AN INTELLIGENT NETWORK TRAFFIC BASED BOTNET DETECTION SYSTEM

D.Gayatri[1], Ravi Kumar Routhu[2]
[1]Student, M.TECH, [2]Assistant Professor
Department of CSE, MVGR College of Engineering

ABSTRACT

Networking is categorized as connectivity and sharing. A network connects computers and users together. "The whole is greater than the sum of its parts"- a phrase that describes networking very well, thereby allows to share information and resources along with cooperation among the devices. While sharing of resources and considering connection issues there leads to attacks regarding security like worms, viruses, Denial-of-Service etc. These days Botnet is one of a major kind of problems in networking which is extremely harmful for computer network security. Botnet is a network of compromised work stations under the control of an attacker. The proposed method involves an algorithm detecting the botnets by analyzing the Network Traffic.

Index terms: Botnet, whitelist, bot queries

I INTRODUCTION:

A network is a collection of devices which are interconnected. The devices share the information among themselves via data links available in network [13]. For a network there exists a client and a server that connect to each other through request sent to each other. Let us consider an example where the server holds a web site and on the other hand web browser is the client. On typing http://www.google.com the browser connects to server which is at http://www.google.com thereby establishing communication. Generally Network security includes provisions and policies by an administrator for controlling the unauthorized access, data manipulation, denial of network and accessible resources. With the growth of attacks thee also evolve defensive methods against these attacks in the Network Security technology. For example physical Network security, password protection, spyware, online privacy. These attacks today mainly include rapid growth of botnet activities which leads to huge loss if they increase in number.

A bot is referred as a zombie which is an individual device connected to Internet Protocol (IP) network [12]. The devices that are vulnerable are the one which become the bot (it may include computers, laptops, printers, routers etc.). Botnets are continuously emerging challenges for user confidence and security as they spread spam and overload websites for crashing and lead to financial hindrance [18]. The frequent applications of botnets include email spams, DOS attacks, spreading spyware and also data theft. Once the device is infected by botnet virus, it gets connected to bot header's command and control server (C&C)[14].

Usually the botnet detection is very typical since the bots were designed to operate without user's knowledge. Some methods do involve in detecting these botnets so far. Some of them are IRC traffic [14](botnets and bot masters use IRC for communications)Connection attempts with known C&C servers, Multiple machines on a network making identical DNS requests, High outgoing SMTP traffic (,Unexpected pop ups, Slow computing/high CPU usage, Spikes in traffic, especially Port 6667 (used for IRC), Port 25 (used in email spamming), and Port 1080 (used by proxy servers)Outbound messages (email, social media, instant messages, etc) which were not sent by problems occurred for users with internet access.

II RELATED WORK:

Wang, Jing, and Ioannis Ch Paschalidis proposed [1] a novel botnet detection method which analyzes the relationship of nodes is proposed. The method consists of two stages in which one determines the anomaly detection in interaction of graphs and the second stage determines the community detection in a social correlation graph. Here botnets are detected with sophisticated C&C channels. The method for detecting botnets are more generalized. The refined modularity also contains some limitations. **Thangapandiyan, M., and PM Rubesh Anand [**2**]** proposed for recognizing botnets in a p2p network. The proposed framework assesses the flow export out utilizing a convention called Net flow. The packet flow is investigated by utilizing exporter, authority and analyzer. Exporter catches the bundles and screens the substance and authority catches the stream activity and the analyzer investigations these movement which is gathered. What's more, this framework gives less time utilization and the recognition of these bots is high when contrasted with other. And furthermore expanded attack counteractive action**. Kong, Xinling, et al [**3**]** botnet detection technique is proposed that analyzes the packet information based on graph structure clustering. This for the most part investigations the data about the data and time stamp stream. Similitude similarity matching algorithm is proposed to quantify the flow matching degree. This is to discover the sender of each cluster that is controlled. In this manner our clustering results have high freedom. Furthermore, the detection rate is high. **Strayer, W. Timothy, et al [**4**]** paper botnets are distinguished by presenting a special architecture. In this looks at the flow characteristics, for example, data transfer capacity, duration and packet timing. It initially dispenses with the activity that is probably not going to be a piece of botnet. And furthermore correspondence designs which propose the activity of a botnet. **Bouguessa, Mohamed, Rokia Missaoui, and Mohamed Talb [5]** proposed in which community detection performed in two stages. The principal stage misuses the covariance of connections amongst hubs and the interclass inactivity keeping in mind the end goal to play out an underlying partitioning of the system. Furthermore, the second stage which recognizes these gathering of hubs. There is an enhanced nature of community detection. **Jianguo, Jiang, et al** [6] attempted to discover what highlight extractor is more powerful in the identification of a botnet. A few measures are taken for assessment of feature extractors. Tranalyzer are extremely sufficiently successful to accomplish perfect botnet detection performance and Tranalyzer is by all accounts marginally better if contrasted with Net Mate. **James R. Binkley [**10**]** introduced an anomaly based algorithm for detecting IRC-based botnet networks. The algorithm combines an IRC mesh detection part with a TCP examine detection heuristic called the TCP work weight. The IRC segment produces two tuples, one for determining the IRC work in view of IP channel names, and a sub-tuple which gathers statistics (counting the TCP work weight) on individual IRC host in channels. This algorithm has been conveyed in PSU's DMZ for over a year and has demonstrated in diminishing the quantity of botnet clients. **Li Sheng Liu Zhiming** [7] proposed a novel frame work and arithmetic for identifying botnets. This approach made out of two segments: data collection and filter. The principal area is conveyed in dispersed has with a specific end goal to catch network traffic information to filter and classify the information. Second area is deployed in centralized put which gathers all information from distributed hosts and distinguish the botnets by utilizing algorithms.

III METHODOLOGY:

A botnet is a system of computers that are compromised frequently called Zombies infected with malware that enables an attacker to control them. Every PC in a botnet is known as a bot. A bot master is the one who take control of botnets. Botnets are normally utilized for Distributed Denial-of-Service (DDOS) attacks, click fraud or spamming. DDOS attacks flood the victim with packets/requests from many bots there by consuming critical resources. On gaining the resources these refuse service to authentic clients. Some basic type of DDOS attacks are UDP, ICMP, Ping of death and so forth. It is attack in which different compromised systems attack a target and cause denial of service for users of targeted resource. Botnet attacks are omnipresent. These cause minimum loss of $10,000. On account of these losses occurred, botnet detection has got significant consideration. Generally intrusion detection

centers around individual hosts but doesn.t concentrate on preventing botnet formation. Botnets make use of IRC [20], a protocol that is developed for internet chat for commanding and controlling their bots. Here botnets are regarded as infected machines. In this case many number of botnet detection methods came into existence. Some methods includes handling botnets using flexible C&C channels. Botnets leads to DDOS attacks that aims to prevent normal communication by disabling the resources or the infrastructure that is providing the connectivity. Some of the common indications of this attack are as follows Unavailability of a resources Loss of access to a website slow performance Increase in spam e-mails

And the types of DDOS attacks include service request flood, SYN attack, ping of death, smurf etc.

The proposed algorithm identifies botnets on considering the network traffic there by differentiating botnets and whitelists individually. Whitelists are not botnets but the one which considered as trusted users. As usual botnets aim for attacking the target devices.

BOTNET DETECTION ALGORITHM:

1. Take Input file from connected Network Server.
2. Classify Network based on ip address.
3. Using packet Handler and based on header ip classify the traffic as UDP & TCP.
4. If the traffic is UDP go to step5.
5. If destination port= = 53 call process Query Method and go to step6
   Else exit/ go to step3
6. call process DNS and go to step10
7. If traffic is TCP call Store work weight method.
8. Check for IRC method
9. If packet has get or post then store http request go to step10.
   Else
   Gotostep3
10. For each host if packet count size>7 then write it into print bot queries
    Else
    Print them as whitelist.

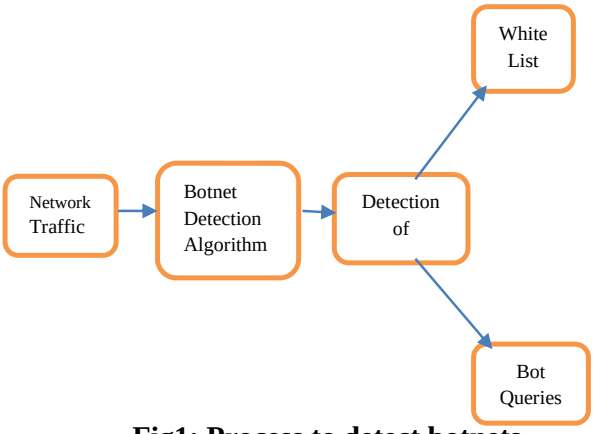

Fig1: Process to detect botnets

We take network traffic into consideration so that to identify bots in the network with the help of Botnet detection algorithm. Some methods have been used for identifying the bots. In the proposed algorithm we process queries and check IRC by using messages (like ping, pong and previous message). We also obtain threshold value and using this value the bots are detected. Finally we obtain bot queries, whitelist and host information. Based on this information obtained some calculations have been performed. The calculations include Accuracy and F-measure.

IV ENVIRONMENTAL SETUP:

The experiment is performed considering Network Traffic of a system using botnet detection algorithm to identify bots using java 1.6 along with the Eclipse Environment.

V RESULTS:

We apply the botnet detection algorithm on network traffic and it detects botnets. It produce a whitelist and bot Queries. Based on these whitelist and bot queries, we perform some calculations. In the field of machine learning confusion matrix is a table that is used for describing the performance of classification model over a set of test data there by knowing the true values. It is easily understood. It contains two rows and two columns reporting the number of false positives, false negatives, true positives and true negatives there by allowing for detailed analysis. The instances are represented by the each row of matrix in the predicted class and the columns indicates the instances of an actual class. Distribution of one variable in rows and another in columns for studying the correlation between the two variables is what contingency table stands for:

Confusion Matrix:

Table1: Confusion Matrix

	P	N
T	95	5
F	0	100

Basically regarding the table we considered true positives and false negatives. A true positive occurs when the proportion of positives were correctly identified. A true negative is one which doesn't detect the condition when the condition is absent. A false positive is on which detects the condition when the condition is absent. A false negative does not detect the condition even though the condition is present.

Precision is measured by the fraction of pairs correctly [16] put in the cluster and **recall** is considered as fraction of actual pairs that are obtained. **F-measure** is the obtained by calculating mean of precision and recall.

$$\textbf{F-measure}=\frac{2PR}{P+R}$$

Precision and Recall are calculated as follows,

$$\textbf{Precision} = \frac{TP}{TP+FP} \qquad \textbf{Recall} = \frac{TP}{TP+FN}$$

The **purity** measures the clustering in both method for recovering known classes which are applicable even though the number of clusters are different from known classes.

$$\text{purity}(\Omega,\mathbb{C}) = \frac{1}{N}\sum_{k}\max_{j}|\omega_k \cap c_j|$$

$$\textbf{Error rate}=\frac{FP+FN}{P+N}$$

$$\textbf{Accuracy}=\frac{TP+TN}{Total}$$

Where, Total = TP+TN+FP+FN

Table 2: **calculations**

Precision	Recall	F-measure	Error rate	Accuracy	Purity
1	0.48	0.65	0.51	0.5	0.97

For the above calculation the regarding graph is represented as follows.

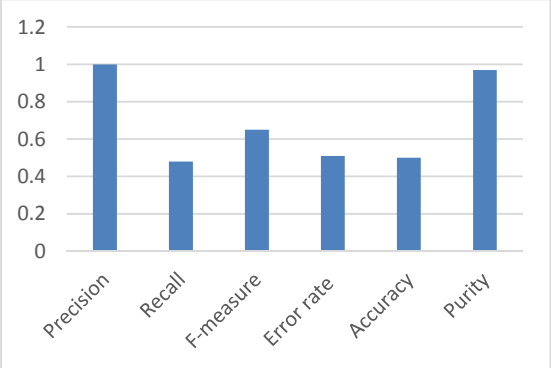

Fig2: Graphical Representation of various Matrix

VI CONCLUSION:

The detection procedure is split into two phases.1identifying Whitelist and 2.Bot queries. Based on network traffic the botnets are identified. The proposed method has shown efficiency in performance based on the purity calculated and also reduced the false positive rate.

REFERENCES:

[1] Wang, Jing, and Ioannis Ch Paschalidis. "Botnet detection using social graph analysis." *Communication, Control, and Computing (Allerton), 2014 52nd Annual Allerton Conference on*. IEEE, 2014.

[2] Thangapandiyan, M., and PM Rubesh Anand. "An efficient botnet detection system for P2P botnet." *Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on*. IEEE, 2016.

[3] Kong, Xinling, et al. "A Novel Botnet Detection Method Based on Preprocessing Data Packet by Graph Structure Clustering." *Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016 International Conference on*. IEEE, 2016.

[4] Strayer, W. Timothy, et al. "Detecting botnets with tight command and control." *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*. IEEE, 2006.

[5] Bouguessa, Mohamed, Rokia Missaoui, and Mohamed Talbi. "A Novel Approach for Detecting Community Structure in Networks." *Tools with Artificial Intelligence (ICTAI), 2014 IEEE 26th International Conference on*. IEEE, 2014.

[6] Jianguo, Jiang, et al. "Botnet Detection Method Analysis on the Effect of Feature Extraction." *Trustcom/BigDataSE/I SPA, 2016 IEEE*. IEEE, 2016.

[7] Sheng, Li, et al. "A distributed botnet detecting approach based on traffic flow analysis." *Instrumentation, Measurement, Computer, Communication and Control (IMCCC), 2012 Second International Conference on*. IEEE, 2012.

[8] Lin, Hsiao-Chung, Chia-Mei Chen, and Jui-Yu Tzeng. "Flow based botnet detection." *Innovative Computing, Information and Control (ICICIC), 2009 Fourth International Conference on*. IEEE, 2009.

[9] Liu, Dan, et al. "A P2P-botnet detection model and algorithms based on network streams analysis." *Future Information Technology and Management Engineering (FITME), 2010 International Conference on*. Vol. 1. IEEE, 2010.

[10] Binkley, James R., and Suresh Singh. "An Algorithm for Anomaly-based Botnet Detection." *SRUTI* 6 (2006): 7-7.

[11] Wang, Jing, and Ioannis Ch Paschalidis. "Botnet detection based on anomaly and community detection." *IEEE Transactions on Control of Network Systems* 4.2 (2017): 392-404.

[12] https://blog.emsisoft.com/en/27233/what-is-a-botnet/

[13] https://community.rsa.com/community/products/netwitness/blog/2017/06/09/an-introduction-to-botnets/

[14] https://www.veracode.com/security/botnet

[15] Grizzard, Julian B., et al. "Peer-to-Peer Botnets: Overview and Case Study." *HotBots* 7 (2007): 1-1.

[16] https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-of-clustering-1.html

[17] https://secure.nic.cz/files/labs/CSIRT-20091102-OS-Botnet-CEPOL.pdf

[18] https://en.wikipedia.org/wiki/Botnet

[19] Ashley, Daryl. "An algorithm for http bot detection." *University of Texas at Austin-Information Security Office* (2011).

[20] Lu, Wei, and Ali A. Ghorbani. "Botnets detection based on irc-community." *Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE*. IEEE, 2008.