Paper presented at 2nd Workshop on Practice and Theory of IT Security, PTITS'2008
Satellite workshop of MCETECH 2008

Analysis of massive backscatter of email spam

Christopher. P. Fuhrman
École de technologie supérieure (ETS), Department of Software and IT Engineering, Montreal, Canada,
e-mail: *christopher.fuhrman@etsmtl.ca*

***Abstract*—Email backscatter is a side effect of email spam, viruses or worms. When a spam or virus-laden email is sent, it nearly always has a forged sender address. If this email fails to reach its recipient, e.g., because the recipient's mailbox is full or the recipient has set up an out-of-the-office auto-responder, the recipient's mail system may attempt to send an automated reply to the forged sender. This creates an unsolicited message, or more generally an email backscatter, which will be sent to the forged sender. On massive email spam runs where the same address is forged as the sender, there can be significant backscatter to the forged address. This may sometimes result in a denial of service, because the victim's inbox or mail system is flooded with backscatter. We consider potential value in the analysis of email backscatter, for example, the prompt detection of zombies that send the spams. We present the results of an analysis approach in the context of an ongoing email backscatter incident. We present results of our approach's effectiveness at finding zombie IP addresses compared to using popular DNS Block Lists. Our results support a hypothesis that spammers limit individual zombie activity by "pulsing" them, perhaps to reduce the chance of a zombie appearing on a block list.**

***Keywords*—Email backscatter, email spam, delivery status notification, zombie detection, DNS Block Lists.**

I. INTRODUCTION

Email spam is a problem that has been addressed from many angles, yet continues to perturb users by arriving in our inboxes and information systems. Most email spams today are sent by zombie computers that are part of so-called botnets [1], i.e., systems that have been compromised by virus-writers and are under the control of criminals. As the scale of the spam problem grows, the anti-spam and anti-virus infrastructure has to keep pace. This has resulted in an "arms race" of spamming [2] and anti-spamming [3] technologies that shows no sign of slowing down any time soon.

A botnet can be seen as a type of illegal Internet Service Provider (ISP) that spammers can use to send emails, to host web sites for their products, etc. Because a single botnet is comprised of thousands of computers from all over the world, it has no geographical boundaries. This characteristic makes it easy for spammers and botnet herders to hide their activities. This is particularly advantageous to them when the computers on a botnet are using a legitimate ISP that is not proactive in stopping or preventing zombie activities on its network.

Because ISPs do not always isolate or disable the zombies on their networks quickly enough, several Domain Name Service (DNS) Block Lists (DNSBL) have been created that keep track of the Internet Protocol (IP) addresses of known zombie machines. DNSBLs are updated dynamically, drawing their information from various sources including spam-trap addresses (bl.spamcop.net), email server logs (backscatterer.org), and user input (njabl.org). DNSBLs can be queried in real time, using the DNS protocol, to determine if a suspected IP address is that of a zombie. This is useful, for example, when an email server receives a connection from a computer and it wants to determine if the sending computer might be a zombie. If the connecting computer's IP address is on a DNSBL, the email server can refuse the connection, or perhaps just use the fact to determine the likelihood that the email it receives is a spam. Studies have been done [4] and are ongoing regarding the anti-spam effectiveness of the DNSBL approach.

It has been observed [3] that spammers try to minimize the on-line footprint of their zombies by a technique described as *pulsing*. In this technique, spammers use individual zombies to only send a short "burst" of spams, possibly to keep the zombies' IP addresses off of DNSBLs.

Spammers use other techniques to try to make their emails look legitimate. This includes using random words in the messages to try to fool Bayesian spam filters, misspelling commonly used words in spam messages, etc. Spammers also attempt to forge SMTP "Received" headers in the headers of the spams [5], possibly to make the spam injection point more difficult to determine [6].

Spammers can also forge the sending "From:" address of spams, which can result in email backscatter. Backscatter is defined as automatically generated emails that come from email servers, but which are a result of a forged sender address in a spam. For example, a spam sent with a forged "From:" address to an invalid "To:" address might bounce back to the forged sender. The "bounce" is typically an automated informational message that the email could not be delivered. Backscatter is typically a small percentage of spam that an average user sees, representing as little as 1% of the spam [7].

The choice of sender address forged by spammers might be considered arbitrary. However, in some cases spammers have intentionally and repeatedly forged sender addresses to harm the reputation of companies and individuals to make it look as if these targeted individuals were behind the spamming operation. This type of attack is known as a *joe job*, because of the infamous incident that occurred to damage the reputation of the web site *joes.com* [8]. As we

will explain later, massive backscatter can result in a denial of service of the email system of the forged sender.

Another hypothesis as to why spammers are now forging real (as opposed to arbitrary) sender addresses is to evade what is known as Sender Address Verification (SAV). In SAV, a mail server that receives an incoming message will try to validate the sender's address with a probe sent via SMTP to the supposed sender's email system. If the forged sender address is not valid, the SAV approach will detect it. Seen from this perspective, SAV and the subsequent forging of real sender addresses are examples of "escalations" of the spam "arms race".

We have access to an email address that has been receiving massive backscatter for at least six months. That is, spammers have been forging this email address in their spam runs since June, 2007, resulting in sometimes more than 900 backscatter emails per day being sent to this address. At first, we thought it to be a *joe job* attack, but evidence found on the Usenet group news.admin.net-abuse.sightings has shown that it is probably a more general spammer tactic that leads to massive backscatter as a secondary effect. The intention of this paper is to consider ways of turning a possibly bad situation into a useful one. That is, to consider the usefulness of analyzing massive email backscatter.

This paper is organized as follows. Section II presents an overview of the email backscatter phenomenon, including causes and the controversy of how backscatter email is considered itself as spam by many users on the Internet. Section III presents the limits and potential value of analyzing email backscatter. Section IV presents some results of an analysis done on the massive email backscatter we have seen at one email address. Section V discusses the conclusions and future directions of our work.

II. Email Backscatter

As illustrated in Fig. 1, email backscatter occurs when spammers forge the sender address in their massive spam runs, resulting in "bounce" messages and other automatic replies arriving in the in-box of the user whose email address was forged.

A backscatter message is always a message caused by an auto-responder. It can have several forms:

- Delivery Status Notification (DSN), indicating the message couldn't be delivered because of a problem, e.g., the destination address was not valid, the user's mailbox was full, etc.;
- Out-of-the-office message, indicating the user at the destination address is not present during a certain time;
- Verification challenge messages, requesting that the sending user verify himself, as part of a challenge/response anti-spam system;
- Anti-virus indication, indicating the message was detected as containing a virus and was blocked;
- Anti-spam indication, indicating the message was detected as spam and was blocked.

A. *Delivery Status Notifications (DSN)*

DSN messages are typically sent to the sender of the message, i.e., the address in the "From:" field. The format of DSN messages is formally specified by Request For Comments (RFC) 3464 [9]. Depending on the mail system that sends them, DSN messages contain some information

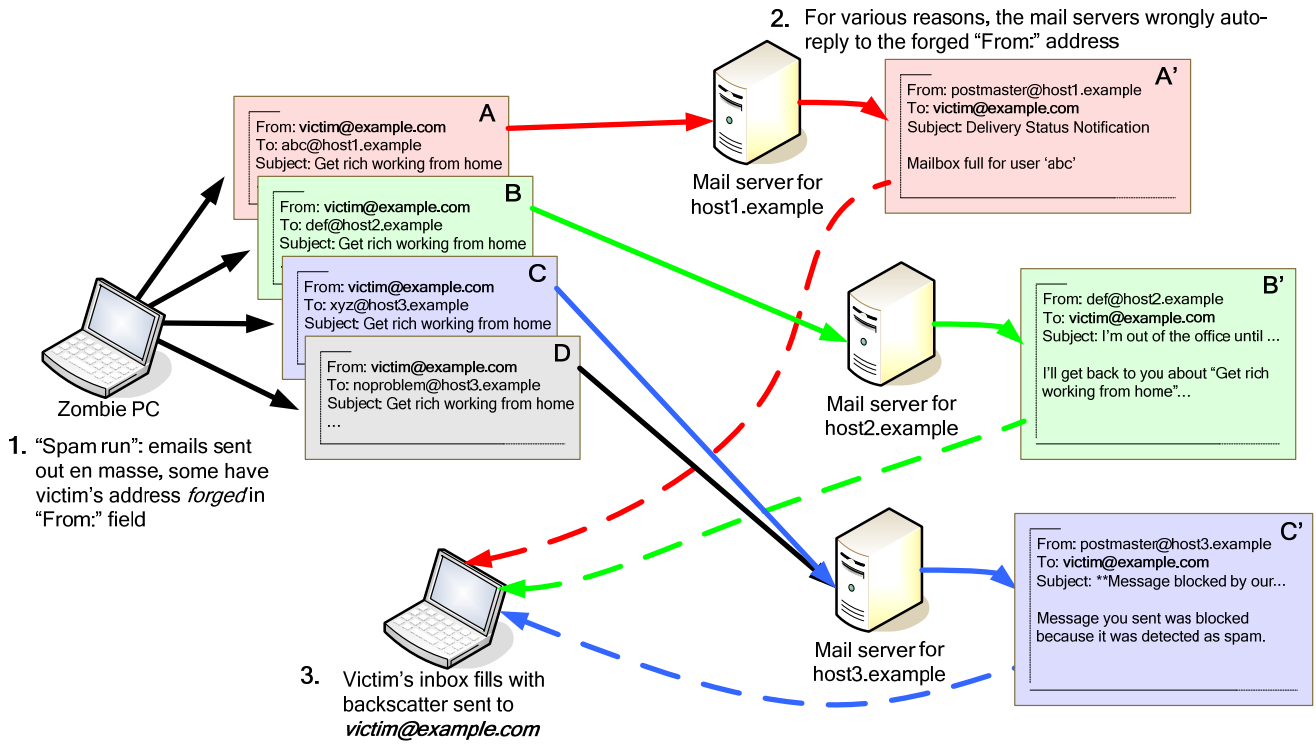

Fig. 1 – Anatomy of email backscatter due to spamming

about the original email, such as the subject, part of the original message, and sometimes the original RFC822 [10] headers. In reality, however, the DSN messages one sees as backscatter do not always adhere to the RFC3464 specification. Furthermore, DSN messages are not supposed to be sent to the "From:" address in the originating email. It states in Section 4.1 of [9]:

> *"Implementers are cautioned that many existing [Mail Transfer Agents] will send non-delivery notifications to a return address in the message header (rather than to the one in the envelope), in violation of SMTP and other protocols."*

However, even if DSN messages are sent to the SMTP envelope return address, it is still possible for systems to forge this address.

To prevent backscatter, it is recommended that mail servers only generate DSN messages to local users. Furthermore, a mail server that accepts an SMTP connection should verify that the connecting system is trustworthy. If a problem occurs such that the mail server cannot deliver the message (e.g., the local user does not exist), the server should reject emails during the SMTP connection [11], leaving the problem of DSN to the connecting system. This approach reduces email spam backscatter, because zombie computers that are used in spamming don't generate DSN messages when a spam message is rejected during SMTP.

Finally, the spam message that provokes a DSN in a backscatter scenario will have possibly passed by a spam firewall. Such firewalls often allow spam messages to pass through, but tag them as spam. The subject of such messages will often contain the word "spam" at the beginning. Some firewalls even add other information such as a spam "score" or concrete reasons why the email was considered spam by the firewall. For example, a subject may be appended with the text "Sending mail server found on xbl.spamhaus.org" to indicate that the sending mail server was on a block list and probably a spam-sending zombie.

B. Out-of-the-office (OOO) messages

Backscatter can originate from a so-called vacation auto-responder that replies indiscriminately to all incoming emails to notify the senders of such emails that the addressee is on vacation or is out of the office and will return by a certain time. Such messages typically contain at most a mention of the subject of the original message, and almost never contain information about the RFC822 headers.

OOO messages, besides being a source of backscatter, can also be a security risk. If a criminal can learn of the duration of a person's absence, he could potentially use that information to his advantage. Some mail systems allow configuring of OOO messages such that they are only sent if the "From:" address of the incoming message is in the address book of the user. Other systems administrators dissuade users from using OOO messages, instead requesting that vacationing users inform all of their contacts of their absence prior to departure.

C. Verification challenge messages

Some anti-spam systems are built on a so-called challenge/response (C/R) approach, in which any incoming message that is not from a trusted sender is immediately replied to, "challenging" the user to verify himself. The verification can be done by simply responding to the challenge message, or by clicking on a hyperlink within the challenge message, depending on the verification system. Although they may be quite effective at not allowing spam to enter into the mailboxes of the systems they protect, they potentially create a lot of backscatter for others, particularly since spammers forge the sender address.

Several problems exist with this approach. First, what happens when a user who has a C/R system attempts to contact another user who has a different C/R system? In theory, the two will never see each other's messages, since the first challenge will be challenged, etc. Another problem occurs when a misdirected challenge (backscatter) is acknowledged. In such a case, the spam, which wasn't sent by the user who was challenged, will be delivered to the C/R user's inbox, thus defeating the C/R system.

D. Anti-virus indications

When a mail server scans its incoming messages for viruses, it can choose to inform the sender that the message contained a virus and was not delivered. Again, if the sender is forged (and such is quite often the case in virus-laden emails), a backscatter message is created. Because of the scale of some virus-based outbreaks, e.g., Sobig.F, most anti-virus companies disable the auto-reply feature [12] of their products.

E. Anti-spam indications

Anti-spam firewalls (also known as "appliances") are becoming commonplace. They work in a similar fashion as anti-virus systems, only they scan incoming messages to determine if they are spam. It turns out that the same problem of backscatter exists in these systems. That is, they reply to forged senders of spam, telling that the message was not delivered because it was identified as spam.

This auto-reply feature is useful in the event that a non-spam message would be mistakenly not delivered, since the legitimate sender is informed (although it's not clear what he can do about it!). However, the backscatter caused by this feature is inconvenient to a much higher number of users whose email address happens to be forged in the spams.

F. Controversy about backscatter being spam

Since one definition of spam is any unsolicited (bulk) email, we can see how email backscatter can be considered by some to be a form of spam. There is even a DNSBL, ips.backscatterer.org, dedicated to listing sites that have produced massive amounts of backscatter. SpamCop.net allows users to report backscattered messages as misdirected bounces, which it considers to be a form of spam. Such backscatter messages can potentially get an email system listed on the SpamCop block list, bl.spamcop.net.

As mentioned above, DSN messages are required by the email-related RFCs. As such, these RFCs do not disallow a

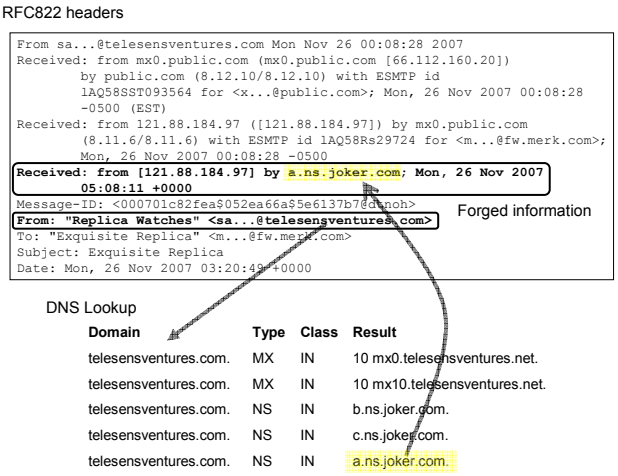

Fig. 2 – Relationship between forged sender and forged Received header

mail system to auto-reply to the sender address, even though this may be forged. In fact, the accepted email-related RFCs do not address a major problem with email today: forged sender addresses. Spammers have exploited this loophole for a long time. Just as non-authenticated email relays had to implement local-user authentication as a result of spammers exploiting them, perhaps email relays will also have to adapt to avoid sending DSN messages to forged senders. The block lists of open relays used by spammers were a tool that got administrators to update their non-authenticated mail relays to authenticated ones. Perhaps, then, the use of block lists will effect a change on the mail servers that cause backscatter. Any systems that have a massive volume of email, e.g., AOL.com, typically do not generate email backscatter, simply because it's too costly.

III. Potential Value in Analyzing Email Backscatter

Email backscatter might be interesting to analyze for several reasons. First of all, the potentially rapid response of backscatter messages might be useful to acquire information about current spam runs. Since DSN messages contain RFC822 headers, it is possible to obtain the IP addresses of the spam-sending zombies, and this can be done relatively quickly when a message arrives. This could be potentially useful for reporting these zombies to DNSBLs.

It might be possible to identify the start (and end) of a spam run using the timing information, assuming that spam runs are done with the same message subjects. This could be useful to government authorities as forensic information if prosecution of suspected spammers or botnet operators is sought. Because backscatter comes from different systems as a result of a small number of spammers, it represents a snapshot of those spamming activities taken from multiple perspectives.

Because typical DSN messages contain the spam's subject and body text, they could be used to feed spam-filtering systems that rely on training of actual spams. Possibly this time-sensitive information could improve the spam detection of these systems [13].

The difficulty with analyzing email backscatter is to acquire a sufficient volume of it. One has to have access to an email address that spammers are frequently, if not intentionally, forging as a sender in at least some of their spam runs. Alternatively, one could do an analysis of mail server logs of the domain of a forged address that is invalid, e.g., bogus_user@example.com. The backscatter sent to the invalid address bogus_user would in fact fail to be delivered at the domain example.com, and the mail server might contain information about the failed messages (including the messages themselves).

This idea brings up yet another consequence of forged sender addresses in spams: backscattered backscatter messages. That is, when a forged sender address is also bad, the bounces send to that address could potentially bounce back to the sending auto-responder. We do not believe there is any value in analyzing this kind of message, since it would only contain information about the failed delivery of a DSN, OOO, etc.

IV. Results of Analysis

A. Background

Since June, 2007, we have been receiving massive email backscatter messages, sometimes more than 900 per day, to a single email address, which we shall refer to as the *victim address*. Upon first receiving the backscatter, we assumed it was a *joe-job* attack, attempting to tarnish the reputation of the victim address. The backscatter messages can arrive quite rapidly, sometimes as many as 10 per minute. Early on in the incident, it was observed that the subjects of many of the spams that caused the backscatter were often identical. For these reasons, we considered that perhaps there may be some use in analyzing backscatter. In many of our particular backscatter messages in the beginning, we found a forged "Received" header that nearly always matched a certain pattern. This behavior changed over the span of our analysis, and in the last weeks of it, we saw very few, if any, of such forgeries.

Since spammers often send the same spam messages to many people, we decided to look for evidence of similar messages reported by other users. The Usenet group *news.admin.net-abuse.sightings* contains samples of spam messages that can be posted by users anywhere on the internet. A search with Google of this newsgroup found that the nearly identical spam messages were appearing on other domains. We also found that they had a similar forged "Received" header.

We determined that the forgeries of the "From" address and the final "Received" header are in fact related in a large number spams in the early stages of this particular attack. When forging the spams, the spammers were apparently doing a DNS lookup for the MX records of the domain name in the "From" address, and using one of the results as the host name in the Received line. An example of this relationship is shown in Fig. 2. Based on this relationship and the presence of other "victim" addresses in the sightings newsgroup, we have determined this is likely not a joe-job attack. It is probably a side effect of a spammer forgery technique.

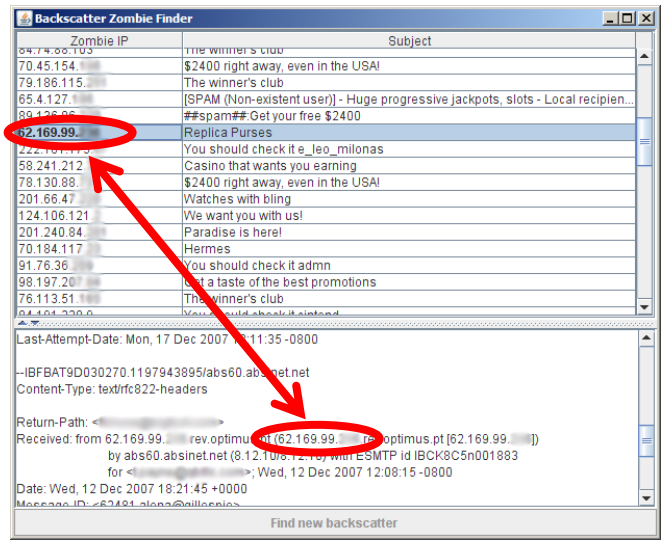

Fig. 3 – Screenshot of real-time backscatter analysis tool

Our victim address is in fact a free mail alias on a popular Internet mail service, which can be configured to redirect emails to the address of choice. For the purpose of our analysis, we have chosen to redirect these emails to Google Mail (Gmail) because it provides an easy access to emails through POP3 and IMAP protocols. Furthermore, Gmail recognizes most backscatter as spam, and classifies it as such quite reliably.

Based on anecdotal accounts on internet news groups and discussion forums, we know that other users experience massive backscatter of this same nature. However, so far we have only studied the incident involving our victim address.

B. Analysis method

Our real-time analyzer is a small program using the JavaMail library that connects via IMAP to a Gmail account's Spam mailbox. It is able to process the backscatter messages as they arrive. Upon arrival of new messages in the Spam mailbox, the analyzer first validates that the email is a backscatter, and then searches (using a regular expression) for either the forged or the final "Received" header in the MIME attachment for RFC822 headers. If it finds a useful header, it extracts the IP address of the zombie used to send the spam, and checks to see if it is on any of the four DNSBLs: cbl.abuseat.org, bl.spamcop.net, dnsbl.njabl.org and list.dsbl.org. A screenshot of the tool is shown in Fig. 3. It demonstrates an example of a backscatter message caused by a spam (whose subject was *Replica Purses*) sent by a zombie whose IP was in none of the DNSBLs and thus shown in bold. The tool displays the subject and RFC822 headers of the spam message that provoked the backscatter, if that information is included in the backscatter. It also groups and highlights the email backscatter messages that are traced to the same zombie.

To study zombie activity over a longer period of time, we did another analysis of a collection of 49,000 emails received at the victim address that were classified as spam by Gmail's filtering method. For the most part, these were backscatter emails. But there were the occasional direct spams, for which we did not attempt to detect zombie IPs, even though this is

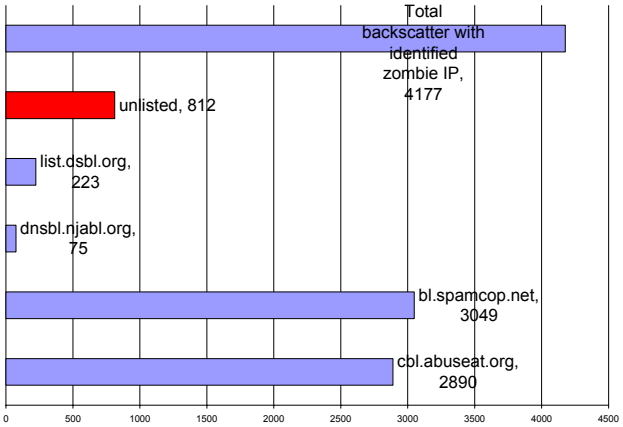

Fig. 4 – DNSBL success for zombie IPs found via backscatter analysis

possible. Furthermore, not all backscatter has useful information for detecting zombies. For example, a vacation auto-responder, verification challenge or even some DSNs do not include the RFC822 headers of the original message (the spam). As such, they can be analyzed to trace zombie activity. In these cases, and in cases where the injection IP address is part of a private network (e.g., 127.0.0.1), our analyzer classifies the zombie IP as "uncertain".

C. Results

Because our analysis technique potentially can receive backscatter within minutes (if not seconds) after a spam has been sent, the identification of zombie IP addresses can potentially be very quick. Fig. 4 shows the results of a sample of email backscatter messages analyzed over a period of roughly 5 days. We found that 19.44% of the IP addresses of zombies found in our analyses were not on any of the four DNSBLs during the processing time of the backscatters.

We also analyzed the activity of the individual zombies used in the spam runs that provoked our email backscatter. In our data set, which included backscatter received over roughly four months (49,000 backscatter messages), we found that there were 42,797 identified zombies IP addresses, and 37,982 of them were unique (89%) as shown in the first part of Fig. 5. In the 4815 cases where backscatter resulted from a zombie IP that had already been identified, we found that in a large number of the cases, it was a question of a zombie being spotted on the same spam run. We reached this conclusion because the delta received time of the messages was small (less than one hour) and the subject of the spam was identical. These are multiple

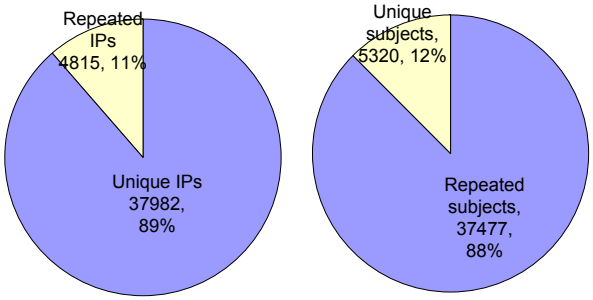

Fig. 5 – Uniqueness of identified zombie IP addresses and spam subjects

backscatter messages received from a single zombie's *burst*. The second part of Fig. 5 shows that the spams with identified IP addresses had a subject that was seen more than once 88% of the time. To normalize the data regarding subjects, we removed as much as possible any annotations or spam tags found in the subjects. Because there is no standard way of tagging a spam, this operation was not perfect. Therefore the value of 88% is likely below the real value.

Finally, these results are unique to our study, and may not be indicative of results from other analyses of email backscatter. Because our victim address was being specifically used by spammers in their spam runs, it is difficult for us to validate our analysis on other cases. Furthermore, it is nearly impossible to know how often spammers forge real email addresses in their spam runs. Therefore, it is very difficult to estimate how representative the results of our study are.

V. Conclusion

In this paper we have presented an overview of email backscatter as well as the results of a limited analysis done on actual backscatter messages. We have shown that there is potential value in analyzing backscatter in real time, as it can potentially yield a prompt identification of zombie IP addresses not already on existing DNSBLs.

We have also shown that in the scope of our backscatter incident, individual zombies appear only once for the most part. There are several explanations possible for this phenomenon. It could be that these zombies are used (rented) by different spammers, and not all of them forge the same sender addresses in the backscatter. It could also be that zombies are rotated in and out of spam sending, in an attempt to keep their IP addresses off of the DNSBLs. Another explanation is that zombies are taken out of commission by their ISPs once they are detected.

The number of unique zombie IP addresses detected in our analysis might indicate the size of the botnet(s) involved. However, we are unable to know from our study how many botnets or groups of spammers were involved. Although, we can assume these numbers are small based on the similarities of the spam emails.

For any empirical study, it is important to know if the results are repeatable. Because we do not control the use of email addresses in a spam run, it is very difficult to repeat our experiment outside the context of our study. However, as future work, we have considered alternative strategies involving catch-all email addresses at specific domain names that would be created to catch email backscatter for analysis.

Alternatively, it would not be difficult to analyze backscatter that comes in to existing email domains, provided one has access to administrator-level email logs or mail boxes containing bounced emails. On the other hand, this would likely pose ethical problems if the emails are addressed to real users. In our study, it was the author's personal email address that was receiving the backscatter.

For other future work, it might be interesting to consider how long it takes an IP address to end up on a DNSBL after we have detected it with our backscatter analysis. We have approached owners of several of the DNSBLs we used in this study to offer to contribute the IP addresses of zombies we identify with our method. As of the time of this article, none of the list owners have taken us up on this offer. One of the problems with this kind of contribution is the verification of the validity of the information.

Acknowledgment

C.P.F. thanks Jean-Marc Robert for discussing some of these ideas with him and for encouraging him to continue with the ideas communicated in this paper.

References

[1] N. Ianelli and A. Hackworth, "Botnets as a vehicle for online crime," 2005. [Online]. Available: www.cert.org/archive/pdf/Botnets.pdf. [Accessed: 7 Dec. 2007].

[2] K. Heyman, "New Attack Tricks Antivirus Software," *Computer*, vol. 40, no. 5, 2007, pp. 18-20.

[3] N. Leavitt, "Vendors Fight Spam's Sudden Rise," *Computer*, vol. 40, no. 3, 2007, pp. 16-19.

[4] A. Ramachandran, et al., "Can DNS-based blacklists keep up with bots?," *Proc. Third Conference on Email and Anti-Spam*, 2006, pp. 55-56.

[5] Michigan State University Computer System and Network Abuse, "Analyzing e-mail headers and tracking e-mail," 17 May, 2004. [Online]. Available: abuse.msu.edu/email-tracking.html. [Accessed: 11 Dec. 2007].

[6] J.S. Park and A. Deshpande, "Spam detection: increasing accuracy with a hybrid solution," *Information Systems Management*, vol. 23, no. 1, 2006, pp. 57-67.

[7] V.C. Gordon and R.L. Thomas, "Online supervised spam filter evaluation," *ACM Trans. Inf. Syst.*, vol. 25, no. 3, 2007, pp. 11.

[8] J. Doll, "Spam attack!," 1997. [Online]. Available: www.joes.com/spammed.html. [Accessed: 11 Dec. 2007].

[9] K. Moore and G. Vaudreuil, "An Extensible Message Format for Delivery Status Notifications," 2003. [Online]. Available: www.ietf.org/rfc/rfc3464.txt. [Accessed: 14 Dec 2007].

[10] D.H. Crocker, "Standard for the format of ARPA Internet text messages," 1982. [Online]. Available: www.ietf.org/rfc/rfc0822.txt. [Accessed: 14 Dec 2007].

[11] M.N. Marsono, et al., "Rejecting Spam during SMTP Sessions," Proc. Communications, Computers and Signal Processing, 2007. PacRim 2007. IEEE Pacific Rim Conference on, 2007, pp. 236-239.

[12] F. Skulason, "Why (some) anti-virus companies are to blame for the recent e-mail flood," 2003. [Online]. Available: www.f-prot.com/news/gen_news/030910_open_letter.html. [Accessed: 17 Dec. 2007].

[13] P. Cunningham, et al., "A case-based approach to spam filtering that can track concept drift.," *Proc. the 5th International Conference on Case-Based Reasoning (workshop on Long-Lived CBR Systems)*, 2003.