

Re: Is predictable spam filtering a vulnerability?

Source: <http://www.derkeiler.com/Mailing-Lists/securityfocus/bugtraq/2004-06/0398.html>

From: Sean Straw / PSE (PSE-L_at_mail.professional.org)

Date: 06/23/04

Date: Wed, 23 Jun 2004 10:07:31 -0700

To: bugtraq@securityfocus.com

At 15:52 2004-06-20 +0200, Luca Berra wrote:

>the problem with your proposed behaviour is the fact that to be able to
>respond with 5xx in the smtp transaction would require the spam filter
>to analyze content on the fly.

Agreed.

There's the additional issue of MX precedence. It's a well known fact that spammers send mail to backup MX'es (often completely skipping even TRYING to deliver to the primary MX) on the basis that they often do not employ the same level of spam protection (such as DNSBLs) as the mail hosts they serve, and generally do not validate recipients (allowing the spammer to deliver a message for hundreds of recipients at your domain, deferring any rulesets which may check for invalid recipients until after the spammer has dropped their message and shot off to their next victim).

>The most common approach for spam (content) filters is to queue messages
>and process them later, in this case the filter **MUST NOT** generate a NDN,
>since there is no way to guarantee that the envelope sender is not
>faked.

If the envelope sender is faked, then rejecting the message at SMTP time (say, due to a DNSBL check) will result in an NDN directed at that faked address anyway, excepting when the sending mail host is really a zombie PC or spamware to begin with, in which case it'd be dropping the NDNs into the ether. The chief difference is that with an SMTP time rejection, YOUR mail server doesn't deliver anything – the server which was attempting to deliver the message to you would be responsible for delivering the bounce based on your SMTP replies during the transaction.

Since we know spammers don't bother to remove bad addresses from their databases (and generally aren't receiving NDNs anyway), one can question whether issuing SMTP-time rejection codes is a wise idea anyway: it allows the spammer to probe your system and possibly determine what type of spam filtering you are using, which may then provide them with information on how to best circumvent that. Not that spammers are operating the same way that a blackhat would relative to any individual target, but what if the

SecurityFocus Bugtraq: Re: Is predictable spam filtering a vulnerability?

spamware were to profile each recipient host and provide that data to the spamware author so that they could have a database identifying different hosts with different exploitable weaknesses in their spam filtering, thus allowing the spamware author to revise their approach?

*>I hold that after suitable training of the spam filter (this includes
>generation of whitelists and such), dropping mail into oblivion is
>perfectly safe.*

A preferred method is dropping suspect content-matched spam into an ARCHIVE, and providing data about the event to the recipient (either in a daily report or making it available via a web query mechanism). It is the recipients option to check that report and determine if there are any false positives. Since the messages haven't been discarded, false positives can be retrieved from the archive and the senders added to the recipients personal whitelist. Even if you discard mail rather than archiving it, providing a LOG of that action, accessible by the intended recipient, is desirable. Spew which has gone unreviewed for some period of time can be purged from the archive to conserve resources.

I still maintain that the original concern of a vulnerability due to "predictable spam filtering" relies upon human action, as well as users A and B somehow being on separate mail systems (such that B would have received the message but A would silently reject it) *AND* that user B would both have forwarded the message verbatim as well as themselves NOT being in A's whitelist to begin with. It supposes entirely too many specific configuration issues and human actions, AND still doesn't provide for any automated exploitation of system resources or exposure of secured data, excepting by human action.

Please DO NOT carbon me on list replies. I'll get my copy from the list.
Founding member of the campaign against email bloat.