Science World Journal Vol. 15(No 1) 2020
www.scienceworldjournal.org
ISSN 1597-6343
Published by Faculty of Science, Kaduna State University

BOTNET DETECTION FROM DRIVE-BY DOWNLOADS

Aisha Garba Bawa[1], Ayuba Peter[2*], Muhammad Aminu Ahmad[1]

[1]Department of Computer Science, Kaduna State University, Kaduna.
[2]Department of Mathematical Sciences, Kaduna State University, Kaduna.

*Corresponding Author's Email Address: ayubng@kasu.edu.ng

ABSTRACT

The advancement in Information Technology has brought about an advancement in the development and deployment of malware. Bot Malware have brought about immense compromise in computer security. Various ways for the deployment of such bots have been devised by attackers and they are becoming stealthier and more evasive by the day. Detecting such bots has proven to be difficult even though there are various detection techniques. In this work, a packet capturing and analysis technique for detecting host-based bots on their characteristics and behavior is proposed. The system captures network traffic first, to establish normal traffic, then already captured botnet traffic was used to test the system. The system filters out HTTP packets and analyses these packets to further filter out botnet traffic from normal internet traffic. The system was able to detect malicious packets with a False Positive Rate of 0.2 and accuracy of 99.91%.

Keywords: Botnet, Command and Control, Drive-by Downloads, Packet Sniffer

1. INTRODUCTION

The connectivity to the internet is increasing on daily basis as the number of users are increasing due to the increase in the activities perform by both humans and machine on the internet. The development of devices with more internet facilities has necessitate the transfer of enormous amount of data in different formats such as text, audio, image, video and animation. These data may be a proprietary information or confidential, private, public, etc. in nature. The confidentiality and security of these data needs to be guaranteed due to the fact that the internet is accessible by virtually any user who has been able to establish an internet connection. This also means that malicious activities are possible over the internet. In an effort to ensure the protection of these transmitted data over the internet, encryption and authentication techniques have been developed to mitigate the challenges. However, these measures have always been circumvented by malicious attackers over the internet who pretend to be someone they are not or install certain software, program or script on other people devices with the sole intend to steal information or compromise the confidentiality, security and integrity of that system (Kirubavathi & Anitha, 2016).

Botnets are one of the ways confidentiality, integrity and security of data being transmitted over the internet are being compromised and their malicious activities are not easy to stop due to the complex nature of their underlying operations been performed autonomously by a botmaster through a command and control (C & C) communication channel (Acarali et al., 2016). A Bot can be defined as either a computer program, an application or a script that automatically executes a task (Khan et al., 2019). However, in this work only the scripts, computer programs or applications with the sole purpose of creating harm to their hosts are being considered. The interconnections of devices (such as Smart Phones, Laptops, Tablets, Personal Computers, Corporate Computers, Networks, etc.) with these illicit scripts and or computer programs and application installed on them to perform malicious attacks forms what is called botnets and are sometimes referred to as bot-clients/host which are remotely control by a botmaster (Haddadi et al., 2014). It should be noted that the host usually performs malicious activities unknowingly and unwillingly.

The common routes through which the botmaster communicates with the botnets (bot-client/host) are Internet Relay Chat (IRC) channels (Schiller & Binkley, 2007), Hyper-text transfer protocol (HTTP) transmission instruments and chatting Apps (Borgaonkar, 2010), Structured Query Language (SQL) and improved paired quick-fluidity service networks (Sood et al., 2016).

Distributed Denial of Service (DDoS) attacks are the most common types of malicious attacks on servers making them unavailable by flooding them with packets, requests or queries. They are also used for unsolicited mails, stealing sensitive data like passwords of debit/credit cards details, bank details and business transaction details (Kirubavathi & Anitha, 2016; Robinson & Martin, 2017). They are also utilised to corrupt information and interrupt numerous businesses even to the extent of demanding ransom from their victims (Kalita, 2017; Graham, 2015).

There are high concerns on the rising number, nature and complexity of the attacks perpetuated through the DDoS on the internet (Wang et al., 2018; Verisign DDoS Report, 2018;). Several attempts have been recorded to address this phenomenon: Gracia & Pechoucek (2016) use the concept of graphs in depicting the interconnection of botnets on the internet. Network traffic analyses are achieved by using a graph to represent each beginning Internet Protocol (IP) with nodes having an ordered list of terminal IP, terminal port and the protocol. The traffic flows are the edges of the graph. An update is performed on each node and edge at each stage of repetition and if self-looping occurs that is also depicted. Stiborek et al. (2018) propose a technique that uses the similarity index of various system resources as a data representation as they interact with malicious traffic activities. This is similar to multiple instance learning. Classification of the data is then achieved using a clustering algorithm. In (Kirubavathi & Anitha, 2016; Bartos et al., 2016) focus is made by extracting several high impact and important features from the network and analyzing the characteristics of the traffic flow. Bijalwan et al. (2016) in an attempt to detect botnets malicious traffic activities aggregated several classifiers in analyzing the network traffic flow by using a free application programming interface (API) named PCAP (packet capture) for capturing network traffic. Khan et al. (2019) propose an adaptive multilayer model for detecting botnets in a network traffic. The decision tree model is constructed using the bagging random sampling technique. The model gave a result of

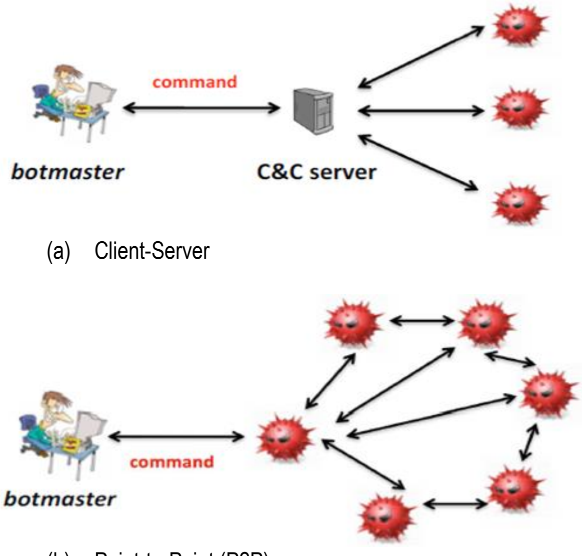
98.7% accuracy in a peer to peer botnets malicious traffic network flow.

Graham (2015) proposed an abstract framework for a botnet protection system for cloud service providers using a non-intrusive detection component. This system would be able to prevent the formation of botnets in virtualized environments that will form the foundation for the Internet of Things. Devices such as routers or switches along the traffic path can generate flow data, based on the traffic that is traversing them. The flow data is sent to a flow collector, which then creates reports and statistics from the flow updates. This process is called flow analysis. The packets sent to a flow collector are not copies of the actual packets in the traffic flow, as in Switch Port Analyzer (SPAN) Port. The flow analysis packets carry statistical data regarding the flow. Flow-based reporting is a good way to understand what traffic is traversing the network. Flow analysis can help to determine traffic statistics overall, but it falls short when you need to analyze a specific conversation in depth. Netflow (version5) does not give any information on the HTTP header. The HTTP header is part of the application layer payload that actually specifies the website and URL that is being requested. Analysis of Port 80 and HTTP Header is important when it comes to the detection of drive by download Bots (Arends, 2017).

This paper proposes a model that detects the presence of a drive by download bot using packet capturing and analysis to secure the confidentiality, integrity and availability of information assets. The system was able to detect the presence of a drive-by download bot, check the network activity and block any communication between the bot and its botmaster. Therefore, HTTP traffic was monitored to detect drive-by downloads.

The organization of the rest of the paper is as follows: A summary of the botnet life cycle and detection is presented in section 2. Section 3 presents the detail methodology, while section 4 gives the results, computations of the performance and discussion of the work. Section 5 concludes the paper and provides direction for future research.

2. BOTNET LIFE CYCLE AND DETECTION



(a) Client-Server

(b) Point-to-Point (P2P)

Figure 1. C&C Botnet Architectures (Bhale, 2016)

Generally, botnet activity can be divided into two phases; the infection phase and the control phase. The infection phase is referred to as the phase in which a clean host is infected by a bot and then becomes a member of a botnet. A fruitful exploitation can prompt the conveyance of a bot to the infected individual host, and such will subvert the host and transform it into an individual from a botnet. The control phase is the phase in which bot-infected hosts are controlled and coordinated by botmaster to launch cyber-attacks (see figure 1.). In the control phase, botmasters rely on command & control (C&C) channels to control and coordinate their bots. Such bots could be used to bombard target servers with network traffic, thereby making it unavailable (see Figure 2.). Different structures could be used by botmaster to build C&C channels (see figure 1.), such as centralized structures and peer-to-peer structures, namely P2P botnets (Zhang, 2012).

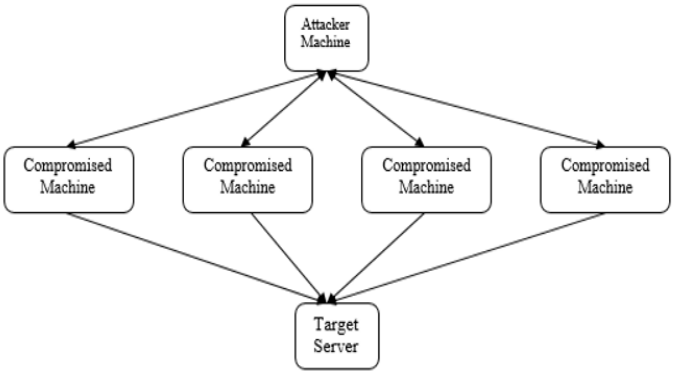


Figure 2: A Botnet used to attack a target server

Depending on the point of deployment, detection methods can generally be classified as Client/Host based or network-based. The Host/Client-based detection approaches are deployed at the client computer targeting bot malware operating at the compromised machine (Masud, 2011). These techniques distinguish the nearness of bot malware by looking at different client level forensics, for instance, application and system logs, active processes, key-logs, usage of the resources and signature of binaries. Furthermore, the client-based detection can also include examination of traffic visible on the computer's network interfaces (See figure 3) (Shin, 2012).

Network-based detection, on the other hand, is deployed on layer three devices (The Network Layer in the Open System Interconnection (OSI) Model) (usually in routers or firewalls). It provides botnet detection by investigating network traffic. This type of method identifies botnets by recognizing network traffic produced by the bots within all three phases of the bots' life-cycle. These techniques are usually denoted to as intrusion detection systems (IDS) or intrusion prevention systems (IPS) (Silva et al., 2013).

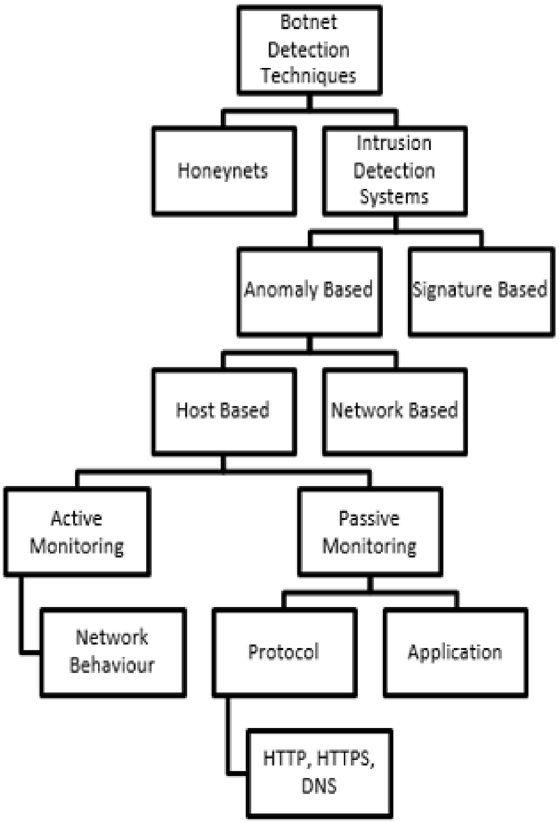


Figure 3: Botnet Detection Techniques (Silva et al., 2013)

Botnets are detected in network traffic by the use of packet sniffers. A packet sniffer is a tool used to capture and analyse packets from the network data, which is used to troubleshoot a network, detect intrusion, control traffic or supervise network contents (Oluwabukola, 2013).

3. METHODOLOGY

This paper focuses on two issues: In the first, a host-based packet sniffer was developed using Java Programming language to capture packets during network traffic flow. Whereas in the second, live botnet traffic is use to test the functionality of the developed packet sniffer (see figure 2) and already captured botnet traffic dataset containing botnet attack in (Garcia et al., 2015) was utilized to run both the existing system (see figure 1) and the proposed system.

The Dataset from (Garcia et al., 2015) is Dataset of botnet traffic that was captured in the CTU University, Czech Republic. The goal of the Dataset was to have a large capture of real botnet traffic mixed with normal traffic and background traffic. The Dataset contains the following features: Time, Source IP, Destination IP, Source Port and Destination Port. The system analyzes each packet, showing the Ethernet Header, IP Header and Payload. The three (3) Datasets used can be described as follows:

Table 1: Overview of the 3 Datasets used

DATASET	CLASS	TOTAL
Dataset1	**Normal**	5923
	Malicious Bot	350
	Total	6273
Dataset2	**Normal**	7704
	Malicious Bot	400
	Total	8104
Dataset3	**Normal**	6945
	Malicious Bot	655
	Total	7600

Dataset1 contains a total of 6273 packets, of which 350 are malicious bots while 5923 are normal network packets; Dataset2 contains a total of 8104 packets, of which 400 are malicious bots while 7704 are normal network packets and finally, Dataset3 contains a total of 7600 packets, of which 655 are malicious bots and 6945 are normal network packets.

The Ethernet Header is Header Information which contains information about Destination MAC Address and Source MAC Address. It also contains the IP version. Two different versions of IP are used in practice today: IPv4 and IPv6

An IP header is header information at the commencement of an IP packet which comprises of details of the following: source IP address, destination IP address, and Protocol Type (which could be TCP, UDP, etc.). The designed application also shows the number of DNS queries performed by the packet.

3.1 Graham System and the Proposed Botnet Detection system (Drive by Download Detection system)

The Proposed (Drive by Download (DD) detection) system uses a different but similar approach to the system in Graham (2015). An additional feature, the Packet Analysis Module was used. The Packet Analysis Module analyses all captured packets and monitor their communication within the network. The packet contents and communication behaviour were analysed to determine if the packet is a bot or not. The designed system analyzes the packets, displays the source MAC address, and destination MAC address and also show the DNS portion of the packet which will further be used to classify a packet as a botnet or not.

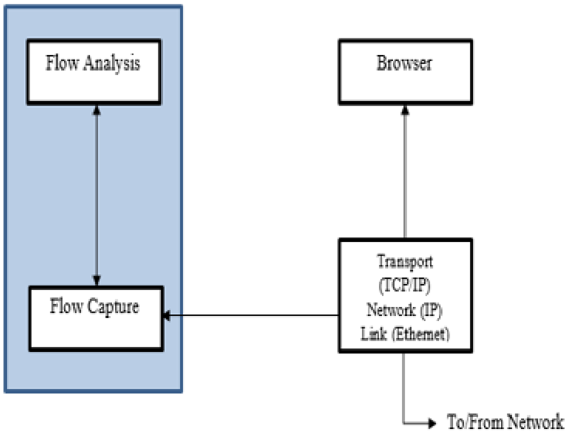


Figure 4: The Existing System (Graham, 2015)

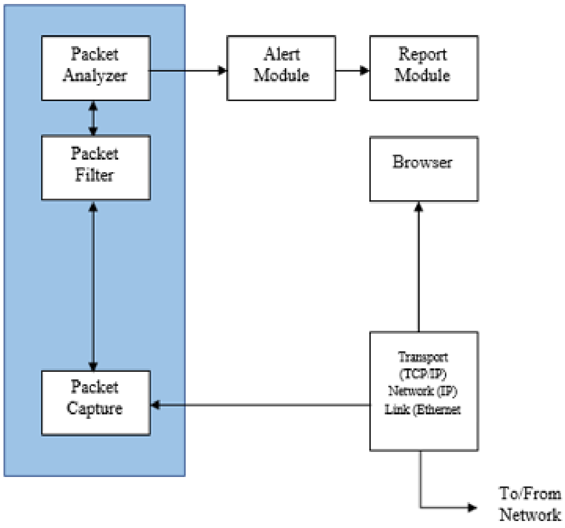


Figure 5: DD-detection system

The approach takes the following tune: traffic dispatch, packet capturing, packet filtering, packet analysis, alerts and reports.

3.2 Traffic Dispatch

Bots have unique characteristics on networks and any bot gotten from a drive by download communicates on HTTP protocol. Once a network is established, communication occurs between the host and various destination servers. According to Arends (2017) it has been established that Drive by Download Packets communicate on Port 80. Therefore, HTTP traffic was monitored to detect drive-by downloads. This was achieved by capturing all network traffic to the system being monitored.

3.3 Packet Capturing

The Capturing Module intercepts the network traffic to and from the host computer. Packets are extracted and stored for future analysis.

3.4 Packet Filtering

The Packet Filtering Module filters out all Hyper Text Transfer Protocol (HTTP) which comprises of Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) traffic for analysis. The packet sniffer captured packets, size of the packet, the source and destination Internet Protocol (IP) and Medium Access Control (MAC) addresses which are party to the packet transfer.

3.5 Packet Analysis

A packet analyzer is a computer program that captures and logs network traffic. As data streams flow through the network, the packet analyzer or sniffer captures each packet and decodes its raw data, showing the values of various fields in the packet, and analyzes its content.

This approach focused on detecting the communication channel between the bot and the botmaster. This step occurs right after infection, so the compromised machine was detected by considering the following parameters:

i. A ***new* connection** represents one whose destination has not been contacted before since the initiation of the process.

ii. When a new connection is built, it is said to be an **intermediate** point.

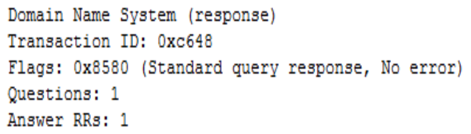
iii. If an intermediate point is reconnected, the connection is updated to be a **sporadic** one.

iv. A connection is said to be a persistent one if an intermediate point or a sporadic connection lasts more than 30 seconds.

A botnet connection is said to be identified when Persistent and Sporadic Connections are flagged.

The analysis and detection system uses the essential properties of botnets and additional functionality of analyzing Domain Name System (DNS) queries and the responses.

The Domain Name System is a tiered and dispersed naming convention for computers, services, or other resources connected to the Internet or a private network. It links various information with domain names allocated to each of the partaking systems. It translates URLs to IP addresses in order for computers to communicate with each other. The DNS response showed in the analysis contains four (4) parts, which are: Transaction ID, Flags, Questions and Answer of Resource Records (RR). A standard DNS response gives, in most cases 1 or 2 answers from the RRs

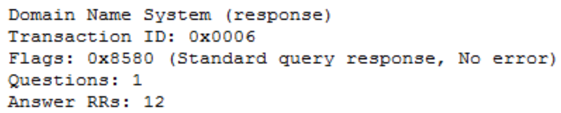


```
Domain Name System (response)
Transaction ID: 0xc648
Flags: 0x8580 (Standard query response, No error)
Questions: 1
Answer RRs: 1
```

(See figure 5).

Figure 5: Benign DNS Query

In the case of Botnet traffic, the DNS response is usually a big number (See figure 6). So, whenever the result of a DNS query is more than 5, it prompts the system to mark that packet as suspicious and when that same packet appears over 10 times in 2.5 seconds (Chen, 2015)



```
Domain Name System (response)
Transaction ID: 0x0006
Flags: 0x8580 (Standard query response, No error)
Questions: 1
Answer RRs: 12
```

Figure 6: Botnet DNS Query

3.6 Alerts

When a packet tries to establish a connection with any server, it makes a Domain Name System (DNS) query to find out what server owns that IP address. For a normal server, the DNS answer usually ranges from 1 to 4, but whenever the answer is more than 5, that connection is also flagged as suspicious. In the case of Botnet traffic, the DNS response is usually a big number. So, whenever the result of a DNS query is more than 5, it prompts the system to mark that packet as suspicious and when that same packet appears over 10 times in 2.5 seconds (Chen, 2018). When a suspicious communication is detected, the user was alerted and the communication severed while the suspected bot was isolated. This system was able to distinguish between benign and botnet traffic and give a low false positive rate.

3.7 Reports

The Report Module considered the following: Number of captured packets within a session, start time and end time, time elapsed, number of suspicious packets and number of connections disengaged.

4. RESULTS, PERFORMANCE AND DISCUSSION

Experiments were conducted using three data sets as obtained in (Garcia et al. 2015) and the results obtained are as tabulated below.

4.1 RESULTS

Table 2: Experimental Results of the Existing system

DATASET	CLASS	CLASSIFICATION		TOTAL
		CORRECT	INCORRECT	
Dataset1	Normal	5523	400	5923
	Malicious Bot	0	350	350
	Total	5563	750	6273
Dataset2	Normal	7154	550	7704
	Malicious Bot	0	400	400
	Total	7154	950	8104
Dataset3	Normal	6555	390	6945
	Malicious Bot	5	650	655
	Total	6560	1040	7600

Table 2 presents the results of the experiment using three (3) datasets with the existing system. In Dataset 1, the existing system correctly classify 5523 packets as normal and incorrectly classify 400 packets as normal packets. It also incorrectly classified 350 packets as malicious bots. In Dataset 2, 7154 packets are correctly classified as normal and 550 packets incorrectly classified. 400 packets are incorrectly classified as malicious bots. In Dataset 3, 6555 packets are correctly classified as normal and 390 incorrectly classified as normal. However, 5 packets are correctly classified as malicious and 650 packets incorrectly classified as malicious.

Table 3: Experimental Results of the DD-detection system

Dataset	Class	Classification		Total
		Correct	Incorrect	
Dataset1	Normal	5563	5	5568
	Malicious Bot	0	5	5
	Total	5565	10	5573
Dataset2	Normal	7153	5	7158
	Malicious Bot	3	4	7
	Total	7156	9	7165
Dataset3	Normal	6555	0	6555
	Malicious Bot	0	6	6
	Total	6555	6	6561

4.2 Performance

The experimental results obtained from both the existing and proposed system are evaluated for their performance using the following performance metric:

True Positive (TP) means correctly classifying a normal packet

False Positive (FP) means wrongly classifying a normal packet as malicious

True Negative (TN) means correctly classifying a malicious packet

False Negative (FN) means incorrectly classifying a malicious packet as a normal packet

False Positive Rate (FPR): The probability of classifying normal packets as malicious packets

$$FPR = \frac{FP}{FP + TN} \qquad (1)$$

Table 4: False Positive Rate Results

Dataset	Number of Bots	Detected	FP Rate
Live Traffic	0	0	0
Dataset1	4	5	0.2
Dataset2	7	5	0
Dataset3	5	6	0.2

False Positive (FP) Rate is a test result which incorrectly shows that a particular condition or trait is present (See equation 2). Table 3 shows the number of malicious bots detected and the False Positive Rates. From the results, it can be seen that even though the system has some False Positives, the rate is considerably low.

Accuracy: this is the general statistics of correctly classified instances, whether normal or malicious.

$$Accuracy(\%) = \frac{TP + TN}{TP + FP + TN + FN} \times 100 \qquad (2)$$

Based on the results of the test, it can be seen that the DD-detection system provides a mechanism for successfully detecting bots from network traffic with an average accuracy of 99.91% as computed using equation 2.

Table 5: Comparison of the Results of the Two Systems:

Dataset	Existing system (%accuracy)	DD-detection system (%accuracy)
Dataset1	88.68	99.91
Dataset2	88.28	99.90
Dataset3	86.32	99.91

Table 5 shows a summary of the experiment carried out with the propose system in comparison with the existing system, In Datasets, the existing system had an accuracy 88.68%, 88.28% and 86.32% respectively while the DD-detection system had an accuracy of 99.91%, 99.90% and 99.91% respectively. This shows that the DD-detection system gives a more accurate result than the system in (Graham, 2015).

5. Conclusion and Future Research Direction

The bot can exhibit various evasion techniques on the machine and only becomes fully active under certain circumstances thereby making it difficult to be detected on a network. In order for bots to start carrying out their objectives, they first need a communication channel back to the attacker. Packet Capture and Analysis is a very effective technique for the detection of bot within a network. The objective of this work is to design and implement a Packet Capturing and Analysis tool that could capture network traffic, filter out HTTP traffic and classify packet communication as malicious or normal and to give a low false positive rate. After the implementation of the system, it was seen that the system gave a low false positive rate of 0.2 and an average accuracy of 99.91%. This showed the potential of the propose system in boosting the effectiveness of the existing detection system by detecting a large number of stealthy drive-by download attacks that were not initially detected by the detection system in Graham (2015) which could be critical in the long run. The system can be further improved to give no false positives if possible. Any malware detection system should be able to accurately detect malware to protect the confidentiality, integrity and availability of information.

REFERENCES

Acarali, D., Rajarajan, M., Komninos, N. and Herwono, I. (2016). Survey of approaches and features for the identification of http-based botnet traffic, *Journal of Network and Computer Applications* 76, 1-15. URL: http://www.sciencedirect.com/science/article/pii/S1084804516302363

Arends, R., Austein, R., Larson, M., Massey, D., & Rose, S. (2017). Resource Records for the DNS *Security Extensions. RFC, 4034*, 1-29.

Bartos, K., Sofka, M. and Franc, V. (2016). Optimized invariant representation of network traffic for detecting unseen malware variants, *25th USENIX Security Symposium (USENIX Security 16)*, USENIX Association, Austin, TX, 807-822. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/bartos

Bhale, K. M. (2016). Botnet Detection Tools and Techniques. *Hyderabad: Centre for Cyber Security.*

Bijalwan, A., Chand, N., Pilli, E. S. and Krishna, C. R. (2016). Botnet analysis using ensemble classifier, *Perspectives in Science* 8, 502-504. Recent Trends in Engineering and Material Sciences. URL: http://www.sciencedirect.com/science/article/pii/S2213020916301422

Borgaonkar, R. (2010), An analysis of the asprox botnet, *2010 Fourth International Conference on Emerging Security Information, Systems and Technologies*, 148-153.

Chen, K., Jiang, J., Huang, P., Chu, H., Lei, C., & Chen, W. (2018). Identifying MMORPG Bots: A Traffic Analysis Approach. *EURASIP Journal on Advances in Signal Processing, 2015*, 1-22.

Garcia, S. (2015). CTU Malware Capture Botnet-3. *CVUT University*.

Garcia, S. and Pechoucek, M. (2016), Detecting the behavioral relationships of malware connections, Proceedings of the 1st International Workshop on AI for Privacy and Security', PrAISe '16, ACM, New York, NY, USA, 8, 1-8:5. URL: http://doi.acm.org/10.1145/2970030.2970038

Graham, M., Winckles, A., & Sanchez-Velazquez, E. (2015). Botnet Detection Within Cloud Service Provider Networks Using Flow Protocols. *2015 IEEE 13th International Conference on Industrial Informatics (INDIN)*, 1614-1619.

Haddadi, F., Runkel, D., Zincir-Heywood, A. N. and Heywood, M. I. (2014). On botnet behaviour analysis using gp and C4.5, Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation', GECCO Comp '14, ACM, New York, NY, USA, 1253-1260. URL: http://doi.acm.org/10.1145/2598394.2605435.

Kalita, E. (2017). WannaCry Ransomware Attack: Protect Yourself from WannaCry Ransomware Cyber Risk and Cyber War, independently published.

Khan, R. U., Zhang, X., Kumar, R., Sharif, A., Golilarz, N. A., and Alazab, M. (2019). An Adaptive Multilayer Botnet Detection Technique using Machine Learning Classifiers. *Applied Sciences,* 9 (2375), 1-22.

Kirubavathi, G. and Anitha, R. (2016). Botnet detection via mining of traffic flow characteristics, *Computers & Electrical Engineering,* 50, 91-101. URL: http://www.sciencedirect.com/science/article/pii/S0045790616000148

Masud, M., Khan, L., & Thuraisingham, B.M. (2011). Data Mining Tools for Malware Detection. *Taylor & Francis Group*.

Oluwabukola, O., Oludele, A., Ogbonna, A.C., Chigozirim, A., & Amarachi, A. (2013). A Packet Sniffer (PSniffer) Application for Network Security in Java. *Issues in Information Science and Information Technology*.

Robinson, N. and Martin, K. (2017). Distributed denial of government: The Estonian data embassy initiative, *Network*

Security 2017(9), 13-16. URL: http://www.sciencedirect.com/science/article/pii/S1353485817301149

Schiller, C. and Binkley, J. (2007), Botnets: The Killer Web Applications, Syngress Publishing.

Shin, S., Xu, Z., & Gu, G. (2012). EFFORT: Efficient and effective bot malware detection. *2012 Proceedings IEEE INFOCOM*, 2846-2850.

Silva, S.S., Silva, R.M., Pinto, R.C., & Salles, R.M. (2013). Botnets: A survey. *Computer Networks, 5*7, 378-403.

Sood, A. K., Zeadally, S. and Enbody, R. J. (2016). An empirical study of http-based financial botnets, *IEEE Transactions on Dependable and Secure Computing,*13(2), 236-251.

Stiborek, J., Pevn_y, T. and Reh_ak, M. (2018). Multiple instance learning for malware classification, *Expert Systems with Applications* 93, 346 - 357. URL: http://www.sciencedirect.com/science/article/pii/S0957417417307170

Verisign DDoS Report (2018). Q2 2018 ddos trends report: 52 percent of attacks employed multiple attack types https://blog.verisign.com/security/ddos-protection/ q2-2018-ddos-trends-report-52-percent-of-attacks-employed-multiple-attack-types/.

Wang, A., Chang, W., Chen, S. and Mohaisen, A. (2018). Delving into internet ddos attacks by botnets:
Characterization and analysis, *IEEE/ACM Trans. Netw* 26(6), 2843-2855. URL: https://doi.org/10.1109/TNET.2018.2874896

Zhang, L., Yu, S., Wu, D., & Watters, P. (2011). A Survey on Latest Botnet Attack and Defense. *2011 IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications*, 53-60.