A Darknet Traffic Analysis for IoT Malwares Using Association Rule Learning

Hashimoto, Naoki
Ozawa, Seiichi
Ban, Tao
Nakazato, Junji
Shimamura, Jumpei

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 144 (2018) 118–123

Procedia Computer Science

www.elsevier.com/locate/procedia

INNS Conference on Big Data and Deep Learning 2018

A Darknet Traffic Analysis for IoT Malwares Using Association Rule Learning

Naoki Hashimoto[a], Seiichi Ozawa[a,*], Tao Ban[b], Junji Nakazato[b], Jumpei Shimamura[c]

[a]*Kobe University, 1-1 Rokko-dai, Nada-ku, Kobe 657-8504, Japan*
[b]*National Institute of Information and Communications Technology, JAPAN*
[c]*clwit Inc., JAPAN*

Abstract

In this paper, we report an interesting observation of the darknet traffic before the source code of IoT malware *Mirai* was first opened on September 7th 2016. In our darknet analysis, the frequent pattern mining and the association rule learning were performed to a large set of TCP SYN packets collected from July 1st 2016 to September 15th 2016 with the NICT /16 darknet sensor. The number of collected packets is 1,840,973,403 packets in total which were sent from 17,928,006 unique hosts. In this study, we focus on the frequently appeared combinations of "window sizes" in TCP headers. We successfully extracted a certain number of frequent patters and association rules on *window sizes*, and we specified source hosts that sent out SYN packets matched with either of the extracted rules. In addition, we show that almost all such hosts sent SYN packets satisfying the three conditions known from the source code of *Mirai*. Such hosts started their scan activities from August 2nd 2016, and ended on September 4th 2016 (i.e., 3 days before the source code was opened).

Keywords: cybersecurity; machine learning; association rule learning; darknet traffic analysis; IoT Malware

1. Introduction

Information technologies (IT) have brought drastic changes in our life and many people have enjoyed new benefits on the Internet. In recent years, in addition to this IT revolution, the great progress of the Internet of Things (IoT), where various services and devices are connected to the network, is about to bring us further revolution. However, along with sophistication of IT and IoT systems, cyberattacks exploiting new system vulnerabilities are becoming serious these days. In particular, the impact of a recent IoT malware *Mirai* was enormous. Mirai is a worm-type malware that finds an IoT device with similar vulnerability for self-replication. An attacker manipulates a number of

cor1 Seiichi Ozawa. Tel.: +81-78-803-6466.
E-mail address: ozawasei@kobe-u.ac.jp

1877-0509
10.1016/j.procs.2018.10.511

IoT devices infected with Mirai as bots and uses this to conduct a Distributed Denial of Service (DDoS) attack by seeding a large number of packets to target hosts.

In order to deal with such a large-scale intelligent cyberattack promptly, it is necessary to construct a mechanism that is capable of observing cyberattacks occurring on the Internet with a wide view. For this purpose, the use of the darknet, known as *network telescope*, has been studied for many years [1]. Darknet is an unused address space. It is considered that no communication occurs because there is no computer installed in the darknet, but many packets are arriving in reality. These packets are mainly caused by scan activity or backscatter of reply packets from hosts targeted by DDoS attack; thus, it can be considered that packet observed in the darknet is generated by malwares. Therefore, through the analysis of Darknet packets, it is possible to observe a part of cyberattacks on the Internet.

In this research, we analyze the behavior of scan attacks from packets observed in Darknet. In particular, we focus on TCP SYN packets characterizing scan attacks and aim to find statistical features in the TCP headers of those packets. For this purpose, we apply the association rule learning to SYN packets and discuss the dynamic features of malware performing scan attacks. As for the destination port information, there have been reported several prior works analyzing SYN packets. Ban et al. [1] and some researchers [2, 3] applied the association rule learning to destination port numbers of SYN packets, and they discovered several association rules related to Carna botnet and other malwares. These rules are currently used as a signature when performing network scan. In this work, we focus on different header information such as window size instead of destination port.

This paper is organized as follow. Section 2 briefly explains the darknet analysis and an association rule learning based on FP-tree/FP-growth algorithms. In Section 3, we present the method to analyze scan attacks by applying the association rule learning to TCP SYN packets, and we show interesting association rules on window sizes in TCP headers that were found for the darknet traffic data observed around September 7th, 2016. Section4 gives our conclusions of this paper and future work.

2. Association Rule Learning for Darknet Traffic Data

2.1. Darknet Traffic Analysis

Darknet represents a reachable and unused IP address space on the Internet. In IPv4 used in most communication on the Internet today, IP address is expressed as 32 bit data; thus, there are about 4.3 billion IP addresses. However, not all of the addresses are assigned to host computers. In fact, a considerable number of packets arrive although packet transmission to an unused IP address does not occur with normal Internet use. There are two main reasons for this fact. One is scan activity by malware and the other is backscatter that corresponds to reply packet sent from a target host suffered by DDoS attack.

A scan attack is an activity to check whether there is security vulnerability in a host. Actually, they attempt to connect by randomly or routing a broad range of IP addresses and destination port numbers, and check the status of the destination hosts by seeing their responses. There are two types of scan attacks, host scan and port scan. Host scan is to designate a range of IP addresses and try to connect in order, and it can know whether a specific host is assigned to an IP address. On the other hand, a port scan is to check if the port is in a communicable state or not. Here, a port is a socket number for identifying an application program used by a computer under TCP and UDP protocols. For example, when browsing a website, the communication between two hosts is conducted using port #80. There are also several types of port scan attacks. The SYN scan is an attack of sending a SYN packet in TCP communication, which is known as a stealth scan attach because it is performed without leaving a log on a server.

In contrast, backscatter is a train of reply packets sent by a victim host of a DDoS attack, and the detection of this backscatter allows us to know if DDoS attacks are carried out. Backscatter is essentially returned to the senders. However, in DDoS attack, infected bot computers send lots of packets to a target host with spoofing their IP addresses, and then some of those backscatters jump into the darknet. Therefore, we only know events of DDoS attacks though this darknet traffic analysis.

2.2. Association Rule Learning

This section briefly explains the association rule learning, a commonly applied technique for discovering interesting relationships hidden in a database.

2.2.1. Frequent Pattern Mining

The problem of association rule learning was originally proposed in the context of market basket data in order to find frequent groups of items that are purchased together [4, 5]. Following the original definition in [4], the problem of association rule learning is defined as follows.

Let $\mathcal{D} = \{T_1, T_2, \cdots, T_N\}$ be a set of N transactions called the *database*. Let $\mathcal{I} = \{i_1, i_2, \cdots, i_M\}$ be the universal set of M all items present in the database. Each transaction in $\mathcal{D}$ has a unique transaction ID and contains a subset of the items in $\mathcal{I}$. The *support* $\mathrm{supp}(X)$ of a set of item (for short item set) X is defined as the number/proportion of transactions in the database which contain the item set.

Frequent pattern mining is to determine all patterns $P \subset I$ that are present in at least a fraction S of the transactions. The fraction S is referred to as the minimum support. It can be expressed either as an absolute number, or as a fraction of the total number of transactions in the database.

An *association rule* is defined as an implication of the form

$$X \to Y, \text{ for } X, Y \subseteq I, \ X \cap Y = \emptyset. \tag{1}$$

The item sets X and Y are called antecedent and consequent of the rule respectively. The confidence of a rule is presented by the conditional probability, $P(Y|X)$, i. e.,

$$\mathrm{conf}(X \Rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X). \tag{2}$$

To select interesting rules from the set of all possible rules, rules that satisfy both a minimum support threshold, S, and a minimum confidence threshold, C, are called strong.

In general, *association rule learning* can be done in the following two steps:

1. Frequent pattern mining: Each of the item sets will satisfy the minimum support, i.e., occurs at least as frequently as S.
2. Strong association rule generation: By definition, rules created from the frequent item sets with guaranteed minimum support must satisfy the minimum confidence constraint.

2.2.2. Frequent Pattern Mining using FP-tree

The first step in association rule learning involves searching in a power set of all possible combinations of items, whereas the size of this set grows exponentially in the number of items n in $\mathcal{I}$. The key to an efficient search algorithm is the so-called a priori property: All nonempty subsets of a frequent item set must also be frequent. Thus for an infrequent item set, all its supersets must also be infrequent. One of the currently fastest and most popular algorithms for frequent item set mining is the Frequent Pattern growth (FP-growth) algorithm [5, 6]. It is based on a prefix tree representation of the given database. By using a prefix tree data structure - the so-called FP-tree - FP-growth can save considerable amounts of memory for storing the transactions. The basic idea of the FP-growth algorithm can be described as a recursive elimination scheme as follows.

1. In the first pass, derive the set of frequent items and their support counts. Delete all items from the transactions which do not satisfy the minimum support constraint. All frequent items are stored in a header table in descending order of their frequency.
2. In the second pass, build an FP-tree by inserting instances into a tree with a root node labeled as ‘null’. To speed up the processing of the FP-tree, items in each transaction are sorted in the same order as in the header table. All nodes referring to the same item are indexed by a list so that all transactions containing the item can be accessed and counted by traversing this list. The header elements to the list are associated with the corresponding items in the header table.

3. Recursive mining of the FP-tree can grows large item sets directly, without generating candidate items and testing them against the entire database. Start from the bottom of the header table, build the conditional item base for the length-1-pattern, which consists of a set of prefix paths in the FP-tree co-occurring with the suffix item. Then, a conditional FP-tree is created, with counts projected from the original tree corresponding to the set of instances that are conditional on the attribute, with each node getting sum of its children counts. Recursive growth ends when no individual items conditional on the attribute meet the minimum support threshold, and processing continues on the remaining header items of the original FP-tree.
4. Once the recursive process has completed, all large item sets satisfying the minimum support constraint are found, and association rule creation begins.

2.2.3. Association Rule Generation from Frequent Item sets

Association rules can be generated based on the frequent item sets in the following steps.

1. For each frequent item set l, generate all nonempty subset of l.
2. For every nonempty subset s of l, output the rule "$s \rightarrow (l-s)$" if its confidence is higher than minimum confidence threshold C.

Since the rules are generated from frequent item sets, all association rules created in such a way automatically satisfy the minimum support.

3. Darknet Analysis for Scan Attacks Using Association Rule Learning

In this section, we establish a darknet analysis for finding traffic patterns of a specific scanning attack using the association rule learning mentioned in Section 2. Although there are various types of scan attacks, here we focus on the ones using TCP SYN packets because they are a majority of darknet packets.

3.1. Proposed Method

As mentioned in Section 1, Ban et al. focused on destination port numbers of SYN packets. Then, they successfully found association rules to detect Carna botnet [1]. However, other than destination port numbers, there are many other information in TCP and IP headers that can feature packet traffic. For example, a TCP header has "source port", "destination port", "sequence number", "9 flags", "window size", and an IP header has "Time to live (TTL)" other than source/destination IP.

The association rule learning can apply to every header fields for mining useful rules. To do this, first we need to check all the values of each field in TCP and IP headers for all collected SYN packets, and define a *transaction set* for each header field. For example, if we focus on the three header fields: "destination port", "sequence number", and "window size", three sets of transactions are obtained from collected darknet SYN packets. Then, the association rule learning is conducted for each transaction set. If interesting association rules are obtained, we would set up a hypothesis that the behaviors of a scan attack are featured by the obtained association rules, and then we would verify the correctness of a hypothesis for darknet packets on the other days. If the correctness of a hypothesis is finally proved, we could use the association rules as a signature of a malware or a scanner program.

3.2. Study on Darknet Traffic before Mirai Outbreak

To evaluate the proposed rule mining method, we use a large set of TCP SYN packets collected from July 1st 2016 to September 15th 2016 with the NICT /16 darknet sensor. The number of collected packets is 1,840,973,403 packets in total which were sent from 17,928,006 unique hosts. The reason why we select the above period is that we might be able to find some indications of a notorious IoT malware "Mirai". We choose *destination port*, *sequence number*, and *window size* as header fields, and the association rule learning is performed for darknet SYN packets collected everyday during the above period. We only consider association rules whose support (#source hosts) and confidence are larger than 2,000 and 90%, respectively.

Table 1. Obtained association rules on TCP window sizes.

Association Rules	Support	Confidence [%]
(1320, 4488) → 792	10051	90.2
(1320, 2376, 4488) → 792	6657	96.0
(1320, 8456) → 792	5885	90.6
(1320, 2376, 8456) → 792	4064	95.8
(1320, 16904) → 792	3329	90.4
(1320, 4488, 8456) → 792	2886	96.3

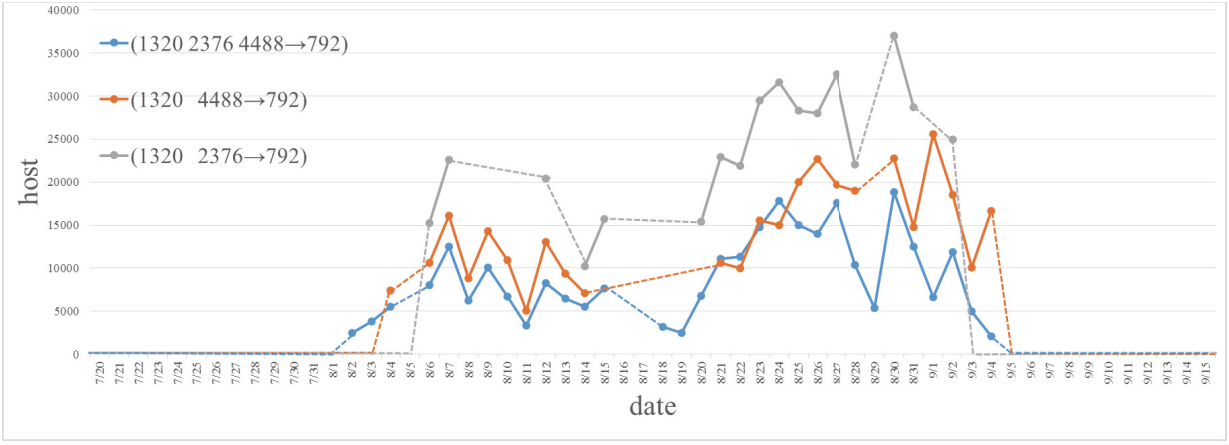

Fig. 1. Transitions of hosts sending darknet SYN packets matched with the three association rules.

As a result, we found new association rules on TCP window sizes from the darknet SYN packets collected from July 2nd to September 4th, 2016. Table 1 shows the obtained association rules on TCP window sizes. Figure 1 illustrates the transitions of hosts sending darknet SYN packets matched with the three association rules. As seen from Fig. 1 , the number of such hosts first appeared on August 2nd and disappeared on September 4th, just 3 days before the source code of *Mirai* was opened. Roughly speaking, there are two peaks with regard to the number of hosts matched with obtained association rules.

To check if the scan activities in Fig. 1 are related to *Mirai*, we verified the following three features of *Mirai* which are clarified by the open of the source code.

Condition 1 sequence number = destination IP,

Condition 2 destination port = 23,

Condition 3 source port > 1024.

Figure 2 illustrates the percentage of hosts satisfying the above three conditions, in addition to the matching of window size features. Surprisingly, almost all hosts matched with window size features satisfies the above three conditions of Mirai. Therefore, the scan activities in Fig. 1 might suggest that attackers were doing some tests or preparation for the actual distribution of *Mirai* malware.

4. Conclusions

In this paper, we developed a new darknet analysis using the association rule learning. In the proposed method, not only destination ports but also other TCP/IP header information are used to create transaction sets for the association

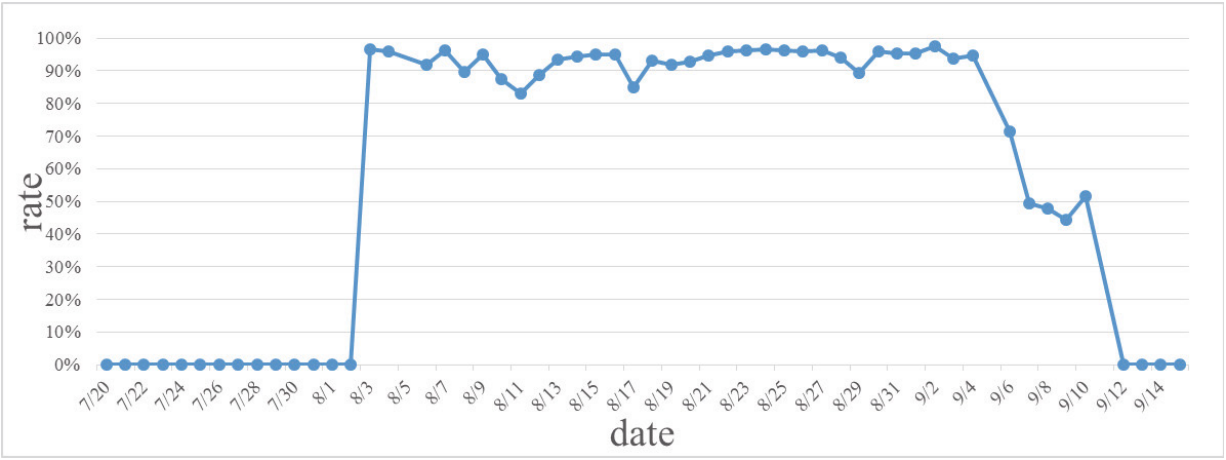

Fig. 2. Relationship TCP window size and Mirai

rule learning. Then, the rule mining for all header information is conducted in parallel to obtain association rules. The proposed darknet analysis was performed to a large set of TCP SYN packets collected from July 1st 2016 to September 15th 2016 with the NICT /16 darknet sensor. As a result, several association rules on TCP window size appeared on 2nd August and disappeared three days before the source code of Mirai was released. The hosts whose scan activities are featured by the obtained association rules have very similar features to Mirai. Therefore, we conjecture that the attackers were doing test or preparation for the actual distribution of Mirai malware about one month before the source code was opened. The result of this paper is very encouraging for us to apply the proposed method for future attack detection.

Acknowledgment

This research was achieved by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Scientific Research (B) 16H02874 and the Commissioned Research of National Institute of Information and Communications Technology (NICT), JAPAN.

References

[1] T. Ban, M. Eto, S. Guo, D. Inoue, K. Nakao, R. Huang, “A study on association rule mining of darknet big data,” *Proc. of Int. Joint Conference on Neural Networks*, pp. 1-7, 2015.
[2] C. Stocker, J. Horchert, “Mapping the internet: A hacker’s secret internet census,” *Spiegel Online*, March 22, 2013.
[3] E.L. Malecot, D. Inoue, “The Carna botnet through the lens of a network telescope”, In: J. Danger, et al.(eds), *Foundations and Practice of Security*, LNCS, vol. 8352, Springer, pp. 426-441, 2014.
[4] R. Agrawal, T. Imielinski, A. Swami, “Mining association rules between sets of items in large databases,” ACM SIGMOD Record, vol. 22, no. 2, pp. 207-216, 1993.
[5] J. Han, J. Pei, Y. Yin, “Mining frequent patterns without candidate generation,” ACM SIGMOD Record, vol. 29, no. 2, pp. 1-12, 2000.
[6] J. HanJian, P.Y. Mao, “Mining frequent patterns without candidate generation: A frequent-pattern tree approach,” *Data Mining and Knowledge Discovery*, vol. 8, no. 1, pp. 53?87, 2004.
[7] C. Borgelt, “Frequent item set mining,” *Data Mining Knowledge Discovery*, vol. 2, no. 6, pp. 437-456, 2012.