ISSN 2278-3091
Volume 9, No.3, May - June 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse79932020.pdf
https://doi.org/10.30534/ijatcse/2020/79932020

Detecting Botnet based on Network Traffic

Nguyen Vuong Tuan Hiep[1], Tisenko Victor Nikolaevich[2], Do Minh Tuan[1], Nguyen The Lam[1], Nguyen Anh Tuan[1]
[1,3,4,5]Information Assurance dept. FPT University, Hanoi, Vietnam, hiepnvtse05065@fpt.edu.vn, tuandmse05518@fpt.edu.vn, lamntse63326@fpt.edu.vn, tuannase62864@fpt.edu.vn
[2]Department Quality Systems, Peter the Great St. Petersburg Polytechnic University, Russia, St.Petersburg, Polytechnicheskaya, 29, v_tisenko@mail.ru

ABSTRACT

Nowadays, to bypass the surveillance of intrusion detection and prevention systems, cyber attackers often find ways to use botnets to connect and control malicious code. If the process of controlling and connecting from malicious code to the control server is detected and prevented, the whole attack will fail. Therefore, the problem of early detection of botnet networks in the system is very necessary today. There have been many methods of detecting botnet based on network traffic using sign sets and behavior sets. In this work, we will introduce the method of using machine learning to detect botnet signals in the system based on their abnormal behavior which collected on network traffic.

Key words: Botnet, abnormal behavior, network traffic, machine learning, detection.

1. INTRODUCTION

Botnet, whose full term is "Bot network", refer to the network of computers directed by someone and controlled by another remote computer [1]. If a computer is part of a botnet, which means it has been infected with one of these malwares such as viruses, worm, bot, etc. In fact, thousands of computers on the Internet are infected with some kind of "bot" that they don't even recognize it. A botnet consists of the following components.

- ✔ The bot is internet-connected devices that are infected with malware and controlled remotely.
- ✔ C & C Server (Command and Control Server) is a server that controls the bots in the network through broadcasting commands.
- ✔ Botmaster is a person who takes control of C & C servers and gives commands to the bots in the botnet.

Botnets use different protocols to connect to C&C [1, 2]. However, in order to bypass the surveillance of intrusion detection systems, botmasters tend to use common communication protocols and applications to connect to bots. Some common types of protocols are often used by botnets such as [1, 2, 3] Internet Relay Chat (IRC), HyperText Transfer Protocol (HTTP), Domain Name System (DNS) or Peer to Peer (P2P). This makes detecting, preventing and researching botnets become much more difficult.

Current research directions for botnet detection focus on two main issues [1, 4, 5] that are detection based on honeypot and detection based on passive network traffic monitoring. For honeypot-based detection, the focus is usually on two main issues [6]. The first is low-level interaction Honeypot. They are built to collect as many malware samples as possible. After malware samples have collected, the experts will analyze them. The second is the high-level interaction Honeypot. They are services, applications, and operating systems. The main purpose of these honeypots is strong interaction to better understand the attack methods and attack behaviors of hackers. The method of botnet detection based on honeypot is highly effective in the research and in the analysis of botnet characteristics. It has been highly appreciated by many experts. However, this method doesn't really make much effect in detecting botnet infections. In botnet detection methods based on passive network traffic monitoring, there are two main techniques that are application-based and protocol-based [1]. In this paper, we propose a method for detecting botnet based on the protocol layer. Accordingly, we conduct a network traffic analysis to obtain information about protocols of the packet and use machine learning algorithms to detect abnormal behavior in network traffic.

2. RELATED WORKS

The study [8] proposed an algorithm to identify C&C server by tracking queries of domain names that have high or abnormal DDNS query ratio (Dynamic DNS - method of mapping domain names to IP addresses that are changed frequently). This approach is similar to the approach proposed in the study [9]. However, both of these techniques can be easily overcome using fake DNS queries.

In the study [10], the authors proposed a DNS traffic monitoring system to detect botnet which have substructures

that form an active group in DNS queries at the same time. They have also developed a mechanism that allows detecting the relocation of C&C servers. This method is more robust than previous methods, and it can detect many types of botnets, even including botnets using encrypted channels because it uses the information in the IP header. However, the biggest weakness of this method is the high processing time, especially for monitoring a large network.

The publication [11] presented a method based on passive traffic monitoring for unusual or suspicious IRC identifiers, IRC servers and uncommon server ports. By using n-gram analysis techniques and scoring system to detect bots using unusual communication channels. However, these approaches have many limitations because IRC identifiers may change to resemble normal. In addition, this method doesn't detect botnets using encrypted communication channels nor botnets that is not an IRC.

The research [12] proposed a mechanism to detect C&C botnet traffic by passive analysis on network information flow. Their approaches are thread-based with characteristics such as duration, number of bytes per packet, bit/s, TCP flag and number of packets pushed into the stream. The system was implemented in 2 steps. Firstly, they distinguish the IRC traffic stream. Then they determine the C&C botnet traffic from there. Even though these techniques are effective to detect some botnets, but they just focus on the detection of IRC botnets. Moreover, for accurate analysis and detection, these techniques require access to the contents of the payload. Therefore, it cannot detect encrypted C&C traffic

3. BOTNET DETECTION BASED ON MACHINE LEARNING TECHNIQUES

3.1 Model overview

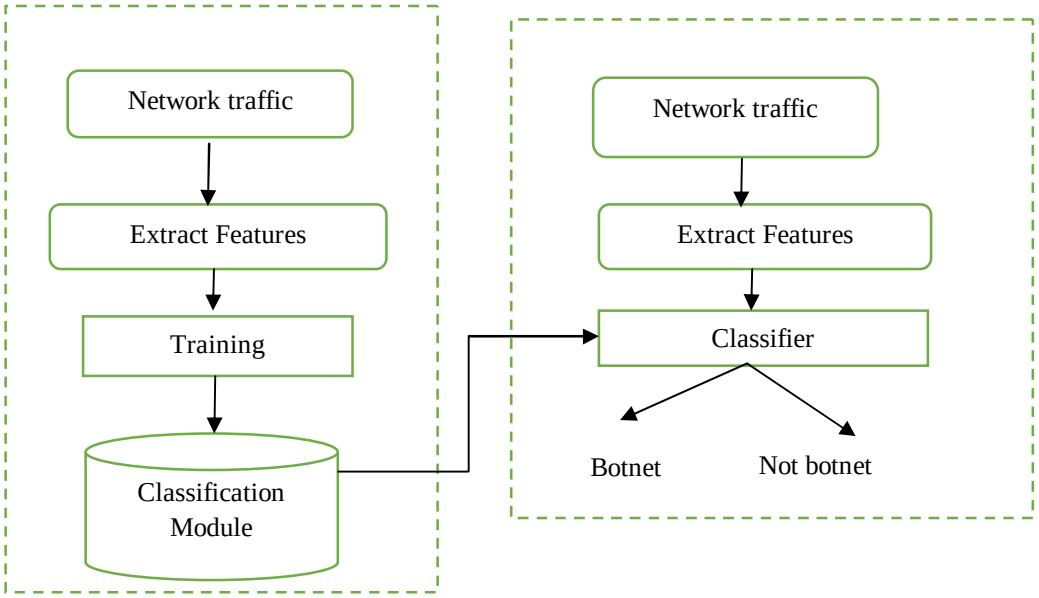

Figure 1: Overview model

Bots of botnet frequently send connections (figure 1) to the DNS system to find the IP addresses of C&C servers by using automatically generated domains. The detection model is implemented in two stages: training phase and detection phase.

In the training phase, DNS query data is collected from network traffic, and then the domain names in the DNS queries are extracted. Next, gather the pre-processed domain names to extract the features for training. In this phase, machine learning algorithms are used to classify normal and abnormal data. Through the evaluation process, machine learning algorithms provide the classification accuracy to be used in the detection model.

In the detection phase of the model, the DNS queries extracted from within Network Traffic will be monitored, extracted domain, preprocessed and classified. This classification use classifier which is built from the training phase to determine whether the domain name is legitimate or botnet.

The preprocessing step for each domain name in the training and detection phases is the same. However, in the training phase, this step is executed in offline mode for all domain names of the training data set, whereas in the detection phase, it is executed for each domain name extracted from the DNS query

3.2 Select and extract features

Some properties of botnet abnormal behavior are shown in table 1.

Table 1: Botnet features

Feature	Description	Data type
1. count_dns_requests	The number of DNS requests	Integer
2. count_distinct_dns_requests	The number of distinct DNS requests	Integer
3. high_request_single_domain	The highest number of requests for a domain	Integer
4. avg_req_per_min	The average number of requests in 1 minute	Integer
5. high_req_per_min	The highest number of requests in 1 minute	Integer
6. count_a_requests	The number of requests containing A variable: map domain with IPv4 address	Integer
7. count_mx_requests	The number of requests containing MX variable: map domain with a mail delivery agent	Integer
8. count_ns_requests	The number of requests containing NS variable: specifies a DNS zone to use a specific name server	Integer
9. count_ptr_requests	The number of requests containing PTR variable: pointer to a CNAME (real name of the hostname of a computer)	Integer
10. distinct_tld_domains	The number of distinct top-level domains	Integer
11. distinct_sld_domains	The number of distinct second-level domains	Integer
12. count_responses	The number of responses	Integer
13. distinct_city_of_ipaddress	The number of cities containing the IP address requested	Integer
14. distinct_subdivision_of_ipaddress	The number of subdivisions containing the IP address requested	Integer
15. distinct_country_of_ipaddress	The number of countries containing the IP address requested	Integer
16. count_response_records	The number of response records	Integer
17. count_response_success	The number of successful responses	Integer
18. count_response_failed	The number of failed responses	Integer
19. avg_ttl_value	Average time to live	Integer
20. high_ttl_value	Max time to live	Integer
21. count_response_ipaddress	The number of responses containing the IP address	Integer
22. flux_ratio	Change domain to keep botnet running	Integer
23. uniqueness_ratio		Integer

3.3 Select machine learning algorithm

We use some of the following algorithms to classify botnets: Random Forest (RF), Support Vector Machine (SVM), k - nearest neighbors (KNN), Naive Bayes. The document [8] presented the operating principles of the above algorithms in detail. In this paper, to see clearly the effectiveness of each algorithm, we will proceed to change some parameters of these algorithms.

4. EXPRIMENTAL AND EVALUATED

4.1 Experimental data

In this paper we use the dataset published at [14]. The dataset consists of 608.738 records in which the number of malicious records is 7.645 records. The number of clean records is 601.093 records. This is a data set for research and testing to detect botnet. The data set is regularly updated about the number of malicious records and clean records. This lead to the research and application of this data set very effectively in practice

4.2 Metrics

The above metrics are calculated by following formula:

$$PPV = \frac{TP}{TP + FP} \times 100\ \%$$

$$FRP = \frac{FP}{FP + TN} \times 100\ \%$$

$$TRP = \frac{TP}{TP + FN} \times 100\ \%$$

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\ \%$$

$$F1 = \frac{2 \times precision \times \mathrm{Re}\ call}{precision + \mathrm{Re}\ call}$$

Where TP is the number of records labeled "Botnet" correctly classified, TN is the number of records labeled "Normal" correctly classified, FP is the number of flows labeled "Normal" incorrectly classified as "Botnet" and FN is the number of flows labeled "Botnet" that are incorrectly classified as "Normal".

4.3 Results and discussion

Table 2: Compare botnet detection results

Algorithm	PPV (%)	TPR (%)	FPR (%)	ACC (%)	F1 (%)
RF	99,61	99,23	0,1767	99,64	99,51
kNN	87,46	97,75	5,4381	95,45	92,32
Naive Bayes	99,98	41,53	0,0252	56,01	58,69
SVM	97,24	94,22	1,27	97,28	95,71

From the results obtained above, we can see that the Naïve Bayes machine learning algorithm gives the lowest classification accuracy (ACC) and the RF algorithm gives the highest classification accuracy in four machine learning algorithms used in testing. This experimental result is completely consistent with the fact because recent studies have shown that the RF algorithm is the best classification algorithm because the algorithm uses a lot of different trees to support the decision. SVM and kNN algorithms have approximately the same classification accuracy. In addition, the experimental results in the paper also show the effectiveness of the features that we use. Although the difference between malicious data and benign data is very large (about 10 times), the rate of false detection and errors is very low.

5. CONCLUSION

In this work, we introduced using machine learning algorithm to detect botnet based on network traffic. Our botnet detection technique proposed in this study is able to detect abnormal connections based on the process of sending and receiving data from an internal machine to external machines if the packets are not is encrypted. However, if cyber attackers use cryptography techniques or the Tor network to hide, it is very difficult to detect them. In subsequent studies, we will study and extract features of connections not only based on the application layer but also based on the data layer to improve the efficiency of botnet detection.

REFERENCES

[1] Manmeet Singh, Maninder Singh, Sanmeet Kaur. **Issues and challenges in DNS based botnet detection: A survey**. *Computers & Security*. Volume 86, pp. 28-52. 2019 https://doi.org/10.1016/j.cose.2019.05.019

[2] Kamal Alieyan, Ammar ALmomani, Ahmad Manasrah, Mohammed M. Kadhum. **A survey of botnet detection based on DNS**. *Neural Computing and Applications*. Vol 28, pp. 1541–1558. 2017. https://doi.org/10.1007/s00521-015-2128-0

[3] Thorsten Holz, Moritz Steiner, Frederic Dahl, Ernst Biersack, and Felix Freiling. **Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm**. In *Proceedings of the 1st Usenix Workshop on Large-Scale Exploits and Emergent Threats*, CA, United States, 2008, pp. 1-9.

[4] Maryam Feily, Alireza Shahrestani, Sureswaran Ramadass, **A Survey of Botnet and Botnet Detection**. In *Proceedings of the Third International Conference on Emerging Security Information, Systems and Technologies*, Athens, Glyfada, Greece, 2009, pp. pp. 268-273

[5] Zhaosheng Zhu, Guohan Lu, Yan Chen, Zhi Judy Fu, Phil Roberts, Keesook Han. **Botnet Research Survey**. *In Proceedings of the 32nd Annual IEEE International Computer Software and Applications Conference*, Turku, Finland, 2008. pp. 967-972

[6] Jihan Barazi, Ahmad Jakalan, Wang XiaoWei. **Botnet Detection Techniques**. *The International Journal of Computer Science and Communication Security*. pp 14-22. 2014

[7] Alparslan, Erdem & Karahoca, Adem & Karahoca, Dilek. **BotNet Detection: Enhancing Analysis by Using Data Mining Techniques**. DOI: 10.5772/48804. 2012. https://doi.org/10.5772/48804

[8] Dagon, D. **Botnet Detection and Response, The Network is the Infection**. OARC Workshop. 2005

[9] Kristoff, J. **Botnets**. In *Proceedings of the 32nd Meeting of the North American Network Operators Group*. 2004

[10] Choi, H., & Lee, H. **Identifying botnets by capturing group activities in DNS traffic**. *Computer Networks*, Vol 56, pp. 20-33. 2012. https://doi.org/10.1016/j.comnet.2011.07.018

[11] Jan Goebel, Thorsten Holz, Rishi. **Identify Bot Contaminated Hosts by IRC Nickname Evaluation**., In *Proceedings of the* 1st Workshop on Hot Topics in Understanding Botnets, USENIX, 2007.

[12] W. Timothy Strayer, David Lapsely, Robert Walsh, Carl Livadas. **Botnet Detection Based on Network Behavior**. *Information Security*. Vol 36, pp. 23-31, 2008

[13] Shai, S.S., Shai B.D.: **Understanding Machine Learning**: *From Theory to Algorithms. Cambridge University Press*. 2014.

[14] REAL-WORLD DNS DATASET (2016). https://ieee-dataport.org/documents/real-world-dns-dataset-2016. doi: 10.21227/9ync-vv09.