A REVIEW OF RECENT PEER-TO-PEER BOTNET DETECTION TECHNIQUES

Priyanka
Department of Computer Engineering
National Institute of Technology, Kurukshetra
Kurukshetra, India
priyanka7165dhayal@gmail.com

Mayank Dave, Senior Member, IEEE
Department of Computer Engineering
National Institute of Technology, Kurukshetra
Kurukshetra, India
m.dave@ieee.org

Abstract— **Peer-to-Peer (P2P) botnets have emerged as a serious threat against the network security. They are used to carry out various illicit activities like click fraud, DDOS attacks and for information exfiltration. These botnets use distributed concept for command dissemination. These botnets are resilient to dynamic churn and to take-down attempts. Earlier P2P botnet detection techniques have some shortcomings such as they have less accuracy, unable to detect stealthy botnets and advanced botnets using fast-flux networks. In this paper, we list recent P2P botnet detection techniques that overcome the weaknesses of previous techniques with higher detection accuracy. We also discuss various such techniques, their advantages, accuracy and the weaknesses they too are having. However, two or more techniques can be used together to have more accurate and robust P2P botnet detection.**

Keywords-botnets; peer-to-peer; DDoS

I. INTRODUCTION

Botnet is a network of bots coordinating over network to launch various illicit activities. The bot is basically the short for 'robot'. It is an automated malware program installed on a machine to perform some specific task, generally any sort of criminal activity over the network. A large scale of bots forms the botnet. These infected computers are jointly remote controlled by an attacker (called botmaster) via command and control (C&C) channels.

The computers are made part of botnet army by exploiting their vulnerabilities and infecting them using virus, worms, Trojan horses and spywares.

There are basically two types of botnets: (i) centralized and, (ii) de-centralized. In centralized botnets, bots get commands from the C&C server. These botnets suffer from single-point of failure problem, because the whole botnetwork can be taken down completely if C&C server is tracked and shut-down.

Peer-to-Peer (P2P) technology thus has been adopted by attackers for botnets to deal with weaknesses of centralized botnets. Unlike traditional botnets, there is no central server i.e., bots in P2P botnets act as both client and C&C server. If a bot or a part of P2P botnet gets identified and is taken down, even then, communication in rest of the botnet won't be disrupted [1]. Thus, these are hard to detect and take-down.

P2P botnets have made various companies' and banks' websites their victim, using DDoS attacks. WordPress, PayPal [2], MasterCard [3], Yahoo [4] are few of their victims. ZeroAccess is the largest P2P botnet in existence today with size of 1.9 million zombies, leading to USD 900,000 of daily revenue loss to legitimate online advertisers [5]. Table 1 provides a list various P2P botnets and their estimated size.

TABLE I. P2P BOTNETS

Year	Name	Estimated Size
2003	Sinit	-
2004	Phatbot	-
2006	Nugache	160,000
2007	Peacomm	5 million
2008	Conficker	10.5 million
2009	Waledec	80,000
2011	TDL-4	4 million
	Zeus GameOver	1.2 million
	ZeroAccess	1.9 million/day

II. P2P BOTNET LIFE CYCLE

Botnet lifecycle consists of four phases: initial infection, secondary injection, connection phase and maintenance. During initial infection vulnerable hosts are infected with Trojan, worms, viruses via various methods for e.g. email spams. The infected hosts are then directed to download and install bot binary from specific servers/hosts. After installing the binary file, the hosts are made to set-up C&C or communication channels with other bots using the list of bots provided to them with the binary, to form botnet. On successful establishment of channels, infected computers become part of the botnet army. Attacker can then send them command by injecting

978-1- 4788-7225 -8/15/$31.00©2015 IEEE

into one or more bots which is then disseminated to all live bots of the botnet either via pull or push method. Then, the recruited bots as a large bot army can launch various coordinated attacks against a target, as directed by the botmaster, shown in Fig. 1. The last phase, maintenance phase is about to maintain the bots alive and updated. It also deals with removal of dead and suspicious bots from the botnet.

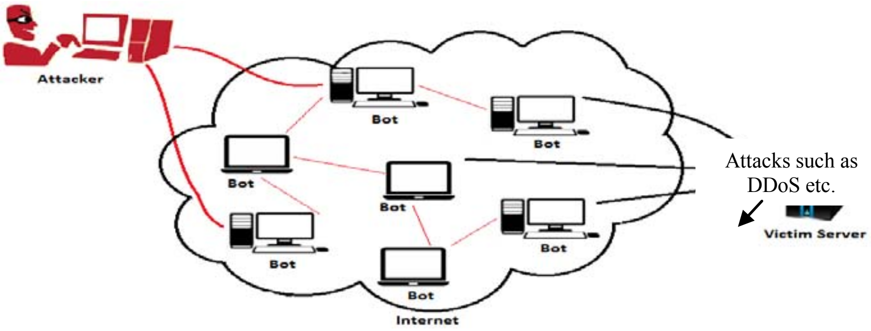

Figure 1: P2P Botnet Operation

Nowadays, P2P botnets are advancing with adoption of new technologies to avoid detection and defense countermeasures. To avoid detection via host-based analysis, rootkits are used. Rootkit is a stealthy software program installed on the infected machine, which hides botnet related processes or programs executing on the bot from various host-based anomaly detectors. Recently, attackers have started utilizing fast-flux service networks (FFSNs) to avoid getting tracked.

III. P2P Botnet Detection and Detection Techniques

As P2P botnets have no central point of failure and are resilient to dynamic churn. They also have high resemblance with legitimate P2P traffic, making them stealthier and thus more difficult to detect. Despite of all these properties it is possible to detect and take-down these botnets. Detection of these botnets is however a difficult task, but not impossible. Researchers are working toward detection of these botnets and in past, have proposed many detection approaches.

Here, we are discussing all the latest detection techniques along with their limitations and vantages. Some techniques discussed are able to detect botnets exchanging benign data, botnets using fast-flux networks and a few are even able to detect botnets in quasi-real time.

A. Big Data Analytics

The authors [6] provides a solution to deal with big-data captured from bots in form of network traffic which is later analyzed for detection, by developing a detection system using various open-source tools as Hadoop, Hive and Mahout. The developed framework is able to detect bots in quasi-real time. Framework consists of three components:

- Traffic Sniffer Module: sniffs packets from the network using dumpcap and saves them to successive pcap files. The files are then submitted to Hadoop Distributed File System (HDFS).
- Feature Extraction Module: Apache Hive extracts the features from the files using group-by clause of HQL, based on map-reduce algorithm. The main vantage of framework is that the features required for detection can be chosen at the run-time, using Tshark. The features are then ranked using Ranker Algorithm by calculating their corresponding information gain w.r.t. the class.
- Machine Learning Module: random forest algorithm is used for detection as it provides high accuracy, less computational complexity and can handle diverse bots and descriptors.

Captured files of various P2P botnets including Conficker, Kelihos-Hlux, Zeus, Storm and Waledac, along with some benign traffic files were used for training the classifier. The developed framework was able to handle big-data in quasi-real time and detected bots with 99.7% accuracy.

B. Dye Pumping

The authors of [7] proposed a new method based on mutual contacts between the bots for botnet detection. A seed bot is required to be known prior to detection and this seed bot is given some dye threshold value. Traffic dispersion graph is used to determine the mutual contacts among the bots. The nodes having mutual contacts with the seed bot or to other suspected bots, are also marked as suspected and are given some fraction of that node's dye on the basis of number of mutual contacts they are having. The nodes having dye value above certain threshold are finalized as bots.

The technique won't work for botnet which has structured P2P topology or if botherder made bots to avoid having mutual contacts. However, the task of making bots to avoid mutual contacts is very complex and will increase unnecessary overhead over the attacker.

C. Emule-like Networks' Periodic Search Packet Based

This detection framework [8] is able to detect eMule-like parasite botnets. As it is well-known that bots execute programmed tasks and send out search requests periodically. The approach works by searching for these periodic sequences in the hybrid sequence traffic generated by the network. Two algorithms: Passive Match Algorithm and Active Search Algorithm were designed for identifying periodic search sequences. The detection framework consists of five components as shown in Fig.2. They also developed a P2P botnet simulator for experimentation, which sends out packets in eMule-format.

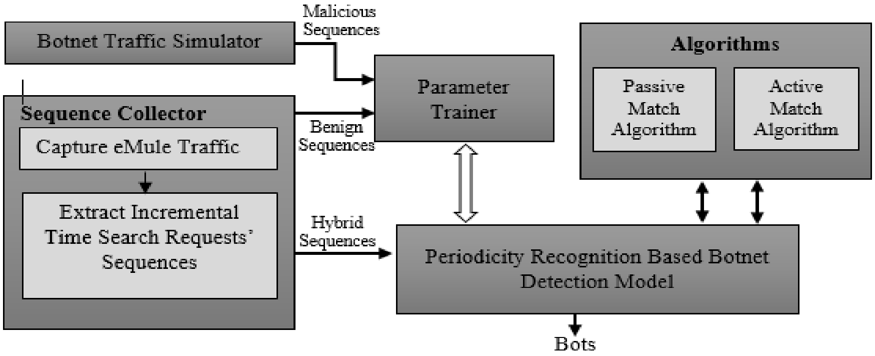

Figure 2: eMule-like parasite network botnet detection framework

Real benign traffic was collected from 100 eMule nodes from a campus network and malicious is generated traffic fusing P2P botnet simulator. The framework was able to detect bots with 98% accuracy. The framework detects bots proactively.

D. *Fuzzy Inference*

The detection approach [9] is based on regularity of bot generated traffic and randomness of user-induced traffic. It is able to detect all types of botnet i.e., IRC, P2P, HTTP Botnets. It classifies destinations contacted by a host as malicious or not, on the basis of entropy of three flow level features extracted from traffic: time_gap, numbers of packets and numbers of bytes in flows to the destination. Data is first captured and is then pre-processed via three steps: filtering, flow aggregation and feature extraction. In feature extraction, for each destination in a flow set, packet count set, byte count set and time gap set are extracted and their entropies are computed separately, using formula:

$$Entropy(X) = -\sum_{i=1}^{n} x_i log x_i$$

Here, x_i represents corresponding parameter of features whose entropy is calculated. Fuzzy rules are then formed for the classification of hosts as benign or malicious, on the basis of entropies calculated from above mentioned three features. And on the basis of fuzzy rules traffic is classified as malicious or not.

E. *Game Theory*

Mobile devices and smartphones are considered as untrusted hosts in the network. Traditional correlation model [10] works well for internal networks only i.e., the networks consisting of trusted hosts only. Approach [11] is a trust-based model, able to detect botnet by correlating host-level and network-level behavior of trusted hosts only, identified from a dynamic network. Trusted hosts are identified by monitoring every hosts by installing host analyzer at each host. Game theory is used to cluster the trusted hosts, reputation level is given to each host on the basis of level of trustworthiness, according to its contribution level.

The behavioral value produced by different group of hosts is then multiplied by the reputation of the corresponding host which is a value from [0, 1]. Thus we can take the value based on the reputation level of a host. Once we calculate the characteristic function of different groups, we combine the data from the trusted contributors. Highest contributors' data is then correlated for setting the detection score.

F. *Irregular Phased Similarity*

Detection approach [12] proposed is based on the irregular intervals during which phased similar pattern traffic is generated by P2P botnets. Detection framework consists of three phases:

Data pre-processing: network flows are extracted for each host and irrelevant flows are removed to increase efficiency. The pre-processing is carried out in three steps: traffic decomposing, traffic filtering and flow extraction (using Argus tool).

Coarse-grained filter: time window having outgoing degree and failed connection rate greater than threshold value is searched in host's time flow. If not found, the corresponding host is then removed from the suspected list.

P2P bot detection: separate flows from the host are considered and similar cluster pairs are searched on the basis of distance between various flow cluster (FC) features. Final distance between two flow clusters is calculated using:

$$dist(FC_i, FC_j) = \alpha dist_{IP}(FC_i, FC_j) + \beta dist_{ND}{}'(FC_i, FC_j) + \lambda dist_{fr}(FC_i, FC_j) + \varphi dist_{IP}(FC_i, FC_j)$$

where, α, β, λ, φ are constants, $dist_{IP}(FC_i, FC_j)$ is distance between two *FCs'* IP address set, $dist_{ND}'(FC_i, FC_j)$ is distance between *FCs'* outgoing degrees, $dist_{fr}(FC_i, FC_j)$

is distance of *FCs'* failed rate and $dist_B'(FC_i, FC_j)$ is normalized average overhead distance between *FCs*.

Host *h* is considered as bot, if final distance is less than some threshold value and *k'* clusters out of *k* clusters of *FC*(*h*) are similar. For evaluation, non-malicious traffic from Traffic Lab of Ericsson Research in Hungary was taken and was mixed with more than 28,000 flows of Storm and Waledac botnets.

Table 2. Summarized Table of discussed detection approaches

Author	Approach	Advantages	Limitations
Coskun et al. (2010)	Bots are detected on the basis of mutual contacts they are having	Can detect stealthy bots. No traffic analysis required	Seed bot required to be known prior to detection Computational Complexity
Yong et al. (2011)	Parasite botnet detected by searching for periodic search requests	Detects bots prior to attack	Can detect malicious nodes in monitored network only
Barthakur et al. (2012)	Botnet detection using SVM	Can detect bots exchanging encrypted traffic	Not able to detect bots exchanging benign data
Huabo et al. (2012)	Proposed detection approach based on botnets irregular traffic patterns	Can detect malicious node in a network containing one bot only	Overhead in pre-processing large amount of traffic captured Approach does not work for stealthy bots
Sheng et al. (2012)	General botnet detection approach based on traffic flows analysis	Can detect both centralized and de-centralized botnets	Can detect bots in the monitored network only Low P2P botnet detection accuracy
Noura et al. (2013)	Detection done using only trusted hosts of a network only, identified using game theory	Bettered detection accuracy compared to traditional model	Each host is required to be monitored individually
Soniya and Wilscy (2014)	Proposed a general botnet detection approach	Limited volume of traffic analyzed	Developed for single host only, can be improved to work at network edge
Zhang et al. (2014)	Botnet detected deriving statistical fingerprints	Able to detect stealthy botnets	Can detect bots in the monitored network only
Kamaldeep et al. (2014)	Framework developed for P2P botnet detection in quasi-real time	Quasi-real time processing and detection	Require continuous monitoring of network

G. *P2P Botnet Detection Using SVM*

The research [13] proposed a proactive detection technique using Support Vector Machine (SVM). SVMs are used as they scale better than others and also provide better detection accuracy. Approach is based on two bot characteristics: maintain minimum packet sizes and bytes transferred and, invariable packet sizes and inter packet arrival time. Detection framework consists of two components:

Flow extraction: from the captured data, flows are extracted and useful flows are labelled. Output is divided for training and testing.

Classification: refined flows are transformed as real vectors and attributes are scaled to range [-1, +1]. SVM model is then selected using Radial Basis Function (RBF) kernel, for classification of flows.

SVM model used detected bots with 99.01% detection accuracy.

H. *Stealthy Botnet Detection*

Reference [14] is able to detect stealthy P2P botnets and proposed technique provides the best result among all the detection techniques present till date with 100% accuracy and negligible false positive rate. It consists of two phases (Fig. 3):

- P2P Clients Identification: the hosts involved in any kind of P2P communication are first identified by DNS traffic analysis and suspected P2P clients are then passed through fine-grained detection for confirmation. They are finalized as P2P clients if they have same flow size and exchange control messages with large number of peers of various networks.
- P2P Bots Detection: finalized peer clients are labelled as suspected hosts on the basis of their activation time and number mutual contacts.

P2P applications like BitTorrent, LimeWire, eMule, Ares and Skype were run on several machines for benign traffic. Network traces of Storm and Waledac bots were also executed in controlled environment. Argus tool was used for collection of information from network. The biggest vantage of approach is that it is able to detect bots sharing legitimate data with high accuracy.

I. Traffic Flow Analysis

Reference [15] introduced a general distributed botnet detection framework. Detection framework (Fig. 4) consists of two components:

- Data Collection and Filter: multiple agents are deployed in monitored network to capture data in a distributed manner. Every agent is responsible for data capturing, filtering (using dynamic whitelisting) and classification. Filtered data is then classified as: TCP-based, UDP-based and other.
- Botnet Detection and Identify: firstly, data fusion of distributed data is done in flow sets. Flow sets are generated corresponding to each SrcIP and to (SrcIP, DestIP, DestPort) pair, and are sent to propagation pattern identify and control pattern identify, respectively. Both modules label host IPs as suspicious or not. Correlation of both modules is then used to finalize a host as bot.

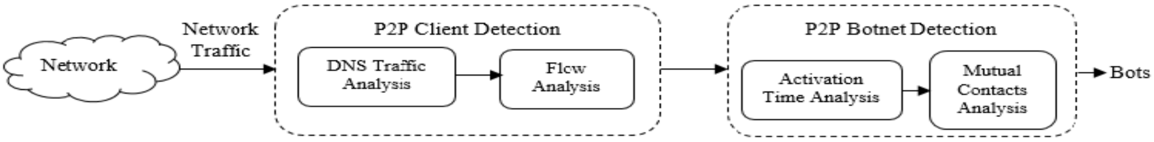

Figure 3: Stealthy Botnet Detection System

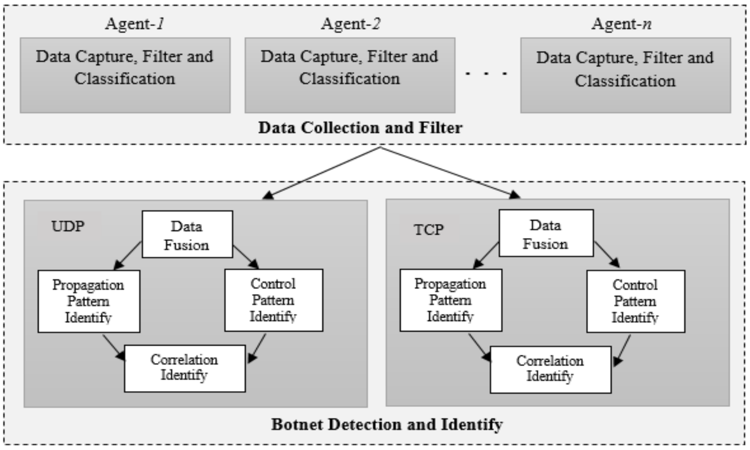
Agent-*1*
Agent-*2*
Agent-*n*
Data Capture, Filter and Classification
Data Capture, Filter and Classification
. . .
Data Capture, Filter and Classification
Data Collection and Filter
UDP
Data Fusion
Propagation Pattern Identify
Control Pattern Identify
Correlation Identify
TCP
Data Fusion
Propagation Pattern Identify
Control Pattern Identify
Correlation Identify
Botnet Detection and Identify

Figure 4: Distributed Traffic-flow based Detection Framework

100 hosts from campus LAN were considered and IRC and P2P botnets were installed on some of them. The developed model detected both type of bots with significant accuracy.

IV. CONCLUSION

Previous detection approaches have some weaknesses such as less accuracy, high FPRs, unable to detect stealthy botnets, advanced botnets using FFSNs and botnets using rootkits. All these problems are solved by various recent approaches described here. Table 2 provides a summarization of all the techniques discussed along with their advantages and limitations. These techniques are able to detect P2P botnets with very high detection accuracy and very less FPRs. But none of the techniques discussed here, is solely able to address all the problems of P2P botnet detection. However, we can develop techniques by combining two or more techniques, which will exclusively address all the detection problems of P2P botnets.

REFERENCES

[1] Ping Wang, Baber Aslam, Cliff C. Zou, "Peer to Peer Botnets", Handbook of Information and Communication Security(Springer), 2010, pp. 335-350.

[2] http://thelede.blogs.nytimes.com/2010/12/08/operation-payback-targets-mastercard-and-paypal-sites-to-avenge-wikileaks.

[3] Operation Payback cripples MasterCard site in revenge for WikiLeaks ban, Dec. 8, 2010, [online] http://www.guardian.co.uk/media/2010/dec/08/operation–payback–mastercard–website–wikileaks.

[4] Yahoo on Trail of Site Hackers, Wired.com, Feb. 8, 2000, [online] http://www.wired.com/news/business/0,1367,34221,00.html.

[5] http://www.symantec.com/connect/blogs/grapplingzeroaccessbotnet.

[6] Kamaldeep Singh, Sharath Chandra Guntuku, Abhishek Thakur, Chittaranjan Hota, "Big Data Analytics framework for Peer-to-Peer Botnet detection using Random Forests", Information Science Journal, Elsevier, 2014, pp. 488-497.

[7] Coskun, B . , Dietrich, S., And Memon, N., "Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts, In Proc. of ACSAC-2010, ACM, 2010, pp. 131-140.

[8] Yong Qiao, Yuexiang Yang, Jie He, Bo Liu, and Yingzhi Zeng, "Detecting Parasite P2P Botnet in *eMule*-like Networks through Quasi-periodicity Recognition", Information Security and Cryptology - ICISC 2011 Springer, 2011, Volume 7259,pp. 127–139.

[9] Soniya B, M Wilscy, "Fuzzy Inference System based on entropy of traffic for bot detection on an endpoint host", IEEE International Conference on Data Science & Engineering (ICDSE), 2014, pp. 112-117.

[10] Y. Zeng, X. Hu, and K. Shin., "Detection of botnets using combined host and network level information", Dependable Systems and Networks (DSN), IEEE, 2010, pages 291–300.

[11] Noura Al Ebri, Hadi Otrok, Azzam Mourad, Yousof Al-Hammadi, "Botnet Detection: A Cooperative Game Theoretical Correlation-Based Model", 3rd IEEE International Conference on

Communication and Information Technology (ICCIT), 2013, pp. 28-32.

[12] Huabo Li, Guyu Hu, Jian Yuan, Haiguang Lai, “P2P botnet Detection based on Irregular Phased Similarity”, IEEE Second International Conference on Instrumentation & Measurement, Computer, Communication and Control, 2012, pp. 79-82.

[13] Pijush Barthakur, Manoj Dahal, Mrinal Kanti Ghose, “A Framework for P2P Botnet Detection Using SVM”, Proc. IEEE International Conference on Cyber-Enabled Distributed Computing and Knowledge Discover, 2012, pp. 195-200.

[14] Junjie Zhang, Roberto Perdisci, Wenke Lee, Xiapu Luo, and Unum Sarfraz, “Building a Scalable System for Stealthy P2P-Botnet Detection”, IEEE Transactions on Information Forensics and Security, January 2014, Vol. 9, No. 1, pp. 27-38.

[15] Li Sheng, Liu Zhiming, He Jin, Deng Gaoming, Huang Wen, “A Distributed Botnet Detecting Approach Based on Traffic Flow Analysis”, IEEE Second International Conference on Instrumentation & Measurement, Computer, Communication and Control, 2012, pp. 124-128.