HTTP BOTNET DETECTION USING FREQUENT PATTERNSET MINING

S.S.Garasia[1], D.P.Rana[2], R.G.Mehta[3]

[1]*M.Tech, Computer Department, SVNIT, Surat, India,* ***p10co969@coed.svnit.ac.in***
[2]*Assistant Professor, Computer Department, SVNIT, Surat, India,* ***dpr@coed.svnit.ac.in***
[3]*Associate Professor, Computer Department, SVNIT, Surat, India,* ***rgm@coed.svnit.ac.in***

Abstract

Among the diverse forms of malware, Botnet is the most widespread and serious threat which occurs commonly in today's cyber-attacks. A botnet is a group of compromised computers which are remotely controlled by hackers to launch various network attacks, such as DDoS attack, spam, click fraud, identity theft and information phishing. The defining characteristic of botnets is the use of command and control channels through which they can be updated and directed. Botnet has become a popular and productive tool behind many cyber-attacks. Recently malicious botnets evolve into HTTP botnets out of typical IRC botnets. Data mining algorithms allow us to automate detecting characteristics from large amount of data, which the conventional heuristics and signature based methods could not apply. Here, a new technique for botnet detection is presented that makes use of Timestamp and frequent patternset generated by the Apriori algorithm. The point that distinguishes our proposed detection technique from many other similar works is that there is no need for prior knowledge of Botnets such as Botnet signature.

Index Terms: *Botnet detection, HTTP botnet, Frequent patternset mining, Cyber-security, Malicious activity*

---------- *** ----------

1. INTRODUCTION

Botnets have become one of the most malicious threats over the Internet. A botnet is a group of compromised computers also called bots or zombies which are controlled by the botmaster's malicious code [1]. Botnets can be facilitated for spamming, traffic sniffing, key-logging, information harvesting and DDoS attacks [2]. More new types of attacks are invented based on botnets. The detection of botnet has been a major research topic in recent years. Different techniques and approaches have been proposed for detection and tracking of botnet.

The main difference between other kind of malware and botnet is command and control (C&C) channel. The C&C channel allows bots to receive commands and perform malicious activity. Botmaster also maintain and update their bot through C&C channel. For instance, they may need to update the bot binary to evade detection techniques or they may intend to add new functionality to their bot army. Moreover, sometimes the updated binary move the bots to a different C&C server. This process is called server migration and it is very useful for botmasters to keep their botnet alive [3]. C&C is the weakest link of the entire botnet. Botmaster must ensure that their C&C infrastructure is sufficiently robust to manage thousands of distributed Bots across the globe, as well as resisting any attempts to shut down the Botnets [4]. Recently, attackers are also continually improving their approaches to protect their Botnets.

This paper presents a new technique for detection of HTTP botnet using frequent patternset mining using Apriori and timestamp. This paper is organized as follows. Related work in this field is discussed in section 2. The proposed work is explained in Section 3. The experimental results are presented in section 4 followed by conclusion is in section 5.

2. RELATED WORK

Many researchers propose botnet detection methods, built by taking into account properties such as the propagation mechanism, the vulnerability exploitation strategy, the type of command and control (C&C) channel used or the set of commands available to the botmaster.

Some existing detection methods based on analysis of DNS (domain name system) queries are developed. In which botmaster connects to C&C channel and send command to their bot army. Choi [5] proposed the botnet detection by monitoring group activities in DNS traffic. He said that the botnet can evade detection algorithms when the botnet uses DNS only at initializing and never use it again (e.g., some HTTP botnets). Jae-Seo Lee [6] proposed a method to detect HTTP botnets based on degree of periodic repeatability. Typical IRC bot maintains connection and doesn't reconnect

after the first connecting to a C&C server. On the other side, HTTP botnets are not maintaining connection with a C&C server. Malicious HTTP bots connect repeatedly with a regular interval, which is configured by botmaster. Degree of periodic repeatability is used to find out repeatability between client and HTTP server. If degree of periodic repeatability is low, it means high periodic repeatability. If not, it is not repeatable periodically.

Data mining aims to recognize useful patterns to discover regularities and irregularities in large data sets. In fact, data mining techniques enables to extract sufficient data for analysis. One effective technique for botnet detection is to identify botnet C&C traffic. Botmasters maintain and redirect their connection through C&C channel and C&C is the weakest link in the entire botnet cycle. However, botnet C&C traffic is difficult to detect. In fact, since botnets utilize normal protocols for C&C communications, the traffic is similar to normal traffic. Moreover, the C&C traffic is not high volume and does not cause high network latency. Therefore, anomaly-based and DNS-based techniques are not useful to identify botnet C&C traffic. A wide range of data mining techniques such as machine learning, data aggregation, classification and clustering can be used efficiently to detect botnet C&C traffic.

Classification is a data mining technique used to predict group membership for data instances. Classification is the problem of identifying which of a set of categories (sub-populations) a new observation belongs, on the basis of a training set of data containing observations (or instances) whose category membership is known. The individual observations are analysed into a set of quantifiable properties. Classification is used to detect abnormal activities in a network. Classification algorithms assume that incoming packet will match one of the previous patterns. Therefore, it is not an appropriate approach to detect new attacks. In other words, detecting new attacks cannot depend on the current set of classification rules [7].

Clustering is a well-known data mining technique where data points are clustered together based on their feature values and a similarity metric. Clustering differs from classification, in that there is no target variable for clustering [8]. The clustering task does not try to classify, estimate or predict the value of a target variable. Instead, clustering algorithms divide the entire data set into subgroups or clusters containing relatively identical features. Thus, clustering provides some significant advantages over the classification techniques, since it does not require a labelled data set for training [9]. In this case, cluster is a collection of objects that are similar to one another and dissimilar to objects in other clusters. In each cluster, the similarity of the objects within the cluster is maximized, whereas the similarity to objects outside this cluster is minimized. The proposals in [10] are good clustering approaches for outlier detection based on distance.

Machine learning, a branch of artificial intelligence, is a scientific discipline concerned with the design and development of algorithms that allow computers to evolve behaviour based on empirical data, such as from sensor data or databases. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables [11]. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviours given all possible inputs is too large to be covered by the set of observed examples (training data). Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new cases.

There is also another data mining technique to obtain a summary count of traffic matching a particular pattern. This method is called data aggregation that enables to get a more complete picture of the information by collecting and analysing several types of records from different channels simultaneously. Furthermore, aggregation summarizes data by combining two or more attributes or objects into a single one. Comparing to other data mining techniques, aggregation is generally more expensive in terms of computation as two or more attributes are involved [12]. Nevertheless, efficiency can be improved by selecting appropriate aggregations of attributes and statistics. Regardless of using more fields, aggregation reduces the downstream volume of data which is helpful for analysing large datasets [13].

Masayuki Ohrui [14] proposed a method for detecting botnet coordinated attacks using Apriori and PrefixSpan algorithms. Apriori deals with subset of events without considering the order of events, it has high false positive ratio and while PrefixSpan considers the sequence of events that was ignored in Apriori. Apriori algorithm is applied to generate frequent item sets and Prefix Span is applied on these frequent item sets to generate sequence of patterns.

3. PROPOSED WORK

This paper presents a technique to identify botnet's command and control (C&C) channel where all the network traffic is directed by botmaster. Proposed framework is based on passively monitoring network traffic. Fig-1 shows the architecture of proposed HTTP botnet detection system, which consist of 4 main components: Traffic Monitoring, Filtering, Network Traffic Separator and Malicious Activity Detector.

3.1 Traffic Monitoring

Traffic Monitoring is responsible to detect the group of hosts that have similar behaviour and communication pattern by inspecting all network traffic. Packet sniffer is utilized to monitor the network flow and to record this information. Each flow record has following information: Source IP address, Destination IP address, Source Port, Destination Port, Number of packets transferred in both directions, Time of packet received and transferred to particular IP address.

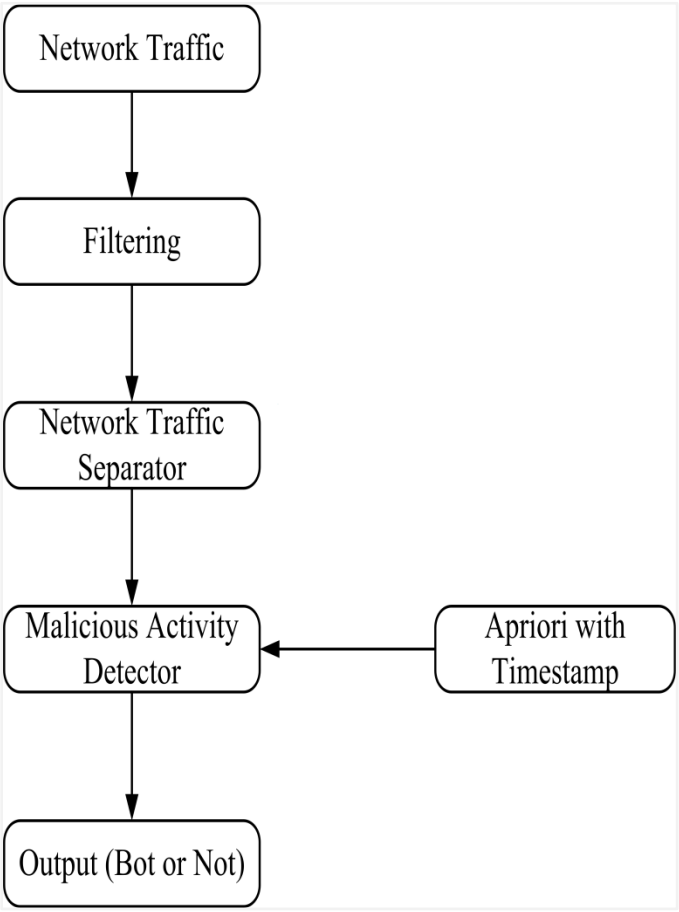

Fig. 1: Architecture overview of proposed detection framework.

3.2 Filtering

Filtering is responsible to filter out irrelevant traffic flows. The main objective of this part is to reduce the traffic workload and makes the rest of the system perform more efficiently. So, the traffic established from external host to internal hosts and from internal hosts to external hosts are kept.

3.3 Network Traffic Separator

Network Traffic Separator is responsible to separate HTTP traffic from the rest of traffic and sends them to centralized part. Like most network protocols, HTTP uses the client-server model and HTTP protocol: An HTTP client opens a connection and sends a request message to HTTP server (e.g. "Get me the file 'www.svnit.ac.in/home.html'"); the server then returns a response message, usually containing the resource that was requested. After delivering the response, the server closes the connection. In the format of HTTP request message, HTTP methods are to be focused. Three common HTTP methods are "GET", "HEAD" or "POST". HTTP bots connects to their C&C periodically (e.g. Kelihos Botnet, Black Energy botnet). Therefore, the traffic is inspected and if the first few bytes of an HTTP request contain "GET", "POST" or "HEAD"; it's the indication of HTTP protocol and separating those flows and redirect them to Centralized part.

3.4 Malicious Activity Detector

This part is to detect malicious activity generated by botmaster. Data mining technique is used for extracting suspicious activity.

Apriori is a well-known algorithm for association rule discover. The Apriori can be used to detect the association rule for the botnet detection. It was designed to detect significant correlation of set of items for extracting rules of items with high support (a fraction of the subset of items). The support is useful feature for detecting all possible behaviours among servers. However, since Apriori deals with *subset of events* without considering the order of events, it has high false positive ratio. For instance, a sequence of events x and then y is equivalent to one of y and x in Apriori. The detected patterns in Apriori contain some false coordination that two independent servers happened to work at almost same time by chance. Hence, its confidence is not so high. So, Timestamp is used to find order of event for frequent patternset generated by Apriori.

3.4.1 Apriori Algorithm

Apriori is a classic algorithm to generate association rules. Apriori is designed to operate on databases containing transactions. As is common in association rule mining, given a set of item sets, the algorithm attempts to find subsets which are common to at least minimum number candidates of the item sets. The algorithm terminates when no further successful extensions are found.

The day is divided into 24 timeslots. So there would be 24 timeslots in a day as a transaction id. After that adding filtered host IP address as item sets according to particular timestamp. The following Table-1 shows a sample network captured on April 24, 2012 with particular timestamp on machine with 172.24.*.* IP address. * is used to hide IP significance.

Time	Source IP Address	Destination Port	Protocol
8:23	172.168.*.*	45662	TCP
8:30	76.12.*.*	80	HTTP
8:45	76.12.*.*	80	HTTP
8:53	67.225.*.*	80	HTTP
11:05	192.168.*.*	80	HTTP
11:12	145.12.*.*	12653	TCP
11:30	76.12.*.*	80	HTTP
11:45	76.12.*.*	80	HTTP
14:30	76.12.*.*	80	HTTP
14:45	76.12.*.*	80	HTTP
17:10	143.12.*.*	2543	TCP
17:17	202.160.*.*	80	HTTP
17:30	76.12.*.*	80	HTTP
17:45	76.12.*.*	80	HTTP
20:02	207.77.*.*	80	HTTP
20:24	64.89.*.*	1534	TCP
20:30	76.12.*.*	80	HTTP
20:45	76.12.*.*	80	HTTP
20:54	192.168.*.*	1678	TCP
23:17	75.21.*.*	80	HTTP
23:30	76.12.*.*	80	HTTP
23:45	76.12.*.*	80	HTTP
23:47	45.12.*.*	80	HTTP

Table-1 Sample Network Capture

The pseudo code for the algorithm is given below for a transaction database D and a support threshold of €. T represents tuples in one timestamp. Usual set theoretic notation is employed; though note that D is a multi-set. C_k is the candidate set for level k. Generate () algorithm is assumed to generate the candidate sets from the large item sets of the preceding level, heeding the downward closure lemma. Count[c] accesses a field of the data structure that represents candidate set c, which is initially assumed to be zero. Many details are omitted below, usually the most important part of the implementation is the data structure used for storing the candidate sets and counting their frequencies.

Apriori (D,€)
$L_1 \leftarrow$ {large 1—item sets}
k$\leftarrow$2
While $L_{k-1} \neq \emptyset$
$C_k \leftarrow \{c | c \in a \cup \{b\} \Lambda a \in L_{k-1} \Lambda b \in \cup L_{k-1} \Lambda b \notin a\}$
for transactions t ∈ T //For each time slot data
$C_t \leftarrow \{c | c \in C_k \Lambda c \subseteq t\}$
for candidates c ∈ C_t
count[c] ← count[c] +1
$L_k \leftarrow \{c | c \in C_k \Lambda \text{count}[c] \geq \in\}$
k← k+1
return $\bigcup_k L_k$

The following Table-2 shows filtered IP addresses on particular Timestamp. Apriori algorithm is applied to each Timestamp to get frequent IP address. Rules generated by Apriori with high support and confidence are kept.

Timestamp	Filtered IP address	Protocol
1	172.24.*. 76.12.*.* 76.12.*.* 172.24.*. 67.225.*.* 172.24.*.	HTTP
2	172.24.*.192.168.*.* 76.12.*.* 172.24.*. 172.24.*. 76.12.*.*	HTTP
3	76.12.*.* 172.24.*. 172.24.*. 76.12.*.*	HTTP
4	172.14.*.* 202.160.*.* 76.12.*.* 172.14.*.* 76.12.*.* 172.14.*.*	HTTP
5	207.77.*.* 72.14.*.* 76.12.*.* 172.14.*.* 172.14.*.* 76.12.*.*	HTTP
6	75.21.*.* 172.14.*.* 172.14.*.* 76.12.*.* 76.12.*.* 172.14.*.* 45.12.*.* 172.14.*.*	HTTP

Table-2 Filtered IP address with Timestamp

4. EXPERIMENTAL RESULT

The proposed system is setup on a machine with the following hardware: Intel Core i3 processor @ 2.26 GHz, 3 GB RAM, 500 GB HDD and software: Ubuntu Linux 11.10 Oneiric Ocelot and Linux kernel 3.0.0

Apriori is used to generate frequent patterns for potential host address and timestamp is used for sequence of patterns. For Apriori, minimum support 20and minimum confidence 20 are taken. Right side of rule represents local machine address, here in our case it is172.24.*.* and left side of rule represents host IP address. For example, On April 24, 2012 to April 26, 2012 Apriori detects following patterns for botnet detection as shown in Table-3.

Table-3 Experimental Results

Date	Rule	Support	Confidence
April 24, 2012	172.168.*.*←172.24.*.*	60	25
	76.12.*.*← 172.24.*.*	80	100
	207.77.*.*← 172.24.*.*	50	40
	67.225.*.*← 172.24.*.*	60	50
April 25, 2012	145.12.*.*← 172.24.*.*	25	50
	76.12.*.*← 172.24.*.*	70	80
	192.168.*.*← 172.24.*.*	60	30
April 26, 2012	76.12.*.*←172.24.*.*	90	80

Table-3 Experimental Results

The experimental results in Table-3 shows that pattern 76.12.*.*← 172.24.*.* has high support and confidence in all three days. So, Host IP address 76.12.*.* could be malicious C&C botnet server.

5. CONCLUSION

With the sharp rise in computer network attacks through botnets, current security monitoring tools will not be insufficient for efficient botnet traffic detection. Therefore, an effective tool that can detect malicious botnet traffic and alert the user is highly demanded. This paper presented HTTP botnet detection system by combining data mining technique and timestamp. In proposed detection technique, incoming and outgoing network traffic is monitored then network traffic filtering and separation is done. Apriori with timestamp is used to detect malicious C&C channel. In addition, further research will be carried out on the time duration and other parameter like response with large dataset.

REFERENCES

[1]. H. Zeidanloo, M. Shooshtari, P. Amoli, M. Safari, and M. Zamani,"A taxonomy of botnet detection techniques," in Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol. 2, pp. 158 –162, 2010

[2]. D. Moore, G. Voelker, and S. Savage, "Inferring internet denial-ofservice activity," in In Proceedings of the 10th Usenix Security Symposium, pp. 9–22, 2011

[3]. G. Gu, V. Yegneswaran, P. Porras, J. Stoll, and W. Lee. " Active Botnet probing to identify obscure command and control channels". In Proceeding of Annual computer security application conferences, asac, pp.241-253, 2009

[4]. Zeidanloo, H.R.; Manaf, A.A. "Botnet Command and Control Mechanisms",. Second International Conference on Computer and Electrical Engineering, 2009. ICCEE '09. Page(s): 564 - 568 , 2009

[5]. Hyunsang Choi, Hanwoo Lee, Heeko Lee, Hyorgon Kim, "Botnet Detection by Monitoring Group Activities in DNS Traffic", 7th IEEE ICCIT, pp. 7 15-720, 2007

[6]. Jae-Seo Lee, Tung-Ming Koo, Hung-Chang Chang, "Construction P2P firewall HTTP-Botnet defensemechanis" , IEEE, PP. 33-39, 2011

[7]. W. Strayer, D. Lapsley, B. Walsh, and C. Livadas, Botnet Detection Based on Network Behavior, ser. Advances in Information Security. Springer, PP. 1-24, 2008

[8]. P. Dokas, L. Ertoz, V. Kumar, A. Lazarevic, J. Srivastava, P. Tan, "Data Mining Methods for Network Intrusion Detection" in *Proc. NSF Workshop on Next Generation Data Mining*, 2002

[9]. E.M. Knorr and R. T. Ng. "Algorithms for Mining Distance-Based Outliers in Large Datasets", in *Proc. 24th Int. Conference on Very Large Databases*, pp. 392-403, 1998

[10]. S. Ramaswarny, R. Rastogi, and K. Shim. "Efficient Algorithms for Mining Outliers from Large Data Sets", in *Proc. ACM Sigmod 2000 Int. Conference on Management of Data*, 2001

[11]. M. M. Masud, T. Al-khateeb, L. Khan, B. Thuraisingham, and K. W. Hamlen, "Flow-based Identification of Botnet Traffic by Mining Multiple Log Files," in *Proc. International Conference on Distributed Frameworks & Applications (DFMA)*, Penang, Malaysia, October 21-22, 2008

[12]. H. Weststrate, "Botnet detection using netflow information Finding new botnets based on client connections" in *Proc. 10thTwente Student Conference on IT*, 2009

[13]. T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "BLINC:multilevel traffic classification in the dark," In Proceedings of the 200S Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 229-240, Philadelphia, Pennsylvania, 2005

[14]. Masayuki Ohrui and Hiroaki Kikuchi, "Apriori–PrefixSpan Hybrid Approach for Automated Detection ofBotnet Coordinated Attcks" International Conference on Network-Based Information Systemsacks, pp 92-97, 2011

BIOGRAPHIES

Samir S. Garasia
M.Tech Scholar,
Computer Department,
SVNIT,
Surat,
India

Dipti P. Rana
Assistant Professor,
Computer Department,
SVNIT,
Surat,
India

Rupa G. Mehta
Associate Professor,
Computer Department,
SVNIT,
Surat,
India