Evidence Fusion for Malicious Bot Detection in IoT

Moitrayee Chatterjee
Computer Science Department
Texas Tech University, TX, USA
Email: moitrayee.chatterjee@ttu.edu

Akbar Siami Namin
Computer Science Department
Texas Tech University, TX, USA
Email: akbar.namin@ttu.edu

Prerit Datta
Computer Science Department
Texas Tech University, TX, USA
Email: prerit.datta@ttu.edu

***Abstract*—Billions of devices in the Internet of Things (IoT) are inter-connected over the internet and communicate with each other or end users. IoT devices communicate through messaging bots. These bots are important in IoT systems to automate and better manage the work flows. IoT devices are usually spread across many applications and are able to capture or generate substantial influx of big data. The integration of IoT with cloud computing to handle and manage big data, requires considerable security measures in order to prevent cyber attackers from adversarial use of such large amount of data. An attacker can simply utilize the messaging bots to perform malicious activities on a number of devices and thus bots pose serious cybersecurity hazards for IoT devices. Hence, it is important to detect the presence of malicious bots in the network. In this paper we propose an evidence theory-based approach for malicious bot detection. Evidence Theory, a.k.a. Dempster Shafer Theory (DST) is a probabilistic reasoning tool and has the unique ability to handle uncertainty, i.e. in the absence of evidence. It can be applied efficiently to identify a bot, especially when the bots have dynamic or polymorphic behavior. The key characteristic of DST is that the detection system may not need any prior information about the malicious signatures and profiles. In this work, we propose to analyze the network flow characteristics to extract key evidence for bot traces. We then quantify these pieces of evidence using apriori algorithm and apply DST to detect the presence of the bots.**

Keywords- Internet of Things (IoT), Botnet, Big data, Bots, Dempster-Shafer Theory, Apriori Algorithm, Cyber Security.

I. Introduction

An attacker can remotely create and control a network of bots, which are typically devices infected with malicious applications. The adversary (i.e., the bot master) can employ this network of bots (botnet) to conduct a series of adversarial activities such as stealing data, launching distributed denial of service (DDoS) attack, spreading spams, and phishing attacks. All the bots in the network are connected to a central communication system and receive commands before executing malicious actions on targeted systems. Hence, botnets require a configuration to obtain directives from the attacker. Internet Relay Chat (IRC) is often implemented in a client server-based botnet architecture but they are prone to easy detection as they rely on a highly centralized architecture. Bots have evolved to act in a Peer to Peer (P2P) architecture in order to yield to a better obfuscation. The Command and Control (C&C) channel uses a legitimate HTTP session to establish communication between the bot master and the computer.

In this work, we propose a network flow analysis approach based on evidence theory for bot detection. We aggregate the behavior of the network activity and classify malicious from benign. This way, bots can be detected and contained even before they can attack the target; that is to say, during the C&C period of a bot life cycle.

Advantages of the proposed approach includes ability to include new evidence. DST has the unique ability to handle uncertainty, i.e. in case of no support or absence of evidence for a new event, it does not assign a likelihood to support the incident. Rather, it assigns probabilistic support *(belief function)* to the evidences, that support an outcome. Hence, in case of polymorphic or dynamic bot detection, often the system lacks prior information about bots where DST can be a powerful identification tool. Use of *DST based robust and flexible classifier* makes the approach extensible and complimentary to other existing approaches. The evidence quantification of the DST-based classifier can be achieved using data mining algorithms or rule based engine or other machine learning approaches depending on the structure of the data. Further in this paper, we present an overview of our proposed approach, followed by experiment details and conclusion.

II. Related Work

Leonard et al. state that botnets usually go through a life-cycle consisting of four phases: formation, Control and Compromise (C&C), attack, and post-attack phases [2]. The first phase of the bot is to discover vulnerabilities of a target system. In next phase the vulnerability is exploited and the host is compromised. This phase gives access to the bot master to control the target machine. Next the bot installs some binaries or executable programs which are essentially malicious and in this phase, the target machine turns into a bot. The last phase is a preventive step, during which the bot employs defense mechanism against detection and removal.

Botnet detection is an ever-evolving extensive area of cybersecurity research. García et al [10] presented a detailed survey on botnet detection. Their topology map for network-based bot identification outlines the various anomaly and finger print based detection methods for bots. The work presented in this paper, falls under the anomaly based bot detection approach.

BotHunter [3] is a closed source framework which associates alerts from the Snort Network Intrusion Detection

978-1-5386-5035-6/18/$31.00 ©2018 IEEE

and Prevention System[1] framework with bot exercises. In particular, BotHunter leverages the way that all bots share a standard set of fundamental activities as a feature of their life-cycle: inbound scanning, attack, binary download, C&C and outbound scanning. BotHunter screens a system and apprehends actions recognized with port examinations, outbound checks and plays out some payload examinations and malware identification based on Snort principles, and afterwards utilizes a correlation mechanism to create a score for the likelihood of a system being infected by bots.

Botminer [5] depends on the group behavior of individual bots inside a botnet for its identification. It exploits the uniform behavior of botnets and recognizes them on various machines in a network, by clustering comparable behavior. Botminer first groups analogous communication in the purported C&C correspondence (referred as C-plane). Once analogous flows have been distinguished, Botminer groups activity (referred as A-plane) stream by means of Snort. By combining the A-Plane and C-Plane, Botminer contrasts and recognizes a botnet from a legitimate machine in network.

III. The Model

The consistency in the botnet behavior and communication mode is notable and can be exploited toward their recognition. The aim of our approach is to exploit the uniform behavior of various machines, and then distinguish machines that are part of a botnet when they start to perform malicious activities simultaneously. The proposed approach works in four different phases:

1) Phase 1: Network traffic analysis and feature extractions.
2) Phase 2: Applying apriori algorithm [4] on the key(features)-value(ranking) pairs to find out the most infrequent feature subsets in the network. The initial hypothesize is that any atypical behavior in the network could be the indication of malicious behavior.
3) Phase 3: The next step involves assigning subjective probability of those subsets, i.e., the most infrequent being the most suspicious.
4) Phase 4: Dempster Rule of combination to identify the presence of bots in the network.

Figure 1 presents the block diagram of the proposed framework. We discuss each of the aforementioned phases in detail in the following subsections.

A. Phase 1: Network traffic analysis and feature extractions.

1) Network traffic analysis: Network traffic analysis uses the idea that bots within a botnet typically demonstrate uniformity of traffic behavior, present unique communications behavior, and that these behaviors may be characterized and classified using a set of attributes which distinguishes them from non-malicious traffic and techniques. Traffic analysis does not depend on the content of packets and is therefore unaffected by encryption. There exists dedicated hardware which may extract this information with high performance without

[1]https://www.snort.org

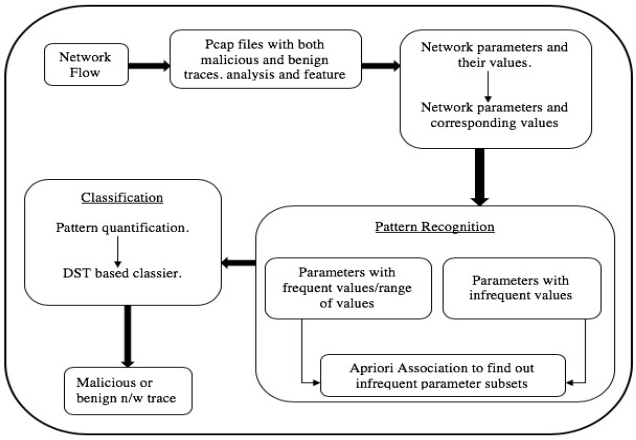

Fig. 1. Proposed bot detection framework.

significantly impacting the network. Typical traffic analysis based detection systems examine network traffic between two hosts in its entirety. While this approach is feasible for offline detection, it is not useful for the detection of botnet behavior in real time. A network flow between two hosts may run for a few seconds to several days, and it is desirable to discover botnet activity as soon as possible. Our approach is based on traffic analysis and detect the presence of bots leveraging the unique malicious traffic.

2) Attribute Selection: Every network traffic flow is characterized by a combination of attributes which are represented as numeric values. The attributes for the proposed approach is described in Table I. A few characteristics, for example, the source and destination IP locations and ports of a stream, might be extricated specifically from the TCP/UDP headers, while others, for example, difference in payload per flow, requires calculation. These traits are then utilized to build feature vectors. These vectors apprehend the attributes of an independent stream. These attributes are not random, but part of extensively used network protocols. Moreover, bot communications are more unvarying as they constantly depend on the commands from the bot-master.

Parameters	**Description**
SrcIp	Source IP address
SrcPort	Source port
DstIP	Destination IP address
DstPort	Destination port
Protocol	Transport Layer Protocol
PcktLen	Packet size in bytes
AvgPcktLen	Average packet length per flow
PcktVar	Mean and standard deviation of payload per flow
Tot	Total number of packets per flow
NumRe	Number of reconnects per flow
Fsz	Size of the first packet in the flow
FlwAddr	Number of flows between two addresses
FlgCnt	Count of flags between two addresses

TABLE I
List of attributes.

Another perception we may have, when a client joins a system for the first time, it has a tendency to behave uniquely

than the rest of the systems in the network (e.g., the first packet size might vary drastically). Also, it is to be noted that the inclusion of source and destination IP addresses might not be as important features as others but they are network independent. A bot may infuse arbitrary packets into its C&C communications. To improve our remedial procedures, we measure the quantity of streams created by an independent address, and the quantity of aggregate streams produced. Then we calculate the mean and variance between the two. This attribute is an indication of an increased flow created by bots. It is helpful in identifying the bot by the number of connects and reconnects it performs. Storing the state history of a connection as a string of letters also contributes as an identification parameter. The FlgCnt parameters can have the following values: a (acknowledge), d (packets with payload), h (handshaking), r (connection reset), s (open ended handshake), t (re-transmitted payload), f (connection termination).

B. Phase 2: Apriori algorithm to find out the most infrequent feature subsets

We hypothesized, rare values of the network parameters are the potential signal of malicious activities. The Apriori algorithm [4] employs Boolean association rule and iterates over the items(attributes) in the item set, to find the the most frequent items. It uses $support$ as the metric for determining if a certain combination of items(network parameters) are frequent or not.

For example, the dataset D_b contains the parameters values extracted from the network packets of *n* bots. Then let:

$$D_{b_i} = \{(SrcIp : v1), (SrcPort : v2), ..., (FlgCnt : v11)\} \quad (1)$$

be an instance in D_b. The calculation of support continues by recognizing the regular individual instances in the D_b and extending them to bigger item sets as long as those item sets yield desired frequency in D_b. If $\{(Fsz : v11)\}$ is a candidate, then the support for $\{(Fsz : v11)\}$ is calculated as:

$$sup(Fsz : v11) = \frac{NumberofPacketsWidth\{(Fsz : v11)\}}{TotalNumumberofPackets} \quad (2)$$

If $sup\{(Fsz : v11)\}$ is greater than δ_{sup} (a user defined minimum threshold), then the item set is considered frequent. According to previous studies and analysis [6] $\delta_{sup} = 0.05$ could be used as the minimum value for support. This algorithm is then used on both benign and malicious flows. Even though they could contain similar values for attributes, malicious bot flows would eventually show different patterns over time.

C. Phase 3 and 4: Assigning subjective probability and applying Dempster Rule of combination to identify the presence of malicious bots in the network

Dempster Shafer Theory [1] provides the ability to group complex set of events into finite number of outcomes based on available evidence. DST is a powerful statistical framework to build a classifier with the ability to handle uncertainty. For example, when we toss a coin, the classical probabilistic approach assigns 50% probability to head and 50% probability to tail. But DST assigns 0% belief in both head and tail and 100% to the possibility to the set $\{head, tail\}$. Our work consists of only two possible outcomes $\{bots, benign\}$, i.e., a traditional classification problem. In DST, the set of possible outcomes are called *frame of discernment* and each item in a frame of discernment is called a *focal element*. Based on the available evidence, each focal element is assigned a *mass* value between 0 and 1. The DST works within the range of an upper limit of probability (i.e., plausibility) and a lower limit (i.e., belief). These evidences are then combined using Dempster Rule of Combination (DRC).

In this work, each item set obtained through apriori algorithm is realized as evidence. The support score is interpreted as mass value. If apriori algorithm calculates support for $\{Fsz, v11\}$ as m_1 from one instance of flow and another instance has a $\{Fsz, v11\}$ as m_2, then we can combine them using the following DRC formula to get the aggregate belief:

$$m_{1,2}(\{Fsz : v11\}) = (m_1 \oplus m_2)(\{Fsz : v11\}) \quad (3)$$

The $\oplus$ notation is the conjunctive operation to combine the mass functions. DRC is a generalized form of Bayes Theorem and can combine evidence from independent sources to formulate a single decision.

IV. Experimentation Details

The experiments for the proposed approach was performed on a MAC OS v10.14, 2.5 GHz Intel Core i5 processor, 8 GB 1600 MHz DDR3 RAM. The apriori algorithm and the DST classifier was implemented in R. we exported the pcap files from the ISOT [7] data set as csv through Wireshark[2] so that we could extract the attributes and their values.

The steps involved are as follows:

1) Step 1: An R program to read and parse the flow information from the csv files to form the attributes.
2) Step 2: These features were then used in the Apriori algorithm to look for patterns and calculating their support or confidence.
3) Step 3: DRC calculation of the DST classifier combined the support for all the rules generated for a particular flow by the Apriori algorithm.

a) Data Set: ISOT HTTP botnet Dataset is a publicly available dataset. It contains malicious as well as benign DNS traffic, generated by various botnets. A subset of this dataset was used in a contained environment to conduct the experimentation of this approach.

b) Pattern Identification and Classification: Based on the attributes mention in Table I, we converted the dataset into a transaction dataset so that we can apply apriori algorithm on it. To identify feature or combinations of features which are more significant than others, the experiment needs to be run on the entire dataset and the traffic behavior needs to be studied in further detail. Some features are necessary for both

[2]https://www.wireshark.org

benign and malicious packet flows (for example source and destination LP address does not contribute much as a feature but the variance in packet size could suggest if the bots are carrying extra binaries for installation on target, or attempts to connect and reconnect to a port that is not usual for a particular protocol). Moreover, the dataset provides two different sets of benign and malicious samples that are already identified. Hence, the proposed framework does not guarantee a resilient solution in the wake of a new type of botnet.

c) Evaluation: The preliminary experiments involved randomly selected network flows with malicious bot traces as well as benign traces. We then extracted the attributes mentioned in Table I for each of the flow and applied Apriori algorithm to find out the set of attributes with rare values (or attributes that are assigned very low δ_{sup}). Then we combined these attributes with low δ_{sup} using DRC to finally classify them as malicious or not. Following is an example of TCP flow features[3] for apriori algorithm: The aforementioned feature

SrcIp=1921685099 SrcPort=52329 DstIP=1921685088 DstPort=53
Protocol=TCP PcktLen=1158 AvgPcktLen=20293035 PcktVar=53681400
Tot=0.015 NumRe=2 Fsz=148 FlwAddr=5 FlgCnt=2

set indicates a communication between 192.168.50.88 and 192.168.50.99. According to *National Vulnerability Database*[4] TCP destination port 53 is an indicator that the source is trying to trigger a DoS attack and then close the connection, causing device restart causing a configuration impairment. The proposed model also classified this flow as malicious.

We analyzed each flow manually and compared with the results from our proposed model and it showed an accuracy of approximately 87.73% for DST based classifier. With minimum support of 5% and minimum confidence of 100% the apriori algorithm generated 9 rules out of the 28 IPv6 flow, each with 13 attributes (features). However, this experimental model needs to be extended for performance tuning and could be compared with the existing tools like BotHunter[3] or BotMiner[5] to present a comparable performance.

Novel Botnet: The proposed approach does not address the feature vector generation required for identification of unforeseen malicious bots, if introduced to the network. The underlying assumption for this approach is that we have bot traces available in the pcap files for feature generation and mass calculation.

V. Conclusion and Future Work

Network traffic analysis strategy employed in this model depends on payload analysis. Payload investigation has least false positive rates when contrasted with different methodologies. However, it comes with two noteworthy issues (i) it is computationally intensive. (ii) it carries an inherent risk to security; especially in the IoT systems where we have to process Big data. The behavior-based analysis depends on recognizing rare network characteristics and can analyze encoded traffic.

The approach was applied to a combination of randomly selected benign and malicious flows. However, theoretical evaluation indicates the proposed model would result in different number of rule generations apart from minimum support/confidence values on benign and malicious data separately. Leading to our assumption that some features are predominant in malicious data. Further, we can conduct an experiment to find out the network parameters that are having out-of-the-ordinary values in the presence of bots and enhance our framework to achieve better accuracy and low error rate. We propose to combine this work with our previous webspam detection foundation [7] for building network anomaly detection framework and observe real time bot messaging for IoT devices. Particularly the application of DRC as a classifier could be advantageous in real world applications as many a time bots with unseen behaviors are introduced in a network and theoretically DST has proven to handle uncertain system behavior. This work is extensible to incorporate security rules and policies and could be implemented easily for various computing platforms like mobile or cloud. No one approach can promise the identification, prevention and removal of bots from the network. Hence, this work can be combined with other existing methods to build a more resilient system. We further aim to combine MapReduce [10] model to make the proposed approach efficiently handle big data.

[3]The Source and destination IP addresses are converted into decimal by removing the dot

[4]//nvd.nist.gov/vuln/detail/CVE-2009-1152

Acknowledgment

This work is supported in part from National Science Foundation under the grant number 1821560.

References

[1] Shafer, G., 1992. Dempster-shafer theory. Encyclopedia of artificial intelligence, pp.330-331.

[2] Leonard, J., Xu, S. and Sandhu, R., 2009, March. A framework for understanding botnets. In Availability, Reliability and Security, 2009. ARES'09. International Conference on (pp. 917-922). IEEE.

[3] Gu,G.,Porras,P.,Yegneswaran,V.,Fong,M.,Lee,W.:Bothunter: detecting malware infection through ids-driven dialog correlation, In: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium, SS07. USENIX Association, Berkeley, pp. 12:112:16 (2007). http://dl.acm.org/citation.cfm? id=1362903.1362915

[4] Agrawal, R. and Srikant, R., 1994, September. Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

[5] Gu, G., Perdisci, R., Zhang, J. and Lee, W., 2008. Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection.

[6] Moonsamy, V., Rong, J., Liu, S.: Mining permission patterns for contrasting clean and malicious android applications. Future Gener. Comput. Syst. 36, 122132 (2014).

[7] Alenazi A., Traore I., Ganame K., Woungang I. (2017) Holistic Model for HTTP Botnet Detection Based on DNS Traffic Analysis. In: Traore I., Woungang I., Awad A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science, vol 10618. Springer, Cham

[8] Chatterjee, M. and Namin, A.S., 2018, July. Detecting Web Spams Using Evidence Theory. In 2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC) (pp. 695-700). IEEE

[9] Dean, J. and Ghemawat, S., 2008. MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), pp.107-113.

[10] Garca, S., Zunino, A. and Campo, M., 2014. Survey on networkbased botnet detection methods. Security and Communication Networks, 7(5), pp.878-903.