Breaking Anti-Spam Systems with Parasitic Spam

Morton Swimmer
IBM Research GmbH
Rüschlikon, Switzerland
swimmer@acm.org

B. Leiba and I. Whalley
IBM T. J. Watson Research Center
Hawthorne, NY, USA
leiba@watson.ibm.com,
inw@us.ibm.com

Nathaniel Borenstein
Lotus Development
Cambridge, MA, USA
nborenst@us.ibm.com

ABSTRACT

The existance of networks of 'bots' raises the possibility of a new type of spam that breaks the current paradigm of spam defense, in which the defence acts purely as a filter. This spam, which we call parasitic spam, looks to a filter very much like spam with scraped text, but contains instead legitimate content going from a legitimate sender to a legitimate recipient. In this problem statement, we will discuss what parasitic spam is and is not, the likelihood of parasitic spam appearing in the wild, and possible countermeasures.

1. INTRODUCTION

This is a problem statement concerning a new type of spam that we call parasitic spam (or 'p-spam' for short). This potential attack on email systems is enabled by 'bot' networks, essentially collections of systems over which a malicious hacker has gained control (so-called 'zombies'). It should not be forgotten that some of these zombies may themselves be email servers or relays. Until now, bots have enabled spammers to distribute the process of spamming, which permits them to hide from law-enforcement and blacklist operators, and to send email at a higher rate than would be possible with a single machine.

However, zombies are useful for more than sending new email messages. Specifically, they could be used to piggyback spam on to legitimate email coming from, or passing through, the zombies. Similar ideas have been expressed by others, e.g., [2]. For the rest of this paper we will look at the various aspects of parasitic spam.

2. PARASITIC SPAM

If a user's machine has been compromized by a spammer (or his proxy), any aspect of the machine can be subverted. While current bots merely use their own SMTP program to send out spam, another alternative is to hook into the existing TCP/IP stack or MAPI infrastructure (on Windows). In a p-spam-bot, this hook would be used to monitor for email being sent from the system, which is then modified to contain spam as well as the original message. The spam has therefore effectively parastically 'infected' the original message, hence the name 'parasitic spam'.

Likewise, if an email server or relay is compromised, the spammer will have access to a greater volume of email to which to attach spam.

The concept of modifying legitimate email is not new. There has been the dubious custom of free email services and some mailing lists to append an advertisement or some other message as a signature to the email (see Fig. 1).

Because these 'signatures' are often perceived as a nuisance in mailing lists, some mailing list software includes tools for stripping these signatures. This works because the ads tend to be fairly predictable and false positives can be kept to a minimum, just like parasitic viruses.

P-spam is similar, but trickier, because the message can be altered in many more ways than just in the signature. We will look at some ways this could be accomplished in the next section.

The spectre that p-spam raises is that the entire p-spam message can not be blocked, just as the the message with the advertisement in the signature in the previous example cannot be blocked. The response must be to remove just the offending spam text.

To the best of our knowledge, p-spam has not been observed yet outside the lab, so we can only speculate about its appearance. In the following section we will outline a sampling of schemes that could be employed by the p-spammer.

2.1 Infection patterns of p-spam

One straightforward scheme is to add the spam to the ham by placing both as parts of a multipart/mixed message. The results are depicted in Fig. 2. A compliant mail user agent (MUA) should render this as both parts occuring in the order that they appear in the message: ham first, then the spam. This is relatively unobtrusive as the spam is effectively merely appended to the ham. It is easier to counter, because each multipart section can be classified seperately and the spam section can be fairly easily removed. Furthermore, if the anti-spam system in use supports user voting, the user may not vote this spam, as he/she may simply ignore the spam section.

A simple variant would be to place the spam first so that it gets rendered first, which would be more obtrusive, but just as easy to counter. A more malicious and blatant approach is to use multipart/alternative. Most MUAs will display just the spam section if it is placed last with an identical content-type as the ham. This is far more disruptive as they will not see the ham and will also likely incorrectly vote it spam. Correct classification by section and removal of the spam will be necessary before the user sees the message.

Other multipart types like digest and parallel can be misused in similar ways.

CEAS 2006 - Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, California USA

```
From: Alice
To: Bob
Subject: Good to hear from you again

We should chat more often, Bob.

Cheers, Alice
--
Get your free email account
at http://free_email.nu
```

Figure 1: Advertising in the signature

```
From: Alice
To: Bob
Subject: Good to hear from you again
Content-Type: multipart/mixed; boundary="break"

--break
Content-Type: text/plain
We should chat more often, Bob.

Cheers, Alice
--break
Content-Type: text/plain

Check out http://sna.fu for all your needs
--break--
```

Figure 2: Spam appended as multipart/mixed

More problematic is if the spam is inserted directly into the ham. Not only is the spam/ham differentiation more difficult for the classifier, but the user may be 'blinded' by the spam and not see the ham, resulting in an incorrect spam vote.

If the objective of the spammer is to get a user to click on a link, then all that really is necessary is to modify an existing link in an HTML email. A filter that uses blacklisted URLs will mark this as spam. It is also impossible to repair this message as the original link contents is gone.

Getting a user to open an attachment, such as a malware executable, also becomes far easier with this approach. With proper setup, password protected attachments that avoid anti-viruses could be more effective than current email worms that try this, like Bagle.

There are a myriad variations on these attack patterns, and we do not consider it necessary to cover them all.

3. LIKELIHOOD OF P-SPAM

The main problem with p-spam, it might appear, is the extremely reduced volume of email, as the rate of spam is bound to the rate of email seen by the zombie.

However, if p-spam is more effective than traditional spam—if using p-spam has a higher return on investment than traditional spam—the use of p-spam is a win for the spammer.

P-spam also requires code that can hook into the zombie in order to observe email traffic. This is far from impossible, but requires additional work.

In conclusion, it is likely that this method will be tried when other methods no longer provide the return on investment the spammers require.

4. COUNTERMEASURES

Although this paper is a problem statement and does not try to solve the problem, we will speculate briefly on what methods could be used to counter p-spam.

MUA/MUA authentication cannot be relied on to protect against a p-spam attack, because Fitzgerald [3] makes the valid point that once malware has compromised a machine all bets are off. Server/server authentication such as SPF [5] and DKIM [1] will also not help, as the message will pass though legitimate channels going from legitimate user to legitimate user.

Attacking the bot problem would be the most positive way of combatting the problem. Programs like Nessus[1] can be used to detect backdoor trojans, but as p-spam bots do not necessarily require direct control and can be made highly adaptable, use of this method is restricted. The same problems apply to network intrusion detection systems such as Snort[2].

More promising is the idea described by Hershkop et al. [4]. This method provides some context to the spam by profiling normal email usage over time. Whereas, Hershkop et al. proposed profiling sender/recipient traffic patterns, the concept could be extended to include structural characteristics of the email ('user X never sends HTML email', for example). These methods are known in the intrusion detection community to have their problems with false positives and false negatives.

By itself, profiling will not help us find which sections are spam and which as ham. Existing spam classifiers can be adapted to detect regions of spamminess and hamness. In combination there may be enough information to tell spam sections from ham.

The next challenge will be to teach classifiers not only to return a score of spamminess, but the modified message with the spam removed (in the same way as virus filters do with email today).

5. CONCLUSIONS

It may well be that this method will never take off, which would be something of a relief. It would certainly require a change in spammers' current practices. However, spammers have proven very adaptable and will use any loophole they can find. If we counteract all current spam methods effectively, they will use new ones, and p-spam is a candidate. The risk for the anti-spam community in not preparing for the potential p-spem problem is that some fundamental technology in email filters need to be modified and it may be too late to start the work when the problem arises.

6. ACKNOWLEDGMENTS

In particular, we would like to thank Jason Crawford, Richard Segal and Mark Wegman of the SpamGuru team at IBM T. J. Watson Research Center, as well as the members of the CARO spam list, in particular Nick Fitzgerald.

7. REFERENCES

[1] E. Allman, J. Callas, M. Delany, M. Libbey, J. Fenton, and M. Thomas. Domainkeys identified mail signatures (dkim). Technical report, IETF, Feb. 2006.

[2] J. Aycock and N. Friess. Spam zombies from outer space. Technical Report TR 2006-808-01, University of Calgary, Computer Science, Feb. 2006.

[3] N. Fitzgerald. Why 'user authentication' is a bad idea. In H. Martin, editor, *Proceedings of the Virus Bulletin Conference 2006*, Oct. 2006. Only the abstract was published.

[4] S. Hershkop. *Behavior-based Email Analysis with Application to Spam Detection*. PhD thesis, Columbia University, 2006.

[5] M. Wong and W. Schlitt. Sender policy framework (spf) for authorizing use of domains in e-mail, version 1. Technical report, IETF, June 2005.

[1] see http://www.nessus.org

[2] see http://www.snort.org