A Hybrid Technique To Detect Botnets, Based on P2P Traffic Similarity

Riaz Ullah Khan
Center of Cyber Security,
School of Computer Science and Engineering,
University of Electronic Science and Technology of China
Email: rerukhan@gmail.com

Rajesh Kumar
Center of Cyber Security,
School of Computer Science and Engineering,
University of Electronic Science and Technology of China
Email: rajakumarlohano@gmail.com

Mamoun Alazab
College of Engineering, IT and Environment
Charles Darwin University
Australia
mamoun.alazab@cdu.edu.au

Xiaosong Zhang
Center of Cyber Security,
School of Computer Science and Engineering,
University of Electronic Science and Technology of China
Email: jhonsonzxs@uestc.edu.cn

***Abstract*—The botnet has been one of the most common threats to the network security since it exploits multiple malicious codes like worm, Trojans, Rootkit, etc. These botnets are used to perform the attacks, send phishing links, and/or provide malicious services. It is difficult to detect Peer-to-peer (P2P) botnets as compare to IRC (Internet Relay Chat), HTTP (HyperText Transfer Protocol) and other types of botnets because of having typical features of the centralization and distribution. To solve these problems, we propose an effective two-stage traffic classification method to detect P2P botnet traffic based on both non-P2P traffic filtering mechanism and machine learning techniques on conversation features. At the first stage, we filter non-P2P packages to reduce the amount of network traffic through well-known ports, DNS query, and flow counting. At the second stage, we extract conversation features based on data flow features and flow similarity. We detected P2P botnets successfully, by using Machine Learning Classifiers. Experimental evaluations show that our two-stage detection method has a higher accuracy than traditional P2P botnet detection methods.**

Keywords: Botnet detection, Feature Extraction, Anomaly Detection, P2P traffic identification

I. Introduction

Nowadays, the network environment is highly complex and the security problem is becoming more and more prominent. As the botnet C & C server has a higher degree of concealment, unknown programs are often used by large-scale network intruders. Almost all of the DDoS attacks and 80% to 90% of the spam attacks are initiated by the botnets. Therefore, the botnet has become a big threat to network security and can not be ignored. Early botnets normally used IRC and HTTP as a communication protocol, with a single failure point, and it has been easy to be detected and destroyed. Today, most of the botnets use P2P technology to create C & C (command and control) mechanisms to enhance network traffic concealment. Compared to botnets with IRC and HTTP protocols, P2P botnets without central nodes have greater threat and concealment. Therefore, P2P botnet is increasingly favored by attackers. P2P botnet detection has also become a hot research area in the field of cyber security.

At present, P2P applications have caused the explosive growth of Internet traffic, which is a huge challenge in terms of data storage and real-time analysis. Therefore, the network of non-P2P traffic filtering is particularly important. This paper aims to examine the features and strategies to detect the botnets. Main contributions of this paper are as follows:

- This article presents a novel approach to classify network traffic and detect botnets through machine learning algorithms.
- We have done classification using two stage technique. This technique covers the limitations of single stage botnet detection e.g. class imbalance.
- We have compared three machine learning algorithms to achieve botnet detection effectively.

A. Structure of paper

The Section I of this paper, discusses the background, problem description, and approach used in this study. Section II gives a brief overview to the literature review. Section III gives a brief overview to the two stage botnet detection scheme and mechanism of two-stage scheme. Section IV discusses the results of the experiments. Finally, Section V concludes the paper.

B. Problem Description

Research works on botnets among our surveyed literature focuses mainly on designing systems to detect command and control (C&C) botnets, where many bot-infected machines are controlled and coordinated by few entities to carry out malicious activities [1]. Those systems need to learn decision boundaries between human and bot activities, therefore ML-based classifiers are at the core of those systems, and

978-1-7281-2600-5/19/$31.00 ©2019 IEEE
DOI 10.1109/CCC.2019.00008

are often trained by labeled data in supervised learning environments. The most popular classifier is support vector machines (SVMs) with different kernels, while spatial-temporal time series analysis and probabilistic inferences are also notable techniques employed in ML-based classifiers. Clustering is mostly used in natural language processing (NLP), to build a large-scale system to identify bot queries [2]. In botnet detection literature, two core assumptions are widely shared:

1) Botnet protocols are mostly C&C [3].
2) Botnet behaviors are different and distinguishable from legitimate human user, e.g., human behaviors are more complex [4].

Other stronger assumptions include that the bots and humans interact with different server groups, and features are independent which are generated by bots and humans, from different messages. Classification techniques, e.g., Weighted Least Square, Binary Classifier and hypothesis testing, are usual system components [5]. Attempts have been made to abstract state machine models of network to simulate real-world network traffic and create honeypots. Ground reality is often heuristic, labeled by human experts, or a combination are used, for example, the game masters visual inspections serve as ground truth to detect bots in online games [6]. In retrospect, the evolution of botnet detection is clear from earlier and more straightforward uses of classification techniques such as clustering and NB, the research focus has been expanded from the last step of classification, to the important preceding step of constructing suitable metrics, that measures and distinguishes bot-based and human-based activities [2], [4].

C. Our Approach

This paper proposes a two-stage detection method for P2P botnets, i.e., the first stage is based on port judgment, DNS query and data flow count in the session to filter non-P2P traffic; and the second stage is based on session characteristics to identify P2P botnet. The method is used on the bases of session feature to effectively reduce the data packets to be analyzed. Furthermore, Machine Learning algorithms are used to classify and identify the traffic. At the same time, we compare our experiments by using three machine learning algorithms on the datasets collected from diverse sources. The experimental results show that the Decision Tree algorithm is the most accurate for P2P botnet detection.

II. Related work

Machine Learning algorithms have been widely used to classify the internet traffic. Irrespective of the class imbalance problem, ML algorithm classifiers such as Decision Trees and Neural Networks, may produce a high accuracy but low byte accuracy. Zhang et al. [7] proposed two algorithms based on feature selection and extended *wsu_auc* selection to apply the best features practically. They achieved more than 94% accuracy with an average byte accuracy of over 80%.

In 2017, Chen et al. [5] proposed a detection method for botnets in high speed network environment. In this PF_RING was used to solve the problem to high packet drop rate and for the extraction of required fields from the traffic data. Random forest algorithm was used by the author on the CTU dataset. They obtained high accuracy but the unimpressive part of this paper is the use of only offline public dataset and no other online or self-generated data.

Zeng and Shen [2] proposed a two-step distributed approach for storm botnet detection which includes a set of heuristics and first-step port numbers and an SVM classifier. Their accuracy of their method was more than 95% with 8 - 12% of FP rate. This scheme works well with 0% FP rate and 8% false negative rate (FN) to detect storm botnets Host. According to Zhang et al. [8], the P2P client is first identified by extracting the statistical fingerprint of P2P communication, and the legal P2P network and the P2P botnet are further distinguished.

Texture-based detection [9], [10] is the design of detection rules by analyzing botnets or communication traffic to extract features such as MD5, PE head format, etc., but the initial detection rules will fail after the botnet application changes their communication mode and packet format. At the same time, if the currently used signature can not effectively represent the characteristics of the zombie program, the detection strategy will have a higher false alarm rate. Ye et al. [11] used a signature-based classification, combined with heuristics and a statistical-based classifier in the clad layer using a C4.5 algorithm built at the flow level and achieved a high accuracy of 97.46%.

Detection based on host behavior [12], [13] detects zombie programs by monitoring changes in the host process, file, network connection and registry content in a controllable environment. The method can not detect new and variant botnet programs. For example, an attacker could use such new detection and hiding techniques such as rootkits, anti-debugging to avoid such detection strategies.

Pattern-based and statistic-based approaches were proposed to overcome the limitations of port-based and signature-based techniques [14]. Wang et al., [15] proposed a P2P storm botnet detection method based on C & C traffic stability. Their approach is to be able to combine storms with C 98% . The flow was "stable" and the false positive rate was 30%. Jiang [14] discovered C & C communication from P2P robots and found flow dependency in C & C flow, but when these flows are few, this method may find it difficult to find flow dependence.

Azab et al. [16] proposed machine learning classifier (C45) and Correlation-based Feature Selection (CFS) for *Zeus V1.x*,*Zeus V2.x* and *benign HTTP* traffic detection in networks. Azab et al., demonstrated 99.3% accuracy with

55.6% Recall results.

In other work, Alazab [17] examined the evolution of malware by the nature of its activity and variants. The paper investigated malware implication on the computer industry and provided a framework using feature extraction from malicious code which reflected the behaviour of the code. Furthermore, Alazab [18] proposed a four-step methodology to develop a fully automated system for suspicious behavior detection. The four-step method classified the behavior of API function calls based on the malicious intent hidden with the packed program.

III. Proposed Scheme for Botnet Detection

A. Two-stage Detection Method

This section describes the methods proposed in this paper to detect bot-bling traffic in two phases. The focus of this method is on non-P2P traffic filtering and the extraction of the characteristics of the session. The architecture of the model is shown in Fig. 1. The first stage of the model will start from the three aspects of, packet filtering rules, session characteristics and classification algorithm. The second stage, classify the traffic as either the traffic is normal P2P or botnet traffic.

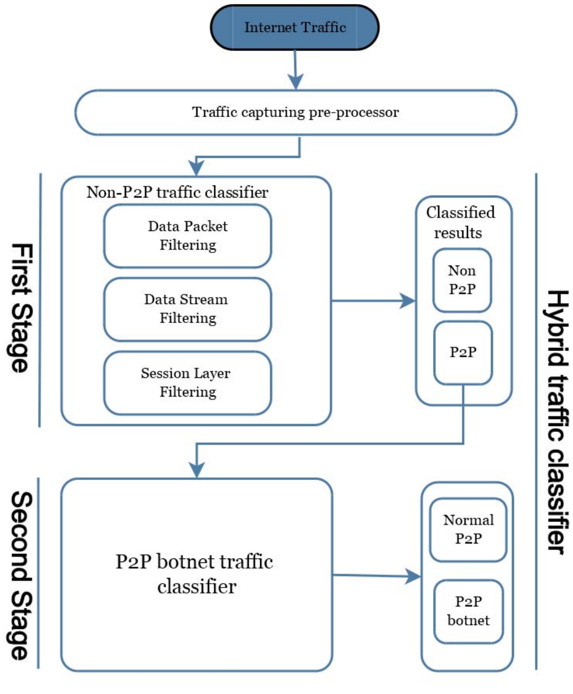

Figure 1. Architecture of the proposed method

1) First Stage of Traffic Classifier:: At present, port identification, signature recognition, and identification are commonly used methods for P2P traffic identification. These methods are based on stream feature [19]. However, the port identification method can not recognize P2P applications with random ports or custom ports. DPI (Deep Packet Inspection Technology), does not recognize encrypted P2P traffic [15]. Stream-based identification methods can only determine P2P applications of the partial flow, and has a high false alarm rate. Therefore, we use non-P2P well-known port filtering mechanism, DNS query, flow counting rules to filter non-P2P traffic, combined with fast heuristic P2P traffic identification method, as shown in Fig. 2.

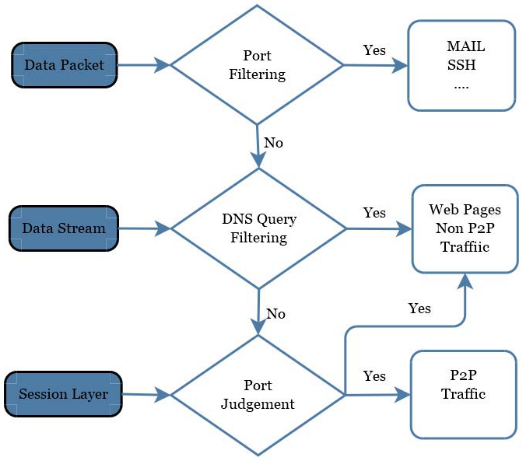

Figure 2. First Stage Traffic Classifier

Port filtering is a packet-level filtering method, mainly filtering the commonly used non-P2P application traffic. DNS query is a stream-level filtering method, flow counting and port judgment is a session-level filtering method, the two rules are mainly filtered web pages and other non-P2P traffic. Among them, the port-based filtering method can identify some common non-P2P application traffic, such as SSH generally use port 22, Telnet (remote login) use port 23. Commonly used applications and their corresponding port numbers are shown in Table I.

Table I
Common applications and their corresponding ports

Application	Port Number
SSH	22
TelNet	23
MAIL	25, 110, 143, 465, 220, 993, 995
NetBios	125, 137, 139, 445
Remote	3389
FTP	20, 21
NTP	123

In general, P2P node communication does not require domain name resolution, but directly read the IPS list stored in the local configuration file to obtain IP. However, for non-P2P applications, DNS domain name resolution must be used to obtain IP. Therefore, one of the criteria for determining non-P2P network data flows such as Web and

Mail etc., is resolved by domain name and may be the destination IP address in the network flow.

When a user sends a Web application service request normally, the Web application uses a multi-port, parallel-requested connection to an IP address on a page. As a result, multiple data streams appear in the same session. The P2P network node communicates each time using a pair of random source and destination ports. Therefore, we can use flow counting and port determination to filter non-P2P traffic. If a session is using the TCP protocol and 80,8080 or 443 port, and the number of sessions in the flow exceeds the threshold, then the session can be considered a web page traffic session. Where the number of valid streams in a session is represented; the threshold is selected based on the number of streams that appear in the normal page access session. Using the capture tool to collect simple and relatively complex web page requests, the analysis results show that the simple web page is generally 3 to 4 connection requests, and the complexity of the page connection request is 5 to 8. Therefore, the threshold is set to 3 in this article.

Although this phase of the method can not accurately detect the identified P2P applications, but it can be in the real network environment to filter out the vast majority of non-P2P traffic and a small amount of secure P2P traffic.

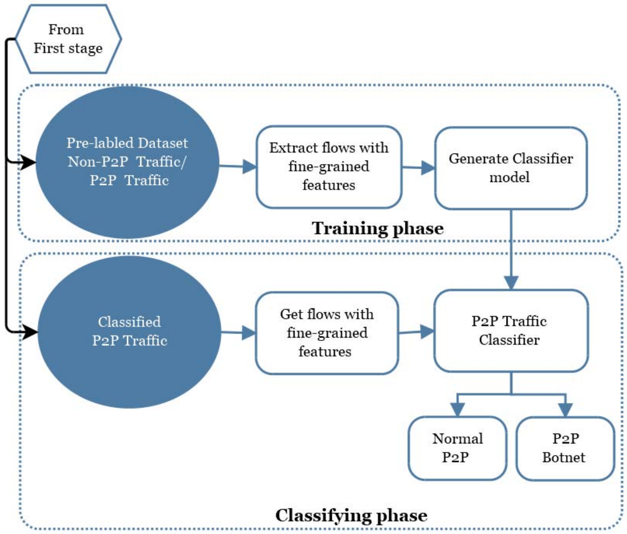

Figure 3. Second Stage Traffic Classifier

2) Second Stage of Traffic Classification:

: **(i) Feature Extraction:**

Through the analysis of the data flow characteristics of P2P botnet, the traffic characteristics between the zombie hosts that join the same botnet are similar. Therefore, this paper uses the session-based strategy for feature extraction, that is, with the same destination address of the data flow in the same session, reducing the number of stream features and the number of data, thereby improving the detection efficiency.

(ii) Session Duration:

P2P zombie host and other zombie host communication process is automatically completed by the zombie program, the flow of the duration is generally short and very fixed. Therefore, you can extract the average, maximum, minimum, and standard deviation of the duration of the session, and the average interval of the upstream (downstream) stream packets in the session as a feature.

(iii) Distribution of The Flow in The Session:

In the process of communication between two nodes in the P2P botnet, the size and transmission quantity of the transmitted packets are relatively small, and the C & C communication flow generated by the zombie host in the same botnet has great similarity. This was observed in our simulations. Therefore, we can distinguish between normal P2P network traffic and P2P botnet traffic by using the distribution of traffic in the session. Fig. 3 describes the role of claffication in the second stage. The average of the maximum packet length of the upstream/downstream in the extraction session, the average of the average packet length, the average of the minimum packet length, the standard deviation of the average packet length, and the average of the number of valid packets, the standard deviation of the number of packets, the average number of bytes transmitted, and the standard deviation of the number of bytes transmitted as a feature. The simulation is shown in Fig. 6, Fig. 7 and Fig. 8.

IV. Experimental Results and Analysis

A. *Evaluating Metrics*

We assessed the execution of our methodologies utilizing 10-fold cross validation. The methodology of k-fold cross validation is shown in Fig. 4. The first example was arbitrarily apportioned into ten equivalent measured sub-tests. Nine sub-tests were utilized for training the model and the remaining one sub-test was held for the testing. The procedure was rehashed ten times, utilizing an alternate sub-sample for testing, every time. The outcomes were then found the average value for single and final result. All tests were utilized once for validation. Furthermore, we used wrapper method for feature selection. The mechanism of wrapper technique is shown in Fig. 5.

1) Accuracy of Detection Model: The percentage of correctly classified instances among the total number of instances.

2) False Alarm Rate: FP False positive rate—the rate of P2P recognized incorrectly.

FN False negative rate—the rate of Normal P2P recognized incorrectly as botnets.

Recall rate is calculated by the following formula with given number of true positives and false negatives.

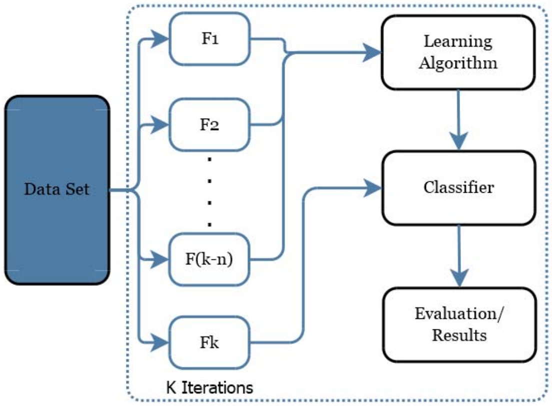

Figure 4. K-Fold Cross Validation

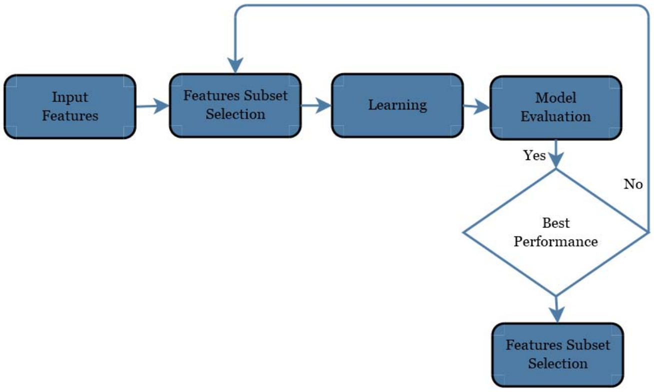

Figure 5. Wraper Method for Features Selection

$$TPR = \frac{TN}{TN + FP}$$

The TPR is referred to "sensitivity" or the "true positive rate" sometimes. Precision is calculated by the following formula which is also known as "positive predictive rate":

$$FPR = \frac{FN}{FN + TP}$$

B. Dataset and Experimental setup

To ensure the reliability and scalability of our proposed model, we trained our model with network flow data from a diverse variety of sources. The dataset includes different types of botnets tested in different kinds of environmental setup. We set a *VMWare* virtual environment in widows 10 and Linux operating systems. *Nfdump* was set up on the system to collect network data. *Nfdump* captures network flow and stores into *nfcapd* files. One instance of the the *nfcapd* file is associated with a flow data record over time. We used *Wireshark* in window environment to capture the network flow because *Wireshark* is an open source software which is available free of cost. The advantage of utilizing *nfcapd* in a linux domain is that it records countless highlights of the network traffic which turn out to be very favorable in further investigation. It likewise runs discreetly out of sight utilizing insignificant handling memory and power, thus an ideal decision as a tool to gather information. Description of the famous datasets which we used in our experiments are discussed below.

CTU-13 Dataset [20]: We used the dataset of CTU-13 project to do experiments because it contains thirteen various captures of different botnet samples i.e., IRC, SPAM, CF (Click Fraud), DDoS, FF (FastFlux), PS (Port Scan), US (Compiled and Controlled by us), HTTP. The capture files are stored in the *pcap* form. The dataset of CTU-13 project is a labeled dataset with background traffic, botnet and normal. We have also downloaded non-malicious packets to combine with CTU-13 dataset. Fig. 6 shows the simulation of the running *Wireshark* environment. Fig. 7 and Fig. 8 are the simulations of *Donbot* botnets and *Sogou* botnets respectively.

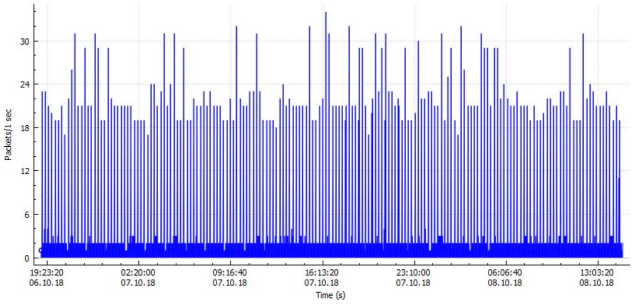

Figure 6. Wireshark Input Output Graph: VMware Network Adapter

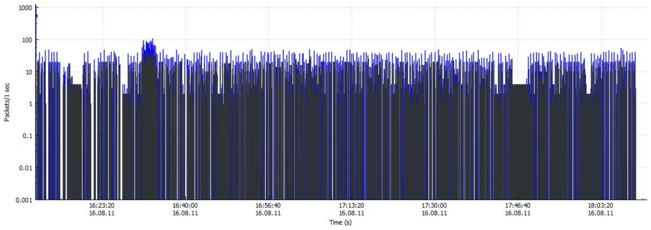

Figure 7. Wireshark Input Output Graph: Donbot Botnet Capture

C. Classifier selection

So far, many supervised machine learning algorithms are used to classify data e.g. Khan et. al. [21], [22] analysed ResNet and GoogleNet models for malware detection using image processing technique. Kumar et. al. [23] used CNN model for malicious code detection based on pattern

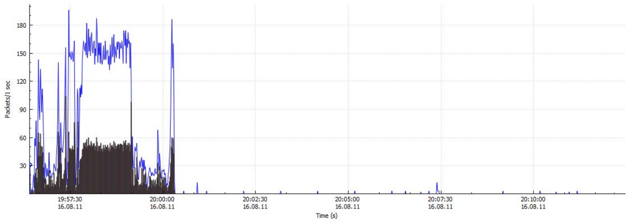

Figure 8. Wireshark Input Output Graph: Sogou Botnet Capture

recognition. In this paper, we have compared following classification algorithms to verify the detection rate of the proposed method;

1) Naive Bayes classification algorithm,
2) Decision Tree classification algorithm,
3) ANN

These algorithms are based on session characteristics to detect P2P botnet traffic, the Decision Tree algorithm shows a high accuracy. The Decision Tree algorithm uses the binary tree as a classification tree. The principle of each classification tree is recursively from top to bottom, and its training set is obtained by returning the original training data set. In order to minimize the occurrence of the fitting phenomenon, the Decision Tree uses the Bagging random sampling method to construct the classification tree. Therefore, this paper uses the Decision Tree classification algorithm for high-speed network environment P2P botnet traffic detection.

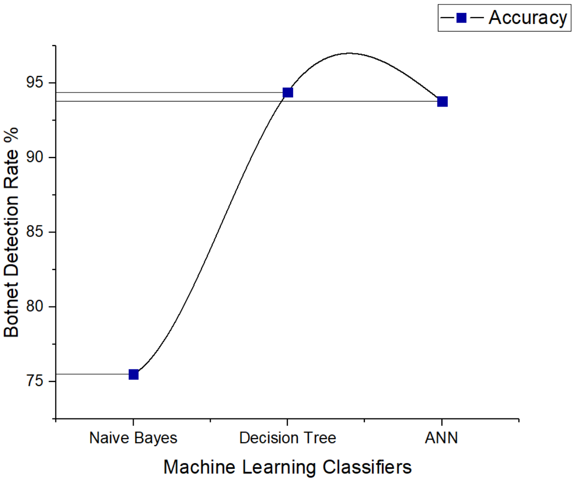

Figure 9. Comparison of three machine learning classifiers on P2P botnet detection

Using Naive Bayes classification algorithm and ANN, the detection rate was 75.5% and 93.8%, respectively, but the results of Decision Tree algorithm was noted as high as 94.4%. Therefore, Decision Tree algorithm for various types of P2P botnet traffic detection is more accurate than the other two classification algorithms. So the Decision Tree detection algorithm based on session feature has greatly improved the detection rate of P2P botnet.

V. Conclusion

In this paper, a hybrid technique for P2P botnet detection is proposed on the basis of session features. Firstly, non-P2P traffic was filtered from packet, stream and session level respectively. Then, P2P botnet classifiers were used to classify the Normal P2P communication and P2P botnet on the basis of session features. This study combines the advantages Detection Method Based on Flow Similarity. The validity of the proposed method is verified by using the open sourced published data set. It is noted from the experimental results, that two-stage technique can effectively detect P2P botnet traffic. We evaluated the model by comparing three different classifiers and noted that the Decision Tree classifier has a higher accuracy.

Aknowledgment: This research was supported by the National Natural Science Foundation of China under grant No. 61572115.

References

[1] Arushi Arora, Sumit Kumar Yadav, and Kavita Sharma, "Denial-of-service (dos) attack and botnet: Network analysis, research tactics, and mitigation", in *Handbook of Research on Network Forensics and Analysis Techniques*, pp. 117–141. IGI Global, 2018.

[2] J Zhang, Y Xie, F Yu, D Soukal, and W Lee, "Intention and Origination: An Inside Look at Large-Scale Bot Queries.", *Ndss*, 2013.

[3] Gregoire Jacob, Ralf Hund, Christopher Kruegel, and Thorsten Holz, "JACKSTRAWS: picking command and control connections from bot traffic", *SEC'11 Proceedings of the 20th USENIX conference on Security*, pp. 22–29, 2011.

[4] Virpi Roto, Antti Oulasvirta, Tuulia Haikarainen, Jaana Kuorelahti, Harri Lehmuskallio, and Tuomo Nyyssönen, "You Are a Game Bot!: Uncovering Game Bots in MMORPGs via Self-similarity in the Wild", *Ndss*, , no. February, pp. 1–19, 2016.

[5] Chia-Mei Chen, Gu-Hsin Lai, and Pong-Yu Young, "Defense Joint Attacks Based on Stochastic Discrete Sequence Anomaly Detection", in *2016 11th Asia Joint Conference on Information Security (AsiaJCIS)*. aug 2016, pp. 74–79, IEEE.

[6] Tammo Krueger, Hugo Gascon, Nicole Krämer, and Konrad Rieck, "Learning stateful models for network honeypots", in *Proceedings of the 5th ACM workshop on Security and artificial intelligence - AISec '12*, 2012, p. 37.

[7] Hongli Zhang, Gang Lu, Mahmoud T Qassrawi, Yu Zhang, and Xiangzhan Yu, "Feature selection for optimizing traffic classification", *Computer Communications*, vol. 35, no. 12, pp. 1457–1471, 2012.

[8] 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*, vol. 9, no. 1, pp. 27–38, jan 2014.

[9] Sheharbano Khattak, Naurin Rasheed Ramay, Kamran Riaz Khan, Affan A Syed, and Syed Ali Khayam, "A Taxonomy of botnet behavior, detection, and defense", *IEEE Communications Surveys and Tutorials*, vol. 16, no. 2, pp. 898–924, 2014.

[10] Junjie Zhang, Roberto Perdisci, Wenke Lee, Unum Sarfraz, and Xiapu Luo, "Detecting stealthy P2P botnets using statistical traffic fingerprints", in *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*. jun 2011, pp. 121–132, IEEE.

[11] Wujian Ye and Kyungsan Cho, "Two-step P2P traffic classification with connection heuristics", in *Proceedings - 7th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2013*, 2013, pp. 135–141.

[12] Raihana Syahirah Abdullah, Mohd Faizal Abdollah, Zul Azri Muhamad Noh, Mohd Zaki Mas'ud, Shahrin Sahib, and Robiah Yusof, "Preliminary study of host and network-based analysis on P2P Botnet detection", in *2013 International Conference on Technology, Informatics, Management, Engineering and Environment*. jun 2013, pp. 105–109, IEEE.

[13] Chunyong Yin and Chunyong, "Towards accurate node-based detection of p2p botnets", *TheScientificWorldJournal*, vol. 2014, pp. 425491, jun 2014.

[14] Hongling Jiang and Xiuli Shao, "Detecting P2P botnets by discovering flow dependency in C&C traffic", *Peer-to-Peer Networking and Applications*, vol. 7, no. 4, pp. 320–331, 2014.

[15] Chunzhi Wang, Xin Zhou, Fangping You, and Hongwei Chen, "Design of P2P Traffic Identification Based on DPI and DFI", in *2009 International Symposium on Computer Network and Multimedia Technology*. dec 2009, pp. 1–4, IEEE.

[16] Ahmad Azab, Mamoun Alazab, and Mahdi Aiash, "Machine Learning Based Botnet Identification Traffic", in *2016 IEEE Trustcom/BigDataSE/ISPA*. aug 2016, pp. 1788–1794, IEEE.

[17] Mamoun Alazab, "Profiling and classifying the behavior of malicious codes", *Journal of Systems and Software*, vol. 100, pp. 91–102, 2015.

[18] Mamoun Alazab, Sitalakshmi Venkataraman, and Paul Watters, "Towards understanding malware behaviour by the extraction of api calls", in *2010 Second Cybercrime and Trustworthy Computing Workshop*. IEEE, 2010, pp. 52–59.

[19] M Abadi R Sharifnya, "DFBotKiller: Domain-flux botnet detection based on the history of group activities and failures in DNS traffic", *Digital Investigation*, vol. 12, pp. 15–26, mar 2015.

[20] Sebastian Garcia, Martin Grill, Jan Stiborek, and Alejandro Zunino, "An empirical comparison of botnet detection methods", *computers & security*, vol. 45, pp. 100–123, 2014.

[21] Riaz Ullah Khan, Xiaosong Zhang, and Rajesh Kumar, "Analysis of resnet and googlenet models for malware detection", *Journal of Computer Virology and Hacking Techniques*, Aug 2018.

[22] Riaz Ullah Khan, Xiaosong Zhang, Rajesh Kumar, and Emelia Opoku Aboagye, "Evaluating the performance of resnet model based on image recognition", in *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence*, New York, NY, USA, 2018, ICCAI 2018, pp. 86–90, ACM.

[23] Rajesh Kumar, Zhang Xiaosong, Riaz Ullah Khan, Ijaz Ahad, and Jay Kumar, "Malicious code detection based on image processing using deep learning", in *Proceedings of the 2018 International Conference on Computing and Artificial Intelligence*, New York, NY, USA, 2018, ICCAI 2018, pp. 81–85, ACM.