Equitable Machine Learning Algorithms to Probe Over P2P Botnets

Pavani Bharathula and N. Mridula Menon

Abstract Cyber security has become very significant research area in line due to the increase in the number of malicious attacks by both state and nonstate actors. Ideally, one would like to properly secure the machines from being infected by viruses of any form. Nowadays, botnets have become an integral part of the Internet and the main drive for creating them is for financial gain. A bot conceals itself using a secret canal to communicate with its governing command-and-control server. Botnets are well-ordered from end to end using protocols such as IRC, HTTP, and P2P. Of all HTTP-based and IRC-based, P2P botnet detection became a challenging task because of its decentralized nature. The paper focuses on the techniques that are predominantly used in botnet detection and we formulate a method for detecting the P2P botnets using supervised machine learning algorithms such as random forest (RF), multilayer perceptron (MLP), and K-nearest neighbor classifier (KNN). We analyze the performance of selected algorithms there by revealing the best classification algorithm for detecting P2P botnets.

Keywords Botnet · Machine learning · Peer-to-peer (P2P)

1 Introduction

Even without our knowledge botnets have become a part and parcel in our Internet life, with an optimistic faith that we are not one amongst them right now. Traditionally, a botnet is described as a group of compromised computers that are controlled remotely using malicious software known as bots by the bot master [1, 15]. There are several reasons for the creation of bots namely financial gain, rivalry among

P. Bharathula (✉) · N. Mridula Menon
TIFAC-CORE in Cyber Security, Amrita Vishwa Vidyapeetham, Coimbatore, Tamil Nadu, India
e-mail: bharathula.pavani@gmail.com

N. Mridula Menon
e-mail: mridula.mmn@gmail.com

S. Das et al. (eds.), *Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015*, Advances in Intelligent Systems and Computing 404, DOI 10.1007/978-81-322-2695-6_2

competing organizations, and many other unethical motives. Botnets are mostly used for profit: Identity theft, email spamming, software piracy [16]. The main aspect lies in the communication architecture of botnet. Most of the communication schemes among the traditional botnets include command-and-control server (C&C) [6].

Bot gets infected with a type of malware known as Trojans and reports to the server through Internet relay chat (IRC). Once infected, bot locates and connects to the server [16]. This connection session is used by the bot master for communicating and controlling the bots. These bots will get new commands from bot master to update its pattern [1]. Depending upon its structure and purpose, C&C server issues commands to perform spamming or to launch distributed denial-of-service (DDoS) attack on a particular target. Thus, the information related to botnet can be used to prevent activities like DDoS and phishing attacks.

Many existing techniques for detecting botnets are based on attack signatures [1]. Even though the signature-based system detects well-known botnets, they suffer from a major drawback that they are unable to detect the variants of the attack signatures and also show high false positive rate. The objective of this paper is to unveil the malicious activities of these botnets and to give the best classification algorithm to detect botnets.

We choose supervised machine learning algorithms because we work on labeled datasets alone. We prefer to use RF, KNN, and MLP algorithms for detecting P2P botnets due to the following advantages, i.e., robust to noisy training data, effective even if training data is large, able to extract patterns even from complicated data, and can handle any number of input variables.

Our contribution starts using the enormous collection of datasets to address the issues like how to identify the malicious contents that originated from botnets and which is the best classification algorithm for detecting P2P botnets among the three chosen algorithms. These questions are addressed in Sects. 3 and 4. Finally, we conclude the paper and outline our future enhancement in Sect. 5.

2 Related Works

Botnet detection involves analyzing capabilities of a botnet and also its network behavior. To detect botnet, there are two approaches, one is static analysis and other is dynamic analysis. Botnet has evolved from single host to distributed in network. P2P botnets were seen for the first time in the mid of 2000. Various types of botnets have been evolving every year that becomes a challenge to all the current detection mechanisms. Botnet named Zotob performed DDOS attack on online banking sites in US in 2005. Another species of Botnet named Kraken which was the worlds largest as of 2008 April infected so many fortune 500 companies and expanded to over 400,000 Bots.

The dangerous botnet Zeus seen in October 2012 was used to steal credit card information of banking customers. Whenever a new bot client is created, hacker

injects malicious scripts to get the details of the computer not limited to memory, network information, and processor speed [5].

Pijush et al. [1] proposed the rule induction algorithm to detect P2P botnets which made the following assumptions: A) For every session of P2P botnets, flow of data occurs in two-way directions. B) Each botnet will have its own set of commands to interact with the bots in C&C. They generated rules based on decision tree classifier. The author measured their algorithm performance with three machine learning classification algorithms such as decision tree, linear support vector machines, and bayesian network. The comparison is done with the help of performance metrics such as accuracy, recall, f-measure, and precision.

Jignesh et al. [13] analyzed several botnet detection techniques and types of botnets. The techniques mentioned are honey pots and honey net, botnet detection using signatures, and host-based detection technique, detection using data mining technique, classification algorithms, clustering, and association rules. Their analysis revealed that data mining algorithms are convenient and effective in detecting botnets based on the characteristic features.

Fariba et al. [4] described botnet analysis system and it is implemented using naive Bayes and C4.5 algorithms for detecting HTTP-based botnet activity. They are detecting botnet behavior by aggregating the network flows and also using domain fluxing techniques.

According to David et al. [16], a theory based on traffic behavior analysis was implemented to detect botnet activity using machine learning. They classified network traffic behavior based on time intervals and did not depend on packet payload. They showed through experimentation that it is feasible to detect the unknown form of botnet activity but their method suffers from false positiveness.

Junjie et al. [15] suggested the technique for detecting the compromised machines using flow-clustering-based analysis. They proposed a method which detects stealthy P2P botnets even if illegitimate traffic is overlapped with legitimate traffic.

3 Methodology

Proposed system follows a hierarchy of three phases which have unique functionalities and are processed in a sequential order as shown in Fig. 1.

3.1 Data Acquisition

On the Internet, the information we communicate across the globe is in the form of packets which are transmitted to and fro the networks and they follow some standard protocols [3]. Network that ships data all over the place in small packets are called packet-switched networks especially in intranet. We capture these packets using Wireshark, which is a well-known network protocol analyzer.

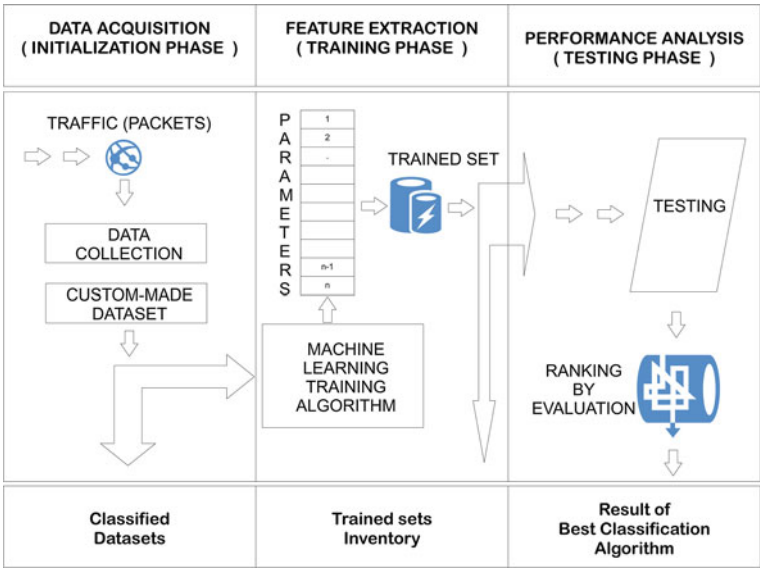

Fig. 1 Architecture

In our Intranet P2P botnet setup, we captured packets using Wireshark. In addition to this, we also collected standard datasets from malware capture facility projects and Information Security Centre of excellence UNB [9, 14]. The two common considerations while capturing packets manually are dealing with "Generic Receive Offload" (GRO) and its workmate "Large Receive Offload" (LRO). It provides a feature for the network card which accomplishes packet reassembling in advance before handled by the kernel. It may also cause issues with stream target-based rebuilding of packets by network interface. It is recommended to knock out these features while accumulating the datasets. The dataset collected contains pcap files for the botnet capture and it includes real botnet traffic. The dataset is mixed with real botnet and normal traffic. Each scenario has been captured in a pcap file. The total number of packets, the number of malicious packets, total number of features, and the size of each pcap file are shown in Table 1. This is directly fed into the next phase for the training of datasets. We train our collected data and then repeat the training process on data for tenfolds, i.e., tenfold cross validation for testing purpose.

3.2 Feature Extraction

Feature extraction is a vital pre-processing phase for analysis of packets. It is regularly disintegrated into feature building and feature assortment [8]. Feature extraction commonly known as flow monitoring, an indispensable component in

Table 1 Datasets information

Number of datasets	24
Number of features	90
Total number of packets	An avg. of 1,20,000
Number of malicious packets	72,000
Number of normal packets	48,000
Size of dataset	Each dataset is an avg. of 68 MB
Proportion of data	70 % from real sources and 30 % from simulated sources

irregularity detection, sums up network behavior from the evaluation of a series of packet stream [11]. Network feature extraction comprises organizing all the network packet symptoms required for analysis. It is mainly used to determine several characteristics of network behavior like entire traffic analysis and mediocre connection parameters [7, 12]. We extract all the essential and appropriate features from the datasets collected to process effectively with machine learning algorithms and train these datasets using supervised machine learning classification algorithms such as MLP (Multilayer Perceptron), KNN (K-Nearest Neighbor classifier), and RF (Random Forest).

We consider the following parameters: source and destination hosts IP addresses, ports, packets information like first and last packet, total packets, resets sent between two hosts, acknowledgement packets sent between two hosts, pure acknowledgement packets sent and received, unique bytes sent and received, actual data packets between two hosts, pushed data packets sent and received, syn and fin packets sent and received, window scale, urgent data packets between two hosts, segmentation size between two hosts, missed data and truncated data information between two hosts, idle time, and throughput. Of all the features, we consider optimized features using supervised attribute selection method for detecting botnets. We consider only bi-directional flow packets. If any single-directional flow is observed in packets, they are filtered out and the remaining are used for training the algorithm. We train the algorithms first by considering all the features and also after optimizing features.

3.3 Testing Phase

We feed these classified and filtered datasets to each algorithm that we have considered and analyze the performance on each algorithm with the datasets based on the parameters like accuracy, recall, precision, and time taken. A number of malicious packets , normal packets, and datasets information are already mentioned in Table 1. For testing purpose, we use tenfold cross validation and the results are tabulated in Table 3. The performance analysis reveals which one is the best classification algorithm to detect P2P botnets. In the test environment, there is a bot master who recognizes that a particular IP is bot-ridden, launches a connection to that IP

address, and uses a dedicated bot control protocol to communicate with the septic computers. The majority of swarms which our approach detects are the infected computers that make prolonged or multiple short-term connections to a command-and-control server located somewhere remotely in the cyber space. The C&C server replies to these connections with a set of instructions to perform attacks. This approach is able to detect bots in network traffic devoid of deep packet examination, while still accomplishing greater detection rates with negligible exceptions.

4 Implementation

For testing, we use three supervised classification algorithms for detecting P2P botnets. First, we present the brief introduction to the algorithms chosen and then specify the analysis of results that are obtained from the algorithms.

4.1 Random Forest (RF)

According to Brieman, random forest algorithm is a large collection of decorrelated decision trees, where each tree is grown with respect to a random parameter. This randomization scheme is blended with bagging [2]. Bagging is to average the noisy and unbiased models to create a model with lower variance in terms of classification. Performing aggregation over ensemble will result in the final prediction. It uses the information of number of training sets and number of classifiers to determine the decision at each node while constructing tree. It uses the bootstrapping to estimate the error of the tree by predicting their classes. The construction of every single tree happens by randomly selecting features from the training set thereby evaluating the best split. This procedure is repeated on all the trees in the ensemble and the average voting of all these trees is considered as random forest prediction.

4.2 K-Nearest Neighbor (KNN)

KNN is a very simple and straightforward algorithm where it takes the input as a set of attributes and output depends upon whether the algorithm is used for classification purpose or regression purpose. It is classified based on the similarity measure. It is also called lazy learner as it does not require building a mode before its actual use. Simply to put, if we are using KNN classifier on set of n attributes, then it is classified based on the majority vote of k-nearest training records on all n attributes. Here, k indicates some user-specified constant and n indicates number of attributes.

4.3 Multilayer Perceptron (MLP)

MLP is a network of neurons called perceptron. It is a neural network with bunch of hidden layers between input and output layers. To train this type of network, it uses back propagation algorithm. Each node in the hidden layer is a function of nodes in the previous layer and the output node is nothing but the function of the node in the hidden layer [10]. With the help of back propagation technique, the input data is fed to the neural network repeatedly. In each iteration, it computes the error by comparing the output results with the desired one. Once the error is computed, it is back propagated to the network for adjusting weights thereby decreasing the error value in each iteration for convergence to the desired one.

We provide our analysis results with the help of three supervised machine learning algorithms namely random forest, KNN, and MLP. We use WEKA, a data mining environment to perform classification. Weka yields a set of machine learning (ML) algorithm which provides visualization tools for analyzing data and also for predictive modeling. Our results show very high true positive (TP) rate and very low false positive (FP) rate.

High true positive rate means that classifiers worked very well in detecting bot flows. Low false positive rate means that only a very small amount of normal web flows was misinterpreted as bot flows. We use the following performance metrics to compare our classification models:

$$Accuracy = TP + TN/TP + TN + FP + FN \tag{1}$$

$$Sensitivity\ or\ Recall\ or\ TP\ Rate = TP/TP + FN \tag{2}$$

$$F - measure = 2 * precision * recall/precision + recall \tag{3}$$

$$FP\ Rate = FP/FP + TN \tag{4}$$

$$Precision = TP/TP + FP \tag{5}$$

Here, sensitivity means true positive rate, that is, correctly detected bot flows. Precision is the proportion of correctly detected bot flows out of total number of flows classified as bot by our classifier. F-measure is test's accuracy measure.

Table 2 shows the effectiveness of different classification approaches. We applied k-cross validation where our observation revealed that k = 10 gives the best accuracy for most of the algorithms. We present the weighted average results obtained for three classification algorithms on our trained data with respect to TP rate, FP rate, and time taken to build the model.

Table 2 Average results obtained for three classification algorithms

Algorithm	FP rate	TP rate	Time taken (s) for classification phase
KNN	0.158	0.942	0.05
RF	0.257	0.937	3.04
MLP	0.083	0.906	7.71

Table 3 Performance results obtained for three classification algorithms

Algorithm	Accuracy	Precision	F-Measure	Recall
KNN	0.891	0.94	0.94	0.94
RF	0.756	0.937	0.93	0.957
MLP	0.911	0.915	0.906	0.906

In Table 3, we discuss the average results of the performance analysis of three classification algorithms on our datasets. Of all the three algorithms, MLP is showing promising results in predicting suspicious bot flows as per TP rate and FP rate.

5 Conclusion

In this paper, we provide the best classification algorithm among KNN, RF, and MLP for detecting suspicious P2P bots using machine learning technique. We perform our experimentation on real botnet datasets and also on our simulated datasets to classify P2P bots using network flow-based features and port-based analysis. Our observation results show that multilayer perceptron classifier gives very good results with an accuracy of 0.911 in detecting botnets. We propose certain features to be considered while detecting suspicious activity in the network.

References

1. Barthakur, P., Dahal, M., Ghose, M.K.: An efficient machine learning based classification scheme for detecting distributed command & control traffic of P2P botnets. p. 9 (2013)
2. Biau, G.: Analysis of a random forests model. JMLR. org, 1063–1095 (2012)
3. Gandotra, E., Bansal, D., Sofat, S.: Malware Analysis and Classification: A survey. Scientific Research Publishing (2014)
4. Haddadi, F., Morgan, J., et al.: Botnet behaviour analysis using ip flows: with http filters using classifiers. In: 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp. 7–12 (2014)
5. Li, L., Mathur, S., Coskun, B.: Gangs of the internet: towards automatic discovery of peer-to-peer communities. In: IEEE Conference on Communications and Network Security (CNS), pp. 64–72 (2013)
6. Lu, C., Brooks, R.: Botnet traffic detection using hidden markov models. In: Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research, p. 31 (2011)
7. Perényi, M., Dang, T.D., Gefferth, A., Molnár, S.: Identification and analysis of peer-to-peer traffic, pp. 36–46 (2006)
8. Rahbarinia, B., Perdisci, R., Lanzi, A., Li, K.: Peerrush: Mining for unwanted P2P traffic, pp. 194–208, Elsevier (2014)
9. Sebastian Garcia, V.U.: Malware capture facility project. http://mcfp.weebly.com/
10. Singh, K., Agrawal, S.: Comparative analysis of five machine learning algorithms for IP traffic classification. In: International Conference on Emerging Trends in Networks and Computer Communications (ETNCC), pp. 33–38 (2011)

11. Stevanovic, M., Pedersen, J.M.: Machine learning for identifying botnet network traffic (2013)
12. Strayer, W.T., Lapsely, D., Walsh, R., Livadas, C.: Botnet detection based on network behavior. Botnet Detection, pp. 1–24. Springer, New York (2008)
13. Vania, J., Meniya, A., Jethva, H.: A review on botnet and detection technique, pp. 23–29 (2013)
14. Victoria, U.: Isot research lab datasets. http://www.uvic.ca/engineering/ece/isot/datasets/
15. Zhang, J., Perdisci, R., Lee, W., Luo, X., Sarfraz, U.: Building a scalable system for stealthy P2P-botnet detection. IEEE, pp. 27–38 (2014)
16. Zhao, D., Traore, I., Sayed, B., Lu, W., Saad, S., Ghorbani, A., Garant, D.: Botnet detection based on traffic behavior analysis and flow intervals. Elsevier, pp. 2–16 (2013)

http://www.springer.com/978-81-322-2693-2

Proceedings of the 4th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA) 2015

Das, S.; Pal, T.; Kar, S.; Satapathy, S.C.; Mandal, J.K. (Eds.)

2016, XX, 729 p. 258 illus., Softcover
ISBN: 978-81-322-2693-2