Security Attacks by the Compromised Machines with Monitoring Outgoing Messages

[1] T. S. Mani Kanth, [2] G. Yedukondalu, [3] U. Shubhangi
M.Tech (CSE) Scholar[1], Associate Professor[2], Assistant Professor[3]
Dept of CSE, Vignan Institute of Technology & Science, Hyderabad

ABSTRACT

Compromised machines create security threats on the Internet by sending spam messages. 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 with Sequential Probability Ratio Test (SPOT) techniques by monitoring outgoing messages of a network. SPOT gives results with less time and improve the execute process. We also adapted Paul-Graham Implementation is used to detect the spams , SPRT which is tracked when a message is passed from the network which is called an outgoing messages finally Bayesian calculation which determines the rating of spam and verifies weather the system is compromised system or not.

Keywords: SPOT, SPRT, Out Going Messages, Compromised machines, Spam Zombies, Spam Detection.

1. INTRODUCTION

A major Security challenge on the Internet is the existence of the large number of compromised machines. Such machines have been increasingly used to launch various security attacks including spamming and spreading malware, DDOS, and identity theft. Two natures of the compromised machines on the Internet sheer volume and wide spread render many existing security countermeasures less effective and defending attacks involving compromised machines extremely hard. 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. 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. A number of recent research efforts have studied the aggregate global characteristics of spamming botnets (networks of compromised machines involved in spamming) such as the size of botnets and the spamming patterns of botnets, based on the sampled spam messages received at a large email service provider.

Rather than the aggregate global characteristics of spamming botnets, we aim to develop a tool for system administrators to automatically detect the compromised machines in their networks in an online manner. we consider ourselves situated in a network and ask the following question: How can we automatically indentify the compromised machines in the network as outgoing messages pass the monitoring point sequentially? The approaches developed in the previous work cannot be applied here. 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. In this paper we develop a spam zombie detection system, named SPOT, by monitoring outgoing messages. SPOT is designed based on statistical method called Sequential Probability Ratio Test (SPRT), developed by Wald in his seminar work. SPRT is a powerful statistical method that can be used to test between two hypotheses (in our case, a machine is compromised vs. the machines is not compromised), as the events (in our case, outgoing messages) occur sequentially. 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. Consequently, users of the SPOT system can select

the desired thresholds to control the false positive and false negative rates of the system.

In this paper we develop the SPOT detection system to assist system administrators in automatically identifying the compromised machines in their networks.

2. PROBLEM

These various security attacks are the major security challenge on the Internet. The existence of the large number of compromised machines. Their approaches are better suited for large e-mail service providers to understand the aggregate global characteristics of spamming botnets instead of being deployed by individual networks to detect internal compromised machines. Moreover, their approaches cannot support the online detection requirement in the network environment considered in this project.

The existing algorithm is less effective. Identifying and cleaning compromised machines in a network remain a significant challenge for system administrators of networks of all sizes.

3. PROPOSED WORK

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.

The nature of sequentially observing outgoing messages gives rise to the sequential detection problem. In this paper will develop a spam zombie detection system, named SPOT, by monitoring outgoing messages. SPOT is designed based on a statistical method called Sequential Probability Ratio Test (SPRT), 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.

An account authentication process is followed in order to send mail to the recipient address. Now these messages are the outgoing messages after the mail is sent and now the system automatically detects the IP Address randomly generated through some logic implemented by code and list of mails and its message files are displayed to the administrator so that he can view the messages and later these are applied for filteration process which implements SPRT. As we discussed SPOT implements the testing procedure to find the problem of detection using Sequential Probability Ratio Test. This includes the filtration process and executes the output with the ratio. This includes the Paul-Graham procedural process which executes based on two hypothesis considering two files. One file contains the records of non-spam that is good message file and another file is a spam which contains records of data usually it is called as a bad message file. These both

In proposed system to develop an effective spam zombie detection system named SPOT is used in monitoring outgoing messages of a network. SPOT is designed based on a statistical method called sequential probability ratio test (SPRT). SPOT can be used to test between two hypotheses whether the machine is compromised or not. SPOT has surpassed the false positive and false negative error rates. It also minimizes the number of required observations to detect a spam zombie.

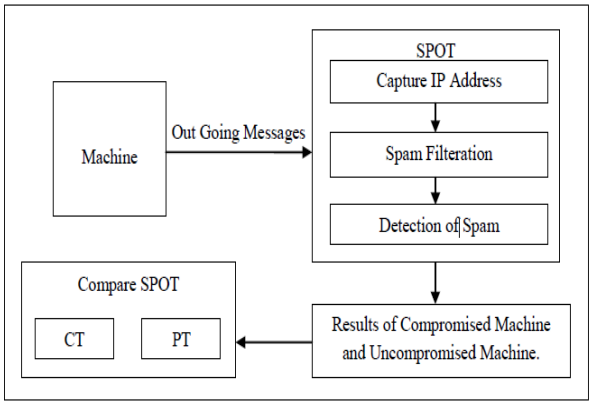

Fig.3.1 Flow of Spam Detection Process

files may contain more number of records which can be greater than thirty thousand lines of text. These text may contain both readable and non-readable data. Whenever the mails are sent than the attachments containing the files are tested between these two trained files, we call this trained files as the two hypothesis and finally rating is measured and based on this rating SPRT decides whether the mail file is spam or genuine. There are more two algorithms bounded to SPOT. One we call as Count Threshold and other is the Percentage Threshold. A particular choice is made when SPRT is finished so as to choose one technique among two and later the control is transferred to that technique. These two thresholds are called as user-defined threshold algorithms because here user will give the constraint limited values to detect the mail spams which are independent for system administrators.

The Count Threshold (CT) detection used to count the number of times the mail messages arrives at each IP Address location. Here user constraints the limited value usually integer. So if the count of the mail message files arrived at particular IP Address is less than this limited value specified and if and only if the records in the file contains greater than twenty lines than it displays as it is the spam mail.

The Percentage Threshold (PT) detection is executed between two limited constraints. One we call it as minimum limit and another is the maximum limit, both are integer values. The minimum limit is used to count the total number of files sent from various address locations and if it exceeds its limit than the mail containing the file is a spam file. Another limit the maximum value is used to check whether the number of mails sent are within its value specified than is less than the limit. But both will be displayed as spam if and only if its records containing lines are greater than twenty lines.

In the above discussion of the spam zombie detection algorithms we have for simplicity ignored the potential impact of dynamic IP addresses and assumed that an observed IP corresponds to a unique machine. In the following we informally discuss how well these algorithms fair with dynamic IP addresses. SPOT can work extremely well in the environment of dynamic IP addresses. To understand the reason we note that SPOT can reach a decision with a small number of observations and shows the average number of observations required for SPRT to terminate with a conclusion. In practice, we have noted that 3 or 4 observations are sufficient for SPRT to teach a decision for the vast majority of cases. If a machine is compromised, it is likely that more than 3 or 4 spam messages will be sent before the (unwitting) user shutdowns the machine and the corresponding IP address gets re-assigned to a different machine. Therefore, dynamic IP addresses will not have any significant impact on SPOT.

So only CT and PT detection are the only two techniques that can support the dynamic behavior. Both the algorithms has surpassed the system SPRT technique in automatically detecting the spam files which are independent of dynamic support. There are no error rates like false positive or false negative in case of SPRT implemented by SPOT. So higher efficiency is maintained by successful execution with less number of observations. It is a very simple process for administrators in detecting the comprised machines because of faster mode of execution.

4. IMPLEMENTATION

4.1 spot implementation using SPRT

The methodology applied in this work is based on Paul-Graham Implementation method using Bayesian calculation. The method uses two files one is **trainGood** and another is **trainBad**. After performing the execution the result of the output from these files are compared with the outgoing message and is applied for filtration to determine the rating. The rating is the key element need to be considered to determine whether the outgoing message is spam or not. This rating performs the Bayesian calculation and finalizes the output.
Methodology performs the following steps to execute the process.

Algorithm of Processing Steps

Step 1) For trainGood and trainBad Messages.
Split the content after reading them into tokens of words and store them in HashMap.

Step 2) Next is to calculate the count of words as below.
Here for both the trainGood and trainBad Message files calculate the Total Words.
For trainGood Message: calculate the ratio as below.
Ratio of Good = 2 * (count of Good Words / Total Words)
Here Total words are considered from trainGood file and ratio is obtained by multiplying with 2 in order to avoid the error rates.
For trainBad Message: calculate the ratio as below.
Ratio of Bad = (count of Bad Words / Total Words)
Here Total words are considered from trainBad file.

Step 3) Determine the probability of spam for each word in HashMap containing both from trainGood and trainBad Message files as follows.
if (ratio of Good + ratio of Bad > 0)
Probability of Spam = (ratio of Bad / ratio of Good + ratio of Bad)
if (probability of Spam < 0.01)
Probability of Spam = 0.01
else if (probability of Spam > 0.99)
Probability of Spam = 0.99

Step 4) Here now the file is dynamically selected this is an outgoing message file and is than tested as below.
Dividing the content into tokens and comparing them with the words in

HashMap. If the words are not present in HashMap than initialize the word with the average probability 0.4 and than these words are stored in ArrayList called **interesting** words.

Step 5) The words in ArrayList are compared in it, in order to get 15 most interesting words and store them sequentially in ArrayList with the following function.

```
interesting ()
{
Return Math.abs ( 0.5 – probability
of spam ) ;
}
```

Step 6) In **interesting** ArrayList each word is compared with the remaining words probability and store the first 15 interesting probabilities in ArrayList.

```
// initialize a new_word from the same
ArrayList.
if ( word.interesting () >
new_word.interesting () )
{
//add that word into interesting ArrayList.
}
```

Step 7) Now here Baye's rule is applied and probabilities of positive product and negative products are calculated.

```
Probability of Positive Product = 1
Probability of Negative Product = 1
for ( i = 0 ; i <  interesting.size () ; i++ )
{
 // Get word from interesting ArrayList.
Probability of Positive Product *=
probability of spam word from interesting
ArrayList.
Probability of Negative Product *= 1 -
probability of spam word from interesting
ArrayList.
}
```

Step 8) Next ratio of Probability of Spam is calculated as below.

```
Probability of Spam = ( Probability of Positive
                        Product ) /
                        (Probability of positive
                        product + probability of
                        Negative product ).
```

Step 9) Determining the Spam File with the following conditions.

```
if ( probability of spam > 0.9 )
// It is Spam mail.
else
// It is Genuine mail.
```

By this the Paul-Graham methodology helps the system administrators to automatically detect the spam files when messages are coming out from their source locations.

4.2 CT DETECTION

The module is used to count the number of files based on threshold limit. The threshold limit is the constant choosed by the user.

Here we set the Threshold limit to 3, if the file exists greater than equal to 3 times in database than this module is set into active state.

It calculates the number of times the message was sent and also detects the message is either spam mail or not.

The Detection of spam is calculated based on the number of lines existing in the file. If the file contains lines greater than or equal to 20 than it displays the output as spam mail else no output is tracked to display.

4.3 PT DETECTION

In this module we define two Threshold user based limits one is let **ca** = 20 and other is **p** = 30. Where **ca** determines the minimum number of messages a machine must send and **p** determines the maximum spam percentage of the machine.

Now we will calculate the count of all similar files existing in database and if the count of files is greater than or equal to 20 than it displays that file as spam mail else not.

Percentage Threshold depends on the current IP address which is randomly generated from the system and detection starts when the file contains lines greater than or equal to 20.

In this module the total number of IP address generated are stored in the MySQL database. Here the total number of files contains the same count of currently generated IP address. So count of the IP address of current machine is compared with the percentage Threshold **p** = 30 which is less than **p**.

Thus spam files are detected from the above two cases and output is generated when there is spam message and output is not tracked when it is genuine message.

5. CONCLUSION AND FUTURE WORK

In this paper we 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 surpassed both the false positive and false negative error rates. It also

minimizes the number of required observations to detect a spam zombie. In addition, we also showed that SPOT outperforms two other detection algorithms based on the number and percentage of spam messages sent by an internal machine, respectively. The main usage is sender can identify the sending mails as either spam or not and weather his system is compromised system or an uncompromised one and the user defined thresholds algorithms which are CT and PT can support the dynamic behavior to detect the spam mails associated with different address locations.

This paper only deals with detecting the spam zombies across the network by monitoring the outgoing messages but not provided a resultant solution to stop or interrupt the hacker or third party from transmission of messages. The whole process is about detecting the spams by using different techniques as SPOT filter, CT Detection and PT Detection but not provided any implementation procedure to resolve the issue of spams that leads to attacks. So our future enhancements are going to be prepared to discover new algorithms to stop these hackers from the transmission of spam mails and providing an efficient utilization of environment in resolving the spam mails also to provide the network environment for SPRT in detecting the spams through various ip address dynamically.

6. REFERENCES

[1] P. Bacher, T. Holz, M. Kotter, and G. Wicherski, "Know Your Enemy: Tracking Botnets," http://www.honeynet.org/papers/ bots, 2011.

[2] Z. Chen, C. Chen, and C. Ji, "Understanding Localized-Scanning Worms," Proc. IEEE Int'l Performance, Computing, and Comm. Conf. (IPCCC '07), 2007.

[3] R. Droms, "Dynamic Host Configuration Protocol," IETF RFC 2131, Mar. 1997.

[4] Z. Duan, Y. Dong, and K. Gopalan, "DMTP: Controlling Spam through Message Delivery Differentiation," Computer Networks, vol. 51, pp. 2616-2630, July 2007.

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

[6] Z. Duan, K. Gopalan, and X. Yuan, "Behavioral Characteristics of Spammers and Their Network Reachability Properties," Proc. IEEE Int'l Conf. Comm. (ICC '07), June 2007.

[7] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection," Proc. 17th USENIX Security Symp., July 2008.

[8] 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.

[9] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting Botnet Command and Control Channels in Network Traffic," Proc. 15th Ann. Network and Distributed System Security Symp. (NDSS '08), Feb. 2008.

[10] N. Ianelli and A. Hackworth, "Botnets as a Vehicle for Online Crime," Proc. First Int'l Conf. Forensic Computer Science, 2006.

[11] J.P. John, A. Moshchuk, S.D. Gribble, and A. Krishnamurthy, "Studying Spamming Botnets Using Botlab," Proc. Sixth Symp. Networked Systems Design and Implementation (NSDI '09), Apr. 2009.

[12] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan, "Fast Portscan Detection Using Sequential Hypothesis Testing," Proc. IEEE Symp. Security and Privacy, May 2004.

[13] J. Klensin, "Simple Mail Transfer Protocol," IETF RFC 2821, Apr. 2001.

[14] J. Markoff, "Russian Gang Hijacking PCs in Vast Scheme," The New York Times, http://www.nytimes.com/2008/08/06/technology/06hack.html, Aug. 2008.

[15] P. Wood et al., "MessageLabs Intelligence: 2010 Annual Security Report," 2010.

[16] S. Radosavac, J.S. Baras, and I. Koutsopoulos, "A Framework for MAC Protocol Misbehavior Detection in Wireless Networks," Proc. Fourth ACM Workshop Wireless Security, Sept. 2005.

[17] A. Ramachandran and N. Feamster, "Understanding the Network- Level Behavior of Spammers," Proc. ACM SIGCOMM, pp. 291-302, Sept. 2006.

[18] Source or More Information: Detecting Spam Zombies by Monitoring Outgoing Messages by Zhenhai Duan, Peng Chen, Fernando Sanchez Florida State University*{duan, pchen, sanchez}@cs.fsu.edu*Yingfei Dong University of Hawaii *yingfei@hawaii.edu* Mary Stephenson, James Barker Florida State University *{mstephenson, jmbarker}@fsu.edu.*