ROBERT HASKINS

ISPadmin

BLOCKING NON-EMAIL SPAM

Robert Haskins has been a UNIX system administrator since graduating from the University of Maine with a B.A. in computer science. Robert is employed by Shentel, a fast-growing network services provider based in Edinburg, Virginia. He is lead author of *Slamming Spam: A Guide for System Administrators* (Addison-Wesley, 2005).

■ *rhaskins@usenix.org*

SPAM HAS BEEN AROUND FOR MANY years, since (at least) the infamous Canter and Siegel green card Usenet message in 1994. The bad news is that the problem of unsolicited commercial messages entered the SMS (mobile/cell phone), Web log (blog), and instant message (IM) space some time ago. However, the good news is that spam activity in these more modern forms of communication is currently much lower than traditional email spam.

Background

Before getting into what can be done about blocking these newer forms of spam, I'll briefly introduce the problem of these spamming methods. In most of these cases, existing anti-email spam methods can be and are applied by the provider and/or the end user to battle the new forms of spam. However, these traditional methods are much more dependent on the precise server and client software tools used by the provider.

The lack of a centralized, open source messaging server as there is in the email space (e.g., Sendmail) makes addressing the problem of non-email spam much more difficult. Another piece of bad news is that the available tools and techniques for addressing non-email spam messaging are in their infancy. However, as the perpetrators adjust their spamming techniques, this will undoubtedly change, as it did in the email spam area.

SMS (CELL PHONE) SPAM

In the case of cell phone spam, the perpetrators often guess the cell phone numbers of the lucky recipients. SMS spammers send their junk to unwitting subscribers using the publicly available email-to-SMS gateways provided by the cell phone service providers. It's a relatively easy and cheap way for the spammers to get their message out to lots of users in an immediate fashion.

The regularity of cell phone numbers (at least for most U.S. carriers) makes guessing recipients trivial. By using publicly available information for the area code and local exchange for the provider in question, the cell phone spammer must simply guess the last four digits for the subscriber.

Directory harvest attacks (where the spammer guesses the email address of the subscriber's

phone) can be effective for identifying potential recipients for the SMS spammer. However, if the service provider has any sort of rate limiting in effect on the SMS-to-email gateway, it can be used to identify a spammer who exceeds preset message sending thresholds (either successful or unsuccessful attempts).

BLOG SPAM

Blog spam (also called link spam or comment spam) is defined by Wikipedia as "any web application that displays hyperlinks submitted by visitors or the referring URLs of web visitors" [1]. The target of the spammers can be any page that accepts comments from the general public, including wikis, blogs, and Web-based discussion boards. (For the purposes of this article, the term "blog spam" is used to denote any discussion-based spam mechanism.) The goal of the spammer is often to increase search engine rankings by increasing the number of link counts to the spammer's target site.

The solutions to the problem of blog spam are closely tied to the software packages used to implement the discussion boards themselves. Without a centralized exchange point (as exists with email spam), it is difficult to generalize a solution for blog spam.

INSTANT MESSAGE

Instant messaging spam (a.k.a. spim) is defined for the purposes of this article as commercial messages received via AOL Instant Message, ICQ, or any similar real-time messaging channel. The advent of protocols like Jabber [2] holds a lot of promise for controlling spim, but much work needs to be done, as the tools are still not very sophisticated.

Solutions

So what can be done about these new forms of spam? In general, the information sent in email spam is similar to the information sent by spammers in other forms. For example, spammers will send a URL or telephone number as part of their message. If the anti-spam solution uses content analysis, then the same information used to filter email spam can be used to filter other types of spam. However, content analysis in the form of header information is not possible, as user-identifiable headers don't exist for most non-email-type communication channels.

CONTROLLING SMS SPAM: PROVIDER SIDE

Regarding cell phone spam, the place to catch SMS spam is before (or at) the email-to-SMS gateway. Traditional content-based email anti-spam methods can be useful to the provider prior to the message entering the provider's SMS system. These methods are well documented elsewhere and are not covered here.

CONTROLLING SMS SPAM: SUBSCRIBER SIDE

Some cell phone providers (e.g., Verizon Wireless) give their subscribers the ability to change the external email address used for sending text messages to the subscriber's cell phone. This change can be made to obfuscate

the subscriber's email address, making it harder for spammers to "guess" potential recipients. Other capabilities may be present in SMS provider networks, such as restricting senders (whitelisting/blacklisting).

CONTROLLING BLOG SPAM

Spam to blogs and similar discussion groups is most often handled by the software that implements the blog itself. This is because there is not much in the way of protocol or other clearinghouse mechanisms with blogs (as there is with email spam in the form of message servers like Sendmail). Some methods used by blog/wiki software to limit spam include:

- Periodically scanning blogs and removing messages associated with known spammer URLs
- Using a CAPTCHA (Turing test) to force the poster to prove that they are a human and not a spammer
- Using whitelists/blacklists of IP addresses posting allowed/disallowed

Blog spam is a difficult problem to solve, as the usual email spam issues such as false positives and what to do with posts identified as spam still apply. How do you allow the moderator to reinstate a blog posting incorrectly classified as spam?

CONTROLLING SPIM: SERVER SIDE

The ability to control instant message spam is arguably the most advanced of the three types of non-email spam handling covered here. One example of the maturity is the simple fact that there is a commercial product in this space, namely, Perimeter Manager for IM [3] from Postini. This service utilizes Postini's email spam processing network to identify IM messages that are potentially spam messages.

On the open source side, there isn't really a solution currently available. While AIM and similar protocols have proxy capability, this author is aware of no firewalls that enable end users to filter instant messages in any way.

Jabber is an open source instant messaging protocol which may improve open source IM spam filtering. There is currently an experimental Jabber standard titled "SPIM-Blocking Control" which enables some level of spim filtering [4]. This is targeted at the large "zombie networks" of machines that often send IM spam (as well as other undesirable messages). Some of the techniques used to control spim include:

- Whitelisting/blacklisting functionality
- Automatically exchanging lists of IDs that have sent spim to users or that don't answer specific challenges to prove the ID is a real person

Unfortunately, existing techniques are simple and don't include complex content analysis such as URL-checking and similar content-filtering capabilities.

CONTROLLING SPIM: CLIENT SIDE

On the IM client side, the controls available are directly dependent upon the IM client that is used. Most allow whitelists and blacklists, but beyond that not much is available.

Conclusion

SMS spam, blog spam, and spim are here to stay and are only going to get worse. The good news is that these forms of spamming are not too widely used, and basic whitelisting/blacklisting techniques can be utilized to filter most of this junk from your daily life. In the case of blog spam, methods exist and are used to help reduce the problem. However, there is a tight integration between the anti-spam blog software and the blog application itself, so stand-alone methods don't exist.

Although there is a commercial solution to the problem of spim and limited anti-spam standards are being developed for the Jabber IM protocol, no open source solution currently exists for spim.

I'd like to thank Todd Underwood of Renesys and Scott Petry of Postini for their input into this article.

REFERENCES

[1] http://en.wikipedia.org/wiki/Blog_spam

[2] http://www.jabber.org/

[3] http://www.postini.com/postini_solutions/im_security.php

[4] http://www.jabber.org/jeps/jep-0159.html

Wikipedia page for SMS spam: http://en.wikipedia.org/wiki/Mobile_phone_spam