Detecting botnet using traffic behaviour analysis and extraction of effective flow features

Sanaz Feizi and Hamidreza Ghaffari*

Department of Computer Engineering,
Islamic Azad University,
Ferdows Branch,
Ferdows, Iran
Email: St.s.feizi@ferdowsiau.ac.ir
Email: hghaffari@ferdowsiau.ac.ir
*Corresponding author

Abstract: Botnets is one of the most serious attacks that cause irreversible damage to systems and networks, and it is important to detect and prevent botnets as attacks using them are constantly occurring. In the paper, the botnet detection method proposed so far was analysed, and based on the analysis, botnet detection method and effective flow features extraction method were proposed. The proposed method in this paper is able to detect and identify them not only during the attack phase, also in the C&C phase of botnet life cycle before they can attack the system or network. The proposed model is based on traffic behaviour analysis to detect botnet-related command and control traffic designed by using classification through selecting effective network flow-based features, which has the advantage that it can also detect encrypted traffic. Accordingly, it is thought that the paper will be an important help for various studies in the method of detecting botnets.

Keywords: botnet detection; network flow; traffic behaviour analysis; random forest; intrusion detection.

Reference to this paper should be made as follows: Feizi, S. and Ghaffari, H. (2022) 'Detecting botnet using traffic behaviour analysis and extraction of effective flow features', *Int. J. Internet Technology and Secured Transactions*, Vol. 12, No. 1, pp.49–60.

Biographical notes: Sanaz Feizi received her MSc in Software Engineering from the Islamic Azad University of Shabestar, Iran. She is currently pursuing her PhD in Software Engineering from the Islamic Azad University of Ferdows, Iran. Her research interests are traffic behaviour analysis, intrusion detection, network security and machine learning.

Hamidreza Ghaffari received his PhD in Intelligent Computing from the Ferdowsi University of Mashhad, Iran. He joined the Islamic Azad University, Ferdows, Iran. His main areas of research interest are data mining, network security, intrusion detection, machine learning and deep learning. He is an academic member of the Departments of Computer and is currently an Assistant Professor.

1 Introduction

Nowadays, internet malware has grown dramatically and is more organised and more profitable than ever before. That is, botnets are a major threat to internet users. These networks are created by infecting a large number of computers with malware (i.e., malicious software) by means of operating system vulnerabilities, USB drives or malicious websites. Once a victim's computer is infected, the botnet software allows an attacker (also known as botmaster) to take control and carry out malicious activities (Cid-Fuentes et al., 2018). Such infected hosts are usually called zombie computers, and they are controlled to infect computers on the internet with the botnet malware itself, via, e.g., e-mail messages, websites and social networking services (SNSs) (Kudo et al., 2018). The many botnets attacks are launching distributed denial of service (DDoS) attacks (Osanaiye et al., 2016), phishing and ransomware distribution (Baldwin and Dehghantanha, 2018), identity theft, or using the powerful computational resources of the network on a wide range of malicious distributed tasks (Kiwia et al., 2017). Due to the automaticity of the attacks carried out by these systems in the network, which are carried out in secret and without the intervention of users and due to their performance similarity to robots, they are referred to as bots (Paxton, 2011), the botnet still represents one of the biggest challenges that security researchers and analysts must face (HaddadPajouh et al., 2018). Due to their irrecoverable and destructive effects, they should be detected before they can impact on the respective system or network. According to the gathered statistics (Silva et al., 2013), about 16% to 25% of computers connected to the internet are members of botnets. The White House's Council of Economic Advisers (CEA, 2018) estimated that malicious cyber activity cost the US economy between $57 billion and $109 billion in 2016.

A botnet needs a communication infrastructure that botnet manager can transmit its commands to the bots and receive their responses. The most significant element of a botnet is the communication architecture. Bots interact through communication channels. Internet relay chat (IRC) was the most common communication scheme among traditional botnets. The botmaster uses command and control (C&C) channels deployed to communicate and control the robots. C&C channels can be centralised or decentralised. New generation bots have started communicating through peer-to-peer (P2P) networks and protocols. They use distributed C&C servers that made them more difficult to detect comparing to centralised (IRC and HTTP)-based botnet. Botmasters are aimed at systems at any points of the internet which can be attacked by the botnet. The systems in the network are infected through security holes and vulnerable points of the system. Botnets can lead to reduce network speed, high bandwidth or providing the information needed to carry out attacks (Paxton, 2011).

The architecture and structure of botnet is determined according to the communication between bot and botmaster in the network, and also, based on the selection of the communication protocol by the botmaster so as to communicate with the bots. A feature which distinguishes botnets from other internet threats is the way in which bot and botmaster establish communications with each other through the C&C communication channel (Correia et al., 2012). Botnets which have been centralised from one topology include IRC and HTTP protocols; they use a central point for transmitting comments and messages by the botmaster. The same central point is their weak point and the cause of their failures. P2P topology uses P2P protocols as C&C channel which overcomes the previous weak point (Cook et al., 2005).

The first botnets were developed based on IRC communication protocol; this structure is a centralised architecture. In this architecture, bot and botmasters communicate with each other by transmitting messages with a chat structure. The organised attacks based on this protocol (Silva et al., 2013) are allowed to establish instantaneous communications among botnets in a large scope. The IRC traffic is uncommon on the internet, and this is considered as a major problem (Zhao et al., 2013). HTTP protocol uses the dynamic feature of selecting the name of DNS protocol range for preventing their detection (Silva et al., 2013). However, due to its centralised communication structure with bots in the botnet and high traffic load which is created on them, it provides the discovery opportunity through traffic analysis. In centralised botnets, when the centre is unable to function, bots try to connect to inaccessible servers while the entire botnet cannot be used. The main idea for using a set of distributed peers is to remove the failure unit point in the centralised botnets. Consequently, P2P protocol is used for the network communication channel (Jiang and Shao, 2012).

From points weak, many existing botnet detection techniques this is that focusing on identifying botnets based on existing signatures of attacks or before knowledge. Whereas, this model has not need to before knowledge. Some techniques detecting bot activity during the attack phase. The proposed method is able to quickly detect the bot at the C&C stage and the attack stage. Some technique cannot detect encrypted packets. Some methods inspection group behaviour of bots but if there is one bot in network cannot detection that. In some techniques where all traffic is inspected, the detection is done slowly, but it is done quickly in the divided traffic in time windows. Through analysing botnet behaviour and normal traffic, this paper proposed a model for detecting traffic of botnet. Traffic behaviour in layer 4, the examined (TCP/UDP protocols) and the proposed method in this paper are not limited to the architecture of the bots with IRC and HTTP protocols. Also, bots with P2P architecture were tested and highly accurate and precise results were obtained. By dividing traffic into time windows, we were able to quickly detect of bots. In each time window, flows were determined and the effective features of flows in each time window were extracted using chi-square method. Then, machine learning was used for classification.

The rest of the paper is organised as follows: related works on botnet detection are briefly reviewed in Section 2. The proposed model is reported and discussed in Section 3. The experimental evaluations and observations of the results are given in Section 4. Comparison and discussion is proposed in Section 5. Conclusions and directions for further research are given in Section 6.

2 Related works

By investigating the studies conducted on botnet, we can categorise them into two dimensions: studies concerned with botnet detection in the network and the studies involved with the ways for defence against botnets. In general, most of the studies on this field are concerned with botnet detection. That is, the significance of botnet detection is higher than defence against them since, at first, botnet management depends on its detection. Botnet detection based on signatures is simplest detection method because packets are compared with a set of existing signatures and it cannot detect complex

communication modes and conditions. Botnet detection techniques based on monitoring and the analysis of network traffic are useful; they do not depend on payload content and there is no concern about personal privacy preservation. Thus, they can work with the communication protocols of encrypted network. The obtained results (Wang and Ramsbrock, 2009) indicate that there are some distinct features among different types of botnets which allow for the detection of their activities. In fact, there are enough similarities within each botnet which distinguish its behaviour from normal traffic. Currently, one of the most popular methods in botnet detection is to extract features at the host or network level for modelling a botnet.

The network traffic analysis was used as an effective method for detecting C&C sessions in Lu et al. (2017) which is limited to investigating DDoS attacks. Flow analysis is based on different behaviour of traffic between C&C session and normal session. A session is a basic unit that based on all the features are extracted from log file. A feature vector includes selective features and a random forest algorithm for creating classification was used.

A detection scheme was proposed in Chen and Lin (2015) for identifying botnets during C&C connection stage and before botnet attack. It focuses on IRC-based botnet problem; it analyses and investigates network traffic. Then, it utilises two anomalous behaviours, homogeneous response and group activity, it assigns an abnormality score to the bots. Next, based on the given scores, it detects suspicious bots before an attack can be setup. That is, botnet can be detected through abnormal traffic. Botnet activities via similarity criterion and periodic features identified and it detects botnet according to anomaly score. For improving detection rate, two-level correlation communication system uses a set of hosts with identical anomaly behaviour. This method distinguishes the traffic of malicious network created by infected hosts from normal IRC clients. In fact, botnet detection based on anomaly score identifies the activities of botnets by using the criteria of similarity and the periodic features of botnets. So, investigation group behaviour cannot detect while there is just only one bot in the network. Since the payload is investigated, it cannot detect encrypted packets.

In Vormayr et al. (2017) was presented an in-depth analysis of all network communication aspects in botnet establishment and operation. Examine botnet topology, protocols, and analyse a large set of very different and highly sophisticated existing botnets from a network communication perspective. Based on analysis, using standardised unified modelling language sequence diagrams a novel taxonomy introduces of generalised communication patterns for botnet communication. Generalised communication patterns provide a useful basis for the development of sophisticated network-based botnet detection mechanisms and can help for building protocol and topology independent network-based detectors. But maybe, new bots using from different communication patterns. But new bots may use different communication patterns.

A real-time collaborative network forensic (RCNF) scheme was suggested in Moustafa and Slay (2017) that can monitor and investigate cyber intrusions. The scheme includes three components of capturing and storing network data, selecting important network features using chi-square method and simple random sampling (SRS) for selecting relevant observation and investigating abnormal events using a new technique called correntropy-variation. In SRS stage remove repeated instances probably to be not useful because removable packets are need.

A general botnet system was proposed in Yahyazadeh and Abadi (2015) which considers malicious activities and the history of group activities. This system, known as BotGrab, detects those suspicious hosts which participate in some group activities. Then, according to their history of participation in those activities, it measures a negative reputation score for the hosts. The hosts detected and identified in this way by this system are introduced as bots. These bots have negative reputation scores or, despite their low negative reputation scores, they have done some malicious activities. The proponents of BotGrab claimed that it can detect different types of botnets. According to the studies which investigated group behaviour among a set of infected machines, in a real network, there might be very few infected bot machines. The assumption that several bot machines present group activities may not executable in real conditions.

A flow-based botnet discovery method was proposed in Pektaş and Acarman (2018) and was deployed deep neural network to classify network traffic whether normal or botnet. In feature extraction stage, flow features are extracted from network traffic and are transformed into a multi-dimensional feature vector. It splits the network traffic between endpoints and represents these flows as a graph toward modelling the interaction and the behaviour of connection. This graph-based model of host interaction is used to extract the feature set. It models network traffic traces between communication endpoints by representing the whole traffic in a graph. Then, the deep neural learning architecture was designed toward botnet detection that consisted of four major parts: embedding, convolution, LSTM and fully connected networks.

Netflow data was used in Amini et al. (2019) to detecting centralised and HTTP botnets. In the centralised botnets, the command-and-control channel and the sent data were considered as the weaknesses of the botnets. The hierarchical clustering, X-means clustering, and rule-based classification were used to discover similar data in a fixed period of time. Hierarchical clustering improved the speed and accuracy rate in the process of separating the flows. The X-means algorithm led to the highest cohesion inside the clusters and the maximum distance between clusters by choosing optimal K. Using rule-based classification, each cluster with the similar flow is placed in a bot cluster, a semi-bot cluster or a normal cluster. The use of several algorithms increases the computational complexity.

For detecting botnet, traffic behaviour analysis was done through classifying network traffic behaviour using of machine learning in Zhao et al. (2013). The possibility of detecting botnet activity without viewing a complete network flow and by means of classifying behaviour based on time distances were examined. For detecting them, the presence of botnets during attack phase and C&C phase are taken into consideration. The network behaviour of a botnet at the TCP/UDP level was investigated. Inside, time windows are separated and a set of extracted features are used for classifying malicious or non-malicious traffic. The considered features are the ones which were obtained at time windows. However, it should be noted that the method proposed in this paper the features of each flow in time windows were taken into consideration.

The botnet detection methodologies presented in the articles mentioned in this section are summarised in Table 1. The items mentioned in Table 1 include: the technique used to detect botnet, dataset used and type of botnet (C&C structure).

Tracked changes

Table 1 A summary of the methodologies of the articles mentioned in the related works section

Author	*Technique*	*Dataset*	*Type of botnet*
Approach (Zhao et al., 2013)	Reduced error pruning algorithm (REPTree)	ISOT	IRC, P2P, HTTP
Approach (Lu et al., 2017)	Flow analysis, classification	CCC dataset	DDoS attacks
Approach (Chen and Lin, 2015)	Investigate group behaviour	Testbed network environments	IRC
Approach (Vormayr et al., 2017)	In-depth analysis of network communication patterns, UML	A diverse list of botnets	IRC, P2P, HTTP
Approach (Moustafa and Slay, 2017)	Correntropy-variation	UNSW-NB15	Nine types of attacks: fuzzers, analysis, backdoors, DoS, exploits, generic, reconnaissance, shellcode, worms
Approach (Yahyazadeh and Abadi, 2015)	Group activities, negative reputation score	Testbed network	IRC, HTTP, P2P
Approach (Amini et al., 2019)	Hierarchical clustering, X-means clustering and rule-based classification	CTU dataset, Alexa dataset	IRC, HTTP
Approach (Pektaş and Acarman, 2018)	Deep neural network (embedding, convolution, LSTM and fully connected networks)	CTU-13, ISOT	IRC, HTTP, P2P

3 The proposed method

The proposed method is based on traffic behaviour analysis which includes three stages:

1 pre-processing

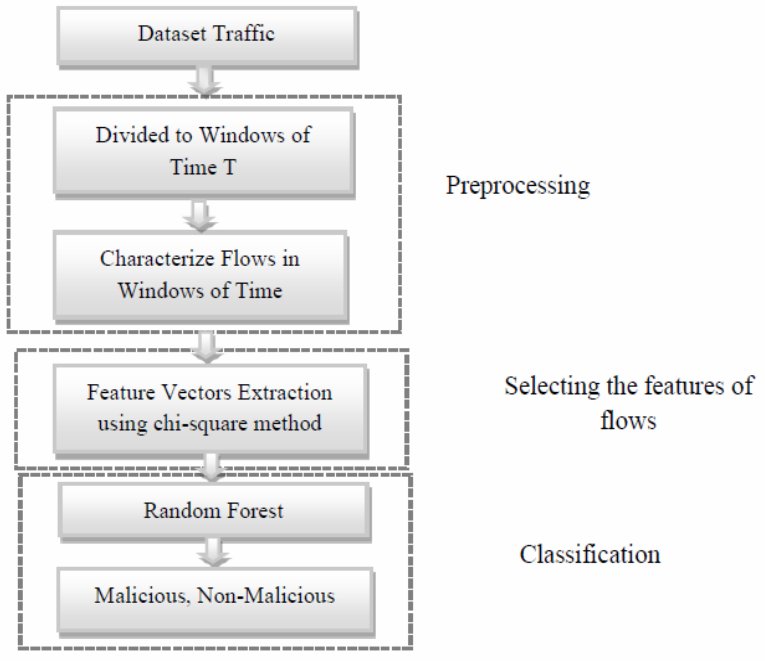
2 selecting the features of flows in the time window

3 classification.

Traffic behaviour is analysed at layer 4 and in TCP/UDP protocols. At first, the traffic is divided into time windows; time windows were practically selected in such a way that they lead to higher speed and accuracy. Then, the flows were determined in each time window and features were selected based on the analysis of the traffic behaviour of flows in certain time windows and chi-square method was used for select effective features. Each flow was defined as a set of packets which are exchanged between two unique IP addresses by means of a pair of ports and by using several layer 4-protocols (Sperotto et al., 2010). Next, for machine learning, random forest algorithm was used for

classification. Dividing traffic into time windows which cause to quick detection and using chi-square method lead to select effective features. The proposed method in this paper is also able to detect them at control and command stage. The related flowchart is given in Figure 1.

Figure 1 Model of the proposed approach



3.1 Pre-processing

In this stage, number of features is obtained by using TCP and UDP headers. Then, traffic into time windows divided that size of time windows were experimentally and practically determined which leads to good results. Within time windows, flows have been specified. At the further stage, features of flows have been extracted.

3.2 Selecting the features of flows in each time window

Features such as the destination and destination IP address, ports of a flow and the size of packets and protocols were directly extracted from TCP/UDP headers at the pre-processing stage. Other features of flows such as the duration of each flow or the average length of exchanged packets in each flow require other measurements. In this stage, attribute vectors including the features of flows are extracted from flows which were produced in time windows in the pre-processing stage. Indeed, these attribute vectors are a set of attributes which have been obtained based on the analysis of the behaviour of botnets in the flows. Then, flows have been labelled into two class, bot or normal for selective more effective features using from chi-square method, a simple and general algorithm.

Tracked changes

The chi-square test is a statistical test of independence to determine the dependency of two variables.

There are feature variables and class variable. The chi-square statistic is calculated between each feature variable and the class variable to determine the relationship between the feature variables and the class variable. If they are dependent, then the feature variable is effective. The chi-square statistic (Yin et al., 2015) calculation formula is:

$$x^2(t.c_i) = \frac{n \times (ad - bc)^2}{(a+c) \times (b+d) \times (a+b) \times (c+d)} \tag{1}$$

$$x^2_{avg}(t) = \sum_{i=1}^{M} P(c_i)\, x^2(t.c_i) \tag{2}$$

The four situations used in the formula to calculate the relative independence of t and c_i are described in Table 2.

Table 2 situations used to calculate the relative independence of t and c_i

Situations used	*Description*
a	No. of times feature t and class label c_i co-occurs
b	No. of times t appears without c_i
c	No. of times c appears without c_i
d	No. of times neither c_i nor 't' appears
n	Total number of records
t	Feature

Tracked changes

Then, in equation (2), the average is calculated. As a result, the top features with the highest chi-square scores are selected. The selected features were considered in Table 3.

Table 3 Selected features

Features	*Explanation*
Ipsrc	Source ip address for each flow
Ipdst	Destination ip address for each flow
Srcport	Source port address for each flow
Dstport	Destination port address for each flow
Protocol	Protocol of transport layer
Flow duration	Duration for each flow
Apl	Average payload packets length for each flow
Sdpsize	Standard deviation size of packets in each flow
Nbytes	Number of bytes in each flow
Total packets	Total packets for each flow

3.3 Classification using random forest

After the effective features of flows of the previous phase were selected, random forest was used as machine learning algorithm for producing classifier that classifies malicious and non-malicious flows. Random forest is regarded as a highly efficient and accurate classifying technique which is capable of fast detection in real time. The main reasons for selecting random forest are as follows:

- This method can prevent high-dimension over-filtering issue at the time of investigation.
- Through measuring OOB rate, random forest can present significant ranking of features which can contribute to the understanding of correlation and relates features to the classification issue (Lu et al., 2017).

Random forest is a combination of classifiers which can deal with problems and issues with high dimension. Also, it can measure the significance of features and contributes to finding key features in each flow. Random forest is a hybrid learning method for classifying, regression and other tasks. It is a combination of a large number of decision trees which selects a random feature. Each decision tree has a uniform input unit and has a unique classification result. A random forest decides about the result of final classification based on the significance of the role of the individual decision tree output. Each tree is similar to only one selector and each class is a candidate. The candidate with the most voters is the winner of the selection.

4 Results of the experimental evaluations

In this section, the respective dataset used in this study is firstly described. Then, the experimental environment, the software used in this study is introduced and the degree of precision and accuracy by using random forest classification are discussed.

4.1 Dataset

ISCX (http://www.unb.ca/cic/datasets/botnet.html) was used for testing and training. This dataset was produced in a physical environment by using real devices which create real traffic and imitate user behaviour. It includes malicious and non-malicious traffics and used bots shown in Table 4.

4.2 Implementation

In this paper, at first, those features, directly obtained from TCP/UDP headers, were extracted by using TSHARK. The remaining features of flows which need other measurements were carried out in SQL Server and features of the flows were selected in determined time windows. Next, selecting important features using chi-square method and detection phase was implemented in R software which included a random forest classification model by using random forest package in R (Breiman et al., 2015). The size of time window was experimentally assumed to be 600 s and ten effective features were used for each flow. After pre-processing and the selection of features phases for each

flow and labelling data classes, a set of non-malicious and malicious data attribute vectors are provided for training classifier so that it can identify malicious and non-malicious traffics in the detection stage. Only 10% of the entire flows were randomly selected for training by the proposed method. The obtained results for the proposed method are given in Table 5 which reveals the detection accuracy. Detection rate is the proportion of total number of correctly predicted instances over total number of instances.

Table 4 Used bots and their types

Botnet name	*Type*
Neris	IRC
RBot	IRC
Menti	IRC
Sogou	HTTP
Murlo	IRC
Virut	HTTP
Zeus	P2P
TBot	IRC
Zero access	P2P
Weasel	P2P
Smoke bot	P2P
Osx_trojan	P2P
Black hole	P2P
IRCbot	P2P

Table 5 Results of the proposed method and its comparison

	Detection rate
The proposed approach	98.7%
Approach (Zhao et al., 2013)	98%
Approach (Amini et al., 2019)	98.2%
Approach (Lu et al., 2017)	96.3%

5 Discussion

To evaluate the performance of proposed method with different botnet detection approaches, the same dataset needs to be evaluated. Because using different datasets may give different results. As a result, comparison with other methods will not be accurate. In this paper, the ISCX dataset is used to compare the proposed method with other methods. This dataset is a combination of malicious and non-malicious traffic. The results are shown in Table 5. In Lu et al. (2017), used 55-dimension feature vector for detecting DDoS attacks and Zhao et al. (2013) used 13 features. In Amini et al. (2019) was focused only on centralised and HTTP botnets. Eight features were selected. The use of several

algorithms increases the computational complexity. The comparison results reveal that the proposed approach achieves better botnet detection accuracy.

6 Conclusions

As discussed earlier, in detecting botnets which investigate the contents of TCP and UDP packets based on suspicious signatures, payload analysis methods need to parse a large number of data packets; hence, they are slow. Also, in the cases in which bots use encryption, payload analysis methods cannot function satisfactorily and the investigation of the packet content is a violation of privacy. In contrast, traffic behaviour analysis does not depend on the content of packets. Hence, it can be easily applied on the encrypted packets. In the traffic analysis method, for distinguishing normal traffic from malicious traffic, a set of attributes on traffic behaviour is investigated. Since bots within a botnet usually present traffic behaviour and uniform communication, the set of attributes is used for making distinctions between these types of traffic.

The proposed method should be noted that it is no limited to architecture and protocol. It is applied on centralised bots including IRC and HTTP and also on P2P bots. By selecting time windows and selecting effective features of each flow in each time window, the proposed method can be used for both offline and online detection.

It should be highlighted that the selection of effective features based on flow led to a better detection of bots and random forests were used for classification. This classification model is able to handle high-dimension classification issues and indicates the significance of features. As a direction for further research, since the behaviour of bots changes as the time passes, devising an adaptive system is regarded as challenge which can be addressed in future studies.

References

Amini, P., Azmi, R. and Araghizadeh, M.A. (2019) 'Analysis of network traffic flows for centralized botnet detection', *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, Vol. 11, No. 2, pp.7–17.

Baldwin, J. and Dehghantanha, A. (2018) 'Leveraging support vector machine for opcode density based detection of crypto-ransomware', in Dehghantanha, A., Conti, M. and Dargahi, T. (Eds.): *Cyber Threat Intelligence. Advances in Information Security*, Vol. 70, pp.107–136, Springer, Cham, https://doi.org/10.1007/978-3-319-73951-9_6.

Breiman, L. and Cutler, A. (Fortran original), Liaw, A. and Wiener, M. (R port) (2015) *randomForest package*, DOI [online] https://cran.r-project.org/web/packages/randomForest/index.html.

Chen, C.M. and Lin, H.C. (2015) 'Detecting botnet by anomalous traffic', *Information Security and Applications*, Vol. 21, No. C, pp.42–51, Elsevier, DOI: https://doi.org/10.1016/j.jisa.2014.05.002.

Cid-Fuentes, J.A., Szabo, C. and Falkner, K. (2018) 'An adaptive framework for the detection of novel botnets', *Computers & Security*, November, Vol. 79, pp.148–161, https://doi.org/10.1016%2Fj.cose.2018.07.019.

Cook, E., Jahanian, F. and McPherson, D. (2005) 'The zombie roundup: understanding, detecting, and disrupting botnets', in *Proceedings of the Steps to Reducing Unwanted Traffic on the Internet*, Vol. 5, pp.39–44.

Correia, P., Rocha, E., Nogueira, A. and Salvador, P. (2012) 'Statistical characterization of the botnets C&C traffic', *Procedia Technology*, Vol. 1, pp.158–166, Elsevier BV, https://doi.org/10.1016%2Fj.protcy.2012.02.030.

Council of Economic Advisers (CEA) (2018) [online] https://www.thompsonhine.com/publications/cea-report-cost-of-malicious-cyber-activity-to-the-us-economy.

HaddadPajouh, H., Dehghantanha, A., Khayami, R. and Choo, K.K.R. (2018) 'A deep recurrent neural network based approach for internet of things malware threat hunting', *Future Generation Computer Systems*, August, Vol. 7, No. 4, pp.320–331, Elsevier BV, https://doi.org/10.1016/j.future.2018.03.007.

Jiang, H. and Shao, X. (2012) 'Detecting P2P botnets by discovering flow dependency in C&C traffic', *Peer-to-Peer Networking and Applications. Springer Science and Business Media LLC*, Vol. 7, No. 4, pp.320–331, https://doi.org/10.1007/s12083-012-0150-x.

Kiwia, D., Dehghantanha, A., Choo, K.K.R. and Slaughter, J. (2017) 'A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence', *Journal of Computational Science*, Vol. 27, No. 2, pp.394–409, https://doi.org/10.1016/j.jocs.2017.10.020.

Kudo, T., Kimura, T., Inoue, Y., Aman, H. and Hirata, K. (2018) 'Stochastic modeling of self-evolving botnets with vulnerability discovery', *Computer Communications*, Vol. 124, No. 6, pp.101–110, https://doi.org/10.1016%2Fj.comcom.2018.04.010.

Lu, L., Feng, Y. and Sakurai, K. (2017) 'C&C session detection using random forest', in *IMCOM '17 Proceedings of the 11th International Conference on Ubiquitous Information Management and Communication*, Article No. 34.

Moustafa, N. and Slay, J. (2017) *RCNF: Real-time Collaborative Network Forensic Scheme for Evidence Analysis*, arXiv preprint arXiv: 1711.02824.

Osanaiye, O., Cai, H., Choo, K.K.R., Dehghantanha, A., Xu, Z. and Dlodlo, M. (2016) 'Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing', *EURASIP Journal on Wireless Communications and Networking*, Vol. 2016, No. 1, pp.1–10, https://doi.org/10.1186/s13638-016-0623-3.

Paxton, N.C. (2011) *Development of a Multi-layered Botmaster based Analysis Framework*, pp.48106–1346, ProQuest LLC, 789 East Eisenhower Parkway P.O. Box 1346 Ann Arbor, MI.

Pektaş, A. and Acarman, T. (2018) 'Deep learning to detect botnet via network flow summaries', *Neural Computing and Applications*, Vol. 31, No. 11, pp.8021–8033, https://doi.org/10.1007/s00521-018-3595-x.

Silva, S.S.C., Silva, R.M.P., Pinto, R.C.G. and Salles, R.M. (2013) 'Botnets: a survey', *Computer Networks*, Vol. 57, No. 2, pp.378–403.

Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A. and Stiller, B. (2010) 'An overview of IP flow-based intrusion detection', *IEEE Communications Surveys & Tutorials*, Vol. 12, No. 3, pp.343–356.

Vormayr, G., Zseby, T. and Fabini, J. (2017) 'Botnet communication patterns', *IEEE Communications Surveys & Tutorials*, Vol. 19, No. 4, pp.2768–2796.

Wang, X. and Ramsbrock, D. (2009) 'The botnet problem', in Vacca, J.R. (Ed.): *Computer and Information Security Handbook*, Morgan Kaufman, Burlington.

Yahyazadeh, M. and Abadi, M. (2015) 'BotGrab: a negative reputation system for botnet detection', *Computers and Electrical Engineering*, Vol. 41, No. C, pp.68–85, Elsevier, https://doi.org/10.1016/j.compeleceng.2014.10.010.

Yin, C., Ma, L., Feng, L., Yin, Z. and Wang, J. (2015) 'A feature selection algorithm towards efficient intrusion detection', *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 10, No. 11, pp.253–264.

Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A. and Garant, D. (2013) 'Botnet detection based on traffic behavior analysis and flow intervals', *Computers and Security*, November, Vol. 32, pp.2–16, Elsevier, https://doi.org/10.1016%2Fj.cose.2013.04.007.