Email Spam Zombies Scrutinizer In Email Sending Network Infrastructures

Sathish Raja S, K.G.S. Venkatesan

Abstract— Email Spammers are constantly pioneering the techniques to bypass anti-spam filters forcing many organizations to invest in spam email prevention mechanisms and solutions. Traditional email spam filters aims at analyzing the email content to characterize the best features that are commonly included in email spams. However, this is observed that a crafty trick which is designed to avoid content-based filters will be endless owing to social and economic benefits of sending email spams. In view of this particular situation, there has been many research efforts towards doing email spam detection based on the reputation of the senders rather than what is contained in the emails. Motivated by the fact that email spammers are prone to have unusual patterns/behavior and specific patterns of email communication, exploring the email social networks to detect email spams which has received much attention. Existing e-mail spam detecting system aims at analyzing the IP address of e-mail to categorize the features that are commonly found in e-mail spam. To resolve this problem an effective research has been made to create a solution that detects how the system generates that spam e-mails. In this paper a method is proposed that identifies the problem of spam e-mails. We present a procedure to generate the e-mail abstraction using HTML content in e-mail, and our newly devised abstraction can more effectively capture the near-duplicate phenomenon of spams. This can be identified by tracking the performance of machines sending e-mail, tracing the e-mail content. Moreover, we design a complete spam detection system (A Multi-level Collaborative Spam Detection System), which possesses an efficient near-duplicate matching scheme and a progressive update scheme.

Index Terms— Compromised Machines, Content based Spam, Email Spam, Network Attacks, Spam Zombies

1 INTRODUCTION

Spam emails is nothing but flooding the Internet bandwidth with multiple copies of the email message which is of same nature, in an attempt to force the message/information on the people who would not otherwise choose to receive the message. Most of the spam emails becomes more commercial advertising, often for dubious unworthy products, get rich schemes, or quasi-legal service.

Spam emails costs the sender very less to send. Most of the costs are paid by the recipients who received the emails or the carriers rather than by the email sender. Email spam lists are created by monitoring or filtering Usenet postings, stealing Internet emailing lists, or search Web for addresses. On top of that, Email spamming costs money for ISPs and online services to transmit spam, and these costs are transmitted directly to subscribers.

It is widely proclaim that identifying the regions that originated malicious traffic on the network. One of challenging security issue on the networking system is the existence of the huge number of Spam spreading machines. Such spam spreading system have been used to implement different security related attacks including spreading the threats.

Spamming is the process of sending the same messages without break. Spam spreading machine is also called as compromised machine, it is defined as the machine which is easily accessible by hacker, and otherwise any malware can be executed without permission of admin. On the other words, detecting and clearing compromised machines in a network is a significant challenge for network administrators for all types of network. In this paper, we propose a system that detects the spam spreading machines in a network that are used for sending spam messages, which are known as spam zombies.

Given that spamming provides a Hypercritical economic benefit for the controllers of the spam spreading machines to recruit these machines, it has been widely observed that many spam spreading machines are involved in spamming process [1], [2], [3]. Current research efforts have studied to identify the cumulative general features of spamming botnets (networks of spam spreading machines involved in spamming) such as the size of botnets are measured and the spamming arrangement of botnets, based on the spam messages received in previous level at a large e-mail service provider [3], [6].Rather than the cumulative general features of spamming botnets, we aim to discover a tool for system admin to significantly detect the spam spreading machines in a linked networks in an online manner.

We consider ourselves situated in a network and ask the following question: How can we automatically identify the spam spreading machines in the network as outgoing messages pass the observation point continuously? The concept developed in the prior work [5], [6] cannot be affix here. The local originated outgoing message in a network does not provide the aggregate large-scale spam view required by these attitudes. Moreover, this concept cannot support the online detection method in the environment we acknowledge.

The Character of observing outgoing messages gives rise to the detection problem. In this paper, we will implement a spam spreading detection system, named SSDS, by monitoring outgoing messages. SSDS is developed based on a statistic method called Constant Presumption Correlation Test (CPCT), developed by Wald in his seminal work [10]. CPCT is a powerful statistical method that can be used to analyze between two axiom (in our case, a machine is compromised versus the machine is not compromised), as the events (in our case, outgoing messages) occur constantly.

As a simple and powerful statistical method, CPCT has a number of appropriate characters. It reduces the required number of observations expected to achieve a resolution among all the sequential and nonsequential statistical tests with less error rates. This means that the SSDS detection system can identify a spam spreading machine instantly. Additionally, both the false positive and false negative probabilities of CPCT can be ponded by user-defined thresholds. According to the SSDS system the user can select the desired thresholds to control the false positive and false negative behaving nature of the system.

In this paper, we develop the SSDS to avail system admin, automatically for identifying the spam spreading machines in their networks. We also estimate the performance of the SSDS system based on a month e-mail trace collected in a large network. Our evaluated studies convey that SSDS is a useful and efficient system which automatically detects spam spreading machines in a network. For example, among the 400 IP addresses observed in the e-mail trace, SSDS identifies138 of them as being associated with spam spreading machines. Out of the 138 IP addresses identified by SSDS, 128 can be either independently confirmed (113) or are highly likely (15) to be spamming. And also, only minimal internal IP addresses associated with spamming machines in the trace are missed by SSDS. In addition, SSDS only needs a Minimum number of observations to identify a spam spreading machine. The majority of spam zombies are identified with as little as two or three spam messages. While Comparing, we design and study two other spam spreading detection algorithms based on the count of spam messages and the percentage of spam messages generated or forwarded by machines, respectively. We compare the performance of SSDS with the two other detection algorithms to illustrate the advantages of the SSDS system.

2 RELATED WORK

In this section, we discuss related work in detecting spam spreading machines. Initially we focus on the studies that handle spamming activities to detect bots and then discuss a number of actions in detecting global botnets. Based on e-mail messages received by the client at a large e-mail service provider, two recent studies [5], [6] are inspected the aggregate global characteristics of spamming botnets including the count of botnets and the spamming arrangement of botnets.

However, their approaches are better suited for e-mail service providers to understand the Cumulative general characteristics of spam spreading machines in case of being deployed by single networks to detect internal Spam spreading machines. Moreover, their approaches cannot support the online detection requirement in the network circumstances considered in this paper. We aim to implement a system to assist system admin in automatic detection of spam spreading machines in their networks in an online manner.

In the following, we discuss a few schemes on detecting global botnets. Botscanner[8], developed by Gu et al., detects spam spreading machines by coordinate the IDS dialog trace in a network. It was designed based on the examination that a complete malware infection process has a number of well-defined stages including inbound scanning, exploit usage, egg downloading, outbound bot coordination dialog, and outbound attack propagation. By correlating inbound attack alarms with outbound communications arrangements, Botscanner can detect the possible affected machines in a linked network.

Unlike Botscanner which commit on the specification of the malware infection process, SSDS focuses on the economic incentive behind many spam spreading machines and their intentness in spamming. An anomaly-based detection system named BotSniffer [9] identifies botnets by exploring the spatial-temporal behavioral similarity commonly observed in botnets. It focuses on IP-based and HTTP-based botnets. In BotTracer, flows are classified into groups based on the global server that they link to.

If the progress within a party expose behavioral likeness, the corresponding hosts involves are detected as being compromised. Botsniffer [7] is one of the first botnet detection systems that are both protocol and structure independent. In Botsniffer, flows are classified into groups based on similar communication patterns and similar malicious activity patterns, respectively. The intersection of the two groups is considered to be compromised machines. Compared to general botnet detection systems such as Botscanner, BotTracer, and Botsniffer SSDS is a lightweight compromised machine detection scheme, by exploring the economic incentives for attackers to recruit the large number of compromised machines. As a simple and powerful statistical method, Constant Presumption Correlation Test has been successfully applied in many areas [2]. In the area of networking security, CPCT has been used to detect portscan activities [2], proxy-related spamming activities [3], anomaly-based spam detection [9], and MAC protocol misbehavior in networks [6].

3 PROBLEM DEFENITON AND ASSUMPTIONS

In this section, we define the spam zombie detection issue in a network. In particular, we discuss the network pattern and assumptions we make in the detection problem. Fig. 1 illustrates the logical view of network model. We assume that messages originated from machines inside the network will pass the deployed spam zombie detection system. This assumption can be attains in a few different scenarios. For example, the traffic of outgoing e-mails (with destination port number of 25) can be counterfeit and redirects to the spam detection system.

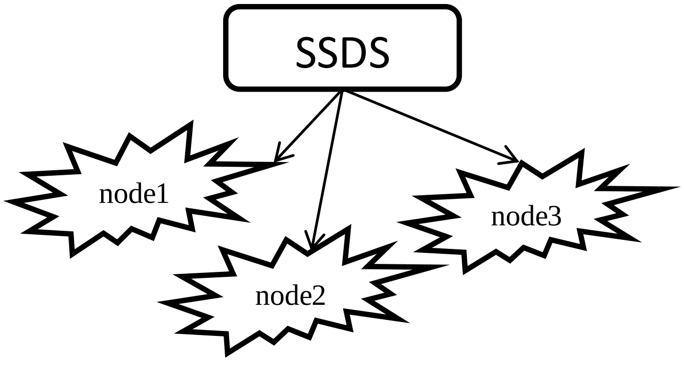

Fig. 1: Network Topology

A machine in the network is assumed to be either spam spreading machine or normal (that is, not compromised). In this paper, we only focus on the compromised machines that are involved in spamming. Therefore, we use the term a spamming machine to denote a spam zombie, and use the two terms conversely. Let Xim for i = 1; 2; . . . indicates the consecutive observations of a random variable X corresponding to the sequence of messages delivered from machine m in a network. We let Xim =1 if message i from the machine is a spam, and Xim= 0 otherwise. The detection system assumes that the nature of a spamming machine is varying from that of a normal machine in terms of the messages they send. Clearly, that a spamming machine will have a higher probability to generate a spam message than a normal machine.

Formally,

$$PR\ (Xim\ = 1 | H1) > PR\ (Xim\ = 1 | H0)$$

Where H1 denotes that machine m is spamming and H0 that the considered as normal machine. The spam zombie detection issue can be formally defined as follows: as Xim arrives constantly at the detection system, the system detects with a high probability if machine m has been spamming. Once a conclusion is reached, the detection system acknowledge the result, and further actions can be proceed, e.g., to clear the machine.

We assume that a (e-mail content-based) spam filter is deployed at the detection system so that a sending mail can be classified as either a spam or non-spam [1]. The existing spam detection filters cannot achieve spam detection accurately, and they all allow both false positive and false negative errors. The false negative rate of a spam filter counts the percentage of spam messages that are misclassified, and the false positive rate counts the percentage of non-spam messages that are misclassified. We denote that all deployed spam filters have very low false negative and false positive rates, and some spam classification errors will occur these are the margin performance of the existing spam detection algorithms.

We consider that the mail sending machine m has been notified by spam detection algorithm, this consideration is just for the advancement of our exposure. The proposed SSDS system focus that were sending message is forwarded by few domestic mails relay servers before leaves from the network.

4 WEB SPAM CLASSIFICATION

4.1 Content Spam

In previous section we discussed more about the content based spam (content spam), The content based spam is the first and most widely spreader form of web spam, the real fact is that spam search engines use information retrieval models based on a mail content to rank web pages, it use a model such as vector space model. In this model the spammers analyze the weaknesses and exploit them.

Consider a document structure into 4 subtypes of content spamming

4.1.1 Banner Spamming

Due to high preference of the title field for information retrieval spammers have a motivation to overstuff it so as to reach overall ranking.

4.1.2 Body Spamming

In this case the body (content) of a mail is modified. This is the most common form of content spam because it is simple and instantly allows assigning various techniques. For instance, if a spammer wants to reach a high ranking of a mail page by only using limited predefined set of queries, they can use the repeated strategy by overstuffing content of a page with strategies that appear in the set of queries. On the other hand, if the aim is to cover so many queries as possible, the method could be to use hit-or-miss approach (using random keywords).

4.1.3 Meta-Definition Spamming

Because Meta-Definition plays a vital role in a content description, search engines estimate them carefully. Hence, the distribution of spam content in this field might be considered as very potential from Spammer view point. Because of the high spamming, currently search engines provide very low priority to this field or even not considered it completely.

4.1.4 URL Linked Spamming

Some search engines also consult a tokenized URL of a web page as a zone. And hence spammers create their own URL for a web page by using words which should be specified in a set of queries. For example, if a spammer wants to be graded high for the query “best deals of laptop", they can create a URL like this, “best-deal-laptop.com/best-laptops.html”

4.2 Link Associated Spam

There are two major categories of associated spam: outgoing link associate spam and incoming link associate spam.

4.3 Outgoing link associate spam

This is the simplest and cheapest method of associate spam because, first, spammers have an open access to his web pages and therefore they can add any kind of items to them, and second, they can easily duplicate the entire web catalogue. Outgoing link associate spamming techniques target mostly HITS algorithm [3] with the goal of getting high hub score.

4.4 Incoming link associate spam

In some case spammers try to increase a Page Ranking or score of a page or simply try to boostup a number of incoming links.

4.5 Own Pages

In such case a spammer has a direct access control for over all the web pages and can be very flexible in their strategies. They can create their own link and carefully fix topology to guarantee the desired assets. The most common link farm has a topology depicted and is named as a honey-pot farm. In this case a spammer creates an own webpage which looks absolutely same as normal web page and may be even authoritative but it directly links to the spammer's targeted web pages. More aggressive form of a honeypot schema is hijacking the website, when spammers first hack a reputable website and then use it as a part of their link farm in their spam linked web pages.

In 2006 a website for CS students was collapsed and spammed with link of obscene nature. Spammers can also plot by inserting a link by exchange schemes in order to achieve a top scale, higher in-link counts. Motivations of spammers are carefully analyzed and optimal properties of link farms are also analyzed to reduce time consumption on a link farm promoted spammers. They are also eager to buy abandoned and expired domain names. They are guided by this principle that is due to the non-instant updation of a domain index and recrawling of expired domains, search engines consider that a domain is still under the control of some other good website owner . Here the spamming method works as follows. First, a honey-pot page achieves high ranking by boosting methods. But when the page is requested by an authorized user, they don't consider it, they get redirected to a target web page. There are different ways to achieve redirection. The simplest approach is to set a page refreshing time to zero and initialize a refresh URL attributes with a target page URL.

4.6 Accessible WebPages

These are web pages which the spammers can modify the content but don't own. For example, the web pages like blog with public comments, Wikipedia pages, or even an open user-maintained web directory, a public discussion group.

5 SPAM SPREADING DETECTION ALGORITHMS:

In this section, we will develop spam spreading detection algorithms. This SSDS algorithm, which utilizes the Constant Presumption Correlation Test presented in the last section. We discuss the impacts of CPCT parameters on SSDS in the context of spam spreading detection. The existing spam Spreading detection algorithm is developed based on the count of spam messages and the cumulative percentage of spam messages received from an internal machine.

5.1 SSDS Detection Algorithm

SSDS is designed based on the statistical tool CPCT we discussed in the last section. In the context of detecting spam zombies in SSDS, we consider H1 as a detection and H0 as a normal machine. That is, H1 is true if the concerned machine is spamming machine, and H0 is true if it is not spamming machine. In addition, we let Xim =1 if the ith message from the particular machine in the network is a spam, and Xim =0 otherwise. Recall that CPCT needs four configure parameters from end users, namely, the false positive probability β, the desired false negative probability α the probability that if a received message is a spam when H1 is true (φ1), and the probability that the message might be a spam when H0 is true (φ0). We discuss how users configure the values of the four parameters after we present the SSDS algorithm. Based on the user-specified values of β and α, the values of the two boundaries A and B of SSDS are computed using (4). In the following, we discuss detail description about the SSDS detection algorithm. Algorithm 1 outlines the steps of the algorithm. When sending message reaches at the SSDS detection system, the IP address of mail sending machine's has been recorded, and the message is under classification as either it is spam or non- spam by the spam filter. For each monitored IP address, SSDS maintains the logarithmic sum value of the corresponding ratio Yn, whose value is upgraded according to message n arrives from the IP address (lines 7 to 13 in Algorithm). Based on the relation between Yn and A and B, the algorithm detects if the corresponding message sending machine is spamming or non spamming, or a decision cannot be achieved and additional observations are needed (lines 14 to 22).

Algorithm 1. SSDS spam Spreading detection system

```
1: Input:
        The outgoing message reaches SSDS
2: Output:
        Trace the IP address of spam sending machine m
3: //The following Specified parameters specific belongs to machine m
4: Let ni be the message index
5: for i=1,2,3…..
6: Let Xni = 1 if message is spam, Xni = 0 otherwise
7: if (Xni == 1) then
8: // spam, 3
8: Yn = $\ln\frac{1-\varphi1}{1-\varphi0}$
 9: else
10: //for nonspam
11: Yn + = $\ln\frac{1-\varphi1}{1-\varphi0}$
12: end if
13: if (Yn ≥ B) then
14: Machine m is spamming machine. Test could terminate for m.
15: else if (Yn ≤ A) then
16: Machine m is considered as non-spamming. Test is reset for m.
17: Yn = 0
18: Test continues with newly observed data
19: else
20: Test continues with an additional observed data
21: end if
22: return output
```

We note that in the context of spam spreading detection, from the viewpoint of tracing IP address in a network monitoring, it is very important to identify the machines that are spreading spam more than the machines that are normal. After identification of the machine is identified as being compromised (lines 13 and 14), it is added into the list of potentially compromised machines that system administrators can go after to clean. The message-sending behavior of the machine is also recorded should further analysis be required. Before the machine is cleaned or removed from the list, the SSDS detection system not need further monitoring the message forwarding characteristics of the machine in a network. On the other hand, a machine that is currently non spamming may get spamming at a later time. Therefore, we need to continuously monitor the machines in a network that are determined to be normal by SSDS. Once such a machine is identified by SSDS, the records of the machine in SSDS are reset, in particular, the value of Yn is set to 0, so that a new session of monitoring phase starts for the machine (lines 16 to 19).

6 EMAIL SPAM DETECTOR

We are trying to focus on multi-level advanced thoughts(Fig. 2) of validating the emails with various mixes up of base algorithms like,

Bayesian spam filtering – Conceptual filtering our spam emails based on the probability of word occurrences (replica).

Topical web crawl Algorithm – A Novel web crawling algorithm to crawl a mix up of email contents which constitute keywords and URL's.

Boyer Moore Exact Pattern Matching Algorithms –
An efficient algorithm to identify the exact patterns which enable the system to filter it as spam.

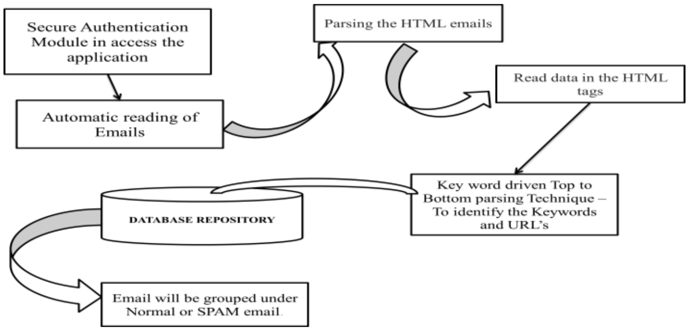
Secure Authentication Module in access the application
Parsing the HTML emails
Read data in the HTML tags
Automatic reading of Emails
Key word driven Top to Bottom parsing Technique – To identify the Keywords and URL's
DATABASE REPOSITORY
Email will be grouped under Normal or SPAM email

Fig 2: Multi level advanced Email Spam Detector

6.1 BAYESIAN SPAM FILTERING

Correlating the usage of tokens (typically words), with spam and non-spam e-mails and then using Bayesian inference technique (below formula) to calculate a probability that an email is or is not spam.

After training the word probabilities (also known as similar/likelihood functions) are used to compute/calculate the probability that an email with a particular set of words in it belongs to either of the category. Each word in the email contributes to the email's spam probability, or only the most interesting words.

$$\Pr(S|W) = \frac{\Pr(W|S)\cdot\Pr(S)}{\Pr(W|S)\cdot\Pr(S)+\Pr(W|H)\cdot\Pr(H)}$$

$\Pr(S|W)$ is the probability that a message is a spam, knowing that the word "replica" is in it;
$\Pr(S)$ is the overall probability that any given message is spam;
$\Pr(W|S)$ is the probability that the word "replica" appears in spam messages;
$\Pr(H)$ is the overall probability that any given message is not spam (is "ham");
$\Pr(W|H)$ is the probability that the word "replica" appears in ham messages.

6.2 TOPICAL WEB CRAWL ALGORITHM

Web Crawl algorithm is used to find an item with specified properties (Keywords and URL's) among the collection of items from Top to bottom. If there is any irrelevant items means is stored individually as records in a Database and Check from Database. A Keyword search looks for keywords anywhere in the record and also for the URL's. We can also use the Guided Keyword search option to combine search elements, group terms, or fields to be searched.

6.3 BOYER MOORE EXACT PATTERN MATCHING ALGORITHMS

A 256 member table is constructed that is initially filled with the length of the pattern.. The 256 members represent the full range of characters in the ASCII character set. A second pass is then made on the table that places a descending count from the original length of the pattern in the ASCII table for each character that occurs. The table constructed in this manner allows the algorithm to determine in one access if the character being compared is within the search pattern or not. The first character compared is the end character of the pattern "M" to the corresponding position in the source. The character being compared is "First Character" which is within the characters that are in the pattern. Character "First Character" has a shift of "String Length" so the pattern is shifted towards string length characters right.

6 PERFORMANCE EVALUATION OF SSDS

In this section, we describe the performance of the detection algorithm based on a month e-mail trace collected on a large network

.

6.1 E-MAIL TRACE OVERVIEW:

The e-mail trace was collected at a mail relay server deployed in the large network between 6/23/2005 and 7/25/2005, In this duration of e-mail trace collection, the e-mail server relayed messages destined for 63 sub domains in that network. The mail relay server ran Spam Assassinated messages [2] to detect spam messages. The e-mail trace contains the following information for each incoming message: the local arrival time, that the IP address of the e-mail sending machine and whether or not that the received message is spam. In addition, if a message has a known virus/worm attachment, it was so indicated in the trace by antivirus software. The antivirus software and Spam Assassinated messages were two independent components deployed on the mail relay server. Due to some privacy issues, we do not have access to the content of those messages in this trace. Uniquely, we have collected all the outgoing mails in order to calculate the performance of the spam detection algorithms.

However, due to logical constraints, we were unable to collect all kind of messages. Instead of that we identified the messages in the e-mail trace that have been forwarded or originated by the internal machines in that own networks , that is, the messages forwarded or originated by an internal machine and destined to an internal network account. We refer to this set of messages as the internal network e-mails and check performance of our spam detection algorithms based on that internal network e-mails. We note the set of that internal network e-mails does not contain all the outgoing messages originated from that internal network , and the spamming machines are detected by our detection algorithms based on the internal network If a message has a known malware/virus attachment, we refer to such a message as a spam mail. We refer that the IP address of a sending machine as a spam-only if the IP address sends spam messages are received from the IP address. In the same way, we consider that the machine is non-spamming, if we only receive non-spam messages, or we receive both spam and non-spam messages, respectively, from the IP address.

Table 1

E-Mail trace summary

Measure	**Non-Spamming**	**Spamming**	**Aggregate**
Duration	6/23/2011 -7/25/2011		
No. of e-mails	6,788,256	18,588,374	2,537,660
No. of Internal network e-mails	46,221,889	58,612,354	104,834,243
No. of infected emails	60,118	162,212	221,330
No .of infected e-mails in internal network	33,181	43,586	76767

Table 1 shows a complete summary of the e-mail trace. As shown in the table, the trace contains more than 20 M e-mails, of which more than 16 M, or about 70 percent, are spam. About half of the messages in the e-mail trace system were originated by internal machines in that network. **Table 2** shows the classifications of the monitored IP addresses. During the same process, we monitored that the internal IP addresses. **Table 3** shows the classification of the observed IP addresses that sent at least one message carrying a malware/ virus attachment.

Table-2
Sending Summary

	Total	**Non-spamming**	**Spamming**	**Combined**
No. Of IP (%)	2,468, 114	120,108	2,222,748(89.1)	113,258(4.2)
No. Of internal network IP(%)	424	182(40.8)	70(15.2)	172(42.8)

Table-3
Summary of malware/virus sending

	Total	**Non-spamming**	**Spamming**	**Combined**

No. Of IP (%)	10,438	1,021	6,805	2,578
No. Of internal network IP (%)	208	21	45	142

Table-4
Permformance of SSDS

Total No of internal network IP	Detected IP	Confirmed IP(%)	Missed(%)
208	188	182(95.8)	6(5.2)

6 CONCLUSION

In this paper, we developed an effective spam spreading detection system named SSDS by monitoring outgoing messages in a network. SSDS was designed based on a simple and powerful statistical tool named Constant Presumption Correlation Test to detect the Spamming machines that are involved in the spam activities. It also reduces the number of required observations to identify the spam spreading machine. Our estimation based on a month e-mail tracing collected on the internal network showed that SSDS is an efficient and effective system that automatically detects spamming machines in a network.

7 REFERENCES

1. Joe St Sauver, “Spam Zombies And Inbound Flows To Compromised Customer Systems”, 2005.
2. Nikita Spirin , Jiawei Han “Survey on Web Spam Detection: Principles and Algorithms”.
3. Malware Recon, “The Making of a Spam Zombie Army”, http://computer.org/security/.
4. “Tracking Dynamic Sources of Malicious Activity at Internet-Scale”
5. John Aycock ,Nathan Friess “Spam Zombies from Outer Space”, January 2006.
6. Sumit Ganguly, Minos Garofalakis, Rajeev Rastogi, Krishan Sabnani “Streaming Algorithms for Robust, Real-Time Detection of DDoS Attacks”.
7. Zhenhai Duan, Peng Chen, Fernando Sanchez,Yingfei Dong Detecting Spam Zombies by Monitoring Outgoing Messages, 2012.
8. Y.Bhavani, P.Niranjan Reddy “An Efficient IP Traceback Throughpacket Marking Algorithm” ,2010
9. Jonathan E. Schmidt, “ Dynamic Port 25 Blocking to Control SPAM Zombies”
10. Abraham Wald "Wald Test" (1939)

8 ABOUT THE AUTHOR

Sathish Raja received BE degree in Computer Science & Engineering from Bharath Institute of Science and Technology, Chennai and currently pursuing his M.Tech degree in Computer Science & Engineering at Bharath University. He has 9 years of Software Development Experience in Telecom and Data Communication domain.

K.G.S.Venkatesan received his B.Tech degree in Computer Science & Engineering from JNT University, Hyderabad and received his M.Tech degree in Computer Science & Engineering from Bharath University. He is currently pursuing his Ph.D in Computer Science & Engineering at Bharath University, Chennai. He has 10 years of Teaching experience and has guided many B.Tech and M.Tech projects. He is having Membership in Indian Society of Technical Education (MISTE).He attended Teaching

Skills Programme conducted by WIPRO MXLA (Mission 10X Learning Approach).