ijecs open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 2 Issue 11 November, 2013 Page No. 3213-32016

Recognizing Spam Zombies by Monitoring leaving Messages

R.Vasantha kumar[1],Mr.K.Ravi Kumar[2]
Research Scholar[1], Asst,professor[2]
Department of Computer Science, Tamil University,
Thanjavur-10

ABSTRACT:

Compromised machines are one of the input security threats on the Internet; they are frequently used to launch various security attacks such as spamming and spreading malware, DDoS, and identity theft. Given that spamming provides a key economic incentive for attackers to recruit the large number of compromised machines, we focus on the detection of the compromised machines in a network that are involved in the spamming activities, commonly known as spam zombies. We 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.

Index Terms—Compromised machines, spam zombies, compromised machine detection algorithms.

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 widespread—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 e-mail 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 identify the compromised machines in the network as outgoing messages pas 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 I the environment we consider. The nature of sequentially observing outgoing messages gives rise to the sequential detection problem. In this paper, we 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).

In this paper, we develop the SPOT detection system to assist system administrators in automatically identifying the compromised machines in their networks. We also evaluate the performance of the SPOT system based on a two-month e-mail trace collected in a large US campus network. Our evaluation studies show that SPOT is an effective and efficient system in automatically detecting compromised machines in a network. For example, among the 440 internal IP addresses observed in the e-mail trace, SPOT identifies 132 of them as being associated with compromised machines. Out of the 132 IP addresses identified by SPOT, 126 can be either independently confirmed (110) or are highly likely (16) to be compromised. Moreover, only seven internal IP addresses associated wit compromised machines in the trace are missed by SPOT. In addition, SPOT only needs a small number of observations to detect a compromised machine. The majority of spam zombies are detected with as little as three spam messages. For comparison, we also design and study two other spam zombie detection algorithms based on the number of spam messages and the percentage of spam messages originated or forwarded by internal machines, respectively. We compare the performance of SPOT with the two other

detection algorithms to illustrate the advantages of the SPOT system.

2, PROBLEM FORMULATION AND ASSUMPTIONS:

In this section, we prepare the spam zombie detection problem in a network. In particular, we discuss the network model and assumptions we make in the detection problem.

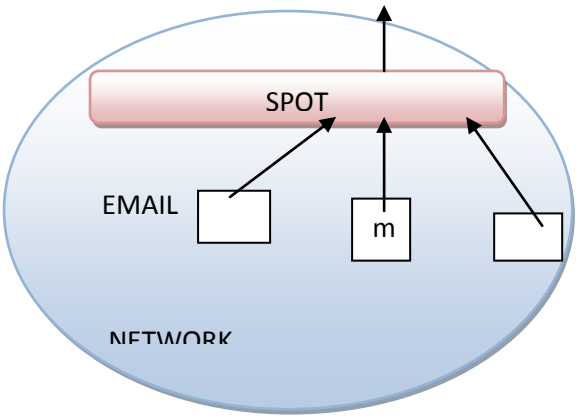

Fig. 1 illustrates the logical view of the network model. We assume that messages originated from machines inside the network will pass the deployed spam zombie detection system. This assumption can be achieved in a few different scenarios. For example, the outgoing e-mail traffic (with destination port number of 25) can be replicated and redirected to the spam zombie detection system. We also assume that a sending machine m as observed by the spam zombie detection system is an end-user client machine. It is not a mail relay server. This assumption is just for the convenience of our exposition. The proposed SPOT system can handle the case where an outgoing message is forwarded by a few internal mail relay servers before leaving the network. In addition, we assume that an IP address corresponds to a unique machine and ignores the potential impacts of dynamic IP addresses on the detection algorithms in the presentation of the algorithms .

3.SPAM ZOMBIE DETECTION ALGORITHMS

In this section, we will develop three spam zombie detection algorithms. The first one is SPOT, which utilizes the Sequential Probability Ratio Test presented in the last section. We discuss the impacts of SPRT parameters on SPOT in the context of spam zombie detection. The other two spam zombie detection algorithms are developed based on the number of spam messages and the percentage of spam messages sent from an internal machine, respectively.

3.1 SPOT Detection Algorithm

SPOT is designed based on the statistical tool SPRT we discussed in the last section. In the context of detecting spam zombies in SPOT, we consider H1 as a detection and H0 as a normality. That is, H1 is true if the concerned machine is compromised, and H0 is true if it is not compromised. In addition, We discuss how users configure the values of the four parameters after we present the SPOT algorithm. Based on the user-specified values of _ and _, the values of the two boundaries A and B of SPRT are computed using (5). In the following, we describe the SPOT detection algorithm. Algorithm 1 outlines the steps of the 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. For each observed IP address, SPOT maintains the logarithm value of the corresponding probability ratio _n, whose value is updated according to (3) as message n arrives from the IP address (lines 6 to 12 in Algorithm 1). Based on the relation between _n and A and B, the algorithm determines if the corresponding machine is compromised, normal, or a decision cannot be reached and additional observations are needed.

Algorithm 1. SPOT spam zombie detection system

1: An outgoing message arrives at SPOT

2: Get IP address of sending machine m

3: // all following parameters specific to machine m

4: Let n be the message index

5: Let Xn ¼ 1 if message is spam, Xn ¼ 0 otherwise

6: if (Xn ¼¼ 1) then

7: // spam, 3

8: _nþ ¼ ln _1

_0

9: else

10: // nonspam

11: _nþ ¼ ln 1__1

1__0

12: end if

13: if (_n _ B) then

14: Machine m is compromised. Test terminates for m.

15: else if (_n _ A) then

16: Machine m is normal. Test is reset for m.

17: _n ¼ 0

18: Test continues with new observations

19: else

20: Test continues with an additional observation

21: end if

We note that in the context of spam zombie detection, from the viewpoint of network monitoring, it is more important to identify the machines that have been compromised than the machines that are normal. After a 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 and removed from the list, the SPOT detection system does not need to further monitor the message sending behavior of the machine On the other hand, a machine that is currently normal may get compromised at a later time. Therefore, we need to continuously monitor machines that are determined to be normal by SPOT. Once such a machine is identified by SPOT, the records of the machine in SPOT are reset, in particular, the value of _n is set to zero, so that a new monitoring phase starts for the machine (lines 15 to 18).

3.2 Parameters of SPOT Algorithm

SPOT requires four user-defined parameters: _, _, _1, and _0. In the following, we discuss how a user of the SPOT algorithm configures these parameters, and how these parameters may affect the performance of SPOT. As discussed in the previous section, _ and _ are the desire false positive and false negative rates. They are normally small values in the range from 0.01 to 0.05, which users of SPOT can easily specify independent of the behaviors of the compromised and normal machines in the network. The values of _ and _ will affect the cost of the SPOT algorithm, that is, the number of observations needed for the algorithm to reach a conclusion.

In general, a smaller value of _ and _ will require a larger number of observations for SPOT to reach a detection. Ideally, _1 and _0 should indicate the true probability of a message being spam from a compromised machine and a normal machine, respectively, which are hard to obtain. A practical way to assign values to _1 and _0 is to use the detection rate and the false positive rate of the spam filter deployed together with the spam zombie detection system, respectively. Given that all the widely used spam filters have a high detection rate and low false positive rate , values of _1 and _0 assigned in this way should be very close to the true probabilities. To get some intuitive understanding of the average number of required observations for SPRT to reach a decision.

3.3 Spam Count and Percentage-Based Detection Algorithms

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. In CT, the time is partitioned into windows of fixed length T. A user-defined threshold parameter Cs specifies the maximum number of spam message that may be originated from a normal machine in any time window. The system monitors the number of spam messages n originated from a machine in each window. Similarly, in the PT detection algorithm, the time is partitioned into windows of fixed length T. PT monitors two e-mail sending properties of each internal machine in each time window: one is the percentage of spam messages sent from a machine, another the total number of messages. Let N and n denote the total messages and spam messages originated from a machine m within a time window, respectively, then PT declares machine m as being compromised if N _ Ca and n N > P, where Ca is the minimum number of messages that a machine must send, and P is the user-defined maximum spam percentage of a normal machine. The first condition is in place for preventing high false positive rates when a machine only generates a small number of messages. For example, in an extreme case, a machine may only send a single message and it is a spam, which renders the machine to have a 100 percent spam ratio. However, it does not make sense to classify this machine as being compromised based on this small number of messages generated.

In the following, we briefly compare the two spam zombie detection algorithms CT and PT with the SPOT system. The three algorithms have the similar running tim and space complexities. They all need to maintain a record for each observed machine and update the corresponding record as messages arrive from the machine. However, unlike 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 userdefine parameters (_, _, _1, and _0) in SPOT is relativel 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 behaviors of the compromised and normal machines in the concerned network and a training based on the behavioral history of the two different types of machines in order for them to work reasonably well in the network. For example, it can be challenging to select the “best” length of time windows in CT and PT to obtain the optimal false positive and false negative rates. We discuss how an attacker may try to evade CT and PT (and SPOT) .

3.4 Impact of Dynamic IP Addresses

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 the three algorithms fair with dynamic IP addresses. We formally evaluate the impacts of dynamic IP addresses on detecting spam zombies in the next section using a two-month e-mail trace collected on a large US campus network. SPOT can work extremely well in the environment of dynamic IP addresses.

4.CONCLUSION

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 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, 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.

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.