EFFECTIVE FEATURE SELECTION FOR BOTNET DETECTION BASED ON NETWORK FLOW ANALYSIS

EFFECTIVE FEATURE SELECTION FOR BOTNET DETECTION BASED ON NETWORK FLOW ANALYSIS

A. Pektaş, T. Acarman

Computer Engineering Department, Galatasaray University, Ortaköy, TR-34349, Istanbul, Turkey.
e-mail: apektas@yandex.com

Abstract: Botnets have been pose one of the most persistent and critical threats across the enterprises and individuals that is not easily identified with traditional methods. The botnet creators constantly developing different hiding techniques, network topologies and communication protocols to survive their bots from detection. Hence, combatting botnet threats is challenging task and various studies have been proposed for the detection of botnet. In the literature, researchers have been employed machine learning methods using different feature sets as a prevalent method. However, there is not enough research performed to explore the effect of the selection of feature set in botnet detection. In this paper, we analyze the most discriminating features for the purpose of building an efficient and effective botnet detection system. To this end, we utilize three different feature selection methods, namely Linear models penalized with the L1 norm (aka Lasso), Recursive Feature Elimination (RFE), Tree-based feature selection (aka random forest feature ranking). To evaluate these methods, we conduct a series of experiments on a public botnet trace by applying three machine learning methods.

Key words: botnet, feature selection, machine learning, information security

INTRODUCTION

Botnet can be described as a collection of compromised computers/devices that are managed remotely by a master node called botmaster or Command and Control (C&C) server. The infected machine actively participate in various malicious activities including but not limited Distributed Denial-of-Service (DDoS) attacks, stealing sensitive data, spreading spam e-mail, gaining financial profit, without the knowledge of the system owner. In the past, the botnet relies on The Internet Relay Chat (IRC) protocol for communicating C&C. As the IRC protocol uses centralized network topology, it is easier to detect and takedown entire botnet.

In order to evade detection mechanisms, botnets have transformed their network structure from centralized to de-centralized topologies. Moreover, the botnet take advantage of the encryption techniques and the more ubiquitous protocols like Hyper Text Transfer Protocol (HTTP) than IRC. For example, although the very first version of the Zeus type botnets uses HTTP-based centralized architecture, the botnet authors adopts and migrates HTTP-based de-centralized architecture using P2P protocol. Moreover, to make the detection harder, the cyber criminals sometimes blend normal web traffic into the botnet communication. Hence, these improvements make the botnet detection very difficult and challenging task that need new mechanisms.

Botnet detection is a challenging research topic that has gained much attention from researchers and security practitioners in the last two decades. Eventually, plenty of experimental botnet detection systems have been proposed in the literature [1], [2], [3], [4], [5]. These proposed approaches exploit different assumptions and methods about the botnet to model and formalize the botnet traffic. One of the most leading and successful types of botnet detection method is based on machine learning (ML) techniques. The main hypothesis of the machine learning-based solutions is that bots produce particular blueprint in the network traffic which can allow researchers to effectively and efficiently detect botnet using machine learning.

Although, much research has been conducted on the discovery of botnet using ML techniques, little research has been undertaken specifically on the effect of the feature set that directly affect the performance of the ML based botnet detection system. In this work, we investigate the three feature selection approaches on public botnet traffic using flow features extracted by open-source Tranalyzer tool [6], [7]. We applied three machine learning algorithm, namely, SVM [8], Logistic Regression [9], and Random Forest [10] on the selected feature set. Our ultimate goal is to determine which features provide better classification performance.

RELATED WORK

Numerous botnet detection methods have been introduced based on various MLAs deployed in diverse configurations and feature set. For example, in a recent study, Yang et al. [11] proposed an approach to detect mobile botnet using a multi-level feature extraction approach. The authors employ not only extracted features from the flow or basic TCP/IP level, but also extracted information from the payload content like HTTP payload content. Finally, Random Forest classifier is evaluated on the feature set and achieved True Positive Rate and False Positive Rate as 0.93, 0.05, respectively.

Discovering the most useful and representative features is of great importance for any machine learning problems. In recent years, some research efforts have been done on the problem of finding favourable feature set for machine learning based botnet detection systems. Alejandre et al. [12] proposed a method to perform feature selection to detect botnets . To this end, they used Genetic Algorithm (GA) to select the set of features and C4.5 algorithm to classify network traffic whether belonging to botnet or normal. The authors evaluate only 19 statistical flow features on two public botnet dataset. According to their experiments, they reduced the number of feature set to 10 and 11 for ISOT [13] and ISCX [14] dataset, respectively and achieved 99.44% and 96.52% accuracy for ISOT and ISCX dataset, respectively. Although authors achieved promising results, the accuracy measure alone is not enough to judge the effectiveness of their method. Thus, they some additional performance metric such as F-score, precision etc. could be calculated to evaluate their method.

Haddadi et al. [15], investigate in their study, the effect of the selection of different network traffic flow exporters. To accomplish this, they evaluate five different traffic flow exporters; Maji, YAF, Softflowd, Tranalyzer and Netmate using five different classifiers C4.5, SVM, ANN, Bayesian networks, and Naive Bayes. They conduct a series of experiments on public botnet datasets. According to their experimental results, the best classification accuracy is achieved by using Tranalyzer with C4.5 classifier.

Beigi et al. [16] also explore the effectiveness of different combination of features to achieve the best classification accuracy. They benchmark flow-based statistical features actively used in the existing studies and analyze their relative effectiveness. They employed three feature selection methods such as Correlation Feature Selection (CFS), Principal Component Analysis (PCA) , and Minimum redundancy-maximum-relevance (mRMR), to eliminate less discriminative features from other candidate features. Although their final feature set showed a high detection rate of with 99% on a training dataset which includes small number of botnets, they achieved 75% detection accuracy on a testing dataset which contains much more different types of botnet.

In this work, we utilize three feature selection algorithms, i.e. Linear models penalized with the L1 norm (aka Lasso), Recursive Feature Elimination (RFE), Tree-based feature selection to choose set of features with high discriminative ability to distinguish botnet from normal traffic. To this end, first we export all flow features by open-source Tranalyzer flow extractors. After applying feature selection on the extracted feature set, three well-known machine learning algorithms namely; SVM, Random Forest and Logistic Regression are evaluated to achieve best botnet detection accuracy.

Our contributions can be summarized as follows:

- The statistical flow features can provide sufficient information for the accurate description of botnet and normal network traffic.
- We conduct extensive experiments on real-world public botnet dataset in order to determine the best set of features for botnet detection.
- The evaluation results show that selected (i.e. reduced) features not only increase the classification performance, but also decrease the computational cost greatly.

METHODOLOGY

In this section, we elaborate our methodology for botnet detection. The proposed system considers flow based static features, including total network packets, duration of the flow, mean, variance and standard deviations of the packet size, etc. to model botnet traffic.

The proposed methodology, as shown in Figure-1, consists of four major steps. The first step is extracting feature set from raw network capture by using an open source flow exporter called Tranalyzer. This tool processes raw network captures and computes flow based statistical features. The feature set extracted is listed in Table 1. The second step is dedicated to the selection of appropriate features from the set of flow features. We use three feature selection methods to choose the most representative subset of the features toward accurate classification of botnet traffic. As a result of this process, network traffic is represented as a feature matrix and a class label indicating traffic category, botnet or normal.

The third step includes the building classification model based on feature set. Since network traffic is vectorised into a feature vector, it is an input to the machine learning algorithms to derive classification model. In our experiments, we investigate three classification methods which are more suitable to high dimensional feature space, including Logistic Regression, Naive Bayes and meta-classifier Random Forest Classification. The final step is evaluation of classification methods.

Feature Set

Network flow can be defined as summary of a connection between two hosts. A network flow is defined based on combination of 5-tuple, source and destination IP addresses, source and destination port numbers, and protocol. Network flows can provide valuable information about the network activities. There are lots of ways to extract flows from the network traffic, one of the popular ways is to use flow extractors. These tools aggregate the 5-tuple into flows and then calculate some statistical features such as the duration of the flow, the number of bytes transferred in the flow, etc. In our study, we extract features from network traces by tracing for both in the forward and backward flow direction. Overall, 27 numeric features are obtained. For each flow, a feature vector is constituted by the features listed in Table-1.

EXPERIMENTS RESULTS AND ANALYSIS

In this section, we elaborate the performance of the selected algorithms with respect to the eliminated feature sets by using three feature selection methods. Our focus in particular is the achievement of a high level in accuracy for detection of botnet traffic. For this purpose, we test the machine learning algorithms on public botnet traffic.

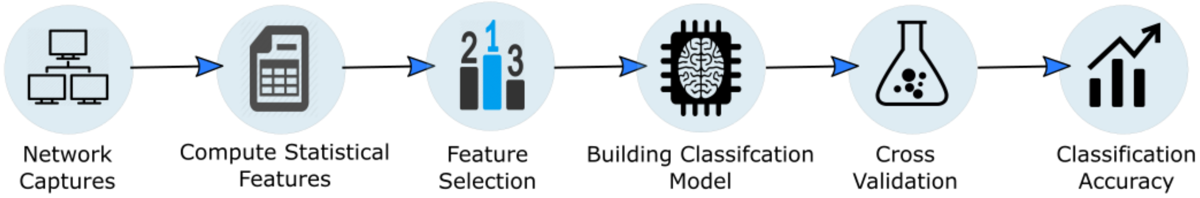

Figure 1. Overview of the proposed methodology

Table 1. The list of statistical flow-based features

Feature	*Description*
Duration	Flow duration
L4Proto	Layer 4 protocol
numPktsSnt	Number of transmitted packets
numPktsRcvd	Number of received packets
numBytesSnt	Number of transmitted bytes
numBytesRcvd	Number of received bytes
minPktSz	Minimum layer3 packet size
maxPktSz	Maximum layer3 packet size
avePktSize	Average packet load ratio
stdPktSize	Filt stddev packet load ratio
pktps	Send packets per second
bytps	Send bytes per second
pktAsm	Packet stream asymmetry
bytAsm	Byte stream asymmetry
tcpPSeqCnt	TCP packet seq count
tcpSeqSntBytes	TCP sent seq diff bytes
tcpSeqFaultCnt	TCP sequence number fault count
tcpPAckCnt	TCP packet ack count
tcpFlwLssAckRcvdBytes	TCP flawless ack received bytes
tcpAckFaultCnt	TCP ack number fault count
tcpInitWinSz	TCP initial effective window size
tcpAveWinSz	TCP average effective window size
tcpMinWinSz	TCP minimum effective window size
tcpMaxWinSz	TCP maximum effective window size
tcpWinSzDwnCnt	TCP effective window size change down count
tcpWinSzUpCnt	TCP effective window size change up count
tcpWinSzChgDirCnt	TCP effective window size direction change count

Feature Selection Methods

We used three different feature selection methods; Linear models penalized with the L1 norm (aka Lasso) [17], Recursive Feature Elimination (RFE) [18], Tree-based feature selection (aka random forest feature ranking) [19].

Random forests algorithm is one the most popular machine learning methods used for classification task. Besides that, it can be used as a feature selection and ranking method. Generally, random forest based feature selection approach expose impurity based feature ranking method. Therefore, generally it requires little feature engineering and parameter tuning.

Feature selection based on L1 regularization / Lasso is to rely on the idea that when there are linearly correlated features, the constructed model becomes unstable. In other words, the small modification in the dataset can produce large variation over the model. L1 regularization approach for feature selection work very well when the data is not noisy and the features are independent.

Recursive feature elimination (RFE) is based on the idea to select features by repeatedly constructing a model and pruning the worst performing feature based on coefficients from the current feature sets. This procedure is repeated on the pruned feature set until the intended numbers of features are reached. The performance of RFE is greatly depends on the model used for feature ranking. In general, the linear models perform better than other models.

Data set

In our study, we employed a public botnet traffic capture called ISOT. The ISOT dataset combines several existing malicious and non-malicious datasets. The dataset contains the botnet traffic involved two different types of botnet; Storm and Waledac. To normalize the botnet traffic, non-malicious everyday network traffic such as web surfing, popular gaming and file sharing are incorporated into the botnet traffic.

Performance metrics

To evaluate the proposed classification method, the following metrics are used: ***precision***, ***recall*** (a.k.a. sensitivity), ***F1-score***, ***classification accuracy***. In binary classification (positive and negative classes), true positives (*tp*) refer to the correctly predicted positive samples, while true negatives (*tn*) are the number of the correctly predicted negative samples. False positives (*fp*) refer to the incorrectly classified positive samples. Similarly, false negatives (*fn*) are the number of incorrectly classified negative samples. Briefly, the terms positive and negative imply the classifier's success while true and false indicates whether or not the prediction is matched with actual (i.e., ground truth) label.

The formulas of the metrics are given as follows:

$$\text{precision} = \frac{\text{tp}}{\text{tp+fp}} \tag{1}$$

$$\text{recall} = \frac{\text{tp}}{\text{tp+fn}} \tag{2}$$

$$\text{F1} - \text{score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision+recall}} \tag{3}$$

$$\text{accuracy} = \frac{\text{correctly classified instances}}{\text{total number of instances}} \tag{4}$$

Evaluation Results

We use 10-fold cross-validation approach. Since benchmark dataset is imbalanced in terms of sample number in each class, we adapt Stratified K-Folds method in the evaluation process. Stratified K-Folds validator splits the data into train and test set by preserving the percentage of samples for each class. All experiments are carried out on a 2.5GHz Intel 4-Core i-7 processor with 8GB physical memory, using scikit learn [20] [21] and MS Windows 10.

We evaluate three different classifiers; Logistic Regression, Naive Bayes and Random Forest. Our particular aim is to achieve best detection. Table-2 gives the general classification accuracy and average recall, precision and F1-score of each machine learning algorithm on the selected features that are eliminated by three feature selection methods. Following the numerical results for each metric, meta-classifier Random Forest applied on the features selected by Random Forest outperforms the other two classifiers and achieves the highest accuracy about %99.9. Random Forest classifier achieves almost perfect classification accuracy for identifying botnet and normal traffic. In other words, the model correctly

Table 2. Performance of the feature selection methods using different machine learning algorithms

	Machine Learning Algorithms											
	Random Forest				Logistic Regression				SVM			
Feature Selection	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy	Precision	Recall	F1-score	Accuracy
Tree Based	0.99	0.99	0.99	0.995	0.72	0.71	0.61	0.710	0.88	0.87	0.87	0.901
RFE	0.94	0.95	0.94	0.943	0.86	0.82	0.84	0.849	0.90	0.92	0.91	0.927
Lasso	0.92	0.92	0.92	0.936	0.85	0.84	0.84	0.891	0.88	0.89	0.88	0.912

predicts the botnet and normal traffic. According to the experimental results, tree based feature selection method chooses 9 of the feature from our initial flow based features, namely Duration, numBytesRcvd, minPktSz, maxPktSz, avePktSize, stdPktSize, pktAsm, bytAsm and tcpMinWinSz. As stated above, these features achieved highest classification performance with Random Forest classification algorithm in terms of precision, recall, F1-score and accuracy.

CONCLUSION

In this paper, we benchmark flow-based statistical feature sets extracted by Tranalyzer flow exporter and empirically examine the impact of these extracted flow features on botnet detection. To this end, we evaluate three feature selection methods and three machine learning classifiers that are well-known and useful in machine learning tasks. Evaluation results show that, the choice of feature set is critical to the success the botnet detection system and greatly affect the performance of these systems. Our results also indicate that Tranalyzer generate very useful statistical feature set which enables the machine learning algorithms to detect botnet with high accuracy and low false positive rates.

From the evaluations performed in this study, the combination of the flow duration, initial, cc, features gives the best classification performance in terms of accuracy, F-score on the public botnet dataset. For the future work, we will investigate the effect of features extracted by other flow exporters. We are also planning to integrate non-numerical features like connection into feature set and benchmark these feature set. We are planning to test the classification system while extending the dataset in terms of flow size and botnet types.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of Galatasaray University, scientific research support program under grant #17.401.001.

REFERENCES

1. Stevanovic M., Pedersen JM. An efficient flow-based botnet detection using supervised machine learning. In Computing, Networking and Communications (ICNC), 2014 International Conference on 2014 Feb 3 (pp. 797-801). IEEE.
2. Zhao D, Traore I, Sayed B, Lu W, Saad S, Ghorbani A, Garant D. Botnet detection based on traffic behavior analysis and flow intervals. Computers & Security. 2013 Nov 30;39:2-16.
3. Kirubavathi G, Anitha R. Botnet detection via mining of traffic flow characteristics. Computers & Electrical Engineering. 2016 Feb 29;50:91-101.
4. Chen R, Niu W, Zhang X, Zhuo Z, Lv F. An Effective Conversation-Based Botnet Detection Method. Mathematical Problems in Engineering. 2017 Apr 9;2017.
5. Azab A, Alazab M, Aiash M. Machine Learning Based Botnet Identification Traffic. InTrustcom/BigDataSE/I SPA, 2016 IEEE 2016 Aug 23 (pp. 1788-1794). IEEE.
6. Burschka S, Dupasquier B. Tranalyzer: Versatile high performance network traffic analyser. In Computational Intelligence (SSCI), 2016 IEEE Symposium Series on 2016 Dec 6 (pp. 1-8). IEEE.
7. https://tranalyzer.com/
8. Wu TF, Lin CJ, Weng RC. Probability estimates for multi-class classification by pairwise coupling. Journal of Machine Learning Research. 2004;5(Aug):975-1005.
9. Yu HF, Huang FL, Lin CJ. Dual coordinate descent methods for logistic regression and maximum entropy models. Machine Learning. 2011 Oct 1;85(1):41-75.
10. Breiman L. Random forests. Machine learning. 2001 Oct 1;45(1):5-32.
11. Yang M, Wen Q. A multi-level feature extraction technique to detect moble botnet. InComputer and Communications (ICCC), 2016 2nd IEEE International Conference on 2016 Oct 14 (pp. 2495-2498). IEEE.
12. Alejandre FV, Cortés NC, Anaya EA. Feature selection to detect botnets using machine learning algorithms. InElectronics, Communications and Computers (CONIELECOMP), 2017 International Conference on 2017 Feb 22 (pp. 1-7). IEEE.
13. http://www.uvic.ca/engineering/ece/isot/datasets/
14. Shiravi A, Shiravi H, Tavallaee M, Ghorbani AA. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. computers & security. 2012 May 31;31(3):357-74.
15. Haddadi F, Zincir-Heywood AN. Benchmarking the effect of flow exporters and protocol filters on botnet traffic classification. IEEE Systems journal. 2016 Dec;10(4):1390-401.
16. Beigi EB, Jazi HH, Stakhanova N, Ghorbani AA. Towards effective feature selection in machine learning-based botnet detection approaches. InCommunications and Network Security (CNS), 2014 IEEE Conference on 2014 Oct 29 (pp. 247-255). IEEE.
17. Zhou Y, Jin R, Hoi S. Exclusive lasso for multi-task feature selection. InInternational conference on artificial intelligence and statistics 2010 (pp. 988-995).
18. Granitto PM, Furlanello C, Biasioli F, Gasperi F. Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products. Chemometrics and Intelligent Laboratory Systems. 2006 Sep 15;83(2):83-90.
19. Chen Y, Miao D, Wang R, Wu K. A rough set approach to feature selection based on power set tree. Knowledge-Based Systems. 2011 Mar 31;24(2):275-81.
20. Scikit-learn: machine learning in Python. http://scikit-learn.org/stable/index.html
21. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research. 2011;12(Oct):2825-30.