JACET

J. ADV COMP ENG TECHNOL, 6(2) Spring 2020

Detecting Bot Networks Based On HTTP And TLS Traffic Analysis

Zahra Nafarieh[1], Ebrahim Mahdipour[2], Haj Hamid Haj Seyed Javadi[3]

1- Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran. (zn.nafarieh@gmail.com)

2- Department of Electrical and Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.

3- Department of Mathematics and Computer Science, Shahed University, Tehran, Iran.

Received (2019-05-03) Accepted (2020-01-12)

Abstract: *Bot networks are a serious threat to cyber security, whose destructive behavior affects network performance directly. Detecting of infected HTTP communications is a big challenge because infected HTTP connections are clearly merged with other types of HTTP traffic. Cybercriminals prefer to use the web as a communication environment to launch application layer attacks and secretly engage in forbidden activities, while TLS (Transport Layer Security) protocols allow encrypted communication between client and server in the context of Internet provides. Methods of analyzing traffic behavior do not depend on payloads. This means that they can work with encrypted network communication protocols. Traffic behavior analysis methods do not depend on package shipments, which means they can work with encrypted network communication protocols. Hence, the analysis of TLS and HTTP traffic behavior has been considered for detecting malicious activities. Because of the exchange of information in the network context is very high and the volume of information is very large, storing and indexing of this massive data require a Big data platform.*

Keywords: *Bot Networks, HTTP Traffic Analysis, TLS Traffic Analysis, Intrusion Detection, Network Security, Security Threats.*

How to cite this article:

Zahra Nafarieh, Ebrahim Mahdipour, Haj Hamid Haj Seyed Javadi. Detecting Bot Networks Based On HTTP And TLS Traffic Analysis. J. ADV COMP ENG TECHNOL, 6(2) Spring 2020 : 61-70

1. INTRODUCTION

Botnet is a coordinated group of malware that are controlled via C&C communication channels. Botnets can be considered as complex threats which are used by sinners to carry out a variety of cyber attacks such as DDOS, spam, phishing, thieving of personal and financial information, information leakage, scareware and sniffing traffic [1].

Detecting of malicious HTTP communication is a really huge challenge since the malicious HTTP communication is transparently merged with other type of HTTP traffic. The difficulty of detecting botnet traffic (especially for HTTP-based botnets) amongst web activity is a current research challenge. It is easy to create concentrated structures of botnets, this structure has a major weakness, for example the command and control server is the single point of failure, and the disabling of this server causes the bot master loses his connection with all the bots [2].

Botmaster controls an organized group of infected devices [3]. A bot master relies on these channels to send commands to the members of its botnet to execute attack activities. Malicious HTTP-based bots always connect to their command and control server periodically in order to get the commands and updates. Bots constantly communicate with their Botmaster via command and control server to receive new commands and updates, in addition to reporting their status. It is important for Botmasters to establish a secure command and control mechanism that plays an important role for all types of Botnet operation [4, 5]. The HTTP protocol is known as the protocol which is used on the Internet and cybercriminals prefer to use the HTTP protocol to perform their forbidden activities. HTTP protocol and port 80 is being used by botnets to impersonate normal web traffic and bypass the network security systems [6].

The use of encryption on the Internet has spread rapidly these last years, a trend encouraged by the growing concerns about online privacy. Businesses, schools, governments and individuals all benefit from the privacy encryption provides and the usage of TLS will certainly continue to grow in the years to come. Today most of the Internet traffic is encrypted with TLS. However privacy does not guarantee security and malware authors have started to leverage TLS to hide their malicious activities. TLS (Transport Layer Security), the standard protocol for packet encryption, is now implemented by every major websites to protect users' messages, transactions and credentials. However cybercriminals have started to incorporate TLS into their activities [7]. Malware have been seen to use TLS to communicate with their command server, either to receive instructions or to send back sensitive data which is collected on the infected machines [8].

Until now, there exists no permanent solution for the detection or mitigation of botnets threats because their techniques and methods keep changing over time [9]. The Botnet detection process stands as an ongoing challenge for researchers and organizations [10].

In this paper, we propose a novel system which is placed at the network egress point that aims to efficiently and effectively detect malware infections based on traffic analysis. We received information about known botnets from reputable resources that provide services under the CTI (cyber threat inteligence) name. These data are generated by the intrusion detection sensor, so that network traffic is given to the intrusion detection sensor and various logs are generated and registered by it. Splunk analysis platform stores and categorizes all generated logs and by generating dashboards, it's possible to monitor the network security status.

The major contributions can be summarized as follows:

- Using a botnet detection technique based on a behavioural and signature-based analysis approach to detect malicious activities in a given network.
- Analyzing HTTP and TLS logs both with random and fixed intervals and the signature-based and behavior-based techniques to effectively detect attacks.
- Momentary, periodic, and online monitoring of network security status and dynamic analytics to find malicious activities on the network.
- In this method not only does not reduce the volume of network traffic which needs to be recorded and analyzed, but also improves the sustainability of the system.
- This method present a novel system which is placed on the edge of the network using a combination of malicious HTTP and TLS detection and intrusion detection technology to detect infected machines inside the network.
- This method defines the specifics to identify the compromised clients that have been remotely controlled.

The rest of the paper is organized as follows: Section 2 presents existing work on identifying malicious HTTP and TLS traffic. Section 3 introduces a method for identifying Botnet activity and infected clients within large-scale networks by monitoring HTTP and TLS traffic. In section 4, intrusion detection methods have been investigated in the network. Section 5 illustrates an example of the HTTP and TLS log reviewed in this paper. Finally, Sect 6 concludes the paper.

2. RELATED WORKS

Several methods have been proposed for detecting botnets, some of the most well regarded methods are referred to below:

Jae-Seo et al. [3] and Tung-Ming et al. [11] proposed a parameter based on one of the properties of HTTP-based botnets. Detecting of the HTTP botnets with a low rate of false alarms has become a notable challenge. They suggested a periodic repeatability degree to show the regular connections pattern from HTTP-based bots to specific servers [6]. The main drawback of this method is that it does not use TLS traffic information and also has false negative. By changing the method of connection distances, botmasters can eliminate this technique and produce false negative results [12].

Pengkui et al. [13], Yadav et al. [14], Sharifnya, Abadi [15] and Kazato et al. [16] focused on DNS traffic and DNS-related failures, and by analyzing them could identify botnets, and mainly focused on detecting malware activity by monitoring and analyzing passive DNS traffic which used of malicious flux service [17]. The major weakness of this approach is that it used passive monitoring of traffic, only known botnets are detected, which is a major weakness in botnet detection methods.

Ibrahim Ghafir et al. [18] proposed a method which applied on packet captured (PCAP) files that contain malicious domains traffic and based on blacklist of malicious domains. This approach detected any connection to malicious domain and can detect the connection to malicious domain and output is correlated with the other detections methods. Blacklist is updated automatically and detection is in real time. In this way, only known botnets are detected, which is a major weakness in botnet detection methods.

Lu et al. in [19] categorised the services and application flows using payload-signature to examine the bit strings in the packets payload as a signature. These signatures were used to separate known traffic from unknown traffic in order to decrease the false alarm rates. The major weakness of this approach is like traditional signature-based techniques the proposed classifier is less effective as it is unable to identify new or encrypted patterns and possibly increase the false negative rate.

Binbin et al. [20] used request byte, response byte, and the number of packets as common features of an HTTP connection, to classify the similar connections generated by a single bot. The primary drawback of this method is that it can detect the small-scale botnets, but some techniques like random request delay or random packet number can evade their detection method and it is more prone to generating false positives.

Assdhan et al. [21, 22] did not propose any new measures to formulate the periodicity of HTTP based Botnet command and control traffic. Instead, they employed an existing technique called periodogram to estimate the Power Spectral Density (PSD) of TCP packets per time intervals. To accomplish this, they divided packet capturing time into discrete time sequence followed by the extraction of a few features such as number of packets per interval (PPI) and bytes per interval (BPI). Finally, several periodograms are illustrated to detect the periodic behaviour of network activities by estimating the distribution of the aforementioned parameters amongst time intervals. However, this technique can easily be evaded by using a random number of packets and bytes. In addition, the time period for each interval is less than one second which is obviously not sufficient to measure the efficiency of the proposed technique in the real world [23, 24].

Li et al. [2] suggest that botnets implementing the HTTP protocol often use poorly formed headers. Therefore, deviations from protocol standards can be used for detection. Network data is first clustered into groups of similar flows. According to the authors, the packets exchanged at the start of C&C communications often contain sensitive details (e.g. IPs, bot IDs, and process names). Therefore, they reduce the volume of data by extracting only the first request and response packets from HTTP flows. The flows are analysed for periodicity, whilst packet headers are examined for missing fields and malicious keywords. The system combines the results of each check to calculate a suspiciousness score for that flow. While this approach can be successful at finding threats, there are several important shortcomings. First, examination of packet headers may not be possible if data is encrypted. Second, this method is computationally expensive and difficult to deploy and maintain.

We propose a new method to detect botnets from the HTTP and TLS traffic both with random and fixed intervals and the signature-based and behavior-based techniques and regardless of the Botnet size and scale. We place the system on the edge of the network and do the network traffic analysis to detect infected machines inside the network. As a result, botnet traffic can be possibly identified and monitored through identifying the anomalies within TLS and HTTP traffic.

No previous work has tried to identify malicious activity by analyzing periodic behavior in TLS and HTTP traffic for servers with the highest number of requests, malware infections activities, and checking status codes in HTTP traffic.

3. Proposed Method

When malicious software by using the HTTP protocol communicates with the command and control server, analyzing HTTP headers is a useful method for detecting malware [25]. HTTP-based communications have also been allowed for flexibility in most networks. Cybercriminals prefer to take full advantage of the Web platform for communicating to launch attacks in the stealth of prohibited and illegal activities. Hence, HTTP and TLS traffic behavior analysis is required to detect infected activities. To detect Bot networks, can perform identification tasks at 2 points: 1) analysis within computers; 2) network traffic analysis. The network traffic of an organization are entered into the Suricata sensor and HTTP and TLS logs are extracted by this sensor, then these logs and information about active bot networks which are received from the CTI (Cyber Threat Intelligence) services are entered into the Splunk platform, and then features of both categories are extracted and compared with each other. Then behavioral and signature-based analysis is done according to the resulting dashboards.

In this research, we proposed a new system that is located at the edge of the network which is the output point of the network and effectively detects malicious behavior by analyzing traffic. In this system, we used signature-based and behavior-based techniques for detecting botnets. Behavioral-based detection techniques attempt to detect botnets based on network traffic anomalies such as high network latency, high volumes of traffic, traffic on unusual ports that could indicate presence of malicious bots. Signature-based detection techniques can be used to detect known botnets. Intrusion Detection Sensor to identify bots uses a signature-based method. A database that contains known bots features, received these information from external authoritative sources which is called Cyber Threat Intelligence (CTI). The experiment has been executed for two months in an extensive network. In this research all the features in the big data volume are investigated. Splunk is a big data platform which is used to store and index data. Network traffic and known malware information entered into the Splunk platform and the results are displayed in the form of dashboards. We saw the pattern of attacks by enter the PCAP (packet capture) traffic which is related to the infected traffic into the Splunk platform. Surikata sensor is an intrusion detection Sensor which received network traffic then extract DNS, HTTP, TLS, and NETFlow logs. Network traffic and known malware information entered into the Splunk platform and the results are displayed in the form of dashboards.

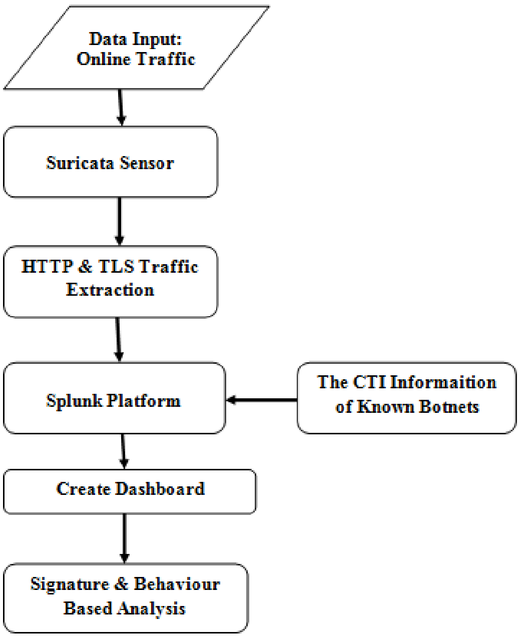


Fig.1. Flowchart of proposed methodology

4. Methods of analysis in intrusion detection systems

Signature-based intrusion detection systems work according to pattern recognition mechanisms. In this way, pre-built intrusion patterns are kept, and if such a pattern is introduced into the system, the influence is indicated. These types of methods can only detect known influences. The advantage of these methods is the precision in detecting intrusions whose patterns have been installed into the system. In behavioral-based intrusion detection methods, the collected information is examined to identify patterns that indicate unusual behavior, and if the behavior of the system does not follow these patterns at the time of the survey, it is considered as a possible

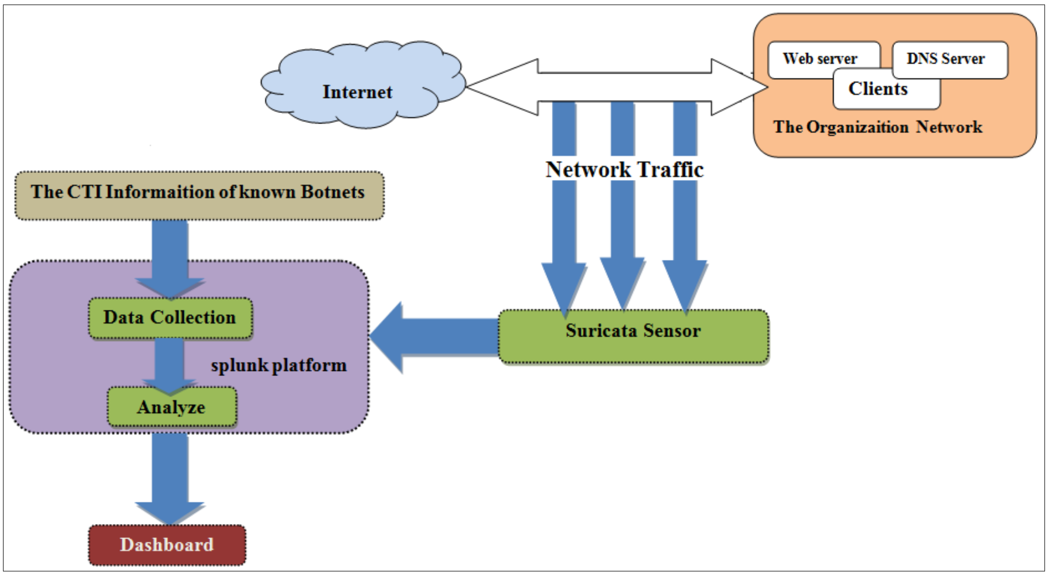


Fig.2. The architecture of the Botnet detection system

influence. Normal traffic in HTTP requests is a variety of HTTP requests generated at random intervals, and the bot traffic is a duplicate HTTP request that is generated at regular intervals. Bots connect permanently to their C & C server to receive new commands and updates. Total time of traffic accumulation is divided into several windows. The length of each window plays an important role in the accuracy of behavior analysis. The number of requests is designed to measure the similarity of group member activity in time windows. In order to determine whether the same number of requests was generated through a group of data in each time window [2].

5. Monitoring and Results

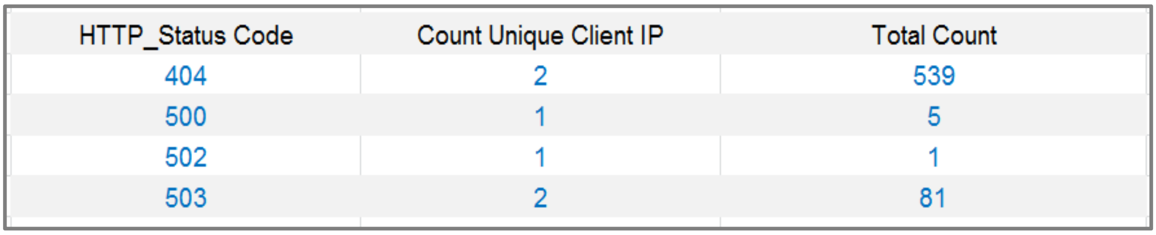
The signature-based techniques detect only known bots and the anomaly-based or behavior-based techniques attempted to detect botnets by analyzing network traffic for example sudden vast amount of traffic, traffic to unusual ports, high network latency and anomalous behavior that may detect the existence of bots in the network. These approaches have the ability to identify new bots [10]. The anomaly-based technique can be categorized into host-based and network-based detection techniques [10]. Botnet can be possibly identified by monitoring the anomalies within the HTTP and TLS traffic. Domain Name System (DNS) represents one of the core components of the Internet that facilitates decoupling of service names and hosting IP addresses [26] in order to detect domain names associated with various types of malicious activities, proposed a large-scale system which analyzes DNS traffic. The proposed method identifies the infected host by monitoring HTTP and TLS traffic.

5.1. Checking Status Codes in HTTP Traffic

In order to prevent detection, most malware traffic data is encrypted. So encrypted data transpire on uncommonly-used status is also likely malicious traffic. Checking status codes in HTTP traffic is a way to identify malicious traffic.

Figure3, shows the number of optimal requests within 24 hours in HTTP traffic. The desired status codes in the sample case of HTTP traffic include codes 200, 204, and 206.

Figure 4, shows the number of clients associated with each undesired HTTP traffic status code, as well as the total number of each undesired HTTP status code in the sample of HTTP event examined in this project. If the number of undesired status codes displayed on the chart within a certain time range is high, it indicates a suspicion of an attacking bot.



HTTP_Status Code	Count Unique Client IP	Total Count
404	2	539
500	1	5
502	1	1
503	2	81

Figure 4. The number of unique clients for each undesired HTTP traffic status code

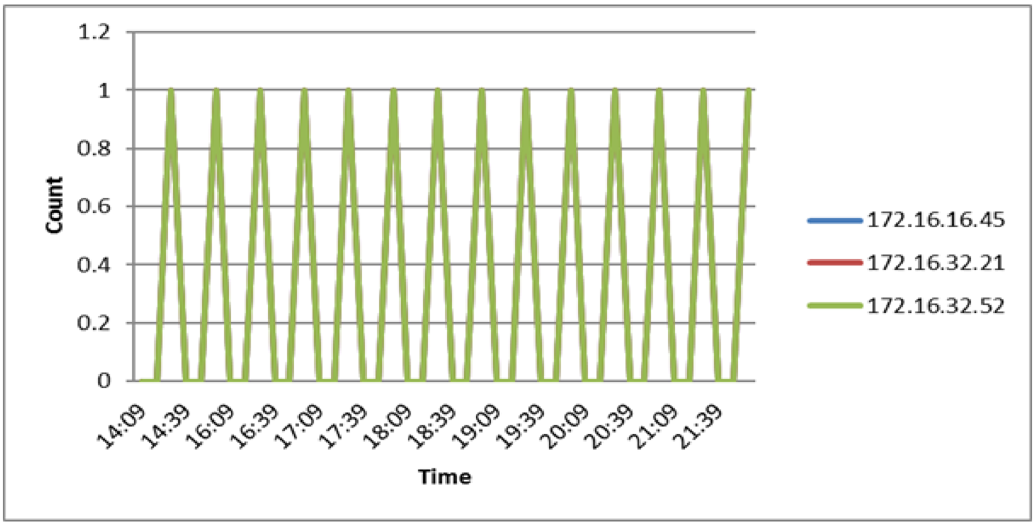


Figure 5. Periodic behavior in infected HTTP traffic

5.2. Detecting periodic behavior in HTTP traffic

Figure 5, represents a periodic behavior for HTTP traffic. Clients with an IP address; 172.16.16.45, 172.16.32.21, and 172.16.32.52 are periodically referred to the server with the IP address of 98.131.132.1. As shown in the figure 5, the client with IP '172.16.32.52' which is infected with the Bot sent a request at intervals of 30 minutes.

5.3. Detecting Dos attack on HTTP traffic

Bots need to communicate periodically with their command and control server. In a vulnerable network that one of the hosts has been infected with a malicious code, the bots will be released to other hosts in that network. Figure 6 shows that the client 147.32.84.167 in December had requests to the server 174.133.57.141. As shown in the figure, this client sent a request to the server on December 25 and December 28 respectively, at 1223 and 2476 times. This client sent a request to the server at short intervals.

5.4. HTTP Traffic for Servers with the highest number of requests

Figure 7 represents 100 web servers that have the highest number of requests. This behaviour can be suspected of Bot activity and within a specified time period indicating a suspicious of bots activity in the organization's network.

When one of the host in the network is infected with bots attempts to infect other computers in the network and start group activities. Figure 8 Represents 10 web servers with the highest number of clients connected to it. It can reflect the behavior of the bot activity in the organization's network.

5.5. Source and destination IPs in HTTP traffic

The source and destination IPs in randomized traffic should be distributed randomly, in order to facilitate access to a variety of services for users. In

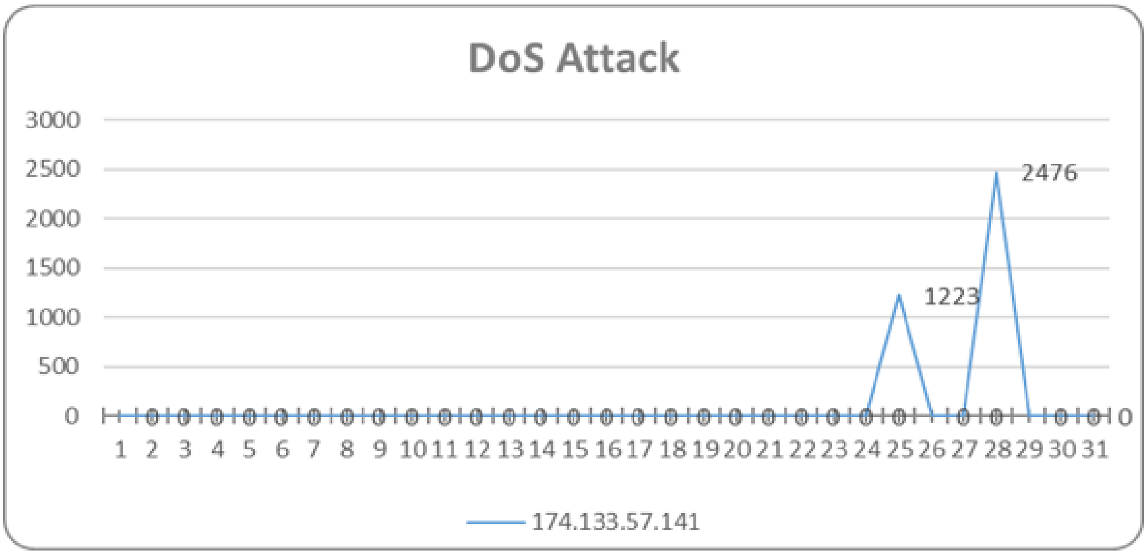


Figure 6. Detecting the Dos attack on the server 174.133.57.141

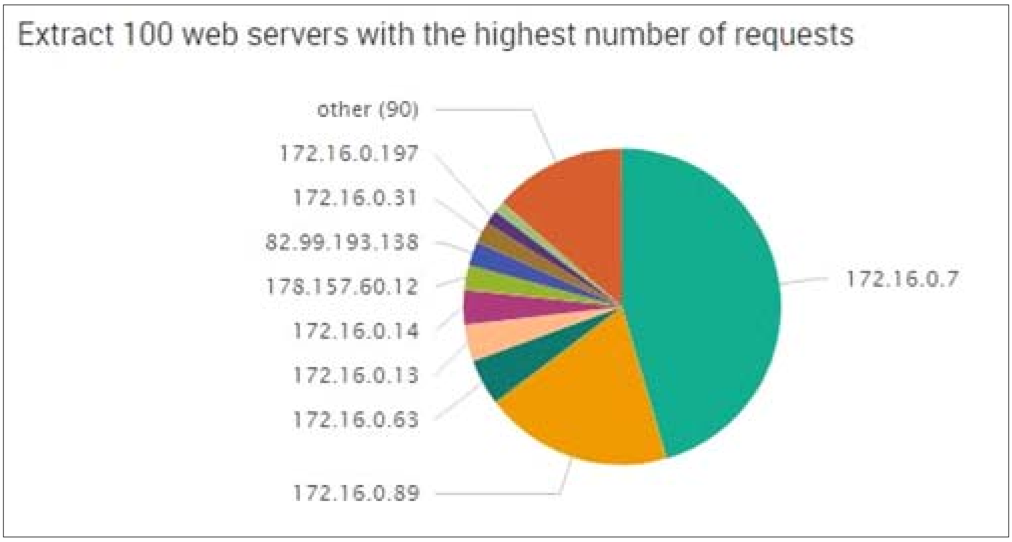


Figure 7. 100 Web server with the highest number of requests

a vulnerable local area network, which has been infected by one of the hosts, the probability of bots spreading to other hosts on the network is also high, and since bots need to communicate with the C & C server, they communicate periodically with them. So, identifying the position of the IPs on the map can identify suspicious behaviors. If IPs that are within a certain range can be accessed periodically and extensively by an anonymous server, this behavior may be the behavior of a bot network or a DDoS attack. By identifying the position of IPs, fake IPs can be detected from valid IPs. Figure 13 represents the position of 10 web servers with the highest number of requests on the map.

5.6. TLS traffic

By analyzing TLS traffic, you can view TLS encrypted traffic without having to decrypt the cargo. If an abnormal domain receives a large

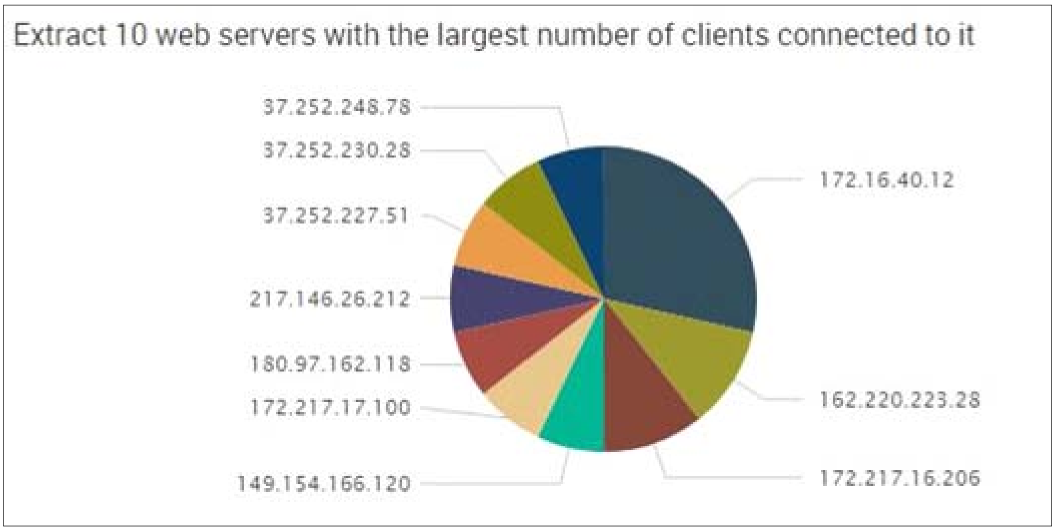


Figure 8. Servers with the highest number of client referrals

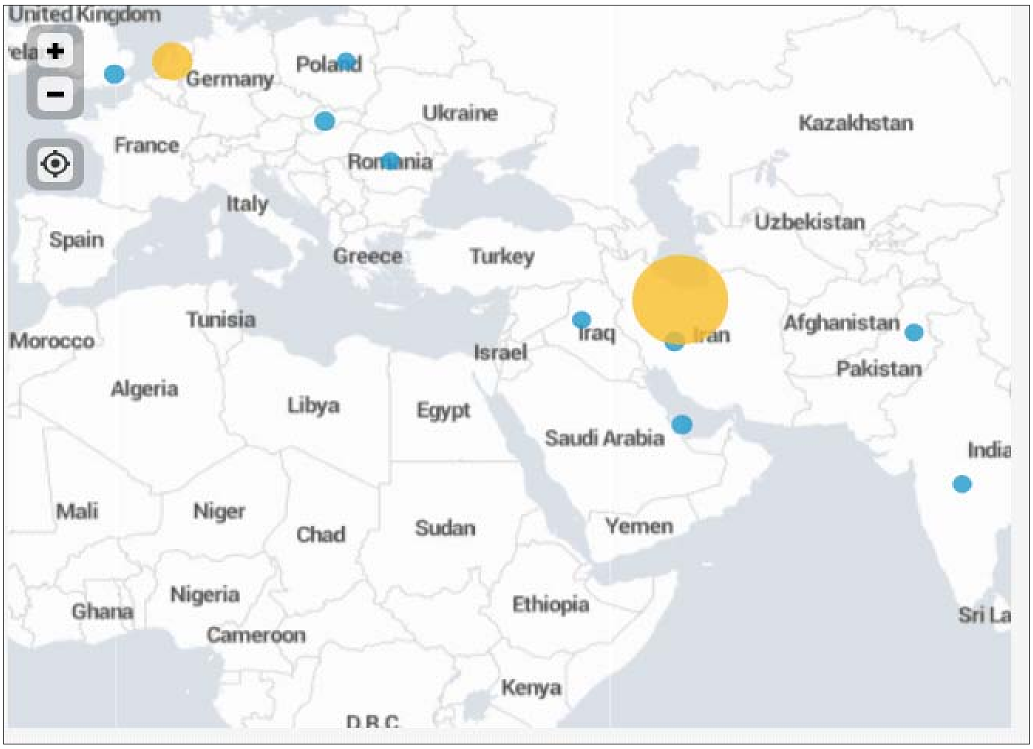


Figure 9. 10 Web server position with the highest number of requests on the map

number of TLS connections from hosts on a network, this connection may be unusual and suspicious traffic. The wave form in Fig. 10 has been transformed abnormally at one time. By examining events, it is possible to receive a number of IPs in a given range over a specified period of time, sending over two thousand requests to a specified server, with the IP address, 98.124.1991, which represents a DDoS attack.

5.7. Checking port in TLS traffic

Since port 443 is the standard TLS protocol port and is open on most networks and access to the outside network is also given through the

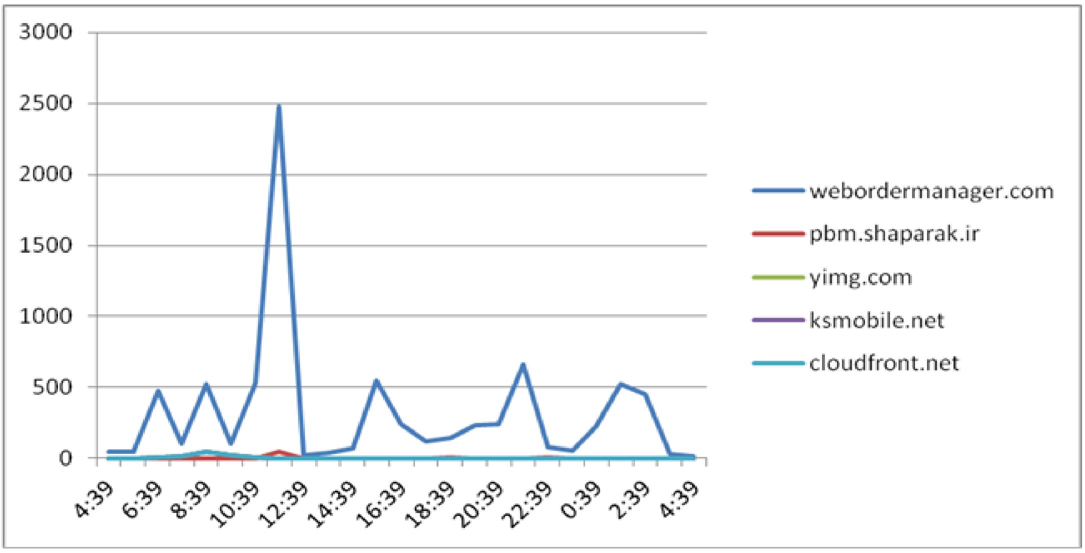
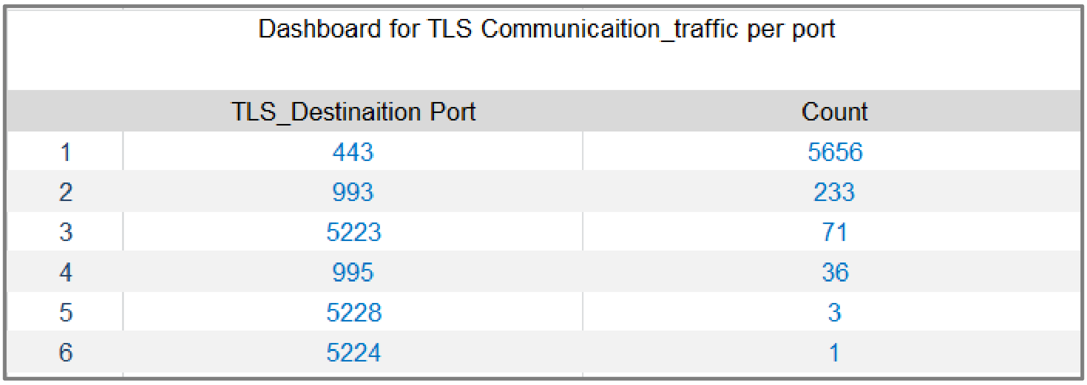


Figure 10. DDOS attack for TLS traffic



Dashboard for TLS Communicaition_traffic per port

	TLS_Destinaition Port	Count
1	443	5656
2	993	233
3	5223	71
4	995	36
5	5228	3
6	5224	1

Figure 11. Destination port traffic in TLS communication

firewall, if the non-TLS connection is established through port 443, it can be a suspicious traffic and should be investigated. Some standard ports used by the TLS protocol include port 993 for encrypted IMAP protocol, port 995 for encrypted POP protocol, port 465 for encrypted SMTP protocol and etc.

6. CONCLUSION

Detecting communication malicious HTTP is a big challenge because malicious HTTP connections are transparently merged with other types of HTTP traffic. The TLS protocol provides an encrypted connection between the client and the server on the Internet. Due to the massive amount of information from network traffic and the amount of malware that is added every day, using a system that uses one of the big data tools to analyze security logs is important. In this paper, with behavioral and signature-based analysis, we proposed a method for detecting online Botnet activity by monitoring HTTP and TLS traffic. The experimental results show that our security approach is good for detecting malware infections.

Acknowledgment

The author would like to thanks the security team and their guideline, suggestions and technical support during the work.

REFERENCES

1. Eslahi, M., R. Salleh, and N.B. Anuar. Bots and botnets: An overview of characteristics, detection and challenges. in 2012 IEEE International Conference on Control System, Computing and Engineering. 2012. IEEE.

2. Acarali, D., et al., Survey of approaches and features for the identification of HTTP-based botnet traffic. Journal of Network and Computer Applications, 2016. 76: p. 1-15; Available from: https://www.sciencedirect.com/science/article/pii/S1084804516302363.

3. Eslahi, M., R. Salleh, and N.B. Anuar. MoBots: A new generation of botnets on mobile devices and networks. in 2012 International Symposium on Computer Applications and Industrial Electronics (ISCAIE). 2012. IEEE.

4. Li, C., W. Jiang, and X. Zou. Botnet: Survey and case study. in 2009 Fourth International Conference on Innovative Computing, Information and Control (ICICIC). 2009. IEEE.

5. Silva, S.S., et al., Botnets: A survey. Computer Networks, 2013. 57(2): p. 378-403; Available from: https://www.sciencedirect.com/science/article/abs/pii/S1389128612003568.

6. Eslahi, M., H. Hashim, and N.M. Tahir. An efficient false alarm reduction approach in HTTP-based botnet detection. in 2013 IEEE Symposium on Computers & Informatics (ISCI). 2013. IEEE.

7. Roques, O., Detecting Malware in TLS Traffic. 2019, Imperial College London.

8. Anderson, B. and D. McGrew. Machine learning for encrypted malware traffic classification: accounting for noisy labels and non-stationarity. in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2017.

9. Karim, A., et al., Botnet detection techniques: review, future trends, and issues. Journal of Zhejiang University SCIENCE C, 2014. 15(11): p. 943-983; Available from: https://link.springer.com/article/10.1631/jzus.C1300242.

10. Alieyan, K., et al., A survey of botnet detection based on DNS. Neural Computing and Applications, 2017. 28(7): p. 1541-1558; Available from: https://link.springer.com/article/10.1007/s00521-015-2128-0.

11. Security 101: Distributed Denial of Service (DDoS) Attacks, 2016. 2016; Available from: https://www.trendmicro.com/vinfo/us/security/news/cyber-attacks/security-101-distributed-denial-of-service-ddos-attacks.

12. Jang, D.-i., et al. Evasion technique and detection of malicious botnet. in 2010 International Conference for Internet Technology and Secured Transactions. 2010. IEEE.

13. Luo, P., et al. Leveraging client-side DNS failure patterns to identify malicious behaviors. in 2015 IEEE Conference on Communications and Network Security (CNS). 2015. IEEE.

14. Yadav, S. and A.N. Reddy. Winning with DNS failures: Strategies for faster botnet detection. in International Conference on Security and Privacy in Communication Systems. 2011. Springer.

15. Sharifnya, R. and M. Abadi, DFBotKiller: Domain-flux botnet detection based on the history of group activities and failures in DNS traffic. Digital Investigation, 2015. 12: p. 15-26; Available from: https://www.sciencedirect.com/science/article/abs/pii/S1742287614001182.

16. Kazato, Y., K. Fukuda, and T. Sugawara. Towards classification of dns erroneous queries. in Proceedings of the 9th Asian Internet Engineering Conference. 2013.

17. Heuer, T., et al. Recognizing Time-Efficiently Local Botnet Infections-A Case Study. in 2016 11th International Conference on Availability, Reliability and Security (ARES). 2016. IEEE.

18. Ichise, H., Y. Jin, and K. Iida. Detection method of DNS-based botnet communication using obtained NS record history. in 2015 IEEE 39th Annual Computer Software and Applications Conference. 2015. IEEE.

19. Lu, W., M. Tavallaee, and A.A. Ghorbani. Automatic discovery of botnet communities on large-scale communication networks. in Proceedings of the 4th international symposium on information, computer, and communications security. 2009.

20. Wang, B., et al. Modeling connections behavior for web-based bots detection. in 2010 2nd International Conference on E-business and Information System Security. 2010. IEEE.

21. AsSadhan, B., J.M. Moura, and D. Lapsley. Periodic behavior in botnet command and control channels traffic. in GLOBECOM 2009-2009 IEEE Global Telecommunications Conference. 2009. IEEE.

22. AsSadhan, B. and J.M. Moura, An efficient method to detect periodic behavior in botnet traffic by analyzing control plane traffic. Journal of advanced research, 2014. 5(4): p. 435-448; Available from: https://www.sciencedirect.com/science/article/pii/S2090123213001410.

23. Wang, K., et al., A fuzzy pattern-based filtering algorithm for botnet detection. Computer Networks, 2011. 55(15): p. 3275-3286; Available from: https://www.sciencedirect.com/science/article/abs/pii/S1389128611002040.

24. Eslahi, M., et al. Periodicity classification of HTTP traffic to detect HTTP Botnets. in 2015 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE). 2015. IEEE.

25. Zhao, G., et al., Detecting APT malware infections based on malicious DNS and traffic analysis. IEEE access, 2015. 3: p. 1132-1142; Available from: https://ieeexplore.ieee.org/abstract/document/7163279.

26. Stevanovic, M., et al., A method for identifying compromised clients based on DNS traffic analysis. International Journal of Information Security, 2017. 16(2): p. 115-132; Available from: https://link.springer.com/article/10.1007/s10207-016-0331-3.