International Journal of Latest Engineering and Management Research (IJLEMR)
ISSN: 2455-4847
www.ijlemr.com || Volume 04 - Issue 06 || June 2019 || PP. 103-109

Botnet Detection Based on Machine Learning Techniques

Ms.Ria A Kurian[1], Mr. Mathews Abraham[2]
[1]*Department of Information Technology*
Rajagiri School of Engineering and Technology
Ernakulam
[2]*Department of Information Technology*
Rajagiri School of Engineering and Technology
Ernakulam

Abstract: Recently, botnet detection has been a hot topic in the research areas due to the drastic increase in the malicious activities. Botnets are a network of devices that are intended to infect a large number of devices constrained by a botmaster utilizing a command and control infrastructure. Users may infect their own devices by opening email attachments, tapping on malevolent popup promotions or by downloading vulnerable software. Once the device gets infected, botnets are allowed to get to and modify individual data, attack different PCs and carry out different violations. Criminal's goal may be monetary profit, malware spread or just interruption of web. Among the methods accessible to alleviate this danger, botnet detection emerges as a relevant solution, since the early detection can diminish the dangers they pose to an extent. This paper introduces a botnet detection model based on machine learning that can identify the botnet accurately before it poses massive threats. Distinguishable patterns created by botnets within the network traffic can be effectively detected by machine learning algorithms. So here classifiers such as Decision Tree, Random Forest and Adaboost are being implemented for botnet detection and experimental results show that Random Forest produces the best overall detection accuracy rates than the other ones.

Keywords: Botnet; Botnet Detection; Feature Selection; Machine Learning.

I. Introduction

A Bot is used to represent a set of scripts or a program intended to perform predefined jobs subsequently after being activated deliberately or through a system malware. Despite the fact that bots began as a useful component for doing monotonous and tedious activities however they are being misused for malicious activities. The amount of botnets has expanded significantly in the previous couple of years and they have turned out to be one of the greatest malware dangers, in charge of a huge volume of destructive activities. Botnet attackers need to make a group of infected gadgets so as to fulfill their needs. Clients may tap on undesirable pop-ups or interfaces by which the infection get introduced on the client gadgets. Tainted gadgets work related to perform false on-line exercises wanted by the assailants. Botnet gives a key stage to cybercrimes, for example, taking of individual data, presenting DDoS assaults, sending spams and other deceitful exercises.

A bot mainly consists of three components such as bot, botmaster and command & control infrastructure. Figure1 shows a botnet architecture. The bot relates to the infected device that is under the control of an attacker. The botmaster is the attacker that claims and controls every one of the bots. The C&C foundation is the most significant piece of a botnet. The botmaster utilizes it to send and get data and directions to the bots. In order to protect themselves from getting caught, the master and the software systems are working in stealth mode which is also responsible for disabling the antivirus. Bot masters were found to be delegate in some of the modern botnet attacks, so they tend to interchange their role with another layer known as botmanagers making them hard to detect. Each bots are given a unique identification number for letting communication with the bot master, which is generally a result of the configuration and location of infected device, yet not necessarily the ip address of the system. The main communication protocols used for bot attacks are IRC (internet Relay Chat) and HTTP (Hyper Text Transfer Protocol). The principle reason behind infusing a botnet into a system is to make an army of infected machines likewise called as zombie machines.

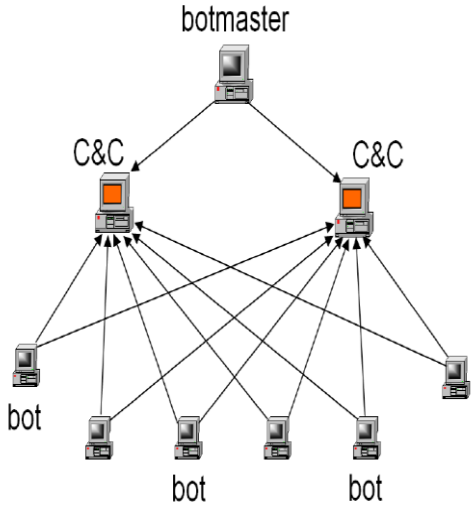

Figure 1: Botnet architecture

The overall purpose of botmaster is to steal data, financial gain or to disrupt internet completely. Since a large number of infected devices are active, the hacker can easily and quickly succeed in achieving his evil intentions, this is because setting up a botnet attack is always a low risk, high profit job. Table1 shows various types of bots and its purpose behind injecting those into the network.

Botnet Type	**Purpose**
DosBot	DoS and Distributes Dos attack utilizing Layer 3 to 7 protocols
SpamBot	Sending spam emails by gathering address books
BrowseBot	Collect user's browsing patterns and fed into advertisement network
idBot	Collect userid and password information
ChatBot	Collect the chat transcripts to find user's chatting patterns
CCBot	Gathers credit card information from ecommerce
PollBot	Manipulate online polls meant for products and services
BruteForceBot	Attack websites with TCP and application layer attacks
NetBot	Attack networks using Layer 2 and 3 protocols

Table1: Various Types of Bot attacks

The most widely recognized Botnet attacks include
Distributed Denial of Service Attacks: It is a kind of Denial-of-service attack where multiple compromised PCs focuses on a single system in order to execute a DOS attack.

Sniffing Traffic: Sniffer is an application that can capture network packets. Attackers uses this application to seize the packets that are not encrypted and finds the sensitive information such as passwords and account information inside it.

Key logging: Keystroke logging also known as keyboard capturing, is the recording of the keys struck on a keyboard, without the knowledge of user. Information can be obtained by the individual operating the logging program. A key logger can be either software or hardware.

Spyware: Any malware that are intended to gain entry over a device that captures data such as passwords and credit card information. It favors the bot herders since they could gain by selling these data in black market.

Installing Advertisement Addons: An ad fraud botnet is used to infect a user system that take over the browser process for directing benevolent traffic to targeted online advertisements.

Attacking IRC Chat Networks: A distinctive use of bots in IRC is to provide definite functionality within a network such as to host a chat-based game or to provide notifications of external events. Some IRC bots are also utilized to perform malicious activities such as denial of service, spamming etc.

Create fake website visitors: There is an automated tool known as Traffic Bot that is able to generate thousands of daily visits to a website.

Manipulating online polls/games: Botnets can be utilized to control online surveys. Each vote threw appears to have same credibility as that of what is done by a genuine individual since each bot has got unique ip address. Web based games can be controlled likewise.

Currently, there are two fundamental approaches for botnet detection. One is through setting up honeynet and other is through passive traffic monitoring. A honeypot is a trap set to identify, redirect, or in some way check counteract attempts at unauthorized utilization of Information Systems. When an intruder breaks into the host, the machine or a system administrator can analyze the interruption strategies utilized by the attacker. Two or more honeypots on a network forms a Honey net. One major application of this is the Spamtrap - a honeypot that controls spam by taking on the appearance of a sort of system mishandled by spammers.

Passive traffic monitoring is classified into signature based, anomaly based and DNS based. In signature based detection, incoming packets are analyzed and compared with a set of predefined signatures of bot binaries. If any match occurs, it will generate alerts to system administrator for taking necessary actions. In anomaly based detection, system activity is monitored to check for variations like high network latency, high volume of traffic, unusual system behavior etc. In DNS based technique, DNS traffic generated by bots is being analyzed. Since a large amount of DNS traffic is generated by users in order to find the servers, attackers find DNS suitable for hiding the malicious data inside it.

II. Literature Review

Kumar et al [2] proposed a nepenthes honeypot based botnet detection. Nepenthes are low interaction honeypots conveyed in a system that are utilized to create alerts to a system administrator for taking necessary actions to fix the security of system. It generates valuable information about bots, such as bot behavior. The result showed that the automated system effectively identified the bots spreading in the system and fix the security against these bots. However it's restricted in their ability to scale and cooperate with malicious bot behavior.

The BotMiner system proposed by Zhang et al [4] gathered all traffic based on the destination address, port number, and the anomalous behavior and events in the logs subsequent to clustering to detect the botnet, however the detection time was long and the measure of computation was also large. BotSniffer can detect botnet dependent on IRC protocol and HTTP protocol by analyzing the events and spatial relationship of zombie host activities in the same LAN , using anomalous event logs and K-Means method, the limitation is that only two explicit botnet systems can be identified, and just the central botnet can be recognized [5]. BotFinder system can recognize numerous botnet activities, yet just three botnets are identified better.

Using an alternate methodology, Bilge et al. [6] introduced the EXPOSURE system that takes into consideration of DNS traffic to detect domain names that are related with malicious practices. They utilized features like time based, TTL value based, DNS answer based, Domain name based and so on. A classifier J48 decision tree was then trained using a set of domains that are known to be malignant or benign. Result showed that EXPOSURE analyzed and classified 100 million DNS queries and recognized new malicious domains that were previously unknown to the system.

Antonakakis et al. [7] proposed a novel detection system, called Kopis which analyzes high-level DNS queries in the DNS hierarchy and considers the patterns of global DNS query responses for detecting malware-related domain names. Kopis is able to recognize malware-related domain names many days before they are incorporated in the public blacklists.

BotGrep, proposed by Nagaraja et al [8] proposed a system for detecting structured P2P botnet by analyzing network traffic behavior. This method combines honeypot and other detection mechanisms, extracts the important features of structured P2P network by gathering traffic flow and then by using random walk clustering algorithm to build sub graph of structured P2P network technology to classify the botnet and non-malicious ones .

Yahyazadeh et al [9] proposed a botnet detection based on bot behavior, BotCatch that is able to identify the bot infected hosts that are part of same botnet. It overcome the limitation of existing techniques for not recording the history of botnet activities that happened before that tends to generate false alarms. Botcatch was conveyed at the edge of the system to catch and break down the traffic between inside and outside hosts. The proposed system can detect bot-infected hosts taking part in some coordinated group activities in the beginning stages of the botnet lifecycle, regardless of whether they have not played out any malicious activities yet. The results showed that BotCatch can effectively recognize botnets with a high average detection rate of 94.45% and a average false alarm rate of 1.64%.

Knysz et al [10] introduced RB-Seeker system that utilizes a combination of linear SVM algorithm and association analysis mechanism that is suitable for sample data set detect the botnets. This strategy relates the system traffic information, spam data and DNS log and uses the linear SVM technology to detect the malicious domain name, and then related with the DNS log to discover the diverted botnet. Likewise, the identification of encoded traffic isn't subject to depth packet detection technology and the key features in C&C infrastructure can be detected by utilizing association rules technology, which can consequently distinguish the infected host. The limitation is that on one hand it can just identify a particular botnet activity, on the other hand the system has got poor scalability.

III. Proposed System

Proposed system focuses on network traffic analysis since bots needs to communicate with network during all phases. Network traffic analysis exploits the idea that bots within a botnet demonstrates consistency of traffic behavior, unique communication behavior categorized using a set of attributes which distinguishes them from benign traffic. It does not depends on content of packets and hence unaffected by encryption. Moreover, bots create unique patterns hidden in network traffic that can be easily detected by exploiting machine learning techniques. It allows automated recognition of bot related traffic without having a previous knowledge of malicious traffic character but by inferring knowledge from available bot behavior. The proposed technique takes into account of models which are Decision Tree, Random forest and Adaboost classifier. These base classifiers are trained on different subsets with the subsets being drawn from the original dataset.

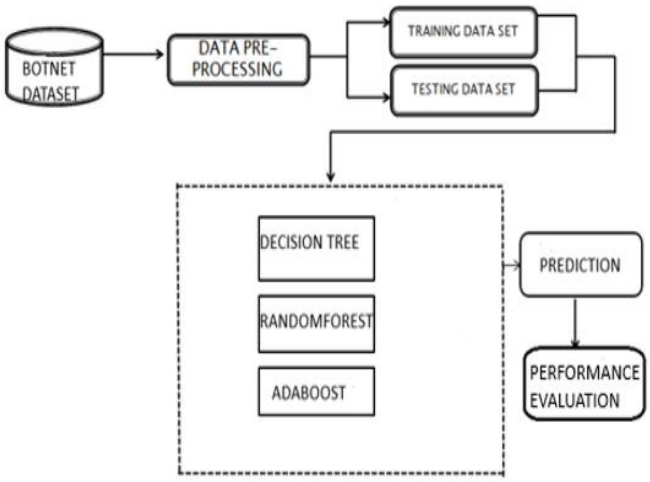

Fig 2: System Architecture

A. Data Source

Botnet Dataset is collected from CIC ISCX botnet dataset and the labelled data from CTU13 dataset. About 45 features such as source and destination ip address, port numbers, protocols, duration, total number of packets in forward and backward direction, total number of bits and bytes per packet sent in forward and backward directions, inter arrival time, flow active and idle time and other subflow features are considered.

B. Data Preprocessing

This step involves transforming the data, which involves removal of missing fields, normalization of data, and removal of outliers. Dataset is labelled malicious and non-malicious by checking the source and destination ip address from the available list of malicious ip's and IRC attacks. Then Down sampling is performed in order to eliminate class imbalance problem that has occurred. Class imbalance is a problem in machine learning where total number of class of data is far less than the total number of another class of data. After applying down sampling, number of majority class will be converted to that of minority class. Finally features that are considered essential for botnet detection such as total number of packets, total number of bits, bytes, bits per second, average inter arrival time, incoming outgoing packet ratio etc are calculated from the available features in the botnet dataset.

C. Feature Selection

Selection of appropriate features is adequate in order to accurately represent the behavior of bots. Bots within a botnet represents uniformity of traffic behavior, unique communication behavior classified using a set of attributes which distinguishes them from non-malicious traffic. Since the botmaster sends instructions to bots

to perform uniform activities, features such as number of packets, total number of bytes exchanged will be same. So here features such as total number of packets and bytes, incoming-outgoing packet ratio, bytes per second, inter arrival time are being selected that demonstrates the similarity of bot behavior and distinguishes them from the benign ones.

D. Machine Learning

Botnet create distinguishable patterns in the network traffic that are effectively identified by machine learning techniques. It allows automated recognition of botnet traffic without having a past knowledge of malicious traffic character but by understanding the knowledge from available traffic bot behavior. Here, mainly three classifiers such as Decision Tree, Random Forest and Adaboost are being used and compared their detection rates.

1. Decision Tree Classifier

Decision Tree creates a training model to predict the class of new set of data based on the decision rules inferred from the training data. Given a data of attributes together with classes, a decision tree requires to calculate its best split node. There are many different splitting criterions that can be used, such as information Gain or using Gini Coefficient. For larger datasets, most preferred is the Gini Coefficient. Gini index of each attribute is estimated and the attribute with largest reduction is taken as the root node. This procedure is repeated until the leaf node having the predicted class label is reached.

2. Random Forest Classifier

Random forest classifier creates a set of decision trees from randomly chosen raining subset. It then totals the votes from different decision trees to determine the final class of the test object. Random forest algorithm can be used for both classification and regression. Moreover over fitting problem can never happen in random forests.

3. Adaboost Classifier

Adaboost or adaptive boosting is a machine learning approach that is able to convert a set of weak classifiers into a stronger one. Combining several classifiers with choice of training set at every iteration and assigning correct amount of weight in ultimate voting can have good accuracy score for the overall classifier.

E. Performance Evaluation

Efficiency of the overall model is measured using a tool known as confusion matrix that measures the performance of each classifier on a set of data for which the true values are known. It forms a matrix layout where each row of the matrix represents the variables in an actual class while each column represents the variables in an predicted class (or vice versa).

Table 2 represents the result variables obtained from a classification.

		Predicted	
		P	N
Actual	P	TP	FN
	N	FP	TN

Table 2: Confusion Matrix

- True Positive (TP): Observation is positive and is predicted to be positive.
- False Negative (FN): Observation is positive, but is predicted negative.
- True Negative (TN): Observation is negative, and is predicted to be negative.
- False Positive (FP): Observation is negative, but is predicted positive

TP, FN, FP, TN values can be applied to evaluate output quality of classification against precision, recall and F1 score. Accuracy is the most instinctive performance measure and it is essentially a ratio of correctly predicted observation to the total observations. Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is the ratio of correctly predicted positive

observations to the all observations in actual class. F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.
Accuracy = TP+TN/TP+FP+FN+TN
Precision = TP/TP+FP
Recall = TP/TP+FN
F1 Score = 2*(Recall * Precision) / (Recall + Precision)

IV. Classification Result

Following shows the accuracies obtained using each classifier. Experimental results show that machine learning algorithms can be used effectively in botnet detection and the random forest algorithm produces the best overall detection accuracy of over 99.68%

1. Decision Tree Classifier

TP=1860	FN=7
FP = 9	TN =1952

Table 3: Performance Evaluation of botnet detection using Decision Tree

ACCURACY	PRECISION	RECALL	FSCORE
0.99582	0.995184	0.996250	0.9957173

Table 4: Performance Evaluation against accuracy, precision, recall and f1score of Decision Tree

2. Random Forest Classifier

TP =1861	FN =6
FP = 6	TN=1955

Table 5: Performance Evaluation of botnet detection using Random Forest

ACCURACY	PRECISION	RECALL	FSCORE
0.996865	0.996786	0.996786	0.996786

Table 6: Performance Evaluation against accuracy, precision, recall and f1score of Random Forest

3. Adaboost Classifier

TP = 1853	FN=14
FP=18	TN = 1943

Table 7: Performance Evaluation of botnet detection using Adaboost

ACCURACY	PRECISION	RECALL	FSCORE
0.991641	0.990379	0.992501	0.991439

Table 8: Performance Evaluation against accuracy, precision, recall and f1score of Adaboost

V. Conclusion

Botnets pose critical and developing risk against digital security. It gives key platform to numerous cybercrimes. As system security has turned out to be vital piece of our life, botnets have turned out to be most serious risk to it. Botnets are intended to infect a large number of gadgets affected by a bot master utilizing the command and control framework. Users may infect their very own systems by opening email attachments, tapping on malevolent popup promotions or by downloading hazardous programs. Once the device gets infected, botnets are allowed to get to and alter individual data, attack different PCs and carry out different breaches. Criminal's ultimate objective is often monetary profit, malware proliferation or only interruption of web. It's anticipated that the pattern will keep bringing about more gadgets infected with mining softwares and digital wallets being stolen. Among the strategies available to moderate this danger, botnet detection emerges as a relevant solution, since the early detection can diminish the dangers they pose to an extreme. So, here a botnet detection model based on machine learning technique is implemented by exploiting the network traffic analysis. Three classifiers mainly Decision Tree, Random Forest and Adaboost were implemented that could effectively identify the specific patterns created by botnets in the network traffic. The experimental results show that machine learning techniques can be effectively used in botnet detection and the random forest algorithm produces the best overall detection accuracy of about 99.6%. In the future, the proposed model can be tested with larger datasets and propose new features to improve the detection accuracy of the proposed model.

References

[1]. M. Mahmoud, M. Nir, and A. Matrawy, "A Survey on Botnet Architectures, Detection and Defences," International Journal of Network Security, vol. 17, no. 3, pp. 272-289, 2015

[2]. S. Kumar, R. Sehgal, P. Singh, Ankit Chaudhary, "Nepenthes Honeypots based Botnet Detection", Journal of Advances in Information Technology, Vol. 3, issue 4, Dec 2012,

[3]. P. Barthakur, M. Dahal, and M. K. Ghose, "CluSiBotHealer: Botnet Detection through Similarity Analysis of Clusters," Journal of Advances in Computer Networks, vol. 3, 2015

[4]. Gu GF, Perdisci R, Zhang JJ, Lee WK. BotMiner: clustering analysis of network traffic for protocol and structure-independent botnet detection. In: Proceedings of the 17th USENIX Conference on Security Symposium. San Jose, USA, 2008:139−154

[5]. Gu GF, Zhang JJ, Lee WK. BotSniffer: Detecting botnet command and control channels in network traffic. In: Proceedings of the Annual Network & Distributed System Security Symposium. San Diego, USA, 2008:1-18

[6]. Bilge, L.; Kirda, E.; Kruegel, C.; Balduzzi, M. Exposure: Finding Malicious Domains Using Passive DNS Analysis; NDSS: New York, NY, USA, 2011

[7]. Antonakakis, M.; Perdisci, R.; Lee, W.; Vasiloglou, N., II; Dagon, D. Detecting malware domains at the upper DNS hierarchy. In Proceedings of the USENIX security symposium, San Francisco, CA, USA, 8 August 2011; p. 16.

[8]. Nagaraja S, Mittal P, Hong CY, et al. BotGrep: finding P2P bots with structured graph analysis. In: Proceedings of the 19th USENIX Conference on Security Symposium. Washington, USA, 2010:7-7

[9]. Mosa Yahyazadeh and Mahdi Abadi, "BotCatch: Botnet Detection Based on Coordinated Group Activities of Compromised Hosts", 978-1-4799-5359-2/14/$31.00.2014 IEEE

[10]. Hu X, Knysz M, Shin KG. Rb-Seeker: auto-detection of redirection botnets. In: Proceedings of the Annual Network & Distributed System Security Symposium. San Diego, USA, 2009:1-17

[11]. J. Kwon, J. Lee, H. Lee, and A. Perrig, "PsyBoG: A scalable botnet detection method for large-scale DNS traffic," Computer Networks, Elsevier, In Press, Jan. 2016

[12]. H. Choi, H. Lee, and H. Kim, "BotGAD: Detecting botnets by capturing group activities in network traffic," in Proceedings of the 4th International ICST Conference on Communication System Software and Middleware, Dublin, Ireland, June 2009

[13]. G. Gu, P. Porras, V. Yegneswaran, M. Fong, and W. Lee, "BotHunter: Detecting malware infection through IDS-driven dialog correlation," in Proceedings of the 16th USENIX Security Symposium, Boston, MA, USA, August 2007

[14]. S. Lysenko, O. Savenko, A. Kryshchuk, Y. Kljots. Botnet detection technique for corporate area network. In: Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS), 2013, pp. 363-368.

[15]. F. V. Alejandre, N. C. Cortés, and E. A. Anaya, "Feature selection to detect botnets using machine learning algorithms," in Electronics, Communications and Computers (CONIELE- COMP), 2017 International Conference on. IEEE,2017, pp.1–7