ISSN 2320 - 2602

Volume 3, No.5, May 2014

International Journal of Advances in Computer Science and Technology

Available Online at http://warse.org/pdfs/2014/ijacst02352014.pdf

A Survey On Botnet Detection Approaches In Peer-To-Peer Network

P.Senthil vadivu [1], K.S.Karthika[2]

[1] Head & Associate Professor, Hindustan college of Arts and Science, India, sowju_sashi@rediffmail.com

[2] Research Scholar, Hindustan college of Arts and Science, India, karthimayav@gmail.com

ABSTRACT

Peer-to-peer network is a decentralized and distributed network where an individual nodes in the network performs as both providers and consumers of resources. This type of network is different from centralized network. In the centralized network, the client requests queries for accessing resources to the central servers. Malware is a harmful effect in the peer-to-peer networks. In the peer-to-peer network, a new type of malware which is called bots has arisen. Bots are distinctive in that they cooperatively preserve communication structures across nodes to robustly distribute commands from a command and control (C&C) node. The capability to organize and upload new commands to bots provides the botnet owner vast power when performing illegal activities, which contains the ability to organize surveillance attacks, execute DDoS extortion, distribution of spam for pay, and phishing. It is very significant for detecting botnets in the peer-to-peer network. In this survey to analyze different methods of detecting peer-to-peer botnets. BotMiner is one of the detection methods in which a group of hosts as bots belonging to the same botnet if they distribute comparable communication patterns. But this detection method is ineffective and there is restricted in scalability. BotGrep is a detection method which analyzes the network flows composed over multiple large networks by analyzing the communication graph formed by overlay networks. In the following survey to analyze different botnet detection methods to improve the detection accuracy in the peer-to-peer network.

Keywords: Peer-to-peer network, Botnet, Network security, intrusion detection.

1.INTRODUCTION

A peer-to-peer network is a network which is generated when two or more PCs are linked and share resources without using a centralized server. It is a communication model where each party has the similar capabilities and either party can initiate a communication session. But other models with which it might be distinctioned which includes the client/server model and the master/slave model. The communication nodes in the peer-to-peer network acts as both server and client. Peer-to-peer has come to illustrate applications in which users can utilize the Internet to exchange files with each other unswervingly or through a mediating server. The peer-to-peer network used to many applications which includes file sharing, instant messaging systems, online chat etc. A general example of a file transfer which utilizes the client-server model is the File transfer protocol (FTP) service in which the client and server programs are dissimilar: the clients instigate the transfer, and the servers persuade these requests.

In order to communicate with other nodes in the peer-to-peer network, the user first download and perform a peer-to-peer networking program. After the beginning process, the user enters the IP address of another computer which belongs to the network. Once the computer identifies another network member on-line, it will connect to that user's connection. Users can select how many member connections to find at one time and decide which files they wish to distribute or password protect.

Due to the decentralized nature in the peer-to-peer networks, it poses distinctive challenges from a computer security perspective. Each node plays a role in routing traffic through the network [1], malicious users can complete a variety of "routing attacks", or “denial of service attacks”. The common routing attacks include incorrect lookup routing in which the malicious nodes damage the routing tables of neighboring nodes by sending false information. In addition to that incorrect routing network partition in which the new nodes are joining the bootstrap through a malicious node, which places the new node in a partition of the network that is inhabited by other malicious nodes.

2. BOTNET CHARACTERISTICS

In the peer-to-peer network botnet is a collection of compromised hosts which are distantly controlled by an attacker 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 [1], identity theft, click fraud, etc. The command and control channel is an important 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.

In this survey, to examine a variety of Botnet detection approaches in the peer-to-peer network. BotMiner [2] finds a group of hosts as bots belongs to the same botnet if they distribute similar communication patterns and temporarily execute similar malicious activities like scanning, spamming, exploiting etc. But this method is inefficient because the malicious activities may be cautious and non-observable. Furthermore, in this method scalability is significantly limited. BotGrep [3] analyze network flows collected over multiple large networks and attempts to detect P2P botnets by examining the communication graph formed by overlay networks. This method needs a inclusive view of internet traffic and a prior detection results from supplementary systems to bootstrap the discovery process. But the drawback is particularly hard to obtain such information in practice.

3. PREVIOUS RESEARCH

To examine the different approaches which is capable of detecting P2P botnets.

Yao Zhao et.al Botgraph is a new type of detection method which identifies the new type of botnet spamming attacks targeting most important Web email suppliers. Botgraphs uncovers the associations among botnet activities by constructing large-user graphs and looking for tightly connected sub graph components. This enables us to find surreptitious botnet users that are difficult to detect when viewed in isolation. The main work is consisting of two folds [4]. The first contribution is to suggest a new graph based method to categorize the new web based abuse attack. This method is based on the observation that bot-users share IP addresses when they log in and send mails. Botgraph discovers the irregular distribution of IP addresses among bot-users by leveraging the random graph theory.

Shishir Nagaraja et.al suggested a botnet detection method which is called BotGrep [5]. This method seperates effectual peer-to-peer communication structures exclusively based on the information about which pairs of nodes communicate with one another.This algorithm iteratively partitions the communication graph into a faster-mixing and a slower-mixing piece, ultimately thinning on to the fast-mixing component. The BotGrep algorithm is content agnostic, thus it is not exaggerated by the choice of ports, encryption, or other content-based stealth techniques used by bots.

On the other hand, BotGrep must be matched with some sort of malware detection method, like anomaly or misuse detection, to be able to discriminate botnet control structures from other applications using peer-to-peer communication. But the drawback of this method is acquiring internet traffic information is a difficult task.

R. Perdisci et.al suggested BotMiner [6] which discovers a group of hosts as bots belonging to the same botnet if they share comparable communication patterns and meanwhile carry out comparable malicious activities, like scanning, spamming, exploiting, etc. Unfortunately, the malicious activities may be stealthy and non-observable thereby making BotMiner ineffective. In addition, BotMiner's scalability is considerably inhibited.

Kang G. Shin et.al suggested a framework to detect botnets by using combined host and network level information [11]. This framework first discovers the mistrustful hosts by identifying similar behaviors among different hosts using network- flow analysis, and authenticate the recognized suspects to be malicious or not by examining their in-host behavior. While bots within the same botnet are probable to obtain the same input from the botmaster and take similar actions, while benign hosts infrequently demonstrate such correlated behavior, this framework looks for flows with similar patterns and labels them as triggering flows.After that associates all consequent flows with each triggering flow on a host-by-host basis, validating the similarity among those associated groups. Whenever a group of hosts is recognized as distrustful by the network analysis, the host-behavior analysis results, based on a history of monitored host behaviors, are reported.

Signature based Detection method

Subhabrata Sen et.al proposed an efficient approach for discovering the P2P application traffic through application level signatures [7]. Firstly, identify the application level signatures by analyzing some obtainable documentations, and packet-level traces. After that utilize the discovered signatures to implement online filters which effectually and precisely track the P2P traffic even on high-speed network links. In this work, a real-time classification method is used in which operates on individual packets in the middle of the network and developed application-level signatures for a number of popular P2P applications. This signatures can be utilized directly to monitor and filter peer-to-peer traffic.

Classification method based detection

A. W. Moore and D. Zuev Network traffic classification is a main process to numerous network activities such as security examining, accounting. This work uses supervised machine learning for the traffic classification in the network [8]. Distinctively, we use data that has been hand-classified to one of a number of categories. Sets of data consisting of the category combined with descriptions of the classified flows are used to train the classifier. The Naïve bayes classifier is used to provide insight into the behavior of this technique itself.The sensitivity of the Naïve algorithm to its preliminary postulations and we plan to display that the use of two techniques, one to break the Gaussian postulations and the other to enhance the quality of discriminators as input, lead to signifigant improvements in the accuracy of the Naive Bayes technique. But the drawback of this method is there is less detection accuracy.

Thomas Karagiannis et.al suggested a new method for the traffic flow classification problem which is called BLINd Classification. The contribution of this work consists of twofolds [9]. First, we shift the focus from classifying individual flows to connecting Internet hosts with applications, and then categorizing their flows consequently. By observing the activity of a host gives more information and can disclose the nature of the applications of the host. Secondly, this method follows a different attitude from the earlier methods attempting to detain the intrinsic behavior of a host at three different levels: (a) social level, (b) network level, and (c) the application level. By merging these two methods, to classify the behavior of hosts at three different levels. Whereas at each level of classification offers increasing knowledge of host behavior, discovering specific applications depends on the unveiled "cross-level" characteristics.

Zhichun Li et.al suggested a design of P2PScope, a measurement tool, to discover and diagnose such unwanted traffic. In this work, to analyze the unnecessary traffic and analyze the major reasons. This will benefit for both end users and internet service providers [10]. For end users, such unwanted traffic affects their P2P system performance and can involve innocent users into DDoS attacks instinctively. Additionally, understanding the behavior patterns of anti-P2P peers will help to identify them, and thus equivocate their tracking to avoid potential warnings and lawsuits. But there is less detection accuracy.

4. BOTNET DETECTION METHODS

Botgraph detection method

The main intent of this method is to capture spamming email accounts used by botnets. There are two components in the Botgraph: One is aggressive sign-up detection and another one is stealthy botuser detection. The primary step of Botgraph is to recognize aggressive signups [4]. The main purpose is to restrict the total number of accounts owned by a spammer. In the second step, BotGraph discovers the residual stealthy bot-users based on their login activities. With the total number of accounts limited by the first step, spammers have to reclaim their accounts, resultant in correlations among account logins. Consequently BotGraph utilizes a graph based approach to recognize such correlations.The aggressive signup detection is based on the principle that signup events occur rarely at a single IP address. Even for a proxy, the number of users signed up from it should be approximately dependable over time. A abrupt amplification in signup activities is mistrustful, indicating that the IP address may be associated with a bot. A simple Exponentially Weighted Moving Average is used to discover the abrupt changes in signup activities. The second component identifies the remaining residual stealthy bot-accounts. As a spammer frequently controls a set of botusers, defined as a a bot-user group, these bot-users work in a collaborative way.

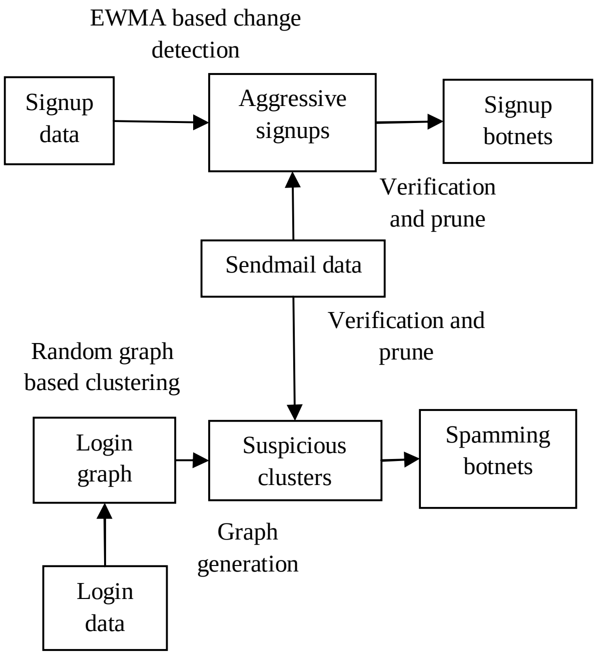

Figure 1. Architecture of Botgraph

They may distribute comparable login or email sending patterns because bot-masters often control all their bot-users using unified toolkits. The user-user graph is used to control the resemblance of bot-user behavior. In the graph each and every vertex is a user. The weight for an edge between two vertices is decided by the features we use to evaluate the similarity between the two vertices (users). The relevant features are selected for similarity measurement, a bot-user group will disclose itself as a connected component in the graph. In this method, the number of common IP addresses logged in by two users as the correspondence feature. Because, the aggressive account-signup detection limits the number of bot accounts a spammer may obtain. To accomplish a large spam-email throughout every bot-account will login and send emails multiple times at various locations, resulting in the sharing of IP addresses.

BotGrep Detection method

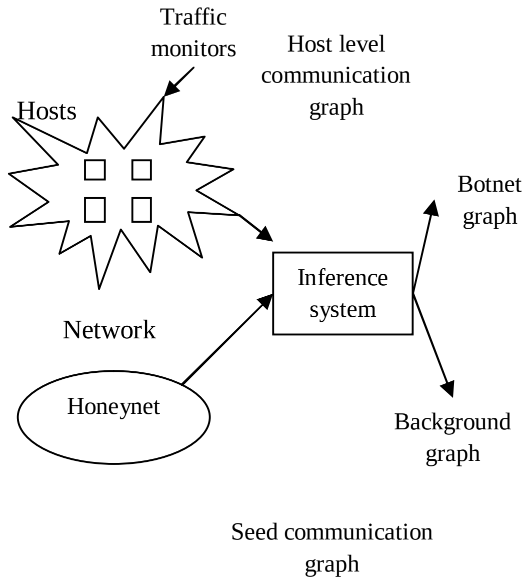

Figure 2. Architecture of BotGrep

The first step requires to gathering a communication graph, in which the nodes represent internet hosts and edges represent communication between them [3]. The internet service provider collects the portions of this graph. Botgrep works on a graph is attained by combining observations across these points into a single graph, which provides significant, though unfinished visibility into the overall communication of Internet hosts. A second source of input is misuse detection. While botnets use communication structures similar to other P2P networks, the communication graph alone may not be adequate to differentiate the two. A list of mischievous hosts can perform as an initial “seed” to speed up botnet classification, or it can be used later to confirm that the detected network is definitely malicious. The next step is to separate communication subgraph. Botnet creators have been turning to communication graphs provided by structured networks, both because of their advantages in terms of effectiveness and flexibility, and due to easy accessibility of well-tested implementations of the structured P2P algorithms.

The general feature of these structured graphs is their quick mixing time, i.e., the convergence time of random walks to a stationary distribution. This algorithm exploits this property by performing random walks to recognize fast-mixing component(s) and separate them from the rest of the communication graph. If there is a problem in sharing of sensitive information, it is possible to complete random walks in a privacy-preserving fashion on a graph that is split among a collection of ISPs.

Once the botnet C&C structure is recognized and established as malicious, BotGrep outputs a set of suspect hosts. This list may be utilized to install blacklists into routers, to organize intrusion detection systems, firewalls, and traffic shapers; or as “hints” to human operators concerning which hosts should be examined. The list may also be disseminated to subscribers of the service, potentially providing a revenue stream.

BotMiner Detection method

The main intent of BotMiner is to detect groups of compromised machines within a monitored network which are part of a botnet [2]. Then passively analze the network traffic in the monitored network. The architecture of the BotMiner detection system consists of five main components: C-plane monitor, A-plane monitor, C-palne clustering module, A-plane clustering module and cross plane correlator.

The traffic monitors in the C-plane and A-plane can be positioned at the edge of the network examining traffic between internal and external networks, similar to BotHunter and BotSniffer. They execute in parallel and monitor the network traffic. The main work of C-plane monitor is to log network flows in a format appropriate for proficient storage and additional analysis, and the A-plane monitor is responsible for detecting suspicious activities.

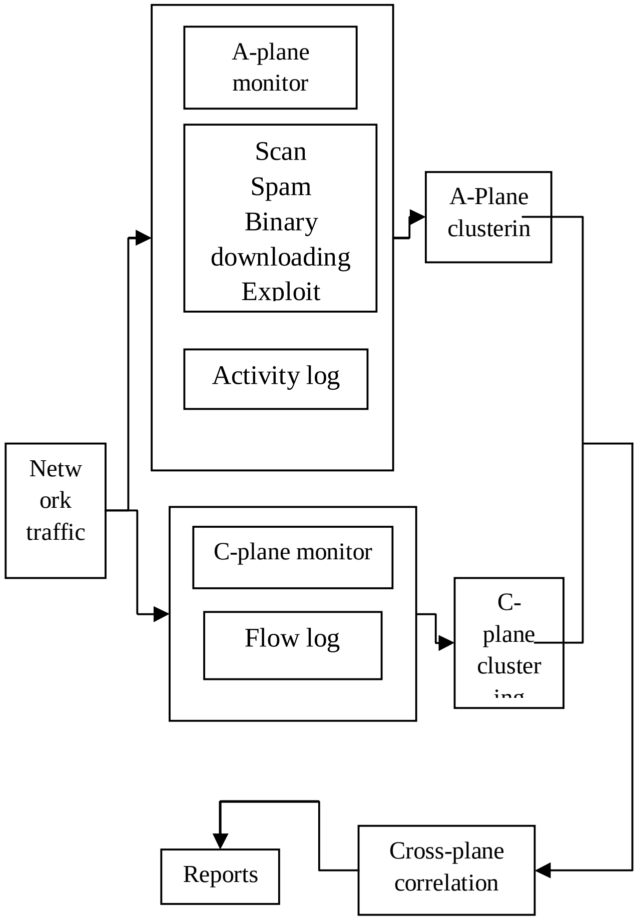

Figure 3. Architecture of Bot Miner

The main work of C-plane and A-plane clustering works is to develop the logs which is formed by the C-plane and A-plane monitors. These two modules take out a number of features from the raw logs and pertain clustering algorithms in order to discover groups of machines that show very comparable communication patterns. At last, the crossplane correlator merge the results of the C-plane and A-plane clustering and makes a last decision on which machines are probably members of a botnet. In an ideal circumstances, the traffic monitors should be distributed on the Internet, and the monitor logs are reported to a central repository for clustering and cross-plane analysis.

Traffic Monitors

C-Plane Monitor: The main work of C-Plane montor is to capture the network flow information and record the information on who is talking to whom. Each and every network flow record contains the information like time, duration, source IP, source port, destination Ip, destination port, and the number of packets and bytes transferred in both directions.

A-Plane Monitor: The A-plane monitor logs information on who is doing what. It examines the outbound traffic via the monitored network and it is capable to distinguish numerous malicious activities that the internal hosts may execute. It is significant to note that A-plane monitoring alone is not adequate for botnet detection purpose. Firstly, A-plane activities are not completely used in botnets. Secondly, because of the moderately loose design of A-plane monitor relying on only the logs from these activities will create a lot of false positives.

C-plane Clustering: C-plane clustering is accountable for reading the logs created by the C-plane monitor and identifying clusters of machines that share comparable communication patterns.

A-plane Clustering: In this module, perform the two-layer clustering on activity Logs. For the whole list of clients that absolute at least one malicious activity during one day, we first cluster them based on the types of their activities. This is the first layer clustering. For every activity type, first cluster clients based on the specific activity features. For scan activity, features could include scaning ports that is two clients could be clustered together if they are scanning the same ports.

Cross-plane Correlation

The results attained from the A-plane and C-plane clustering is utilized to perform cross-plane correlation. The main idea is to crosscheck clusters in the two planes to find out intersections that reinforce evidence of a host being part of a botnet.

5.CONCLUSION

Peer-to-peer (P2P) networks have many dissimilar features that are different from conventional client-server networks. The most important point in the peer-to-peer network is that every peer acts as both server and client roles in the peer-to-peer network. There is no central server which is used to store the files. Due to decentralized nature peer-to-peer network is vulnerable to different types of attacks. So, there is variety of cyber-crimes such as spamming, distributed denial-of-service (DDoS) attacks, finding theft, click fraud etc.Botnet is a gathering of compromised hosts that are distantly prohibited by an attacker through a command and control (C&C) channel. So, to detect the botnets in peer-to-peer network different detection approaches are analyzed. The signature based detection method is used for discovering the P2P application traffic through application level signatures. A few approaches have been proposed like Botgraph, BotGrep, BotMiner are capable of detecting P2P botnets. However, these approaches cannot address all the aforementioned challenges. At the end of this survey we conclude that effectual mechanism is proposed to detect the botnets to improve the detection accuracy and improve the scalablity in the peer-to-peer networks.

REFERENCES

[1] Lin Wang, "Attacks Against Peer-to-peer Networks and Countermeasures," in Proc. TKK T-110.5290 Seminar on Network Security, 2012.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] A. W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," in *Proc. ACM SIGMETRICS*, 2005, pp. 50–60.

[9] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC: Multilevel traffic classification

in the dark," in *Proc. ACM SIGCOMM*, 2005, pp. 229–240.

[10] Z. Li, A. Goyal, Y. Chen, and A. Kuzmanovic, "Measurement and diagnosis of address misconfigured P2P traffic," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1–9.

[11] Yuanyuan Zeng, Xin Hu, Kang G. Shin," Detection of Botnets Using Combined Host- and Network-Level Information", In: Proceedings of the IEEE Symposium on Security and Privacy. (2001).