

Detecting Spamming Activities by Network Monitoring with Bloom Filters

Po-Ching Lin*, Ping-Hai Lin*, Pin-Ren Chiou*, Chien-Tsung Liu[†]

*Department of Computer Science and Information Engineering

National Chung Cheng University, Chiayi, Taiwan 62102

Email: {pclin,lph99m}@cs.ccu.edu.tw, littlecho@cloud.littlecho.tw

[†]CyberTrust Technology Institute

Institute for Information Industry, Taipei, Taiwan 10622

Email: netsaga@iii.org.tw

Abstract—Spam delivery is common in the Internet. Most modern spam-filtering solutions are deployed on the receiver side. They are good at filtering spam for end users, but spam messages still keep wasting Internet bandwidth and the storage space of mail servers. This work is therefore intended to detect and nip spamming bots in the bud. We use the Bro intrusion detection system to monitor the SMTP sessions in a university campus, and track the number and the uniqueness of the *recipients' email addresses* in the outgoing mail messages from each individual internal host as the features for detecting spamming bots. Due to the huge number of email addresses observed in the SMTP sessions, we store and manage them efficiently in the *Bloom filters*. According to the SMTP logs over a period of six months from November 2011 to April 2012, we found totally 65 dedicated spamming bots in the campus and observed 1.5 million outgoing spam messages from them. We also found account cracking events on 14 legitimate mail servers, on which some user accounts are cracked and abused for spamming. The method can effectively detect and curb the spamming bots with the precision and the recall up to 0.97 and 0.96.

keyword: spamming activities, network monitoring, botnet, Bloom filters, detection.

I. INTRODUCTION

Email delivery has become an indispensable approach to communications in daily life. Due to its popularity and nearly zero cost, it is commonly exploited to carry advertisements, malware, phishing messages, and so on. According to a recent report from [1], around 90% of email messages are unsolicited commercial ones, namely *spam*. Even though modern spam filtering techniques can filter out spam with high accuracy and rare recipients click the links in the spam messages, this problem persists because spammers can still capitalize on spamming due to the huge number of spam messages [2].

Spammers turn to the botnet infrastructure to efficiently deliver spam nowadays. Botnet is a collection of compromised hosts, namely *bots*, commanded by a bot master to perform malicious activities such as spamming. Delivering spam through *spamming botnet* has the following advantages:

- 1) An individual bot does not have to deliver a large number of spam messages, and thus can reduce the chances of being detected. The total number of spam messages is still huge due to the large number of bots.

- 2) The botnet can rapidly infect more systems under control for spamming, making blocking spamming bots by IP blacklisting ineffective [3]. For example, a system may get infected if the user clicks a harmful link or executes the attached malware in a spam message [4].

Most practices to reduce spam are filtering on the receiver side. Common solutions include cloud-based mail security products such as Symantec MessageLabs and Google Postini, as well as personal security products such as Kaspersky Internet Security and Avast Internet Security. Mail clients such as Microsoft Outlook and Mozilla Thunderbird, as well as mail service providers, also support spam filtering. The solutions receive mail before filtering, so spamming activities still exist, and spam messages still waste Internet bandwidth and the storage space of mail servers.

We consider that if the spamming bots can be detected and cracked down on the sender side as early as possible, the number of spam messages can be significantly reduced. A crackdown of the Rustock botnet in 2008 alone once temporarily reduced 75% of the total volume of Internet spam (en.wikipedia.org/wiki/Rustock_botnet). This case demonstrated that cracking the spam sources, if it succeeds, will be rather effective to reduce the number of spam messages.

We deployed the Bro network intrusion detection system (NIDS; see www.bro-ids.org) to monitor the outgoing SMTP sessions initiated from the internal hosts in the campus of National Chung Cheng University over a period of six months from November 2011 to April 2012. Bro can record the SMTP sessions and extract the *recipients' email addresses* (REAs) from each session. The rationale behind the detection method is simple. The REAs from a spamming bot tend to be unique to each other to diversify the recipients of spam messages, while those from a normal user tend to be repetitive because they usually belong to familiar persons. We therefore conduct the statistical analysis of the number and the uniqueness of the REAs from the internal hosts within a period, and classify the hosts by the features. We use *Bloom filters* to track the REAs due to a large number of them. Moreover, we also study the cases of spamming through the legitimate mail servers. The contributions of this work are summarized as follows:

- 1) We present an simple yet effective detection method with high accuracy based on the diversity of REAs. This method is proved to be effective in a real environment.
- 2) The detection method found 65 spamming bots and observed around 1.5 million spam messages over a period of six months. The list of spamming bots was reported to the network administrators in the computer center for them to investigate and crack down the hosts.
- 3) The detection method also found account cracking events on 14 mail servers in the campus. The events are critical, and should be detected and cracked down like spamming bots.

The rest of this paper is organized as follows. In Section II, we review prior studies of detecting spamming botnet. In Section III, we describe the network monitoring and analyze the diversity of REAs from each individual internal host in the detection method. In Section IV, we analyze and verify the correctness and accuracy of this method. We conclude this work in Section V.

II. RELATED WORK

We focus on the studies of detecting spamming botnets rather than generic botnets in the review. For botnet tracking, the authors in [5] studied the command and control (C&C) operations of the MegaD botnet by reverse engineering, and revealed the operations such as spam composition and delivery. The authors in [6] also presented the Botlab architecture to launch various spamming bots in a controlled environment for studying their behavior. Unlike the studies, this work passively monitors suspicious SMTP traffic without interacting with the malware binaries, so the bots are unaware of being monitored and will behave as usual without any evasion.

Besides active botnet tracking, passively monitoring and analyzing network traffic are common for botnet detection. A common assumption is that the bots in the same botnet will exhibit similar behavior. SpamTracker in [3] is a behavioral blacklisting algorithm to cluster the hosts having similar patterns of recipients' domains. The methods in [4] and [7] look for similar mail content, delivery time and so on to detect spamming bots in the same botnet. BotMagnifier in [8] can detect spamming bots that behave similarly to the initial set of seed hosts derived from the source hosts delivering spam messages with similar subject lines or destined for similar IP addresses. A common limitation of these methods is that they can detect only spamming bots behaving similarly in an environment, and the assumption of similarity may not hold with evasion.

BotGraph [9] can detect abuse of web-based email accounts for spamming. The authors correlated the user accounts behaving similarly with a graph algorithm, and identified the abuse on Hotmail. In contrast, this work aims at detecting spamming bots, not web-based email accounts abused for spamming. SPOT in [10] can redirect outgoing messages to a spam filter, and use the sequential probability ratio test to detect whether an internal host constantly sends spam. We do not rely on an external spam filter for two reasons. First, an SMTP session

is likely to fail in the transaction due to blacklisting or invalid recipients, and not even a spam message will be sent out in the session. The spam filter thus becomes useless in this case. Second, a user may configure automatic forwarding on a mail server, which will forward the received mail, including spam messages, to an external account specified by the user. The spam filter will therefore see many spam messages forwarded from the mail server, but they are not originated from the server. The work in [11] assumes that most spamming bots are from end-user hosts, and separates them from legitimate mail servers with support vector machine (SVM). We find the assumption is not always true, since legitimate mail servers may also send spam messages due to account cracking.

III. DETECTION OF SPAMMING BOTS

We introduce the features to detect spamming bots and the design issues in this section. The rationales behind the design will be also discussed in depth.

A. Problem Analysis

According to prior studies such as [12], spamming bots derive the list of REAs from the botmaster, send spam to the recipients, and report back the delivery status to the bot master. The REAs in the list should be unique and diverse to efficiently distribute spam to a large number of recipients.

SpamTracker in [3] refers to the recipients' domains to detect spamming bots with the assumption that the bots in the same botnet will target at similar domains and form a large cluster. The assumption could be imprecise due to the popularity of some mail services such as Gmail and Yahoo, which own a huge number of users. It is likely that the recipients' domains are similar, but the REAs are mostly different. Moreover, the spamming bots may not send spam to similar domains due to deliberate evasions, or simply because they are not in the same botnet.

We consider the REAs rather than the domains to characterize the diversity of the recipients of the mail messages from each individual host. We analyze the numbers of total REAs (counting duplicated ones) against unique REAs (not counting duplicated ones) from a host over a specific period. A normal user usually sends mail to a relatively small set of familiar persons, such as colleagues, friends, students and so on. There are occasionally exceptions (e.g., when the user replies to queries from unfamiliar persons), but the recipients that the user sends to are largely fixed and are likely to appear more than once. In contrast, a spamming bot should deliver spam messages to a wide range of unique REAs for efficient spam distribution.

Figure 1 compares the numbers of total REAs, unique REAs and unique recipients' domains in the spam messages from the spamming bots we found. The former two numbers are mostly very close, while many of the recipients' domains are duplicated. Therefore, the REAs can reflect the diversity of the recipients better than the recipients' domains.

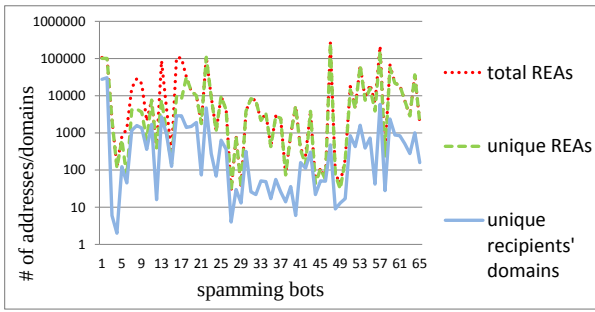


Fig. 1. The difference between the numbers of REAs, unique REAs, and unique recipients' domains.

B. Application of Bloom Filters

The number of REAs in the outgoing mail messages, especially spam messages, can be large, so an efficient data structure to store them is essential. We use a Bloom filter [13] to maintain the REAs from each individual internal host. A Bloom filter consists of an m -bit array to store n objects, which are the REAs of the outgoing mail messages in this work. The bits in the array are all initialized to 0. Each REA A_i of an outgoing message is stored into the Bloom filter for the host sending the message by setting the bits at the positions $h_1(A_i), h_2(A_i), \dots, h_k(A_i)$ to 1, where $h_1, h_2, \dots, h_k$ are k independent hash functions. We also check whether the same REA has been in the Bloom filter before storing A_i . If one of the bits at the positions $h_1(A_i), h_2(A_i), \dots, h_k(A_i)$ is equal to 0, A_i must be unique in the Bloom filter. Otherwise, A_i may have been in the Bloom filter because some of the bits may be set by the hash functions of the other REAs. Simply put, looking up the Bloom filter cannot result in any false negatives, but a false positive is probable. Suppose the mapping of each hash function is uniform, the false-positive rate [13] is

$$(1 - (1 - \frac{1}{m})^{kn})^k \approx (1 - e^{-\frac{kn}{m}})^k. \quad (1)$$

Given proper values of n and m , the false-positive rate can be minimized to approximately $(0.6185)m/n$ when $k = \ln \frac{2n}{m}$.

Notice that a Bloom filter may be filled up, if too many REAs appear. An expiration mechanism is therefore necessary. An intuitive idea is marking the storage time of each REA. If the lifetime of an REA is longer than a pre-defined expiration time, the address will be purged. Nevertheless, more than one REA may be mapped to the same bits in the Bloom filters. We cannot simply set the bits mapped from the REA to be purged to 0 because that may incorrectly purge another REA. The problem can be easily solved by counting Bloom filters presented in [14], but the required storage can be several times larger than the original Bloom filters.

We refer to the work in [15] to implement the expiration mechanism by creating two Bloom filters for each individual host. When an REA appears, the two Bloom filters are looked up simultaneously to see whether it is stored in either one. If the REA is unique, we update one of the Bloom filters by

adding the REA to it. When the Bloom filter for updating is nearly full, we reset the other Bloom filter, and switch their roles. In other words, the other Bloom filter becomes the one for updating and querying, and the original one for updating and querying becomes the one for querying only. Following the principle, we constantly switch the roles of both Bloom filters. The expiration mechanism requires only twice the memory space for the original design of Bloom filters, and saves more memory space than counting Bloom filters.

C. Classification of SMTP activities

According to our observations, the SMTP activities of a host can be classified into the following four types.

- 1) *Infrequent outgoing email deliveries and the REAs likely to repeat*: The host is likely to be normal. If a spamming bot behaves in this way, it must send spam very slowly to a relatively small set of recipients. Even though that can evade the detection, the capability of a spamming bot will be seriously restrained. Like the work in [10], we may additionally forward the mail messages to a spam filter in this case to check whether the host is spamming or not. The additional check will not burden the filter since the mail deliveries are infrequent.
- 2) *Infrequent outgoing email deliveries and the REAs unlikely to repeat*: The host can be considered a low-profile spamming bot that avoids drawing attention because the REAs that a normal host sends to are otherwise likely to repeat according to our preceding discussions. For example, we found a low-profile bot that sent only 393 spam messages in a month, and the number of unique REAs of these messages is also 393. The recipients never repeat, which is unusual in a normal case.
- 3) *Frequent outgoing email deliveries and the REAs likely to repeat*: The host is likely to be a mail server with many users sending mail through it. The high repetition of REAs implies the users on the mail server are fixed. We will discuss the case in which spammers crack normal accounts on the mail servers in the later sections.
- 4) *Frequent outgoing email deliveries and the REAs unlikely to repeat*: The host is likely to be a spamming bot, even though we found that relatively few normal hosts behave in this bot-like manner, e.g., a host that regularly delivers birthday cards to the alumni. Since such false positives are few and fixed, the hosts can easily be excluded by a white list after investigation and will not bother the detection in practice.

D. Detection of spamming through the mail servers

We also discuss detecting spamming through the mail servers. A spamming bot may send spam through a normal mail server in the following three ways.

- 1) A mail server becomes an open mail relay due to poor configurations. Internet users, rather than just those from the permissible domains, can send mail through it [16]. We suggest the best solution to this problem be

```

Received: from mail.cs.ccu.edu.tw (unknown [127.0.0.1])
  by localhost (Postfix) with SMTP id 9EAB24B3267
  for < >; Wed, 14 Dec 2011 14:09:39 +0800 (CS
Received: from mail.cs.ccu.edu.tw (localhost [127.0.0.1])
  by mail.cs.ccu.edu.tw (NOPAM 20071006(G2)) with ESMTP id 42528E3
  Wed Dec 14 14:09:39 2011
  (envelope-from < >)
Received: from [140.123. . ] ( .cs.ccu.edu.tw [140.123. .
  (using TLSv1 with cipher DHE-RSA-CAMELLIA256-SHA (256/256 bits))
  (No client certificate requested)
  (Authenticated sender: )
  by mail.cs.ccu.edu.tw (Postfix) with ESMTPSA id 9A18D4B3267
  for < >; Wed, 14 Dec 2011 14:09:39 +0800 (CS

```

Fig. 2. The Received path in the mail header.

regularly scanning the mail servers to see whether they are accidentally configured to be open relays or not.

- 2) The spammer cracks user accounts beforehand, e.g., by social-engineering techniques and brute-force password guessing, and sends mail by impersonating the users. Because normal users also send mail through the accounts, we can parse the header of an outgoing mail message for its real source by the Received path (see an example¹ in Figure 2). We can see whether the REAs of the mail (or spam) messages from each real source are likely to repeat or not by the Bloom Filters. If many unique REAs are found, some accounts on the mail server are likely to be cracked for spamming.
- 3) A user configures to automatically forward his/her mail on a mail server to an external account. If numerous spam messages are received, they will be also forwarded to the external account from the mail server. In the perspective of network monitoring, the spam messages are delivered from the mail server. The REAs of the forwarded spam messages are rather fixed (to the external accounts), so this case will not raise an alarm in the detection. It is not a problem because detecting external spamming bots is beyond the scope of this work.

E. System Flow

Figure 3 presents the system flow in this work. First, the Bro NIDS monitors the SMTP sessions initiated from the campus. The collected information in the SMTP activities includes the delivery time, session identifier, source/destination IP address/ports, and mail headers, as well as the mail subjects for manually judging whether a host is spamming or not.

The collected SMTP logs are separated by hosts. We judge whether a host is a mail server or not by probing its port 25 or checking in the Bro logs whether its port 25 was passively connected. If it is not a mail server, we proceed to detect whether it is a spamming bot or not; otherwise, we extract the IP address of the real source from the Received path in each mail header, and detect whether the source is a spamming bot in the campus. This work is dedicated to detecting the spamming bots in the campus, not external ones, as we do not have the authority to crack down the latter. A white list is established by listing the normal hosts that are found incorrectly classified after investigation.

¹The partially sanitized IP address in the path is the real source.

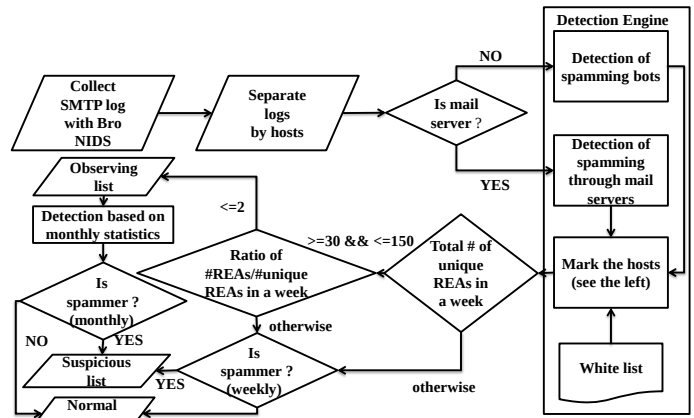


Fig. 3. The stages of monitoring and detecting spamming activities.

For each host, we use the numbers of total REAs and unique REAs, as well as the ratios of them observed in a week and a month as the detection features. If the number of unique REAs in a week is smaller or larger than the given thresholds, or if the number is not, but the REAs in a week are likely to repeat (i.e., the number of total REAs divided by that of unique REAs is higher than the threshold), we can judge whether a host is spamming or not based on the weekly features, and classify it into either the suspicious list or the normal list. Otherwise, the host is put into the observing list due to its ambiguous behavior, and will be judged based on the monthly features. The thresholds in Figure 3 are empirical values from our observations in the campus. The classifiers based on the weekly and monthly features are both trained with the J4.8 decision-tree algorithm implemented in Weka (www.cs.waikato.ac.nz/ml/weka). We will describe the training sets for both classifiers in Section IV. Finally, we provide the IP addresses of the hosts in the suspicious list to the computer center for investigation and crack-down.

Considering both weekly and monthly features allows the classification to be fast if the host behaves like a normal host or a spamming bot, while sufficient evidence of the features can be accumulated in a month if the host behaves ambiguously. Classification based on daily features is an option to speed up the detection, but the collected features may be insufficient, particularly for a low-profile spamming bot. We consider the weekly features an acceptable balance between efficiency and accuracy in this work.

It is possible that a bot master distributes a small and separate set of REAs to individual bots for spamming, and each bot then repeatedly delivers spam messages to just few REAs in a low-profile manner to evade the detection. Since this behavior is very similar to that of normal hosts, it is difficult to identify spamming activities except examining the mail content with a spam filter like that in [10]. Despite the possibility, the spamming bots that constantly deliver spam messages to a small set of fixed recipients will increase the risk of being blacklisted. The method can be also complemented

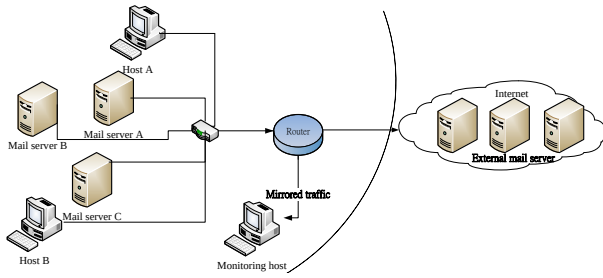


Fig. 4. The deployment to monitor the SMTP traffic in the campus.

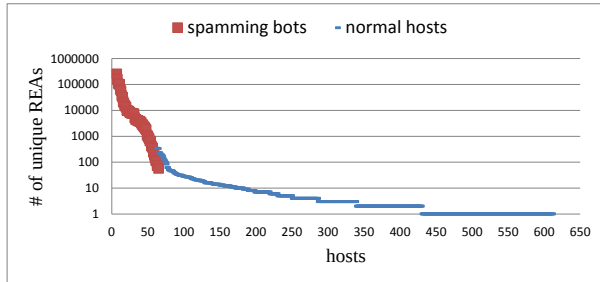


Fig. 5. Difference of the numbers of unique REAs between spamming bots and normal hosts.

with a spam filter to avoid the evasion, as discussed in Section III-C.

IV. EXPERIMENTAL ANALYSIS AND EVALUATION

We deployed a monitoring host with Bro and an Endace DAG 7.5G2 Network Monitoring Card in the computer center to monitor the SMTP activities (see Figure 4) over the period from November 2011 to April 2012. The total size of the SMTP logs is approximately 100 GB. The training set and test set, as well as the accuracy of the detection will be discussed in Section IV-A. The issue of spamming through the mail servers will be discussed in Section IV-B.

A. Evaluations of the Detection Method

Figure 5 compares the numbers of unique REAs between spamming bots and normal hosts (sorted in a descending order) over the period of six months. Totally 615 internal hosts initiated at least one SMTP session, and 65 out of them are found to be spamming bots after manual verification. According to the figure, the spamming bots generally send to a much larger number of REAs than the normal hosts. The observation again demonstrates the distinction of spamming bots and normal hosts in terms of the number of unique REAs, which can serve as an effective feature for classification. Figure 6 compares the the difference of the average delivery times per REA. The average delivery times per REA (derived from dividing the number of total REAs by that of unique REAs) for the spamming bots are mostly few. Only three spamming bots have the value larger than 2.

We then evaluate the classification accuracy based on both weekly and monthly features discussed in Section III-E. In the first evaluation, we select the instances characterized by the

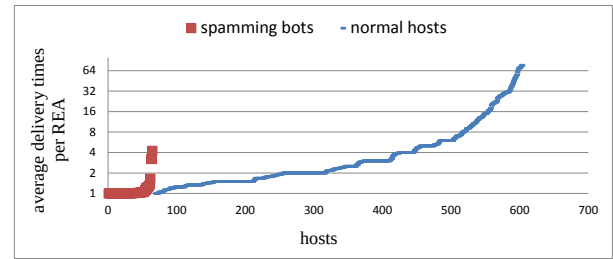


Fig. 6. The average delivery times per REA.

weekly features from the first 12 weeks over the observation period as the training set and those from each week over the rest of the period as the test sets. In the second evaluation, we select the instances characterized by the monthly features from November 2011 to February 2012 as the training set and those from each month in the rest as the test sets. The J4.8 algorithm is executed in both evaluations.

The mail subjects of the outgoing mail (or spam) messages are checked to manually verify whether a host is spamming or not. The first evaluation indicates the average precision is 0.91, and the average recall is 0.97 for the test sets. In the second evaluation, the average precision and recall are 0.89 and 0.82 for the test sets. When the detection goes through the flow in Figure 3, the average precision and recall become 0.97 and 0.96. Although the recall is slightly lower than that in the first evaluation, the overall accuracy in terms of precision and recall is rather high. The results prove that the classification can find spamming bots with high accuracy. We found totally 65 spamming bots in the campus network, and observed 1.5 million spam messages over the period of six months.

B. Case Study for Mail Servers

We also studied the issue of the mail servers abused for spamming, and found totally 14 such mail servers over the observation period. In one case, for example, we found an internal host not only delivered 109,979 spam messages from itself, but also delivered 322,950 spam messages through an official mail server in the dormitory over the period. This case is interesting because the host can deliver spam from itself and by abusing a mail server at the same time.

In another case, we cooperated with the administrator of a mail server abused for spamming, so that we can look into the access log on that server. We found that several overseas spamming bots regularly accessed the web mail interface for spam delivery through few user accounts, and confirmed that the accounts are cracked for spamming. The administrator have notified the users of the accounts after the investigation.

We analyzed the SMTP logs about the mail server in the second case. Figure 7 indicates that more than 60% of total messages sent out from the mail server are spam messages. The result implies that spam messages are more than normal ones, and thus abusing a mail server for spamming by cracking user accounts is a serious problem that cannot be neglected. Figure 8 compares the numbers of unique REAs

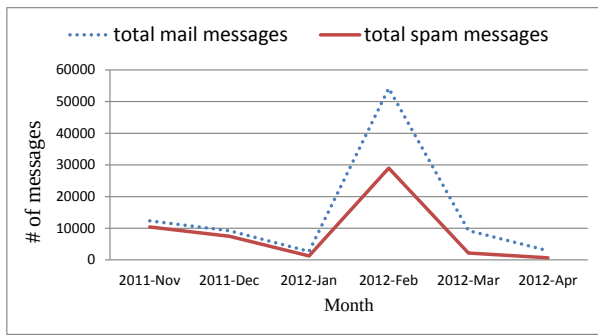


Fig. 7. Difference of recipient addresses between normal hosts and spammers.

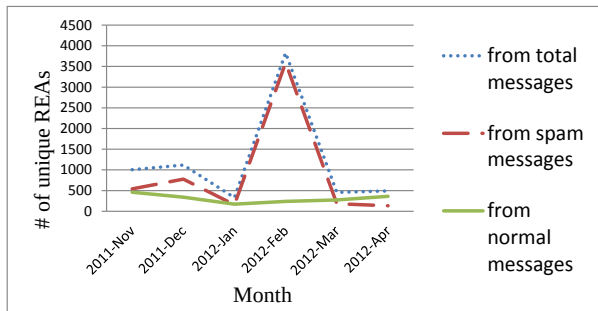


Fig. 8. Difference of unique recipient addresses between normal hosts and spammers.

from total messages, spam messages, and normal messages. The spam messages contributed a significant proportion of the unique REAs, especially in February 2012 (3,590 out of 3,829 unique REAs were from spam messages). In contrast, the number of unique REAs from normal messages is largely fixed because the recipients are mostly persons that the users on that server are familiar with. The observation demonstrates that the number and uniqueness of REAs are effective features to identify mail servers abused for spamming.

V. CONCLUSION AND FUTURE WORK

In this work, we present a method to detect spamming bots on the sender side. The detection features based on the number and uniqueness of REAs are simple yet effective. We monitored the SMTP sessions initiated from a large campus network for six months, and analyzed the SMTP logs by tracking the features with Bloom filters to detect the internal spamming bots. The accuracy of the detection is rather high. The average precision is 0.97, and the average recall is 0.96. The detection method has found 65 spamming bots and 14 legitimate mail servers abused for spamming. Besides a campus network, this method can be also deployed in any network of an institute to detect the resided spamming bots. It will benefit the network administrators to crack down the spamming bots as soon as possible.

Spamming bots may access web mail interfaces or deliver via secure SMTP for spamming. Since the packets are encrypted, the detection method cannot identify the spamming

bots in this case. This issue will be left to the future work.

ACKNOWLEDGMENT

This study is conducted under the National Science Council (NSC) Project 101-2219-E-194-001- the "Cloud computing systems and software development Project" of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

REFERENCES

- [1] Messaging Anti-Abuse Working Group, Email metrics program: Report #15 – first, second and third quarter 2011, Tech. rep., http://www.mawg.org/sites/maawg/files/news/MAAWG_2011_Q1Q2Q3_Metrics_Report_15.pdf (First, Second and Third Quarter 2011).
- [2] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson and S. Savage, "Spamalytics: an empirical analysis of spam marketing conversion," *Comm. of the ACM*, 52(9), pp. 99-107, Sept. 2009.
- [3] A. Ramachandran, N. Feamster and S. Vempala, "Filtering spam with behavioral blacklisting," In *Proceedings of the 14th ACM conference on Computer and Communications Security (CCS)*, Oct. 2007.
- [4] Y. Xie, F. Yu, K. Achan, R. Panigrahy, G. Hulten and I. Osipkov, "Spamming botnets: signatures and characteristics," In *Proceedings of ACM SIGCOMM*, Aug. 2008.
- [5] C. Y. Cho, J. Caballero, C. Grier, V. Paxson and D. Song, "Insights from the inside: a view of botnet management from infiltration," In *Proceedings of USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Apr. 2010.
- [6] J. P. John, A. Moshchuk, S. D. Gribble and A. Krishnamurthy, "Studying spamming botnets using Botlab," In *Proceedings of the 6th USENIX symposium on Networked Systems Design and Implementation (NSDI)*, Apr. 2009.
- [7] W. K. Ehrlich, A. Karasaridis, D. Liu and D. Hoeflin, "Detection of spam hosts and spam bots using network flow traffic modeling," In *Proceedings of the 3rd USENIX Conference on Large-scale Exploits and Emergent Threats: botnets, spyware, worms, and more (LEET)*, Apr. 2010.
- [8] G. Stringhini, T. Holz, B. Stone-Gross, C. Kruegel, G. Vigna, Botmagnifier: locating spambots on the internet, in: *Proceedings of USENIX Security Symposium*, 2011.
- [9] Y. Zhao, Y. Xie, F. Yu, Q. Ke, Y. Yu, Y. Chen, E. Gillum, Botgraph: large scale spamming botnet detection, in: *Proceedings of Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.
- [10] Z. Duan, P. Chen, F. Sanchez, Y. Dong, M. Stephenson, J. M. Barker, Detecting spam zombies by monitoring outgoing messages, *IEEE Trans. Dependable and Secure Computing* 9 (2) (2012) 198–210.
- [11] F. Sanchez, Z. Duan and Y. Dong, "Blocking spam by separating end-user machines from legitimate mail server machines," In *Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-abuse and Spam Conference (CEAS)*, Sept. 2011.
- [12] C. Kreibich, C. Kanich, K. Levchenko, B. Enright, G. Voelker, V. Paxson, and S. Savage, "On the spam campaign trail," In *Proceedings of USENIX Workshop on Large-scale Exploits and Emergent Threats (LEET)*, Apr. 2008.
- [13] B. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Comm. Of ACM*, vol 13, no. 7, pp. 422-226, July 1970.
- [14] F. Li, P. Cao, J. Almeida and A. Z. Broder, "Summary cache: a scalable wide-area Web cache sharing protocol," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 281293, June 2000.
- [15] S. D. Paola and D. Lombardo, "Protecting against DNS reflection attacks with Bloom filters," In *Proceedings of Conference on Detection of Intrusions and Malware & Vulnerability Assessment (DIMVA)*, July 2011.
- [16] M. Xie, H. Yin and H. Wang, "Thwarting E-mail Spam Laundering," *ACM Trans. Information and System Security (TISSEC)*, vol.12 no.2, pp.1-32, Dec. 2008.