Detecting Botnets Using Cisco NetFlow Protocol

Royce Clarenz C. Ocampo[1, *], and Gregory G. Cu[2]

[1] Computer Technology Department, College of Computer Studies, De La Salle University, Manila
[2] Software Technology Department, College of Computer Studies, De La Salle University, Manila
**Corresponding Author: royce_ocampo@dlsu.edu.ph*

Abstract: One of the threats that organizations need to combat to maintain strong information security is the evolution and increased notoriety of botnets. Detection systems currently rely on payload-based data to detect the presence of botnets in a network; however, user privacy compromise, and performance and storage overhead problems hinder their efficiency. Recent studies have introduced the use of flow-based data as an alternative to payload-based detection systems.

In this paper, a model similar to previous studies that detects botnets using flow-based data and machine learning techniques was studied and explored. To achieve this, a separate dataset of malicious and benign NetFlow data that was generated and clustered had a set of features extracted. Multiple learning algorithms were then tested on the extracted clustered dataset in order to build the detection model. Cross-validation was used to evaluate the contribution of the features towards detection, and detection models were built by using extracted features separately or in combinations. The evaluation exhibited high accuracy when using all features, while having varied results when being used separately or in combinations. Having used a fully behavioral approach in detecting malware infected nodes in the network, the model is promising as it provided a favorable detection accuracy.

Key Words: botnet, J48, machine learning, NetFlow, SVM

1. INTRODUCTION

Botnets are currently considered as one of the most sophisticated and popular types of cyber threats (Ban, 2015) by slowly becoming a powerful medium for spam, distributed denial-of-service (DDoS), identity theft, and phishing (Amini, Azmi, & Araghizadeh, 2014). Having been present since the late 90's, they were recently considered as one of the malicious cyber threats that caused huge financial losses and serious damages to companies worldwide (Paganini, 2013).

Detection methods and countermeasures against further proliferation of botnets were proposed through the years. At least three types of detection mechanisms are being proposed and used up to this day, and these are: host-based honeypot-based, and network-based detection (Gross, 2015). Among these, host-based and network-based detection are more common as they are more economical. These detection systems can address increased runtime overhead while complementing their usage by monitoring network traffic to see any indications of bot-infected machines in the network. An Intrusion Detection System (IDS) is an example of a network-based detection system which attempts to identify anomalous behavior based on network traffic (Alparslan, Karahoca, & Karahoca, 2012). IDS and other similar packet-based systems, although accurate, pose problems as new intrusion types, zero day threats, and encrypted traffic can cause inefficiencies not to mention the high network speed and resource overhead. Payload interception can also be a possibility, causing another information security breach.

The introduction of flow-based botnet detection has given the IT field an alternative solution to the problem carried about by payload-based detection systems. Existing flow-based systems have been helpful to network administrators in understanding network behavior, and utilization of network resources, network anomalies, and security vulnerabilities (So-In, 2009). Numerous studies were conducted through the years in order to build a model that will detect botnets by using NetFlow and machine learning. Bilge et. al (2012), and Kheir, et. al (2013) both proposed approaches that involved the use of NetFlow and machine learning. Both studies used detection rates to evaluate the algorithms using a labeled set of clusters.

In this paper, an approach similar to the aforementioned studies was studied, examined, and validated.

2. METHODOLOGY

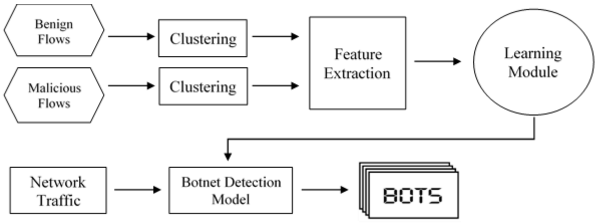

Figure 1. The model will be created and tested by following this system architecture. Labeled training datasets will be used to build detection models in the training phase (upper half), which will then be applied on network traffic to detect bots in the detection phase (lower half)

2.1 Data Collection

Separately collecting malicious and benign NetFlow data was performed as the first step. Malicious NetFlow data was synthetic, that is, the series of flows can be tagged as benign or malicious based on IP addresses. A rat/bot malware simulator (Shinog◌ۜi, 2017), and a virtual machine which served as the C&C server and bot, respectively, were used to generate malicious traffic.

Since there lacks a ground truth about the legitimacy of benign flows, benign NetFlow data was collected from the well-protected network of a computer laboratory. Terminals connected to this network abide to strict security policies, and access outside the network is monitored using a proxy server. Additionally, a Python program was created to generate traffic from a list of legitimate websites provided by Alexa (Alexa Internet, Inc., n.d.).

2.2 Clustering

The goal of this next step was to group together NetFlow records that are similar or are likely to implement the same functionalities. As shown in the work of Karagiannis et. al (2003), each protocol had different preferred packet sizes that correspond to control packets, shown by spikes in each histogram. NetFlow records were clustered using high level features: total bytes, total packets transferred, and the time difference between the start and end of each flow. The distance between two flows can be defined as the Euclidean distance between their respective features. K-means clustering algorithm was tested against the dataset of both benign and suspicious NetFlow records in order to find the best number of clusters. The Davies-Bouldin index (DBI) (Davies & Bouldin, 1979) was then used to assess the intra-cluster similarities and inter-cluster differences and to determine the optimal k number of clusters for both benign and malicious data. Clustering was then executed, and the output, a CSV file, showed and indicated the cluster where each flow data belonged.

2.3 Feature Extraction

Using high-level NetFlow data is often criticized because it provides only generic information such as port numbers and contacted IPs, and this usually leads into overfitted models. A set of features proposed in BotSuer (Kheir & Wolley, 2013) that went beyond the intrinsic characteristics of every single NetFlow record was then used. These features better describe the relationship and common trends among all records within a single cluster. The following features that were extracted from the clusters obtained from the previous step can be grouped into three categories:

Time-based features capture unusual sequences of flows and periodicities in a given cluster, and is based on the observation that observing flows for a long period of time may reveal periodicities that are unlikely to exist in benign P2P flows. Periodicities were leveraged by using the recurrence period density entropy (RPDE), a normalized metric in time series analysis that determines a signal's periodicity. It is 0 when a signal is perfectly periodic and 1 if there are white random noise signals. The mean and standard deviation for inter-flow arrival times in each cluster were also computed.

Space-based features characterize the way a P2P node contacts other peers in the network. During bootstrap, infected nodes often use hard-encoded lists of peers, implying a lower rate of new IP addresses and ports contacted by an application. They are characterized by the mean and standard deviation for the distributions of new IP addresses and destination ports contacted.

Flow size-based features characterize the number of bytes and packets transferred in a flow. These features capture specific control operations for a given P2P application. Unique and statistical flow size features were both extracted, with the former representing the distribution of unique flow sizes against the number of flows that have a given size in a cluster, and the latter characterizing the regularity of flow size behavior over time within a cluster. Both mean and standard deviation of the new distributions were computed.

2.4 Machine Learning

Upon extracting a new clustered dataset, multiple learning algorithms were tested on the dataset to build the detection model, including Support Vector Machine (SVM), an extension to nonlinear models based on statistical learning theory, and J48 and C4.5 decision tree classifiers, representing information from a machine learning algorithm that offer a way to express structures in data. To determine which algorithm is used to build the final detection model, detection rates are used to evaluate the detection rates of the learning algorithms using the labeled set of clusters.

3. RESULTS AND DISCUSSION

Cross-validation was used to evaluate the contribution of the features towards detection. Detection models were built by separately using each class or combinations of these classes. The detection accuracy, including false positives and negatives, were then evaluated.

When J48 or C4.5 was used as the algorithm, the model that used all features achieved almost 72% accuracy with a false positive rate (FPR) of 29%. When evaluated separately, time and flow size-based features achieved the same accuracy and FPR, while space-based features achieved a low detection accuracy of 43% and high FPR of 83%. The high detection rate of time and flow size-based features may be caused by the amount of time it took to execute the botnet communications. On the other hand, using SVM as the algorithm on all feature sets, although having gained the highest accuracy levels, yielded the same accuracy of 84% and an FPR of 28%. A possible reason for the similarities in accuracy rates and FPR is that the dataset that was used involved only one botnet communication simulated in the network, which effected the easy detection of the number of botnet communications. The execution time of botnet communications may have also contributed to the accuracy of the algorithm.

In comparison to BotSuer (Kheir & Wolley, 2013), where malware execution took only an hour, time and flow size-based features provided low accuracy rates, which may be an indication that the execution time was not long enough to characterize periodicities in flow data accurately. Additionally, the high detection rate by flow size-based features presented in this paper may be caused by the absence of paddings or noise to flows which only few malwares implement.

4. CONCLUSIONS

This paper studied and explored a model similar to previous studies that detects botnets that uses flow-based data and machine learning techniques. The approach that was used is fully behavioral in detecting malware infected nodes in the network, not to mention that it does not use any deep packet inspection or intrusion detection. This model was tested against synthetic malware and benign NetFlow records, and showed that the model is promising as it provided a favorable detection accuracy.

5. ACKNOWLEDGMENTS

The authors thank Dr. J. Azcarraga, G. Franco, A. Ong, and K. Solomon for their help.

6. REFERENCES

Alexa Internet, Inc. (n.d.). *The top 500 sites on the web*. (Alexa Internet, Inc.) Retrieved February 10, 2017, from http://www.alexa.com/topsites

Alparslan, E., Karahoca, A., & Karahoca, D. (2012). BotNet Detection: Enhancing Analysis by Using Data Mining Techniques. In *Advances in Data Mining Knowledge Discovery and Applications* (pp. 349-366). Rijeka: InTech.

Amini, P., Azmi, R., & Araghizadeh, M. (2014). Botnet Detection using NetFlow and Clustering. *Advances in Computer Science: an International Journal, 3*(2), 139-149.

Ban, E. (2015, May 21). *How Important Are False Positives in Measuring the Quality of an Antimalware Engine?* (Bitdefender) Retrieved July 14, 2016, from http://oemhub.bitdefender.com/importance-of-false-positives-for-antimalware-engine-quality

Bilge, L., Balzarotti, D., Robertson, W., Kirda, E., & Kruegel, C. (2012). DISCLOSURE: Detecting Botnet Command and Control Servers Through Large-Scale NetFlow Analysis. *Proceedings of the 28th Annual Computer Security Applications Conference*, 129-138.

Davies, D. L., & Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence, 2*, 224-227.

Gross, G. (2015, November 3). *Botnet Detection and Removal: Methods & Best Practices | AlienVault*. (AlienVault, Inc.) Retrieved July 16, 2016, from https://www.alienvault.com/blogs/security-essentials/botnet-detection-and-removal-methods-best-practices

Karagiannis, T., Broido, A., Brownlee, N., Claffy, K., & Faloutsos, M. (2003). File-sharing in the Internet: A characterization of P2P traffic in the backbone. *University of California, Riverside, USA, Tech. Rep.*

Kheir, N., & Wolley, C. (2013). BotSuer: Suing Stealthy P2P Bots in Network Traffic through Netflow Analysis. *International Conference on Cryptology and Network Security*, 162-178.

Paganini, P. (2013, April 8). *Botnets and cybercrime - Introduction - InfoSec Resources*. (InfoSec Resources) Retrieved July 13, 2016, from http://resources.infosecinstitute.com/botnets-and-cybercrime-introduction/

So-In, C. (2009). A Survey of Network Traffic Monitoring and Analysis Tools. *Cse 576m computer system analysis project*. Retrieved from http://www.cs.wustl.edu/~jain/cse567-06/ftp/net_traffic_monitors3/#Section1.0

Shinog◌i, S. (2017, Jan 13). *ShinoBOT -the rat/bot malware simulator-*. Retrieved from http://shinobot.com/top.php