International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Impact Factor (2012): 3.358

Malware Seeker: A Network Intrusion Detection and Correlation Technique against Peer to Peer Botnet

A. Shameem.[1], M. Parveentaj[2]

[1]MCA, (M.Phil) Jahendra Sarawathy Vidhyalaya Art and Science College, Coimbatore, India

[2]Assistant Professor, M.C.A., ADCA, M.Phil Jahendra Sarawathy Vidhyalaya Art and Science College, Coimbatore, India

Abstract: ***Current research has been carried out against Malware propagating in the Peer to Peer parallel distributed system is challenging and cumbersome task .In Existing solutions , lot of efforts has been carried against the malware evolution and activities but solutions are ineffective against the detection rate and accuracy in detection due to growing of high traffic calls to the networks. In this paper, we propose a novel Solution to mitigate the malicious activities of peer to peer Botnet attackers through the detection mechanism and countermeasure strategies named as Malware Seeker. To prevent vulnerable Host machines from being compromised by the peer to peer Malware, we propose a multiphase distributed vulnerability detection through the Principle of component analysis of each traffic data, measurement and countermeasure selection mechanism called Malware Seeker which is built on attack graph-based analytical models based on classification process and reconfigurable against update solutions to virtual network-based countermeasures with respect to command and Control established by botmaster . The proposed framework leverages hierarchical models to build a monitor and control process to classify the network traffic data to the virtual machine to significantly improve attack detection and mitigate attack consequences such as spamming , scanning an exploitation . Extensive Evaluation will demonstrate the behaviors of the proposed System against the Malware in file sharing process with legitimate and illegitimate and Malware causes in the peer to peer network process with huge amount of network information.***

Keywords: Intrusion Detection, Peer to Peer Network, BOTNET, DDOS, Network Security, Attack Correlation

1. Introduction

Network Security is real challenging in day to today life .There were several attacks exploiting in the network applications. BOTNET is a collection of compromised hosts (a.k.a. bots) that are remotely controlled by an attacker (the botmaster) through a command and control (C&C) channel. Botnets serve as the infrastructures responsible for a variety of cyber-crimes, such as spamming, distributed denial of-service (DDoS) attacks, identity theft, click fraud, etc. The C&C channel is an essential component of a botnet because botmasters rely on the C&C channel to issue commands to their bots and receive information from the compromised machines. However, a fundamental disadvantage of centralized C&C servers is that they represent a *single point of failure*. In order to overcome this problem, botmasters have recently started to build botnets with a more resilient C&C architecture, using a peer-to-peer (P2P) structure [1]–[3] or hybrid P2P/centralized C&C structures [4]. Bots belonging to a P2P botnet form an overlay network in which any of the nodes (i.e., any of the bots) can be used by the botmaster to distribute commands to the other peers or collect information from them. Notable examples of P2P botnets are represented by Nugache [5], Storm [2], Waledac [4], and even Confiker, which has been shown to embed P2P capabilities [3]. Detecting botnets is of great importance. However, designing an effective P2P-botnet detection system is faced with several challenges. First, the P2P file-sharing and communication applications, such as Bittorrent, emule, and skype, are very popular and hence C&C traffic of P2P botnets can easily blend into the background P2P traffic. Second, as the volume of network traffic grows rapidly, the deployed detection system is required to process a huge amount of information efficiently. To date, a few approaches capable of detecting P2P botnets have been proposed [7]–[9]. However, these approaches cannot address all the aforementioned challenges. For example, BotMiner [7] identifies a group of hosts as bots belonging to the same botnet if they share similar communication patterns and meanwhile perform similar malicious activities, such as scanning, spamming, exploiting, etc. Unfortunately, the malicious activities may be stealthy and non-observable, thereby making BotMiner ineffective. In addition, BotMiner's scalability is significantly constrained [10]. Yen et al. [8] proposed an algorithm that aims to distinguish between hosts that run legitimate P2P file sharing applications and P2P bots. Nevertheless, this algorithm [8] does not take into account the fact that a bot may coexist with a legitimate P2P application on the same host. In this paper, we present a novel *scalable* botnet detection system capable of detecting *stealthy* P2P botnets. The high scalability of our system stems from the parallelized computation with bounded computational complexity.

We redesigned the clustering-based P2P client detection algorithm to enhance its efficiency, which decreases the processing time by at least 68% compared to the original design. Third, we parallelize our system to boost its scalability and demonstrated its efficiency. Finally, we have empirically evaluated the extent to which configurable parameters affect the detection performance and manifested that our system is effective over a large range of parameter values. organization of the proposed work is as follows , it is been started with Background and literatures in section 2 , proposed system in Section 3 with Design and algorithm, Experimental section 4 with Results and analysis and finally Conclusion and suggestion for future enhancement.

2. Related Work

A few approaches capable of detecting P2P botnets have been proposed [7]–[9], [12]–[14]. Compared with the existing methods [7]–[9], the design goals of our approach are different in that: 1) our approach does not assume that malicious activities are observable, unlike [7]; 2) our approach does not require any botnet-specific information to make the detection, unlike [9]; 3) our approach needs to detect the compromised hosts that run both P2P bot and other legitimate P2P applications[11] at the same time, unlike [8]; and 4) different from [7]–[9], our approach has high scalability as a built-in feature. Other methods [12]–[14] use machine learning for detection, which require labeled P2P botnet data to train a statistical classifier. Unfortunately, acquiring such information is a challenging task, thereby drastically limiting the practical use of these methods.

To achieve the aforementioned design goals, our system includes multiple components. The first one is a *flow clustering*- based analysis approach to identify hosts that are mostly likely running P2P applications. In contrast to existing approaches of identifying hosts running P2P applications [15]–[19], our approach differs in the following ways:

1) unlike [16], our approach does not need any content signature because encryption will make content signature useless;
2) our approach does not rely on any transport layer heuristics (e.g., fixed source port) used by [15], [17], which can be easily violated by P2P applications;
3) we do not need training data set to build a machine learning based model as used in [18], because it is very challenging to get traffic of P2P botnets before they are detected;
4) in contrast to [19], our approach can detect and profile various P2P applications rather than identifying a specific P2P application (e.g., Bittorrent); and
5) our analysis approach can estimate the active time of a P2P application, which is critical for botnet detection.

3. Proposed System

3.1 DDOS Characteristics Based on Time and identity (IP address) for Data access and File sharing Characterize the Threat Models –Supervised DDOS

It is an attempt to make a machine or network resource unavailable to its intended users. Although the means to carry out, motives for, and targets of a DoS attack may vary, it generally consists of efforts to temporarily or indefinitely interrupt or suspend services of a host connected to the Internet. DoS (Denial of Service) attacks are sent by one person or system.It makes a network service or host server unusable, usually by overloading the server or network, common causes of DDOS is;

Consume host resources
- TCP SYN floods
- ICMP ECHO (ping) floods

Consume bandwidth
- UDP floods
- ICMP floods

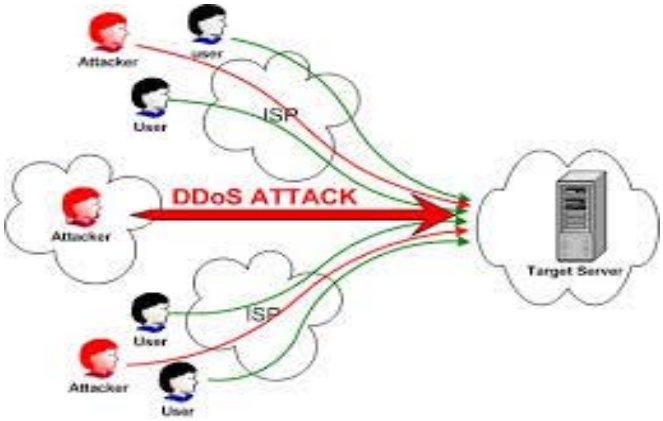

Figure 1: Workflow of the DDOS attack

3.2 Essential Characteristics and Manifestation of Attack

- Unusually network performance (opening files or accessing web sites) is slow.
- Unavailability of a particular web site
- Inability to access any web site
- Dramatic increase in the number of spam emails received—(this type of DoS attack is considered an e-mail bomb)
- Disconnection of a wireless or wired internet connection
- Long term denial of access to the web or any internet services

Denial-of-service attacks can also lead to problems in the network 'branches' around the actual computer being attacked. For example, the bandwidth of a router between the Internet and a LAN may be consumed by an attack, compromising not only the intended computer, but also the entire network or other computers on the LAN. If the attack is conducted on a sufficiently large scale, entire geographical regions of Internet connectivity can be compromised without the attacker's knowledge or intent by incorrectly configured or flimsy network infrastructure equipment.

Algorithm1: Monitoring the Application Process:
Input: Ip Address to be read,
Process:
If (IP! = Registered address)||(Current Process==suspicious)
Return Operation To Alert correlation Mechanism
For (IP segment == Malicious list; IP segment is included)
If (IP==Non trusted Loop in Benign Graph)
Statement: Attach the IP in the Alert Graph Correlation
Else
Statement: copy the protocol malware into suspicious (Stack) for Blocking
Then
Return (permit the IP to Table for Prevention)

Algorithm2: Detecting the Malware Process:
Input:
If (Suspicious stack == Incoming Node)
Statement: Block the data access in the Network or Host

Condition for Blocking
$\alpha = \sum \beta(x)$
Where α is the Response of the system
β is the suspicious list In terms of x (IP data or packet header of the request in particular Time)
Else
Statement: Allow the Request to Process.

3.3 Characterize the threat models –Unsupervised (unlabelled)

Novel Threats in the network and host system is difficult to detect. To enable the novel attack detection set of attributes from each event (IP packet or TCP connection), including strings in the payload, and induce a set of conditional rules which have a very low probability of being violated in a no stationary model of the normal network traffic in the training data.

Figure 4.4: Botnet Propagation in the Network

3.4 Learning Rules for Anomaly Detection

We extend the network traffic model to include a large number of attributes, including the application payload. We introduce a non stationary model, in which the probability of an event (an attribute having some value) depends on the time of its most recent occurrence, and not on its average frequency. We introduce an efficient algorithm for selecting good rules for anomaly detection from a rule space that is exponentially large in the number of attributes.

3.5 Attack Correlation based on the Vulnerability Ratio and Attack Strategies:

Vulnerability of networks is been estimated through considering the geometric properties of regional failures and network nodes. To evaluate the criticality of node locations and determine the critical areas in a network, we propose the concept of α-critical-distance with a given failure impact ratio α, and we formulate two optimization problems based on the concept. By analyzing the geometric properties of the problems, we show that although finding critical nodes or links in a pure graph is a NP-complete problem, the problem of finding critical areas has polynomial time complexity. We propose two algorithms to deal with these problems and analyze their time complexities. Additionally, we find that with the same impact ratio α, the topologies examined have larger α-critical-distances when the network performance is measured using the giant component size instead of the other two metrics. Similar results are obtained when the network performance is measured using the average two terminal reliability and the network efficiency, although computation of the former entails less time complexity than that of the latter

3.6 Developing the flow based Clustering Technique as Detection Technique

Cluster is been utilized for parallel detection in order to increase the detection speed in large networks.

3.7 Developing a PSO based Clustering Technique as detection Technique

Particle has taken time or identity or data or file. Velocity has been no. of occurrence in specified time. It measures the type of the file and size of file of same content or data.

4. Experimental Results

We evaluated the system against botnet and DDOS attacks with screens as follows. All but a few attacks show some evidence in the inside network traffic, so to simplify the results we just report all detections.

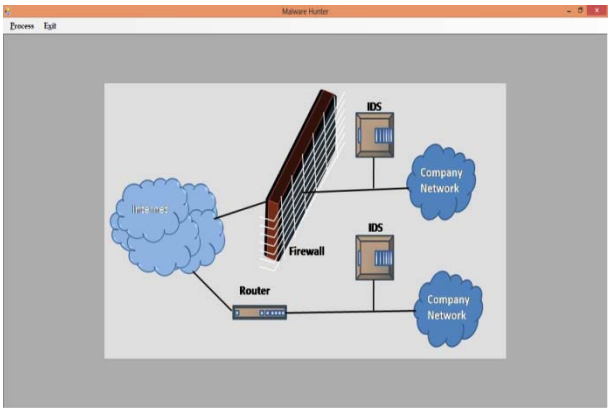

Figure: Overall process outlet Screen

We expected to prove the proposed solution, thus achieving our goal "establish dynamic defensive mechanism-based software defined networking approach that involves multiphase intrusion detections.

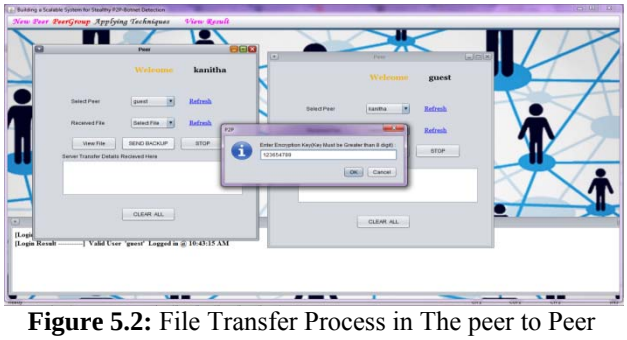

Figure 5.2: File Transfer Process in The peer to Peer Networks

The Data Communication by the BotNet attacks is shown as follows

Figure 5.3: Spam Detected in the file transferring by the BOTMASTER

In each concerned system call, we set up one or more checkpoints, each of which is responsible for checking the behaviors belonging to the same operation with the support of a modifiable behavior list in memory.

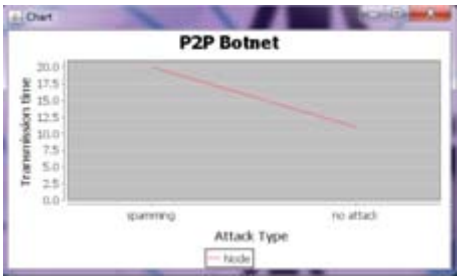

Figure: Detection Accuracy of the Malware Seeker for PEER to PEER BOTNET

The performance results provide us a benchmark for the given hardware setup and shows how much traffic can be handled by using a single detection domain.

Data Recovery Process

Figure: Rate of Successful data recovery

To scale up to a data center-level IDS, a decentralized approach must be devised. We utilized the synthesis data for the experimental evaluation and proposed a multiple technique random traffic data and Experimental results has been derived.

5. Conclusion and Future Work

We implemented a multiphase distributed vulnerability detection through the Principle of component analysis of each traffic data under the dynamic attack evolution area, measurement and countermeasure selection mechanism called Malware Hunter which is built on attack graph-based analytical models based on classification process and reconfigurable against update solutions to virtual network-based countermeasures. The classification process has been carried using the principle component analysis to establish the efficient detection mechanism against the attacks. The System has been included with different characters for modeling with attack detection solutions for botnet attacks. The proposed framework leverages hierarchical models to build a monitor and control process to classify the network traffic data to the virtual machine to significantly improve attack detection and mitigate attack consequences. Hence malware hunter achieves the good detection performance against all types of network and host based intrusion evolving. As a future work, We introduce novel Solution to mitigate the malicious activities of Botnet attackers through the detection mechanism and countermeasure strategies .To prevent vulnerable virtual machines from being compromised in the cloud, we propose a multiphase distributed vulnerability detection for each traffic data, measurement and countermeasure selection.

Reference

[1] S. Stover, D. Dittrich, J. Hernandez, and S. Dietrich, "Analysis of the storm and nugache trojans: P2P is here," in *Proc. USENIX*, vol. 32.2007, pp. 18–27.
[2] P. Porras, H. Saidi, and V. Yegneswaran, "A multi-perspective analysis of the storm (peacomm) worm," Comput. Sci. Lab., SRI Int., Menlo Park, CA, USA, Tech. Rep., 2007.38 IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, VOL. 9, NO. 1, JANUARY 2014
[3] P. Porras, H. Saidi, and V. Yegneswaran. (2009). *Conficker C Analysis*[Online].Available: http://mtc.sri.com/Conficker/addendumC/index.html
[4] G. Sinclair, C. Nunnery, and B. B. Kang, "The waledac protocol: The how and why," in *Proc. 4th Int. Conf. Malicious Unwanted Softw.*, Oct. 2009, pp. 69–77.
[5] R. Lemos. (2006). *Bot Software Looks to Improve Peerage* [Online] Available: http://www.securityfocus.com/news/11390
[6] Y. Zhao, Y. Xie, F. Yu, Q. Ke, and Y. Yu, "Botgraph: Large scale spamming botnet detection," in *Proc. 6th USENIX NSDI*, 2009, pp. 1–14.
[7] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: Clustering analysis of network traffic for protocol- and structure-independent botnet detection," in *Proc. USENIX Security*, 2008, pp. 139–154.
[8] T.-F. Yen and M. K. Reiter, "Are your hosts trading or plotting? Telling P2P file-sharing and bots apart," in *Proc. ICDCS*, Jun. 2010, pp. 241–252.
[9] S. Nagaraja, P. Mittal, C.-Y. Hong, M. Caesar, and N. Borisov, "BotGrep: Finding P2P bots with structured graph analysis," in *Proc. USENIX Security*, 2010, pp. 1–16.
[10] J. Zhang, X. Luo, R. Perdisci, G. Gu, W. Lee, and N. Feamster,"Boosting the scalability of botnet detection using adaptive traffic sampling," in *Proc. 6th ACM Symp. Inf., Comput. Commun. Security*,2011, pp. 124–134.
[11] J. Zhang, R. Perdisci, W. Lee, U. Sarfraz, and X. Luo, "Detecting stealthy P2P botnets using statistical traffic fingerprints," in *Proc.IEEE/IFIP 41st Int. Conf. DSN*, Jun. 2011, pp. 121–132.
[12] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, *et al.*, "Detecting P2P botnets through network

behavior analysis and machine learning," in *Proc. 9th Annu. Int. Conf. PST*, Jul. 2011, pp. 174–180.
[13] D. Liu, Y. Li, Y. Hu, and Z. Liang, "A P2P-botnet detection model and algorithms based on network streams analysis," in *Proc. IEEE FITME*, Oct. 2010, pp. 55–58.
[14] W. Liao and C. Chang, "Peer to peer botnet detection using data mining scheme," in *Proc. IEEE Int. Conf. ITA*, Aug. 2010, pp. 1–4.
[15] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevelm traffic classification in the dark," in *Proc. ACM SIGCOMM*, 2005, pp. 229–240.
[16] S. Sen, O. Spatscheck, and D. Wang, "Accurate, scalable in-network identification of P2P traffic using application signatures," in *Proc. 13th ACM Int. Conf. WWW*, 2004, pp. 512–521.
[17] T. Karagiannis, A. Broido, M. Faloutsos, and K. Claffy, "Transport layer identification of P2P traffic," in *Proc. 4th ACM SIGCOMM Conf. IMC*, 2004, pp. 121–134.