Network flow-based botnet detection using supervised learning

Praveen Keshavamurthy
Department of Computer Science
Northeastern University, Boston, MA, USA - 02115
Email: keshavamurthy.p@husky.neu.edu

ABSTRACT

Botnets continue to be one of the most crucial problems on the Internet. One of the main difficulties preventing from building a timely and effective botnet detection system is the unavailability of raw network data due to administrative restrictions. In contrast, there is an abundant availability NetFlow data which demands for sophisticated solution to perform accurate botnet detection.

In this paper we have proposed a botnet detection system to classify malicious from legitimate destination IPs by using NetFlow data set. We try to capture 35 features by analyzing CTU-13 Bidirectional NetFlow dataset. The solution is evaluated against 3 machine learning models indicating the best one.

1. INTRODUCTION

Botnets are one of the most serious threats and challenging topics in the field of computer and network security. Botnets are usually a collection of very large number of hosts which are controlled by a remote adversary. The attacker communicates with the botnets using a special Command & Control(C&C) communication channel and thereby controlling the hosts. This makes the botnet very different when compared to other malwares such as virus, trojans, worms, etc. The attacker may use a wide variety of transport protocols for C&C communication such as TCP, UDP, ICMP, GRE, etc. In essence by controlling all the hosts remotely, this is an effective distributed platform which the adversary can use to send spams, launching DDOS attacks, click fraud, privacy leakage by making it hard for the defender to trace back to the adversary.

In order to successfully neutralize such attacks we need to develop an efficient system which provides high accuracy and low false positives in real time. One of large data sets available for such a system is NetFlow data set. Almost all ISPs deploy distributed network sampling systems to analyze and monitor performance across the backbone networks. In this paper we try to build classifiers to distinguish malicious destination IPs from the legitimate. We identify unique 35 unique features for each of the destination IPs by analyzing the Network flows involved. We evaluate the results by comparing 3 classifiers: Logistic Regression model, Neural Network model and Random Forest model.

The rest of the paper is organized as follows. Section 2 presents the feature description. Section 3 gives an overview of the high level approach of the implementation. Section 4 analyzes the results of the experiments. Section 5 concludes the paper by summarizing the findings and discussing future work.

2. FEATURE SELECTION

The botnet detection system proposed in this paper makes use of features from the NetFlow data set to distinguish the malicious IPs. The system has 2 phases: training phase and testing phase. During the training phase the system identifies different features per destination IP and a classification model is trained based on these feature inputs. During testing phase the classifier is run on the unlabeled data set and the prediction is verified against the ground truth data.

2.1 NetFlow Attributes

Network flows are collected by the backbone routers and forwarded to the NetFlow data collectors. These network flows include both bidirectional flows and unidirectional flows. Each flow include various statistics about the network traffic which helps in characterizing the traffic in general. Few of the attributes which are used during feature extraction include: starting timestamps of the flow, source and destination IPs, source and destination ports, protocols, total packets and total bytes exchanged during the flow, total source bytes in the flow. Since the solution is focused on distinguishing legitimate destination IPs from the malicious rather than the flow itself, the feature extraction stage extracts the necessary details for each unique destination IP by analyzing the NetFlow dataset and building an intermediate feature dataset. Also pre-processing stage identifies features related to /24 subnets which will give a key insight about the adversarial infrastructure details.

2.2 Feature Extraction

This is the pre-processing stage where 35 key features are identified from the NetFlow dataset.

2.2.1 Generic Features

The first group of features extracted from NetFlow data are general features which give information with respect to the communication protocols and different types of communications. The group consists of total number of source IPs communication involved in the communication, total number of transport protocols used, total number of bidirectional flows involved, total number of unidirectional flows involved (Client only & Server only) and a bit string representing the protocols used. The intuition for capturing these features are mainly the fact that botnet traffics are originated from only a subset of machines which are compromised and contact the C&C Servers designed over a specific protocol. On the other hand benign traffic originates from many different machines in the network and may use varying protocols.

2.2.2 Statistical Features

These group of features tries to capture the irregularities of traffic with respect to flows, packets, bytes and source bytes for both benign and C&C servers. In particular we extract total, min, max, mean, standard deviation and variance for all the four network statistics per each destination IP. Botnets communicate with C&C Servers to receive the control information

2.2.3 Subnet Features

These set of features extracts total number of IPs present in the /24 destination IP subnet along with the total number of packets and flows originating from the subnet. The intuition is adversaries normally use small set of random IPs in a given subnet. Also the traffic involved between the botnets and the C&C servers should be significantly small when compared to benign servers.

2.2.4 Periodic Communication Features

These features include the total number of source IPs which is involved in the periodic communication with the destination IPs and the ratio of total periodic source IPs to total source IPs. Botnets mainly communicate with the C&C servers to receive commands and the intuition is there might be a specific pattern in communication. This rarely occurs with benign traffic as it involves the user randomness. These features are extracted by calculating the Inter Arrival Time (IAT) of the flows. We group all the flows from a source IP to each destination IP involved in each of the 13 simulated botnet scenarios. Then the standard deviation of the IAT of the flows to particular <dst IP, dst Port> is calculated. If the standard deviation is less than the pre-defined threshold the source IP is considered to be involved in periodic communication with the destination.

Features /dst IP	Type	Feature Group
Total Source IPs	Numerical	F1
Total Protocols	Numerical	F1
Total Bidirectional Flows	Numerical	F1
Total Client Flows	Numerical	F1
Total Server Flows	Numerical	F1
Protocol Information	Bit String	F1
Total, Max, Min, Mean, Std Dev, Var of Packets	Numerical	F2
Total, Max, Min, Mean, Std Dev, Var of flows	Numerical	F2
Total, Max, Min, Mean, Std Dev, Var of Bytes	Numerical	F2
Total, Max, Min, Mean, Std Dev, Var of Source Bytes	Numerical	F2
Total IPs in each /24 subnet of dst IPs	Numerical	F3
Total Flows in each /24 subnet of dst IPs	Numerical	F3
Total Packets in each /24 subnet of dst IPs	Numerical	F3
Total periodic communications	Numerical	F4
Ratio of Total Source IPs with Periodic commn. / Total Source IPs	Numerical	F4

Table 1: List of features extracted for each destination IP in the Network Flow

3. IMPLEMENTATION AND MACHINE LEARNING MODELS

To extract the features from the NetFlow dataset we developed a map reduce program. The algorithm runs two map reduce operations in sequence: first operation groups all the destination IPs belonging to the same /24 subnet and extracts the subnet features and second operation groups the flows based on destination IPs and generates the final feature set for each unique destination IP. Using this feature set as the input to the machine learning models, we experimented with three different models namely Random Forest, Neural Networks and Logistic Regression. Random Forest algorithm which is known to be one of the most accurate algorithms out performed both the neural net and logistic regression models. We also evaluated the model by building two layer and three layer neural networks but resulted in no increase of accuracy. At last we performed a feature analysis to identify the effectiveness of the feature groups. The feature sets we evaluated are F1, F2, F3, F4, (F1,F2), (F1,F3), (F1, F4), (F2, F3), (F2, F4), (F3, F4), (F1,F2,F3), (F1,F3,F4), (F2,F3,F4) and all features. Results show that individual features are not so effective in reducing the false positives. Also F1, F3 effectively contributed to the detection of malicious destination IPs.

4. EVALUATION

The system has been realized by implementing the pre-processing stage of feature extraction using hadoop on AWS. The machine learning models and the data visualization are implemented in python using jupyter. For the machine learning library, we used scikit and tensor flow.

4.1 Dataset

For the dataset, we made use of CTU-13 bidirectional NetFlow dataset. The CTU-13 is a dataset of botnet traffic that was captured in the CTU University, Czech Republic, in 2011. The goal of the dataset was to have a large capture of real botnet traffic mixed with normal traffic and background traffic. The CTU-13 dataset consists in thirteen captures (called scenarios) of different botnet samples. On each scenario we executed a specific malware, which used several protocols and performed different actions.The dataset was collected over 9 days and contains nearly 20M flows. Among these flows 400k flows are labelled as Botnet and ~350K are labelled as Normal. The rest of the flows are labelled as Background which implies the unknown traffic during the capture. After the pre-processing stage, the final feature set consists of ~52K data samples out of which 48k are Botnet labelled and the rest are labelled as Normal. This forms the ground truth dataset for training the different machine learning models.

4.2 Experimental Setup and Evaluation Metrics

Since the feature set derived out of the pre-processing stage contained biased labelled data with 90% being botnet, we need to upscale normal data to prevent classifiers suffering from overfitting of botnet data and under-fitting of normal data. To achieve this we sampled 48k botnet data into groups of 4k. And during each iteration, we feed 4k botnet data and 4k normal data to train the classifier using 10-fold cross validation. We collect the metrics by taking the mean out of all the iterations.

We executed 1 layer neural networks with different number of hidden nodes and realized the optimal number of nodes which resulted in high accuracy was 4096 nodes. Beyond this the results saturated at ~92% accuracy. We also executed with 2-layer neural net and different combinations of hidden nodes (4096/2048, 2048/1024, 1024/512, 4096/1024) in both the layers. The result was no better than the single layer neural networks. Infact the results were equivalent to that of logistic regression classifier. For the Random Forest model, the total number of decision trees configured was 100. The rest of the configurations was default with respect to the scikit package.

The classification performance are expressed by performance metrics that describe accuracy of the models. The accuracy is expressed by following metrics.

1. Precision = TP / (TP + FP)
2. Recall = TP / (TP + FN)

where TP is total True Positives, FP is total False Positives and FN is total False Negatives.

4.3 Experimental Results

The results of experiments are shown in the table 2. It clearly shows Random Forest model clearly performed better than the neural network with an accuracy of ~99.6%. Table 3 shows the results of feature analysis.

Classifiers	Accuracy(%)	Precision(%)	Recall(%)
Logistic Regression	83.7	78.3	90
Neural Networks (1 Layer)	91.1	86.7	97.1
Random Forest	99.8	99.9	99.8

Table 2: Classification Performance of different machine learning models

Features	Logistic Regression			Neural Network			Random Forest		
	ACC	PRC	REC	ACC	PRC	REC	ACC	PRC	REC
F1	77.6	71.2	92.1	84.3	76.4	99.4	99.7	99.8	99.6
F2	50.8	21.4	0.4	54.8	62.3	60	99.6	99.9	99.3
F3	68.5	59.7	88.5	72.6	66.3	92.6	94.7	94.6	94.6
F4	50.6	0	0	50.9	45.5	90	50.6	11.2	20
F1, F2	78.2	71.8	92.1	86.9	80	98.5	99.7	99.9	99.3
F1, F3	82.8	77.2	89.3	89.9	85.5	96.2	99.5	99.9	99.1
F1, F4	78.7	72.4	92.2	85.2	77.7	98.7	99.7	99.8	99.6
F2, F3	68.9	60	88.5	71.3	70.3	83.4	99.5	99.9	99.2
F2, F4	61.8	56.6	98.4	53.8	64.6	49.9	99.6	99.9	99.3
F3, F4	68.7	59.8	88.6	73.8	67.5	92.1	94.7	94.5	95
F2, F3, F4	69	60.1	88.6	75	68.3	93.7	99.6	99.9	99.2
F1, F3, F4	83.4	78	89.9	90.4	85.8	97	99.5	99.9	99
F1, F2, F4	79.1	72.9	92.2	87.2	80.2	99	99.7	99.9	99.6
F1, F2, F3	83.2	77.9	89.6	90.8	86.7	96.5	99.7	99.9	99.4

Table 3: Results of feature analysis

We observe that generic feature group F1 and subnet feature group F3 are contributing much more when compared to aggregate features F2 & periodic communication features F4. Also when periodic communication feature set F4 is fed alone into the logistic regression classifier, the precision and recall is 0. This clearly indicates the feature set is not contributing to overall classification. This might be due to the threshold of standard deviation of IATs OR the flow IATs itself which was used as a periodic communication detector which needs to be analyzed further.

5. CONCLUSION

Botnets, as one of the most serious cyber security threats require efficient detection in order to be effectively neutralized. This paper explores how flow-based traffic analysis and supervised machine learning can be used to provide that. We developed a botnet detection system that relies on flow-level network traffic analysis and supervised MLAs for capturing patterns of malicious botnet traffic. The future work shall be devoted in analysis and identification of new and effective features with respect to periodic flows.

REFERENCES

[1] Bilge, Leyla, et al. "Disclosure: detecting botnet command and control servers through large-scale netflow analysis." Proceedings of the 28th Annual Computer Security Applications Conference. ACM, 2012.

[2] Stevanovic, Matija, and Jens Myrup Pedersen. "An efficient flow-based botnet detection using supervised machine learning." Computing, Networking and Communications (ICNC), 2014 International Conference on. IEEE, 2014.

[3] Tegeler, Florian, et al. "Botfinder: Finding bots in network traffic without deep packet inspection." Proceedings of the 8th international conference on Emerging networking experiments and technologies. ACM, 2012.

[4] Garcia, Sebastian, et al. "An empirical comparison of botnet detection methods." computers & security 45 (2014): 100-123.

[5] Stevanovic, Matija, and Jens Myrup Pedersen. "An analysis of network traffic classification for botnet detection." Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), 2015 International Conference on. IEEE, 2015.