proceeded to reach millions of USENET subscribers with their marketing messages. The lure of free marketing combined with the lack of protocol security set the stage for the inevitable war that rages on to this day.

Almost immediately, two technological methodologies appeared to combat spam. The first type focused on the content of the message in question, and the second type on indicators such as the email or IP address of the sender. These divergent paths have evolved largely independent of each other as the spam attacks have become more

frequent and increasingly complex. Content filters eventually moved toward statistical learning, whereas delivery countermeasures evolved increasingly sophisticated challenge/response mechanisms. Let's examine the lineage of each front independently, beginning with content filters.

## Content Filters

The earliest example of automated content filtering might be Nancy McGough's procmail filtering FAQ, published in 1994 and still available at http://www.faqs.org/faqs/mail/filtering-faq/. Early spam messages were very much singular events [7]. The defenders of the time knew who would be sending spam, and sometimes even when, so early content filters needed to do little more than look for static strings in the message body.

Static string searching continued to work well for many people until around the year 2000, when various content obfuscation techniques became prevalent in spam [6]. The word obfuscation games continued for a number of years, with spammers using misspellings, character spacing. and a multitude of other HTML- and MIME-based [8] techniques to bypass word filters. For a while the defenders followed in step, adding html parsers and character transformation algorithms to their content filters.

In late 2000 and early 2001, based on an idea from Paul Vixie, an innovative content filter was created which worked by taking a fuzzy checksum of a message and comparing it to a database of known spam checksums. Two implementations of this idea exist today in The Distributed Checksum Clearinghouse (http://www.rhyolite.com/anti-spam/dcc/) and Vipul's Razor (http://razor.sourceforge.net/). Spammers responded with attempts to poison the checksum database by reporting legitimate messages to the abuse lists and by adding unique gibberish to the messages in an effort to make the checksums more "different."

Then, in August 2002, Paul Graham published his seminal paper, "A Plan for Spam" [9], in which he publicly made the case for Bayesian learning algorithms for spam classification. Prior work existed [10, 11], but ironically Graham's less mathematically rigorous approach made the technique far more effective [12]. At least a dozen Bayesian implementations exist today.

Since 2002, several improvements have been made to Graham's core idea; these include the addition of several classification algorithms, such as Robinson's inverse chi-square [13], and data purification techniques, such as Bayesian noise reduction [14]. But overall, naive Bayesian classifiers have been unanswered by the spammers for four years and are therefore considered a category killer for content-based filters. Where researchers have had success against Bayesian filters, it has been by training other Bayesian filters to use against them [15].

## Delivery Countermeasures

On the delivery countermeasure front, examples of blacklists date back to November 1994 [3], when USENET "remove lists," consisting of malicious sender addresses, were used to remove unwanted messages. In those early days, reporting abuse to a spammer's ISP yielded swift results [5]. By 1995 the USENET community's outrage over spam abuse had outweighed its censorship fears, and UDPs (USENET Death Penalties) were used to block all posts from malicious sites. At the time, the sites weren't necessarily considered malicious; the UDP was most often used to "persuade" the admin-

istrators of a particular site to take care of its abuse problems after more diplomatic means had failed [16].

In non-USENET circles, sender address-based filtering was becoming more and more common, forcing spammers toward joe-job attacks. By 1996, the abuse reports and sender filtering resulted in the spammers' use of relay systems to deliver their mail. In 1997 the term "open relay" was coined to describe a mail server that would relay mail for any recipient from any sender. The first real-time spam blacklists (RBLs) appeared the same year [6].

From 1998 to 2002, so many delivery countermeasures had been proposed and beaten that they are too numerous to mention. Attempts at using blacklists were thwarted by relays, whitelists were met by directory harvest attacks and more joe-job spoofing, and greylists still showed promise but wreaked havoc with noncompliant MTAs. A few payment-based systems were proposed in this period, including micropayment-based "e-stamps" [17] and the CPU-based idea that eventually became hashcash [18]. These are not particularly effective against spoofing attacks and never gained widespread adoption. Direct challenge/response systems were ineffective, owing to forged "from" headers, and were generally considered rude. Most of the other solutions of the time were in one way or another thwarted by spammers simply adopting someone else's net identity, through open relays, or with header spoofing, or using a combination of the two.

The ease with which the spammers abused valid credentials was clearly frustrating to those designing delivery countermeasure systems. Many in this period became convinced that the problem with SMTP was the lack of sender authentication. In June of 2002, Paul Vixie wrote a paper entitled "Repudiating MAIL FROM" [19] which became the basis for SPF, or Sender Policy Framework. SPF is a DNS-based authentication mechanism which calls for the use of MX-like DNS records for mail servers that send mail, instead of those that receive them.

Although SPF was in clear violation of the SMTP RFCs and broke important functionality such as forwarding, many (especially in the business community) lauded SPF as the silver bullet that would once and for all solve the spam problem, especially when it was embraced by Microsoft and AOL. But, alas, SPF too has met with defeat [20] (although many vendors still encourage its use).

The year 2000 witnessed the first large-scale distributed denial-of-service attack against multiple prominent Web sites, including Yahoo! and eBay. The attack, launched by a Canadian teenager, brought public attention to the problem of botnets. A recent IronPort study found that 80% of the spam currently sent on the Internet is sent through similar collections of zombied PCs [2]. In one way or another, zombies make moot most of the remaining challenge/response systems of today. By directly making use of "grandma's" PC to send their message, spammers are nearly assured of success in a challenge/response scenario.

Today, RBLs remain the largest and most widely used tool on the delivery countermeasures front, despite the questionable ethics and legal entanglements of the RBL managers themselves [21, 22, 23]. The ease of setup, combined with a quantifiable reduction in spam, makes RBLs a popular choice with system administrators looking for a quick fix so that they can get back to their "real" work.

## BGP Prefix Hijacks

However, a recent paper by Anirudh Ramachandran and Nick Feamster [24] may change all that by providing a sneak peek at the next battlefield on the delivery countermeasures front. The Feamster paper provides the first documented evidence of spammers using short-lived BGP prefix hijacks against RBLs to get their mail delivered. Since you may not be familiar with the technique, I'll briefly summarize.

Prefix hijacking can happen a couple of different ways. In the first scenario, the hijacker advertises a huge netblock, for example, 12.0.0.0/8. Much of the space in this netblock is unallocated, or allocated but unused. More specific advertisements in this netblock will take precedence over larger ones, so in practice, the attacker won't interrupt legitimate traffic. For example, a legitimate company advertising 12.10.214.0/24 will not be affected by the hijacker's less specific advertisement.

The second scenario is more of a direct prefix hijack, whereby the hijacker advertises a legitimate netblock (yours, for example), and routers closer to the hijacker who don't or can't filter bogus announcements from their BGP peers simply believe the hijacker. This is less common in practice right now, because this sort of thing is easier to spot and has less of a payoff; some routers still believe the legitimate Autonomous System.

Prefix hijacking has been used in the past by profiteers who would combine bogus BGP announcements with RIR social engineering to take control of blocks of IP space they did not own. They would then sell these bogus netblocks to unwitting organizations. In the past few years, however, the network engineering and security communities have become aware of a different kind of prefix hijack. These hijacks are very short-lived, lasting 15 minutes or less.

Why would someone hijack a route for such a short amount of time? For the readership of this magazine, it's probably not a huge test of the imagination. In fact, pretty much any illicit behavior you happen to fancy would benefit from the technique, because using addresses nobody owns makes you harder to track. If you wanted to nmap the NSA, DoS the RIAA, P2P MP3s, or perform whatever other acronyms might get you in trouble, and you wanted to do it in a quasi-untraceable manner, this might be for you. Spamming people is of course a behavior generally assumed to be associated with short-lived prefix hijacks, but while speculation abounds, very little in the way of actual evidence has been available until the Feamster paper.

Prefix hijacking attacks directly target countermeasures such as RBLs by using netblocks nobody has seen yet. It's simply a new take on the same old trick of using someone else's credentials to deliver unwanted mail. Prefix hijacks can also target SPF by making it so that large portions of the Internet might actually send their SPF DNS authentication requests right to the spammers. The bottom line is that if your anti-spam solution depends on IP addresses, you lose.

## Conclusions

I believe prefix hijacking may prove to be the proverbial nail in RBL's coffin, but even as you read this the arms race escalates on the delivery countermeasures front. RBLs for their part are evolving lower into the networking stack and I'm quite sure that this is not a good thing.

For example, the MAPS RBL now offers a BGP feed that your Cisco router can consume [25]. Given the history of the war thus far, I am skeptical that further forays toward filtering spam using incidental indicators such as IP address are going to be effective without incurring additional collateral damage. Finally, given that NBCs (naïve Bayesian classifiers) remain an effective and unanswered weapon in the fight, I find it curious that there are two fronts at all.

## REFERENCES


[1] http://www.templetons.com/brad/spamreact.html.

[2] http://www.ironport.com/company/ironport_pr_2006-06-28.html.

[3] http://www-128.ibm.com/developerworks/linux/library/l-spam/l-spam.html.

[4] http://en.wikipedia.org/wiki/Canter_&_Siegel.

[5] http://catless.ncl.ac.uk/Risks/15.79.html#subj12.

[6] http://keithlynch.net/spamline.html.

[7] http://www.l-ware.com/wsj_cybersell.htm.

[8] http://www.jgc.org/tsc/.

[9] http://www.paulgraham.com/spam.html.

[10] http://citeseer.ist.psu.edu/sahami98bayesian.html.

[11] http://citeseer.ist.psu.edu/pantel98spamcop.html.

[12] http://www.paulgraham.com/better.html.

[13] http://radio.weblogs.com/0101454/stories/2002/09/16/spamDetection.html.

[14] For example, http://freshmeat.net/projects/libbnr/.

[15] http://www.jgc.org/SpamConference011604.pps.

[16] http://www.stopspam.org/faqs/udp.html.

[17] http://www.templetons.com/brad/spam/estamps.html.

[18] http://www.hashcash.org/papers/hashcash.pdf.

[19] http://sa.vix.com/~vixie/mailfrom.txt.

[20] http://www.theregister.co.uk/2004/09/03/email_authentication_spam/.

[21] http://www.internetnews.com/dev-news/article.php/10_995251.

[22] http://csifdocs.cs.ucdavis.edu/tiki-download_wiki_attachment.php?attId=431.

[23] http://www.peacefire.org/stealth/group-statement.5-17-2001.html.

[24] http://www-static.cc.gatech.edu/~feamster/publications/p396-ramachandran.pdf.

[25] http://www.pch.net/documents/tutorials/maps-rbl-bgp-cisco-config-faq.html.