Bots C&C Traffic Detection Using Decision Tree Based Classifier

Beena J Stuvert, Soniya B

Abstract—In recent years, the root cause of many security problems on the Internet are botnets. A botnet is a network of compromised computers under the control of bot code. When accessing a bot infected sites, these bot code are installed into the victim machine. Once the bot code affects a victim machine, it became part of the botnet. These botnets are the major cause of cyber-crimes such as spamming, phishing, click fraud etc. Bot is a type of malware and it differ from other class of malware is its command and control (C&C) channels. Thus the effective way to detect botnet is based on the command and control channels. This work presents a system that detects botnet based on the statistical features of the communication between bot and its botmasters without performing packet payload inspection. The proposed system uses machine learning technique to identify the features of the command and control channel. Based on the extracted feature a model is created to detect unknown bot traffic. Both classification and clustering methods are used to create the models and the detection accuracy and false positive rate of these methods are compared. The detection accuracy of the model is evaluated on standard real dataset, CTU-13 dataset. The experimental result shows that, both algorithms provide very good detection rate in CTU-13 dataset. Also, the false positive rate of the model is evaluated using another standard dataset, LBNL dataset. The evaluation results shows that the classification algorithm has less false positive rate compared to clustering.

Index Terms—Bot, Botnet, Command and control, Machine learning, Malware.

I. INTRODUCTION

The major attack on today's internet is through malware. Malware are used to disrupt computer operations or gather sensitive information or gain control over other computers. The most harmful malware in recent time is bots. Bot is a software program capable of performing malicious action. The network of compromised bots under the control of bot-code is called botnet. The common channel for developing cybercrimes is through botnets. These botnets are the major cause of many cybercrimes and security problems such as spam, data theft, click fraud etc.

Within the botnet, the bots are controlled by botmaster or botherders through command and control (C&C) server [1]. The bot differ from other class of malware is the use of C&C channel. Through these channels, the botmaster can update or direct the bot. These botnets are used for large number of criminal activity. Prevention of these attacks is very important. Thus the first step is to identify the C&C traffic.

The C&C traffic identification is difficult because it use normal protocol for communication; the traffic volume is less and may use encrypted communication. There are various techniques to detect bot based their C&C traffic. The bot detection can be mainly classified into signature based, anomaly based, DNS based and mining based detection. Signature based detection is an IDS based detection system that applies behavior and signatures of known botnets [2]-[4]. In anomaly based detection techniques network traffic anomalies are used to detect botnets. The detection method based on the DNS information generated by a botnet is DNS-based detection [5],[6]. The mining based detection methods use machine learning algorithms to detect bots C&C traffic.

Some of the botnet detection techniques are based on the payload inspection [3],[6],[7]. Botzilla [6] automatically extracts signatures in form of characteristic recurring payload substrings from network traces of repeated execution of a malware sample.It does not address encrypted protocols which do not exhibit characteristic strings in the cipher text. The payload based detection system is effective when botherders use normal communication. Also, the payload inspection based approach shows very high detection rate and a limited false positive rate and can be circumvented by encrypting the C&C communication. Thus, the existing botnet detection system becomes ineffective. The strength of the botnet depends on the communication with its C&C server, thus the common characteristics of the botnet depend on the network activities of the C&C channels. Thus, the better way to detect botnet or individual bot in the botnet is based on the features of the packet header information of their communication with the C&C channel.

The proposed work presents a payload independent approach that can detect unknown bot traffic even in case of encrypted C&C traffic. Machine learning algorithms are used for classification and prediction of C&C traffic flows generated by the botnets. Decision tree based algorithm is used for the classification of bot traffic. The system extracts features from the C&C network traffic of available known bot traffic and a model is created using J48 decision tree algorithm. The created model can be applied to the unknown traffic to detect bot traffic. The proposed model is compared with previous work and the detection accuracy is evaluated. The detection accuracy is evaluated using standard real dataset, CTU-13 dataset and the false positive rate is evaluated using LBNL datasets.

II. RELATED WORKS

A Command and Control system is set-up by the botmasters to communicate with his bots indirectly because it does not want its identity to be published and want to cover the command sent. Within the botnet, the bots are controlled by botmasters using the C&C server. Different types of C&C servers exist such as centralized and peer to peer. The centralized C&C architecture provide simple, low-latency

Revised Version Manuscript Received on August 19, 2015.

Beena J Stuvert, Department of Computer Science, Sree Chitra Thirunal College of Engineering, University of Kerala, Trivandrum, India.

Soniya.B, Department of Computer Science, Sree Chitra Thirunal College of Engineering, University of Kerala, Trivandrum, India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

and efficient real time communication to the botmasters. The centralized C&C use existing protocols such as IRC, HTTP for communication. The different botnet detection systems using the C&C are discussed below.

Botsniffer [7] is an anomaly based detection method mainly used for detecting IRC and HTTP botnets in LAN without any prior knowledge of signatures. The detection approach is based on the fact that the bots in the same bot family respond to the botherders command and performs activities in similar fashion. Correlation and similarity analysis algorithms are used to identify the crowd of hosts that exhibit similar response or activity pattern. Botsniffer mainly consist of two components: monitor engine and correlation engine. To examine network traffic, generates connection record of suspicious command and control protocols, and detects activity and response behavior are the main functions of monitor engine. The function of correlation engine is to analyses the spatial-temporal correlation, activity similarity or message similarity observed by monitor engine. The main advantage of this system is it does not require the previous knowledge of content signatures. However, this method has some disadvantages such as evasion by encryption, evading protocol matcher, evasion by misusing the whitelist, evasion by using very long response delay.

Botminer [8] is a detection system that uses data mining techniques for detecting botnet command and control traffic. Botminer is an improved form of Botsniffer, which is independent of the command and control protocol, structure, and infection model of botnets. Botminer mainly consist of C-plane monitor, A-plane monitor and cross-plane correlator. The similar communication pattern is captured using C-plane monitor. The similar malicious pattern is captured using A-plane monitor. Then, it clusters similar communication activities and clusters similar malicious activities. Cross–cluster correlation is performed to identify the hosts that shares both similar communication patterns and similar malicious activity patterns. Botminer can detect IRC-based, HTTP-based and P2P botnets and it also detect real-world botnets. However, it has some disadvantages such as it doesn't detect stealthier bots and can be evaded by bots using normal servers to hide the activity.

Disclosure [9] is a botnet detection technique based on the analysis of NetFlow. It is a large scale, wide area approach to detect botnet C&C servers. In this approach, the features such as flow size, client-access pattern and temporal behavior are used to differentiate C&C channel from normal traffic. These features are effective in detecting current C&C channel and also it is relatively robust against the counter measures of the future botnets. The false positive rate is reduced by incorporating an external reputation score. It is the only systems that use NetFlow data and it does not assume a prior knowledge of the particular C&C protocols.

CoCoSpot [10] is a method to group similar botnet C&C channels. Recent botnets C&C protocols can be detected using traffic analysis features such as message length, the carrier protocol as well as the encoding scheme. Thus a fingerprint can be derived using this approach. The C&C candidates can be identified using the periodicity of the message. The main advantage of this approach is that, it does not dependent on payload byte signature which enables the detection of C&C protocols with encrypted or obfuscated message contents and the system is able to produce human-readable reports, thus analysis is easy. One disadvantage of this approach is the flow clustering of C&C, which is used to discover relationships between malware families based on the distance of their C&C protocol.

BotFinder [11] is a detection system that detects bots in the network traffic without performing packet content inspection. It is a system that detects malware infection in the network traffic by comparing the statistical features of the previously-observed bot activity. The packet header information is used to create the model. It works by automatically building multi-faceted models for C&C traffic of different malware families. The high level information about the network connection is required for this approach. The advantage of this system is, it identify bots in the network even the bots use encrypted C&C communication.

III. System Overview

The proposed method detects bots based on the packet header information of the communication between bots and its botmasters. The features of packet header information are very effective in bot traffic detection even if botmasters use encrypted communication. The packet header information used to detect bot C&C traffic are average duration of the flows, the average time interval between the flows, the source byte and destination byte, the average number of packet in subsequent flows, and the fast Fourier transform. Based on these features a model is created using machine learning algorithms.

The machine learning process is divided into learning phase and testing phase. In the learning phase, the training datasets are analyzed and the underlying principles are derived. The input given to the system is network traffic packet. These packet level data is reduced into flows and features are extracted from the flows. Then a model is created in the learning phase. In the testing phase, unknown traffic is given as input to the system. The system performs the task and applies the learned knowledge to classify unknown data. The figure 1 shows the overview of the system.

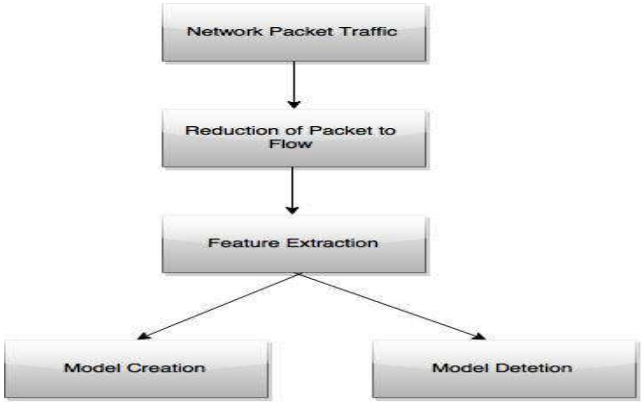

Figure.1 : Overview of the system

A. Data preprocessing

The input given to the system is traffic capture. During the training phase, malware samples are executed in a controlled environment, and all network traffic is recorded. In this step, it is important to correctly classify the malware samples from the normal traffic.

B. Reduction of packets to flow

The traffic date obtained from the network traffic monitor is packet level data. In this step, these packet level data are reassembled as flows. A flow is defined as five tuples, source

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

IP address, destination IP address, source port, destination port and transport protocol. For each flow, the properties such as start time and end time of the flows, the number of bytes transferred in total and number of packets are extracted. The data similar to NetFlow is obtained in this step. These flows are ordered chronologically to obtain an aggregated flow. The further processing steps are operates on these aggregated flow data.

C. Feature extraction

The statistical features such as time interval, duration, source byte and destination byte and fft are extracted from the aggregated flow data.

1. The average time interval between the start time of two subsequent flows in the trace. The botmasters can control its bots and give command to the bots using C&C servers. Also the botmasters makes ensure that all bots receive new commands and updates frequently. Thus most bots use constant time interval between C&C connection.
2. The average duration of the subsequent flows. The duration of the C&C trace shows similar pattern because the bots must frequently communicate with the botmasters.
3. The average number of source byte and destination bytes. By splitting up the two directions of communication using source and destination bytes, it is able to separate the request channel from the command transmission. That is, the request for an updated spam address list might always be of identical size, while the data transferred from the C&C server, containing the actual list, varies. As a result, a C&C trace might contain many flows with the same number of source bytes.
4. The Fast Fourier transformation (FFT) is calculated to find out the communication regularity. The FFT is calculated over a binary sampling of the C&C communication by assigning 1 to each connection start and 0 in between the connection.
5. The average number of packet transferred in each flow.

D. Model creation via classification

A classification technique is a systematic approach to build classification model from an input data set. The main objective of classification algorithm is to build models that can accurately predict class labels of unknown records.

J48 decision tree classifier to create model based on the extracted features. It is an entropy-based approach that build decision tree based on C4.5 algorithm [12]. The model can be created based on various features such as average duration, average time interval, average source and destination byte, fft and packet count. It builds the decision tree from labeled training data set using information gain and the splitting attribute can be chosen based on maximum information gain. The algorithm repeats for all other attributes. Then the algorithm recurs on smaller subsets. The splitting procedure stops if all instances in a subset belong to the same class. Then the leaf node is created in a decision tree telling to choose that class.

E. Model detection

Algorithms are used in detection phase, in order to classify normal traffic data and bot traffic data. Classification and prediction are the main parts in detection phase. J48 decision tree algorithm is used for model detection. The model detection is performed using Weka tool.

In the detection phase, an unknown packet traffic data is given to the system. The packet level data is reduced into flow level data and the system extracts feature from the unknown flow level data. The extracted unknown traffic feature is re-evaluated using the previously created model. If the model matches the already created model, it predicts whether the unknown traffic is bot affected or not.

IV. EXPERIMENTAL EVALUATION AND RESULTS

A. Data collection

In the training phase, malware samples are collected by executing malware.exe in a controlled environment. The malware.exe is available in open malware site. Each malware samples are executed for 6-7 hours and their traffic is recorded using Microsoft network traffic monitor 3.4. Network traffic monitor is a tool used for capturing and analysing network data. These are the hash values of some of the bot binaries used in our work.

MD5: d3da39a0f2f61ff91a16e9c78e523adc
MD5: 70d2cd94d7bbcd08e1ca0ed7e4195120
MD5: 5bde5f9acde5f74b9a597c580d977341
MD5: 5576dd168d4cef4eab642ce832f810d4
MD5: e48c44ed37f927e82864342adfc16760

B. Dataset preparation

A dataset refer to the data in a collection of closely related tables, corresponding to a particular experiment or event. In statistics, datasets usually come from actual observations obtained by sampling a statistical population, and each row corresponds to the observations on one element of that population. In every machine learning algorithms, both train dataset and test dataset are used.

a. Train dataset

The training dataset is a set of data used to discover potentially predictive relationship. In a dataset, a training set is implemented to build up a model. Our training dataset consist of the features of both bot traffic and normal traffic. The bot traffic data obtained is discussed in section 4.1. From that data, the features are extracted. Similarly, the normal traffic features are extracted. The features used to prepare train dataset are, time interval, duration, source byte, destination byte, fast Fourier transform and packet count.

b. Test dataset

A test set is a set of data used to assess the strength and utility of a predictive relationship. The test dataset given to the system is a standard real datasets, CTU-13 dataset and LBNL dataset. The CTU-13 is a dataset of botnet traffic that was captured in the CTU University. It consists of large real botnet traffic. The false positive rate of the model is evaluated using LBNL datasets. Lawrence Berkeley National Lab (LBNL) dataset is a non-malicious dataset. The LBNL is a research institute with a medium-sized enterprise network. The dataset contains trace data for a variety of network activities spanning from web and email to backup and streaming media.

C. Evaluation results and analysis

In this section, we evaluate our proposed work using machine learning algorithms to identify whether the given unknown traffic is bot or not. The final classification model can be created based on features described in section 3.3.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

WEKA Data mining environment is used for model creation. Weka provides a collection of Machine Learning (ML) algorithms and several visualization tools for data analysis and predictive modeling. High true positive rate means that the machine learning classifiers worked well in prediction of actual bot flows. Very low false positive rate shows that very few normal flows were confused as bot generated flows. We consider the following performance metrics to evaluate our model.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$Recall = \frac{TP}{TP + FN}$$

$$False\ positive\ rate = \frac{FP}{FP + FN}$$

$$F - measure = \frac{2 * recall * Precision}{recall + Precision}$$

$$Precision = \frac{TP}{TP + FP}$$

Recall is the proportion of correctly identified bot flows. False positive rate is the number of non-bot traffic is predicted as bot in the given test sample. Precision is the proportion of correctly identified bot flows out of total number of flows classified as bot by the classifier. TP is the number of instances correctly classified as a given class. FP is the number of instances falsely classified as a given class. FN is the number of false negatives and TN is the number of true negatives.

The area under ROC curve provides a alternative and better measure for machine learning algorithms. The ROC curve is given by TP rate and FP rate. ROC curve drawing algorithm use decision threshold values and construct the curve by sweeping it across from high to low. This gives rise to TP rate and FP rate at each threshold level which can be interpreted as points on the ROC curve. Area under ROC provides a good measure of comparing the performances of ROC curves in particular to the cases where dominance of one curve is not fully established. In case of perfect predictions the area under ROC is 1 and if it is 0.5 the prediction is random.

To determine the detection capabilities of the system, a cross-validation experiment is performed based on the training data. Cross validation is a model validation technique for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used for prediction and to estimate how accurately a predictive model will perform in practice. Multiple rounds of cross-validation are performed using different partitions to reduce variability. The cross-validation results of the training datasets shows 91% accuracy with false positive rate of 9%. Figure 2 shows the ROC curve of the proposed model.

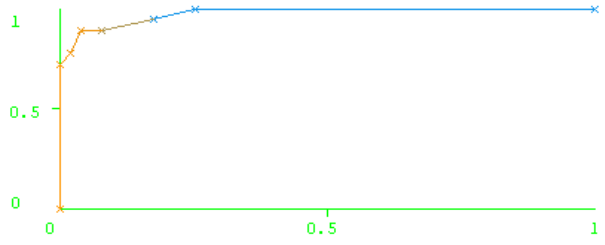

Figure 2: ROC curve of J48 decision tree (class: Botnet)

The detection rate of the proposed model is evaluated using standard dataset, CTU-13 dataset. The detection rate is the proportion of correctly identified bot flows. The evaluation result of CTU-13 based on classification and clustering is shown in table 1. For clustering, density based clustering algorithm is used. Figure 3 shows the ROC curve of CTU-13 dataset for class botnet.

Dataset	Technique	Detection rate	F-measure
CTU-13	J48 decision tree classification	1	1
	Density based clustering	1	0.857

Table 1: Detection rate of CTU-13 dataset

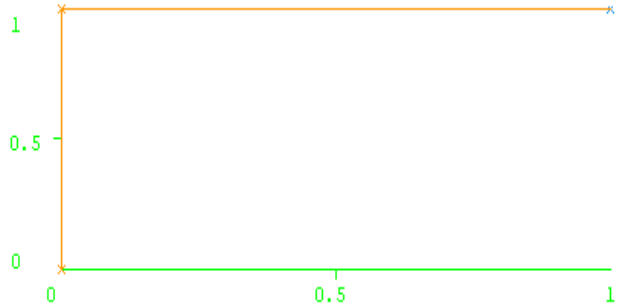

Figure 3: ROC curve of CTU-13 dataset (class: BOTNET)

The evaluation result shows a very good detection rate for CTU-13 dataset, in both classification and clustering. The ROC curve shows a straight line, ie., the area under ROC curve is 1. This shows a perfect prediction result.

The false positive rate of the proposed model is evaluated using LBNL dataset. False positive rate is the number of non-bot traffic is predicted as bot. Both, classification and clustering algorithms are used to evaluate the false positive rate of the proposed work. The evaluation result of LBNL dataset is shown in table 2. Figure 4 shows the ROC curve of LBNL dataset for class normal.

Dataset	Technique	False positive rate
LBNL	J48 decision tree classification	0.056
	Density based clustering	0.123

Table 2: False positive rate of LBNL dataset

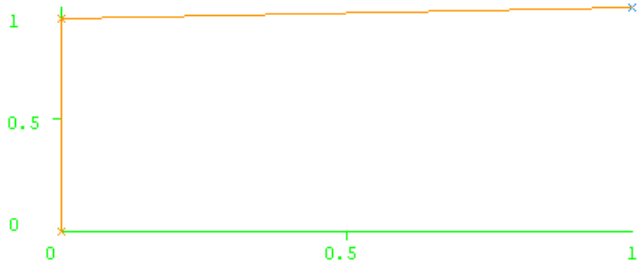

Figure 4: ROC curve of LBNL dataset (class:Normal)

The evaluation results of LBNL dataset shows that,the classification algorithm have less false positive rate compared to clustering. The area under ROC curve is 0.972. This shows a false positive rate of 0.056.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.

V. CONCLUSION

Botnets have becomes the most prominent threats on the internet and it provides the key platform for many cyber-crimes such as distributed denial of service, sending spam, stealing personal information, computing resources, identity theft etc. The strength and stability of the botnet depends on the existence of its command and control channel. Thus, the effective way to detect botnet is based on the features of command and control channels.

The proposed work presented a botnet detection system based on the features of the packet header information of the communication of the bot with its C&C server. The packet header information is very effective in botnet detection if botmasters use encrypted communication to make their bots stealthier. Based on the extracted feature, a model is created using J48 classification algorithm. The detection rate and false positive rate of our model is evaluated using two standard real-world datasets, CTU-13 and LBNL datasets respectively, and compared the result with clustering algorithm. The experimental result shows that, the detection accuracy is well in both method but, the false positive rate is less in classification algorithm compared to clustering.

REFERENCES

1. P.V. Amoli M. Safari M. Zamani H.R. Zeidanloo, M.J. Shooshtari. A taxonomy ofbotnet detection techniques. in: 3rd IEEE International Conference on ComputerScience and Information Technology (ICCSIT), 2:158–162, 2010
2. M. Dacier F. Pouget. Honeypot-based forensics. Asia Pacific Information technology Security Conference, 2004.R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” in Proc. of ACM CCS’06, 2006.
3. T. Holz J. Goebel. Rishi: identify bot contaminated hosts by irc nickname evaluation. Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, USENIX Association, Berkeley, CA, USA, page 8, 2007.
4. T. Holz J. Goebel C. Kruegel E. Kirda P. Wurzinger, L. Bilge. Automatically generating models for botnet detection. in: M. Backes, P. Ning (Eds.), ComputerSecurity – ESORICS 2009, Lecture Notes in Computer Science, vol. 5789,Springer, Berlin/Heidelberg,, page 232–249, 2009.
5. L. Khan B. Thuraisingham K. Hamlen M. Masud, T. Al-khateeb. Flow-based identification of botnet traffic by mining multiple log files. First InternationalConference on Distributed Framework and Applications,,page 200–206, 2008 .
6. T. Limmer T. Holz K. Rieck, G. Schwenk and P. Laskov. Botzilla: Detecting the phoning home of malicious software. In Proceedings of the 25th ACM Symposiumon Applied Computing (SAC), March 2010.
7. W. Lee G. Gu, J. Zhang. Botsniffer – detecting botnet command and control channels in network traffic. in: 15th Annual Network & Distributed System SecuritySymposium, The Internet Society (ISOC), San Diego, 2008.
8. J. Zhang W. Lee G. Gu, R. Perdisci. Botminer: clustering analysis of network traffic for protocol-and structure-independent botnet detection. in: Proceedings ofthe 17th Conference on Security Symposium, USENIX Association, Berkeley, CA,USA, page 139–154, 2008.
9. William Robertson EnginKirdaLeyla Bilge, DavideBalzarotti. Disclosure: Detectingbotnet command and control servers through large-scale netflow analysis.ACM, 2012.
10. Norbert PohlmannChristainJ.Dietrich, ChristainRossow. Cocospot: Clustering and recognizing botnet command and control using traffic analysis. Computernetworks, Elsevier, 2012.
11. Giovanni Vigna Christopher Kruegel Florian Tegeler, Xiaoming Fu. Botfinder: Finding bots in network traffic without deep packet inspection. ACM, 2012.
12. J. R. Quinlan, “C4.5: Programs for Machine Learning”, San Mateo CA:Morgan Kaufman, 1993.

Beena J Stuvertis currently doingher post-graduation in Computer Science and Engineering at Sree Chitra Thirunal College of Engineering under Kerala University, Trivandrum, Kerala, India. Beena received her under graduation in Computer Science and Engineering from P.A.AZIZ College of engineering under Kerala University, India in 2013. Her area of interest includes network security, computer networks and data mining.

Soniya B is working as associate professor in the department of computer science and engineering, Sree Chitra Thirunal College of Engineering, Trivandrum, Kerala. Her research interests include Intrusion Detection, Port Scan and Botnet Detection, Neural Networks and Fuzzy Systems

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication Pvt. Ltd.