Effective Discovery Of E-mail Spam Using Spot Detection System

M. A. Hema[1]

[1](Post Graduate) Dept. of Computer Science & Engineering,
S.A.Engineering College, Chennai, India.
hemajul14@gmail.com

ABSTRACT: Compromised machines on the Internet are generally referred to as bots, and the set of bots controlled by a single entity is called a botnet. Botnets have multiple nefarious uses: mounting Distributed denial of service attacks, stealing user passwords and identities, generating click fraud, and sending spam email. Compromised machines are one of the key security threats on the Internet. Given that spamming provides a key economic motivation for attackers to recruiting the large number of compromised machines, and focus on the detection of the compromised machines in a network that are involved in the spamming activities, commonly known as spam zombies. Develop an effective spam zombie detection system named SPOT by monitoring outgoing messages of a network. SPOT is designed based on a powerful statistical tool called Sequential Probability Ratio Test, which has bounded false positive and false negative error rates. The number and the percentage of spam messages originate by spam detection technique.

Index Terms—Compromised machines, spam zombies, spam detection techniques, spot detection system.

1.INTRODUCTION

A major security challenge on the internet is the existence of the large number of compromised machines. This compromised machines involved in the spamming activities are referred as spam zombies. Network of spam zombies are recognised as one of the most serious security threats today. This paper presents the study of the discovery of e-mail spam using the spam detection technique. Compromised machines are generally referred as bots and the set of bots that are controlled by a single entity are called botnets[17]. In this, identifying and cleaning of compromised machines in a network remain a significant challenge for system administrators of network of all sizes. Botnet have multiple nefarious uses such as generating click fraud, stealing user passwords and identities and sending spam email. There is an anecdotal evidence that spam is a driving force. Bot malware which is used to infect and control the systems in the botnet. The process in this are maintained by the botmaster. . The key factor is all the botnet have the common requirement for a command and control channel infrastructure and protocol for the botmaster to direct the activities of the bots through the internet. Early bots typically used Internet Relay Channel for communication, as bots initially grew out of the IRC community. (botnet analysis using c c) Botmasters have expanded their capabilities to use hypertext transfer protocol (HTTP) and peer-to-peer(P2P) for botnet command and control. A number of recent efforts have studied the aggregate global characteristics of spamming botnets such as the size of the botnets and the spamming patterns of the botnets, based on the sample emails received at the large e-mail service provider. To aggregate the global characteristics of spamming botnets, we developed a tool for the administrators to automatically detect the compromised machines in a network in an online manner. Such machines have been increasingly used to launch various security attacks including spamming and spreading malware, DDoS, and identity theft [19],[20].Compromised machines are one of the key security threats on the Internet. On the other hand, identifying and cleaning compromised machines in a network remain a significant challenge for system administrators of networks of all sizes. In this paper, we focus on the detection of the compromised machines in a network that are used for sending spam messages, which are commonly referred to as spam zombies and also detecting the spam messages. Given that spamming provides a critical economic incentive for the controllers of the compromised machines to recruit these machines, it has been widely observed that many compromised machines are involved in spamming. We consider ourselves situated in a network and ask the following question:How can approaches developed in the previous work cannot be applied. The locally generated outgoing messages in a network normally cannot provide the aggregate large-scale spam view required by these approaches. Moreover, these approaches cannot support the online detection requirement in the environment we consider. The nature of sequentially observing outgoing messages gives rise to the sequential detection problem. As a simple and powerful statistical method, SPRT has a number of desirable features. It minimizes the expected number of observations required to reach a decision among all the sequential and non-sequential statistical tests with no greater error rates. This means that the SPOT detection system can identify a compromised machine quickly. Moreover, both the false positive and false negative probabilities of SPRT can be bounded by user-defined thresholds.

2. RELATED WORK

Large networks of exploited computers that are under the control of a remote master (botmaster, bot herder) who can manage the lifecycle and activities of the exploited computers in the botnet to conduct a potentially large variety of malicious activities using a variety of effects. Botnets are used by malicious actors for many purposes such as spam campaigns, key logging, clickfraud, scareware schemes, spyware, distributed denial of service (DDoS), fast-flux phishing support, and other criminal endeavours[7]. The number of current and abandoned botnets is not known however there are a few that are better known such as Rustock, Mega-D, and Storm. These botnets and others have provided insight into command and control (C2 or C&C) methods that havebeen used by security researchers to help build detection algorithms, however it is only natural that as researchers and network security analysts become more proficient at detecting and disabling botnets, botmasters have become moreskilled and creative at hiding their malware and communication channels. The bot malware that has been used to infect and control the computers in the botnet can be deployed with a number of built-in capabilities and can be updated, refocused, or even deleted by the botmaster. The key factor that all botnets have in common is the requirement for a C2 infrastructure and protocol for the botmaster to direct the activities of the bots through the Internet. Early bots typically used Internet Relay Chat (IRC) as their communications channel as bots initially grew out of the IRC community. Over time botmasters have expanded their capabilities to use hypertext transfer protocol (HTTP) and peer-to-peer (P2P)[13] for botnet command and control; some moreskilled developers have programmed custom C2C protocols for their botnets to help obfuscate their activities.

2.1. Problem Formulation

E-Mail spam detection is a key problem in Cyber Security; and has evoked great interest to the research community. Various classification based and signature based systems have been proposed for filtering spam and detecting viruses that cause spam. However, most of these techniques require content of an email or user profiles, thus involving in high privacy intrusiveness. In the existing system, we address the problem of detecting machines that behave as sending spam. Our approach involves very low privacy intrusion as we look at only the border network flow data[13]. We propose two kinds of techniques for detecting anomalous behavior. The first technique is applicable for single instance network flow graph. The second technique involves analyzing the evolving graph structures over a period of time. We have run our experiments on University of Minnesota border network flow. Our results on this real data set show that the techniques applied have been effective and also point to new directions of research in this area.

In the MailRank system, it investigates the feasibility of MailRank, a new email ranking and classification scheme exploiting the social communication network created via email interactions. The underlying email network data is collected from the email contacts of all MailRank users and updated automatically based on their email activities to achieve an easy maintenance. MailRank is used to rate the sender address of arriving emails such that emails from trustworthy senders can be ranked and classified as spam or non-spam[10]. The paper presents two variants: Basic MailRank computes a global reputation score for each email address, whereas in Personalized MailRank the score of each email address is different for each MailRank user. The evaluation shows that MailRank is highly resistant against spammer attacks, which obviously have to be considered right from the beginning in such an application scenario. MailRank also performs well even for rather sparse networks, i.e., where only a small set of peers actually take part in the ranking of email addresses.

2.2. Research Design

The various kinds of data that can be analyzed from e-mail traffic, and the levels of privacy involved. Secondly, it gives a brief overview of link analysis techniques that can be applied for network security. Further, our approaches are explained in detail. Results of experimental evaluation of our approaches are presented.

In this, we address the issue of identifying the machines that are sending spam, or machines that have been compromised and are being used as a spam relay. Note that our focus is not on identifying individual users who send spam, or filtering an e-mail as spam based on its content. There has been work in such areas which is not directly related to ours. Recent work on detection of spam trojans suggests the use of signature and behavior based techniques.

In this they propose MailRank, a new approach to ranking and classifying emails according to the address of email senders. The central procedure is to collect data about trusted email addresses from different sources and to create a graph for the social network, derived from each user's communication circle. There are two MailRank variants, which both apply a power-iteration algorithm on the email network graph: Basic MailRank results in a global reputation for each known email address, and Personalized MailRank computes a personalized trust value. MailRank allows to classify email addresses into 'spammer address' and 'non-spammer address' and additionally to determine the relative rank of an email address with respect to other email addresses. And alsoanalyzes the performance of MailRank under several scenarios, including sparse networks, and shows its resilience against spammer attacks.

They investigated the feasibility of MailRank, a new email ranking and classification scheme, which intelligently exploits the social communication network created via email interactions. On the resulting email network graph, a power-iteration algorithm is used to rank trustworthy senders and to detect spammers. Mail-Rank performs well both in the presence of very sparse networks: Even in case of a low participation rate, it can effectively distinguish between spammer email addresses and non-spammer ones, even for those users not participating actively. MailRank is also very resistant against spammer attacks and, in fact, has the property that when more spammer email addresses are introduced into the system, the performance of MailRank increases. Based on these encouraging results we are currently investigating several future improvements for our algorithms. We intend to move from a centralized system to a distributed one to make the system scalable for a large-scale deployment. We are currently investigating a DNS-like system, in which the computation is handled in a distributed manner by several servers. Finally, another approach would be to consider each email client as a peer in a P2P network[10], and run a distributed approach to MailRank as such Spam filtering problem can be seen as a particular instance of the Text Categorization problem, in which only two classes are possible: spam and legitimate email or ham. In this, present spam filtering based on the MRF Model with different weighting schemes of feature vectors for variable neighborhood of words are presented. We present theoretical justification for our approach and conclude with results. Recently Sparse Binary Polynomial Hash (SBPH), a generalization of the Bayesian method and Markovian[15] discrimination have been reported for spam filtering. The classifier model in uses empirically derived ad-hoc super increasing weights. We develop more on correlate it with MRFs, and choose variable neighborhood windows for features using Hammersley-Clifford theorem and present different weighting schemes for the corresponding neighborhood window. We tested these weighting schemes in CRM114 Discriminator Framework . Our results reflect the effect of neighborhood relationship among features and provide evidence that this model is superior to existing Bayesian models used for spam filtering.

3. SPAM DETECTION TECHNIQUE

In this section, thus the spam zombies and spam messages can be identified.Normal Machine generates the original message. Original message enter in to the network and received by the server . Spam Zombie produces the spam messages and the spam message enters into the network. Server, first identifies the which message is Spam. By using Spam detection technique, Spam message is identified by the following process.Tagextraction,Tagreordering process,Anchor tag formation,Spam detection and elimination process.

3.1. Detecting The Compromised Machines

Compromised machines are the machines that are involved in spamming activities. Compromised machines on the Internet are generally referred to as bots, and the set of bots controlled by a single entity is called a botnet. Botnets have been widely used for sending spam emails at a largescale, By programming a large number of distributed bots, spammers can effectively transmit thousands of spam emails in a short duration. To date, detecting and blacklisting individual bots is commonly regarded as difficult, due to both the transient nature of the attack and the fact that each bot may send only a few spam emails. Furthermore, despite the increasing awareness of botnet infection and their associated control process little effort has been devoted to understanding the aggregate behaviours of botnets from the perspective of large email servers that are popular targets of botnet spam attacks. Spam filter is deployed at the detection system so that an outgoing message can be classified as either a spam or non spam. Spam filter and also identify the machines involving spamming activities[3].

The primary contributions of our work are:

- We are the first to analyze entire botnets (in contrast to individual bot) behaviour from spam email messages. We propose and evaluate methods to identify bots and cluster bots into botnets using spam email traces.
- Our work is the first to study botnet traces based one economic motivation and monetizing activities. Our approach analyzes botnets regardless of their internal organization and communication. Our approach is not thwarted by encrypted traffic or customized botnet protocols, unlike previous work using IRC trackers or DNS lookup.
- We report new findings about botnets involved in email spamming. For example, we report on the relationship between botnets usage and basic properties such as size. We also confirm previous reports on capabilities of botnet controllers and botnet usage patterns.

3.2. Tag Extraction And Reordering

3.2.1. Tag Extraction phase

In Tag Extraction Phase, the name of each HTML tag is extracted, and tag attributes and attribute values are eliminated. In addition, each paragraph of text without any tag embedded is transformed to <mytext/>. <anchor> tags are then inserted into AnchorSet, and the first 1,023 valid tags are concatenated to form the tentative e-mail abstraction. Note that we retain only the first 1,023 tags as the tag sequence. The main reason is that the rear part of long e-mails can be ignored without affecting the effectiveness of near-duplicate matching. Subsequently we preprocess the tag sequence of the tentative e-mail abstraction. One objective of this preprocessing step is to remove tags that are common but not discriminative between e-mails. The following sequence of operations is performed in the preprocessing step.

1. Front and rear tags are excluded.
2. Nonempty tags that have no corresponding start tags or end tags are deleted. Besides, mismatched nonempty tags are also deleted
3. The pairs of nonempty tags enclosing nothing are removed.

3.2.2. Tag Reordering Phase

On purpose of accelerating the near-duplicate matching process, we reorder the tag sequence of an e-mail abstraction in Tag Reordering Phase. If we consider two e-mail abstractions which have the same tag length and differ only in their last tags, the difference cannot be detected until the last tags are compared. To handle this problem, we destroy the regularity by rearranging the order of tag sequence to lower the number of tag comparisons. Note that this process ensures that the newly assigned position numbers of e-mail abstractions with the same number of tags are completely identical. As such, the matching process can be accelerated without violating the definition of near-duplicate, each tag is assigned a new position number by function ASSIGN_PN (PN denotes for position number) The final e-mail abstraction is the concatenation of all tags with new position numbers.

3.3. Spam Detection

The complete Spam Detection System is introduced here. Three major modules, Abstraction Generation Module, Database Maintenance Module, and Spam Detection Module are included in our system. In Abstraction Generation Module, each e-mail is converted to an e-mail abstraction by Structure Abstraction Generator with Abstraction Generation procedure. Three types of action handlers, Deletion Handler, Insertion Handler, and Error Report Handler, are involved in Database Maintenance Module.

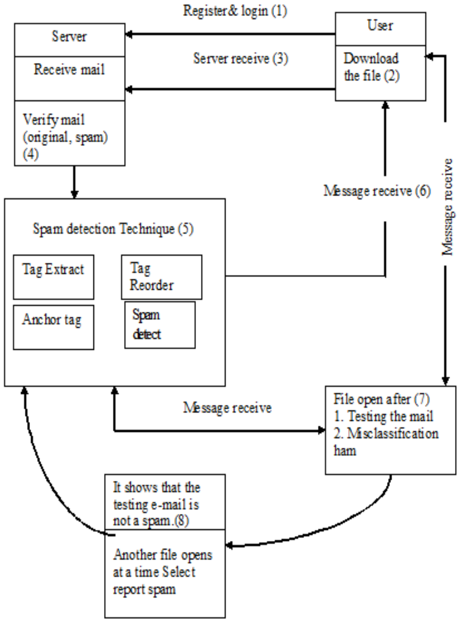

FIGURE: SYSTEM ARCHITECTURE

Note that although the term "database" is used, the collection of reported spam's can be essentially stored in main memory to facilitate the process of matching. In addition, Matching Handler in Spam Detection Module takes charge of determining results. There are three types of e-mails, reported spam, testing e-mail, and misclassified ham, required to be dealt with by Spam Detection System.

3.4. Percentage Count And Spam Based Technique

For comparison, in this section, we present two different algorithms in detecting spam zombies, one based on the number of spam messages and another the percentage of spam messages sent from an internal machine, respectively. For simplicity, we refer to them as the count-threshold (CT)detection algorithm and the percentage-threshold (PT) detection algorithm respectively. SPOT, which can provide a bounded false positive rate and false negative rate, and consequently, a confidence how well SPOT works, the error rates of CT and PT cannot be a priori specified. In addition, choosing the proper values for the four userdefined parameters ($\alpha,\beta,\theta_1,\theta_2$) in SPOT is relatively straightforward . In contrast, selecting the "right" values for the parameters of CT and PT is much more challenging and tricky. The performance of the two algorithms is sensitive to the parameters used in the algorithm. They require a thorough understanding of the different behaviours of the compromised and normal machines in the concerned network and a training based on the behavioural history[1]of the two different types of machines in order for them to work reasonably well in the network.

3.5. Spot Detection Algorithm

The SPOT detection algorithm, when an outgoing message arrives at the SPOT detection system, the sending machine's IP address is recorded, and the message is classified as either spam or nonspam by the (content-based) spam filter.

3.6.Algorithm:

```
An outgoing message arrives at SPOT
Get IP address of sending machine m
// all following parameters specific to machine m
Let n be the message index
Let Xn = 1 if message is spam, Xn = 0 otherwise
if (Xn == 1) then
// spam, 3
ʌn+= ln$\theta_{1/}\theta_2$
else
// nonspam
ʌn+=ln 1-$\theta_1$/1-$\theta_0$
end if
if (ʌn >=B) then
Machine m is compromised. Test terminates for m.
else if (ʌn <=A) then
Machine m is normal. Test is reset for m.
ʌn = 0
Test continues with new observations
else
Test continues with an additional observation
end if
```

4. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the spam detection techniques based on a two-month e-mail trace collected on a large US campus network. We also study the potential impact of dynamic IP addresses on detecting spam messages

4.1. Perfromance Of Spot

In this section, we evaluate the performance of SPOT based on the collected FSU e-mails. In all the studies, we set α=0.01, β=0.01, $\theta_{1=}$0.9, and$\theta_{0=}$0.2Table1 shows the performance of the SPOT spam zombie detection system. There are440 FSU internal IP addresses observed in the e-mail trace. Out of the 132 IP addresses[6] identified by SPOT, we can confirm 110 of them to be compromised in this way. For the remaining 22 IP addresses, we manually examine the spam sending patterns from the IP addresses and the domain names of the corresponding machines. If the fraction of the spam messages from an IP address is high (greater than 98 percent), we also claim that the corresponding machine has been confirmed to be compromised. We can confirm 16 of them to be compromised in this way. We note that the majority (62.5 percent) of the IP addresses confirmed by the spam percentage are dynamic IP addresses, which further indicates the likelihood of the machines to be compromised. For the remaining six IP addresses that we cannot confirm by either of the above means, we have also manually examined their sending patterns.

TABLE 1 PERFORMANCE OF SPOT

This is confirmed by the low percentage of infected messages in the overall e-mail trace. Infected messages are more likely to be observed during the spam zombie recruitment phase instead of spamming phase. Infected messages can be easily incorporated into the SPOT system to improve its performance.

4.2. Performance Of CT And PT

CT is a detection algorithm based on the number of spam messages originated or forwarded by an internal machine, and PT based on the percentage of spam messages originated or forwarded by an internal machine. For comparison, it includes a simple spam zombie detection algorithm that identifies any machine sending at least a single spam message as a compromised machine. In this,, we set the length of time windows to be 1 hour, that is, T ¼ 1 hour, for both CT and PT. For CT, we set the maximum number of spam messages that a normal machine can send within a time window to be 30 $(C_s=3)$, that is, when a machine sends more than30 spam messages within any time windows, CT concludes that the machine is compromised. In PT, we set the minimum number of (spam and nonspam) messages within a time window to be 6 $(C_a=6)$, and the maximum percentage of spam messages within a time window to be 50 percent (P=50%). That is, if more than 50 percent of all messages sent from a machine are spam in any time window with at least six messages in the window, PT will conclude that the machine is compromised. We choose the values for the parameters of PT in this way so that it is relatively comparable with SPOT. The minimum number of observations needed by SPOT to reach a detection is 3 (when α=0.01, β=0.01,θ_0=0.2, and θ_1=0.9).It shows the performance of CT and PT.. We use the same methods to confirm a detection or identify a missed IP address as we have done with the SPOT detection algorithm. From the table we can see that, CT and PT have a worse performance than SPOT. The antivirus software and Spam Assassin[2]were two independent components deployed at the FSU mail relay server, and a small number of messages carrying virus/worm attachments were not detected as spam by the spam filter.

Due to the difference in the methods of confirming a detection or identifying a missed IP address, the four detection algorithms observe different number of confirmed and missed IP addresses[6]. the simple detection algorithm can detect more machines (210) as being compromised than SPOT,CT, and PT. It also has better performance than CT and PT in terms of both detection rate (89.7 percent) and false negative rate (10.3 percent).

	Total#FSU IP	Detected	Confirmed (%)	Missed (%)
CT	440	81	79(59.8)	53(40.2)
PT	440	84	83(61.9)	51(38.1)
Simple	440	210	157(89.7)	18(10.3)

Total #FSU IP	Detected	Confirmed(%)	Missed(%)
440	132	126(94.7)	7(5.3)

TABLE 2. PERFORMANCE OF CT AND PT

5. CONCLUSION

This project has developed an effective spam zombie detection system named SPOT by monitoring outgoing messages in a network. SPOT was designed based on a simple and powerful statistical tool named Sequential Probability Ratio Test to detect the compromised machines that are involved in the spamming activities. SPOT has bounded false positive and false negative error rates. It also minimizes the number of required observations to detect a spam zombie. Our evaluation studies based on a two-month e-mail trace collected on the FSU campus network showed that SPOT is an effective and efficient system in automatically detecting compromised machines in a network. In addition, it showed that SPOT outperforms two other detection algorithms based on the number and percentage of spam messages sent by an internal machine, respectively.

REFERENCES

[1] Z. Duan, K. Gopalan, and X. Yuan, “Behavioral Characteristics ofSpammers and Their Network Reachability Properties,” Technical Report TR-060602, Dept. of Computer Science, Florida State Univ.,June 2006.

[2]G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee,“BotHunter: Detecting Malware Infection through Ids-Driven Dialog Correlation,” Proc. 16th USENIX Security Symp., Aug. 2007.

[3]L. Zhuang, J. Dunagan, D.R. Simon, H.J. Wang, I. Osipkov, G.Hulten, and J.D. Tygar, “Characterizing Botnets from Email Spam Records,” Proc. First Usenix Workshop Large-Scale Exploits and Emergent Threats, Apr. 2008.

[4]SpamAssassin, “The Apache SpamAssassin Project,” http://spamassassin.apache.org, 2011.

[5]K.V.SrinivasaRaoS.SrinivasuluandA.Amrutavallli“Detection of Spam through E-mail Abstraction Scheme” IJERT june 2012

[6]Y. Xie, F. Xu, K. Achan, E. Gillum, M. Goldszmidt, and T. Wobber,“How Dynamic Are IP Addresses?” Proc. ACM SIGCOMM, Aug. 2007.

[7]“Botnet Analysis Using Command and Control Channels” 15 December 2011 Carleton University

[8]A. Ramachandran and N. Feamster, “Understanding the Network- Level Behavior of Spammers,” Proc. ACM SIGCOMM, Sept. 2006.

[9]F. Sanchez, Z. Duan, and Y. Dong, “Understanding Forgery Properties of Spam Delivery Paths,” Proc. Seventh Ann. Collaboration, Electronic Messaging, Anti-Abuse and Spam Conf. (CEAS ’10), July 2010.

[10]Paul AlexandruChirita, JorgDiederich, Wolfgang Nejdl“ MailRank: Using Ranking for Spam Detection”.

[11]PrasannaDesikan and JaideepSrivastava“Analyzing Network Traffic to Detect E-Mail Spamming Machines”

[12]Ming-wei Chang, Wen-tau Yih, Christopher Meek “ Partitioned Logistic Regression for Spam Filtering”

[13]Ernesto Damiani, Sabrina De Capitani di Vimercati, Stefano Paraboschi, Pierangela Samarati”P2P-Based Collaborative Spam Detection and Filtering”

[14]Christopher Kruegel, Giovanni Vigna, GianlucaStringhini,”Detecting Spammers On E-Mail Spam Record”University of California, Santa

[15]ShalendraChhabra, William S. Yerazunis, Christian Siefkes “Spam Filtering using a Markov Random Field Model with Variable Weighting Schemas”

[16] Zhenhai Duan, Senior Member,Peng Chen, Fernando Sanchez,Yingfei Dong, Member,Mary Stephenson, and James Michael Barker”Detecting Spam Zombies By Monitoring The Outgoing Messages”

[17] N. Ianelli and A. Hackworth, “Botnets as a Vehicle for Online Crime,” Proc. First Int’l Conf. Forensic Computer Science, 2006.

[18] R. Droms, “Dynamic Host Configuration Protocol,” IETF RFC 2131, Mar. 1997.

[19] P. Bacher, T. Holz, M. Kotter, and G. Wicherski, “Know YourEnemy: Tracking Botnets,” http://www.honeynet.org/papers/bots, 2011.

[20] J. Markoff, “Russian Gang Hijacking PCs in VastScheme,”TheNewYorkTimes,http://www.nytimes.com/2008/08/06/technology/06hack.html, Aug. 2008.