Machine Learning For HTTP Botnet Detection Using Classifier Algorithms

Rudy Fadhlee Mohd Dollah[1], Faizal M. A.[1], Fahmi Arif[2], Mohd Zaki Mas'ud[1] and Lee Kher Xin[1]
[1] Information Security, Digital Forensic, and Computer Networking (INSFORNET), Faculty of Information Technology and Communication, Universiti Teknikal Malaysia Melaka, Ayer Keroh, Melaka, Malaysia
[2] Department of Industrial Engineering, Institut Teknologi Nasional (Itenas) Bandung. Indonesia
faizalabdollah@utem.edu.my

***Abstract*— Recently, HTTP based Botnet threat has become a serious problem for computer security experts as bots can infect victim's computer quick and stealthily. By using HTTP protocol, Bots are able to hide their communication flow within normal HTTP communications. In addition, since HTTP protocol is widely used by internet application, it is not easy to block this service as a precautionary approach. Thus, it is needed for expert finding ways to detect the HTTP Botnet in network traffic effectively. In this paper, we propose to implement machine learning classifiers, to detect HTTP Botnets. Network traffic dataset used in this research is extracted based on TCP packet feature. We also able to find the best machine learning classifier in our experiment. The proposed method is able to classify HTTP Botnet in network traffic using the best classifier in the experiment with an average accuracy of 92.93%.**

***Index Terms*— Botnet Detection; Classification; Classifier; HTTP Botnet; Machine Learning; Malware.**

I. Introduction

Nowadays, the cybercriminal uses bot malware tirelessly to infect victim's computer and make them as part of their bot armies (zombie PC) which known as Botnet. The infected machine is controlled by botmaster to commit their crimes and achieve their evil intentions. A botnet can be defined as a collection of computers or devices that have been infected by malware, allowing the attacker to perform malicious activities by sending instructions through command and control (C&C) server. There are various types of Botnet communication channels and the earliest Botnet uses centralized network architecture of Internet Relay Chat (IRC) for C&C server to communicate with bot zombies. To date, Botnet has adapted to several attack pattern and using various type of network protocols to commit malicious activity. One of example is peer-to-peer (P2P) Botnet that use the P2P application to carried out C&C server command. However, P2P Botnet has the drawback of complexity in managing bots for decentralized network architecture, so the Hypertext Transfer Protocol (HTTP) Botnet is introduced to overcome the issue. HTTP Botnet operating in centralized network architecture, similar to IRC Botnet with some detection evasion features like DNS fast-flux and using HTTP protocol resulting difficulties in detection. HTTP Botnet responsible for committing several attacks famously known for distributed denial-of-service (DDoS) attack, stealing information, spamming, fraud and malware spreading in the digital world. According to Ref. [1], it is found that HTTP-DDoS was a common attack by Botnet. MyCert Incident Statistics Report 2017 [2] stated that the number of malware infection caused by Botnet increased from Jan 2017 to April 2017. The threat on common HTTP protocol (Port 80) which is used by the normal user to access web page motivates us to study the detection of HTTP Botnets and minimize its threat in the future.

Intrusion detection system (IDS) is a system device or software that is used to monitor computer network or system from malicious behavior and violation in security policy [3]. There are two main categories of IDS which are network IDS (NIDS) and host IDS (HIDS) [4]. NIDS are located at a certain point in computer networking system to monitor network traffic to and from all network devices connected to the network. Meanwhile, HIDS is setup in an individual node in network traffic usually in mission critical devices, for example on the servers. IDS has two main detection methods namely signature-based and anomaly-based. A signature-based IDS is an IDS that detect the attack based on specific known attack signatures. The drawback of signature-based IDS is that the system not able to detect a new attack as no attack signature available in IDS knowledge database. For anomaly-based IDS, the detection system main purpose is to detect any malicious activities based on malicious behavior as set in the IDS rule sets. Anomaly-based IDS basically implement machine learning approach to create a detection model (normal and malicious detection model) for detecting new unknown malicious behavior. The anomaly-based IDS may produce a false alarm if there are unknown legitimate behavior in the system. Hence, in this research, we use anomaly-based IDS which implement machine learning classification to classify normal and malicious behavior in network traffic.

Machine learning is a term that a computer has been programmed, giving the ability to learn by studying the data pattern and make a prediction on new data in artificial intelligence (AI). Machine learning has two main categorizations namely clustering and classification. In clustering, the data input is group into their similarities to each other without learning model. This type of learning known as unsupervised learning. The examples of clustering techniques are k-means [5] and power spectral density [6]. Meanwhile, in classification, has two phases which are training phase and testing phase. The data is labeled by assigning the class to the data input. Then the machine will learn data pattern using classifier algorithm in training phase and produce learning model. In the testing phase, the new data is used and the machine will classify using classifier algorithm together with learning model. This type of learning known as supervised learning. The examples of classification techniques are decision tree and Naïve Bayes. Thus, in this experiment, we use classification as our data is labeled with malicious or normal classes for each network packet.

In this paper, the purpose of the study is to implement machine learning classifiers to detect HTTP Botnet in network traffic. The rest of the paper is organized as follows. Section II discusses related work that has been done by the previous researchers which related to this paper topic. Section III discusses on the methodology of the experiment. Section IV describes obtained result. Finally, the conclusion is stated in section V.

II. Related Work

This section discusses detection of HTTP Botnet that has been done before. Various techniques have been used to detect Botnet. C. Livadas et al. in 2006 [7] are among the earliest work to study about detection of Botnet using machine learning.

Reference [8], use various type of classification algorithm. The algorithms are Sequential Minimal Optimization (SMO), Bayes Theorem-based algorithms, J48 – Decision Tree, Random Forest, Voted Perceptron, K-Nearest Neighbour and Multilayer Perceptron. The author highlighted the ratio number of the packet corresponding to benign traffic with malicious traffic is ranged from 4:1 to 80:1. The highest accuracy achieved is 82.48 % by using random forest classifier. Interestingly, the ratio number of the packet also discussed by C.Chen and H. Lin [9] that use 5:5 ratio for individual malicious traffic to normal traffic. Thus, our experiment use, 1:1 ratio traffic as the previous study show a good result for individual malicious traffic to normal traffic.

Another researcher that achieved high detection rate by using C4.5 Decision Tree classifier is [10], employs C4.5 Decision Tree and Naïve Bayes learning algorithm to detect HTTP based Botnet. The study uses flow-based network traffic (NetFlow) and using HTTP filters. The highest detection rate achieved is 97% with a very low false positive rate of 3% with C4.5 Decision Tree as the classification algorithm.

Meanwhile, Venkatesh, G.K. and Nadarajan, R.A. [11] identifies anomalies in network flow by extracting TCP packet features. Extracted TCP packet is based from communication web-based botnet in specific time intervals. The researchers did a comparison between multilayer Feed-Forward neural network model with C4.5 Decision Tree, Random Forest and Radial Basis Function (RBF). The study found that neural network classifier has better average detection accuracy of 99.025% on SpyEye and Zeus Botnet. The accuracy of the experiment is then compared with [12] and [13].

Although detection accuracy shows the promising result, our experiment does not implement both neural network algorithm namely multilayer Feed-Forward neural network and RBF due to several reasons. Firstly, the neural network requires a lot of computational processing resources and Graphic Processing Units (GPU) is used to decrease the training duration. However, our experiment PCs have a low specification in term of GPU and processing resources which limit the capabilities of the neural network. Secondly, the neural network also requires a large set of features during training phase compared to decision trees. Our detection features have been reduced during data preprocessing phase and not suitable to implement neural network classifier as it may give out false detection accuracy. The justifications are discussed based on work by Tabarez-Paz et al. [14] and supported by Pouliakis et al. [15] that highlighted the same issues. Hence, in this paper, the classifier algorithms are used to detect HTTP Botnet in network traffic based on TCP packet features.

III. Methodology

This paper implement classifier algorithm of machine learning to classify the normal and bot-infected communication in network traffic. In this section, we discuss the methodology of the research using machine learning to detect HTTP Botnet. The methodology that has been carried out is depicted as in Figure 1.

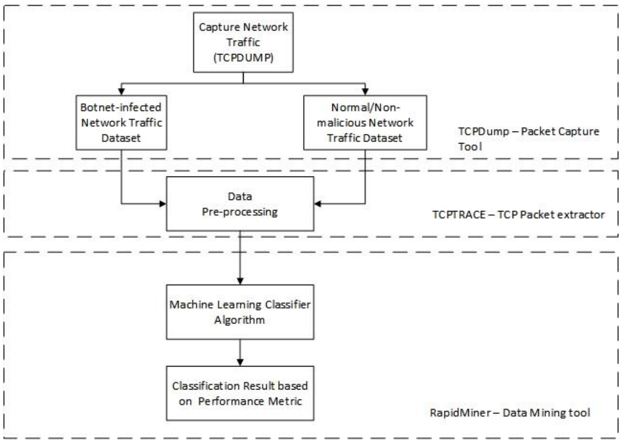
Capture Network Traffic (TCPDUMP)
Botnet-infected Network Traffic Dataset
Normal/Non-malicious Network Traffic Dataset
TCPDump – Packet Capture Tool
Data Pre-processing
TCPTRACE – TCP Packet extractor
Machine Learning Classifier Algorithm
Classification Result based on Performance Metric
RapidMiner – Data Mining tool

Figure 1: Research methodology for HTTP Botnet detection

A. Data Collection

First, a test bed environment is setup to generate data set for HTTP Botnet detection analysis. This test bed environment aims to obtain real malicious traffic. The design of network test bed depicted as Figure 2. The network design of test bed consists of five desktop PCs installed with Windows 7 operating system that becomes Botnet zombies by executing bot binaries in the PCs. There are five types of HTTP Bots used in this study namely Dorkbot, Zeus, Citadel, SpyEye, and Cutwail. A sniffer server also connected to the same network to capture network traffic log that incoming and outgoing at the default gateway.

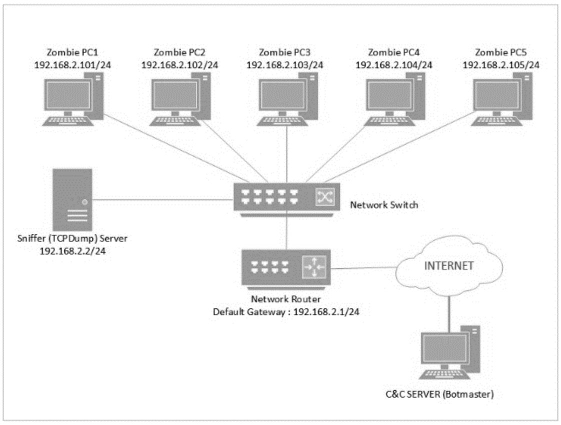
Zombie PC1 192.168.2.101/24
Zombie PC2 192.168.2.102/24
Zombie PC3 192.168.2.103/24
Zombie PC4 192.168.2.104/24
Zombie PC5 192.168.2.105/24
Network Switch
Sniffer (TCPDump) Server 192.168.2.2/24
INTERNET
Network Router Default Gateway : 192.168.2.1/24
C&C SERVER (Botmaster)

Figure 2: Network design for HTTP Botnet test bed environment

Any network communication between bot and C&C server is collected using tcpdump tool. The tcpdump data for the different type of HTTP Botnet are collected to analyze the network activity of the HTTP Botnet. The five type of HTTP bots is released for seven days. After seven days, the tcpdump data that are collected will continue to the next phase. Malicious traffic is combined with non-malicious (normal) traffic. Normal traffic is obtained by carrying out with the same test bed design without executing bot binary in the PCs and perform web browsing activity to simulate normal user activities.

B. Data Pre-Processing

In data pre-processing, the network traffic of both malicious and normal is extracted into TCP traffic log parser (.csv file) using TCPTRACE tool. Malicious and normal log parser is combined and labeled with “0” for normal traffic and “1” for bot traffic. The aggregation traffic is then undergoing data cleaning process which is carried out manually to reduce error, meaningless noise in the obtained result and avoid miscalculation of detection accuracy in classification. Data cleaning also includes ignoring the source and destination IP and port number due to inefficiency in general Botnet detection and less effective on Botnet’s IP-flux attack [16].

C. Machine Learning Classification

Then, the labeled data is run through classifier algorithm using data modeling tool, RapidMiner Studio [17]. Classifier algorithms used in this research are summarized in Table 1.

Table 1
Classifier description

Classification Algorithm	Description
Decision Tree	A decision algorithm is a machine learning model that consist internal and leaf nodes. The internal node contains the attribute or feature of data. Meanwhile, the leaf nodes show the class label. The branches of internal nodes connected to leaf nodes to create a model of a learning tree.
k-Nearest Neighbour (KNN)	KNN classifies unknown input data based on the class of the attributes that closest to training dataset. The KNN algorithm measures the distance between training data and unknown data in order to classify the attribute.
Naïve Bayes (NB)	Naïve Bayes classifier is a derivation from Bayesian Theorem by using all attribute contained from the data and conditionally analyzed the attribute independently as to assume that all attribute are equally important.
Random Forest (RF)	RF classifier algorithm is an ensemble machine learner method provides works by constructing many individual decision trees on various sub-sample of data and decide the best parameter by selecting the output class that appears most often or by mean prediction of classes in decision trees nodes.

The k-fold cross-validation(x-validation) is used in this experiment to validate the performance of the learned model. The number of fold used is set to 10. 10 fold x-validation is the method where the input data is divided into 10 sets of data. When 9 sets of data is used for learning in training phase, the other 1 set of data are used as a test set in the testing phase. The validation method is repeated 10 times according to number of the folds and the classifier performance is evaluated by using performance metrics.

In the performance metrics, the labeled data that had been classified using classification algorithm will give a result on True Positive Rate (TPR), False Positive Rate (FPR), Accuracy and Precision [18].

IV. Result and Discussion

In order to evaluate the proposed approach, several Botnets datasets were used as shown in Table 2. Botnet datasets are consist of seven large datasets with one dataset signify one day for each HTTP Bot, executed in the test bed.

Table 2
The result of HTTP Botnet detection using our approach

HTTP Bot Family	Classifier	Accuracy (%)	Precision (%)	Recall/TPR (%)	FPR (%)
Dorkbot	Decision Tree	87.75	86.86	99.99	66.14
	KNN	90.07	93.69	94.07	26.88
	Naïve Bayes	70.10	91.54	69.46	27.22
	Random Forest	81.47	81.37	99.99	97.12
Zeus	Decision Tree	83.61	82.16	99.82	65.07
	KNN	86.96	91.21	91.42	26.44
	Naïve Bayes	51.84	84.85	43.58	23.35
	Random Forest	78.08	77.42	99.93	87.51
SpyEye	Decision Tree	90.41	96.33	90.86	11.03
	KNN	95.26	96.84	96.94	10.09
	Naïve Bayes	65.51	98.31	55.66	3.06
	Random Forest	76.84	76.68	99.98	96.95
Cutwail	Decision Tree	93.73	94.91	97.94	31.52
	KNN	97.88	98.66	98.87	8.06
	Naïve Bayes	87.76	95.42	90.04	25.95
	Random Forest	86.38	86.33	99.93	94.93
Citadel	Decision Tree	91.70	90.39	98.41	23.10
	KNN	94.46	95.92	96.04	9.01
	Naïve Bayes	73.23	85.51	73.55	27.47
	Random Forest	71.88	71.02	88.89	89.85

Table 2 shows the result of the performance of using four type of classifier algorithms. The random forest classifier shows promising TPR with Dorkbot, Zeus, SpyEye and Cutwail detection achieved an average above 90%. However, the FPR for random forest classifier also high which shows that the detection using random forest classifier may produce false alarm during HTTP Botnet detection.

Surprisingly, the accuracy produced by KNN classifier are highest for each type of bot family with good performance of FPR. In another word, KNN is able to classify the bot and normal traffic due to high detection accuracy and produce low false alarm during detection. Hence, we conclude that the best classifier to detect HTTP bot for this experiment is KNN classifier algorithm. KNN has good performance in term of high accuracy, good bot detection rate (TPR) and low false alarm compared to other classifiers.

Interestingly, although the result of the experiment shows that our approach is able to detect HTTP Botnets activities in network traffic, in some circumstances the approach may falsely detect normal behaviors as malicious activities in real traffic. For example, sometimes when the user does keep on reloading the same web pages, it sends repeated HTTP request packet to the web server. This activity resembles the behavior pattern of HTTP Botnets attack [19]. Thus, to ensure that our approach is able to detect Botnet effectively, we will look for selecting proper network features in the future to increase detection effectiveness.

V. Conclusion

The number of HTTP Botnet threat has increased year by year. Hence, there is a need of finding solutions to overcome this threat. This paper aims to implement machine learning classifier to detect HTTP Botnet. The detection is carried out by detecting HTTP Botnet in network traffic based on TCP traffic features. The proposed methodology is evaluated based on true positive rate, false positive rate and detection accuracy on five different HTTP Botnets. The classifiers used in the experiment are four classifiers namely Decision Tree, Naïve Bayes, K-Nearest Neighbour and Random Forest. We achieve our objective to detect HTTP Botnet using machine learning classifier algorithm. Moreover, the result showed significant readings on classification detection of malicious activities of HTTP Botnet in their network traffic. The best classifier for this experiment, K-Nearest Neighbour classifier achieving average detection accuracy of 92.93% with TPR of 95.47%. The result shows that the KNN is able to detect HTTP Botnet in network traffic and with low false alarm compared to other machine learning classifier. The result achieved in the experiment may contribute to the body of knowledge in computer network security field that machine learning classifier is capable and convincing to detect HTTP Botnet. In the future, we will perform a selection of network attribute. The attribute selection purpose is to reduce the number of the feature while getting similar or better result as without attribute selection.

Acknowledgment

This work has been supported under Universiti Teknikal Malaysia Melaka (UTeM) research grant GLUAR/CSM/2016/FTMK-CACT/I00013 and KPT MyBrain15 postgraduate scholarship. The authors would like to thank you to reviewers and members of INSFORNET research group for their incredible supports and guides in the making of these paper.

References

[1] Kaspersky Lab, "Statistics on Botnet-Assisted DDoS Attacks in Attacks in Q1 2015," 2015. [Online]. Available: https://securelist.com/statistics-on-botnet-assisted-ddos-attacks-in-q1-2015/70071/. [Accessed: 12-Jul-2015].

[2] MyCERT, "MyCERT Incident Statistics 2017," 2017. [Online]. Available:https://www.mycert.org.my/statistics/2017.php. [Accessed: 20-May-2017].

[3] J. Jabez and B. Muthukumar, "Intrusion Detection System (IDS): Anomaly Detection Using Outlier Detection Approach," *Procedia Comput. Sci.*, vol. 48, pp. 338–346, 2015.

[4] M. A. Khan, "A survey of security issues for cloud computing," *J. Netw. Comput. Appl.*, vol. 71, pp. 11–29, 2016.

[5] C. Fachkha, E. Bou-Harb, and M. Debbabi, "Inferring distributed reflection denial of service attacks from darknet," *Comput. Commun.*, vol. 62, pp. 59–71, 2015.

[6] J. Kwon, J. Lee, H. Lee, and A. Perrig, "PsyBoG: A scalable botnet detection method for large-scale DNS traffic," *Comput. Networks*, vol. 97, pp. 48–73, 2016.

[7] C. Livadas, R. Walsh, D. Lapsley, and W. T. Strayer, "Using machine learning techniques to identify botnet traffic," in *Proceedings - Conference on Local Computer Networks, LCN*, 2006, pp. 967–974.

[8] F. Brezo, D. Puerta, X. Ugarte-pedrero, I. Santos, P. G. Bringas, and D. Barroso, "A Supervised Classification Approach for Detecting Packets Originated in a HTTP-based Botnet," vol. 16, no. 3, pp. 1–13, 2013.

[9] C.-M. Chen, Y.-H. Ou, and Y.-C. Tsai, "Web botnet detection based on flow information," *2010 Int. Comput. Symp.*, pp. 381–384, 2010.

[10] F. Haddadi, J. Morgan, E. G. Filho, and a. N. Zincir-Heywood, "Botnet behaviour analysis using IP flows: With http filters using classifiers," *Proc. - 2014 IEEE 28th Int. Conf. Adv. Inf. Netw. Appl. Work. IEEE WAINA 2014*, pp. 7–12, 2014.

[11] G. Kirubavathi Venkatesh and R. Anitha Nadarajan, "HTTP Botnet Detection Using Adaptive Learning Rate Multilayer Feed-Forward Neural Network," *Inf. Secur. Theory Pract. Secur. Priv. Trust Comput. Syst. Ambient Intell. Ecosyst. SE - 5*, vol. 7322, pp. 38–48, 2012.

[12] Nogueira, P. Salvador, and F. Blessa, "A Botnet Detection System Based on Neural Networks," Digit. Telecommun. (ICDT), 2010 Fifth Int. Conf., pp. 57–62, 2010.

[13] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner : Clustering Analysis of Network Traffic for Protocol- and Structure-Independent Botnet Detection," *Proc. 17th Conf. Secur. Symp.*, pp. 139–154, 2008.

[14] Tabarez-paz, N. Hernández-Gress, and M. G. Mendoza, "Improving of Artificial Neural Networks Performance by Using GPU 'S : A Survey," in Third International Conference on Advances in Computing & Information Technology, 2013, no. 1943, pp. 39–48.

[15] Pouliakis, E. Karakitsou, N. Margari, P. Bountris, M. Haritou, J. Panayiotides, D. Koutsouris, and P. Karakitsos, "Artificial Neural Networks as Decision Support Tools in Cytopathology: Past, Present, and Future," Biomed. Eng. Comput. Biol., no. 7, pp. 7–1, 2016.

[16] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards Effective Feature Selection in Machine Learning-Based Botnet Detection Approaches," pp. 247–255, 2014.

[17] RapidMiner inc, "RapidMiner: Data Science Platform," 2016. [Online]. Available: https://rapidminer.com/. [Accessed: 23-Sep-2016].

[18] M. Z. Mas'Ud, S. Sahib, M. F. Abdollah, S. R. Selamat, and R. Yusof, "Analysis of features selection and machine learning classifier in android malware detection," *ICISA 2014 - 2014 5th Int. Conf. Inf. Sci. Appl.*, 2014.

[19] M. Eslahi, H. Hashim, and N. M. Tahir, "An efficient false alarm reduction approach in HTTP-based botnet detection," *IEEE Symp. Comput. Informatics, Isc. 2013*, pp. 201–205, 2013.