ML-Based Early Detection of IoT Botnets

Ayush Kumar[1(✉)], Mrinalini Shridhar[1], Sahithya Swaminathan[1], and Teng Joon Lim[2]

[1] National University of Singapore, Singapore, Singapore
{ayush.kumar,e0269748,e0269724}@u.nus.edu
[2] University of Sydney, Camperdown, NSW 2008, Australia
tj.lim@sydney.edu.au

Abstract. In this paper, we present EDIMA, an IoT botnet detection solution to be deployed at the edge gateway installed in home networks which targets early detection of botnets. EDIMA includes a novel two-stage machine learning (ML)-based detector which first employs ML algorithms for aggregate traffic classification and subsequently Autocorrelation Function (ACF)-based tests to detect individual bots. Performance evaluation results show that EDIMA achieves high bot scanning detection accuracies with a very low false positive rate.

Keywords: Internet of Things · IoT · Malware · Mirai · Botnet detection · Machine Learning · Anomaly detection · Intrusion detection

1 Introduction

The Internet of things (IoT) refers to the network of low-power, limited processing capability sensing devices which exchange data with each other and/or systems (e.g., gateways, cloud servers). IoT devices are used in a number of applications such as wearables, home automation and industrial automation. Unfortunately, hackers are increasingly targeting IoT devices using malware (malicious software) for a number of reasons such as legacy devices connected to the Internet with little or no security updates, low priority given to security within the development cycle, weak login credentials, etc.

In a widely publicized attack, the IoT malware Mirai was used to launch the biggest Distributed Denial-of-Service (DDoS) attack on record in 2016 through infected IoT devices such as IP cameras and DVR recorders. The source code for Mirai was leaked in 2017 and since then, there has been a proliferation of IoT malware. These malware are usually Mirai variants using a similar brute force technique of scanning random IP addresses for open TELNET ports and attempting to login using a built-in dictionary of commonly used credentials (e.g., Remaiten, Hajime), or more sophisticated ones that exploit software vulnerabilities to execute remote command injections on vulnerable devices (e.g., Reaper, Satori, Masuta, Linux.Darlloz, Amnesia). Bots compromised by Mirai or similar IoT

N. Park et al. (Eds.): SecureComm 2020, LNICST 336, pp. 254–260, 2020.
https://doi.org/10.1007/978-3-030-63095-9_15

malware can be used for DDoS attacks, phishing, spamming and bitcoin mining. These attacks can cause network downtime for long periods which may lead to financial loss to Internet Service Providers (ISP), leakage of users' confidential data, and unauthorized exploitation of computational resources. Furthermore, many of the infected devices are expected to remain infected for a long time.

We propose an IoT botnet detection solution, EDIMA (Early Detection of IoT Malware Scanning and CnC Communication Activity), which is designed to be deployed at the edge gateway installed in home networks and targets the detection of botnets at an early stage of their evolution (scanning and propagation phase) before they can be used for further attacks. EDIMA employs a two-stage detection mechanism which first uses machine learning (ML) algorithms for aggregate traffic classification based on bot scanning traffic patterns, and subsequently Autocorrelation Function (ACF)-based tests which leverage bot-CnC messaging characteristics at the per-device traffic level to detect individual bots. We only target IoT botnets with centralized Command-and-control architecture in this work.

2 EDIMA Architecture

EDIMA is designed to have a modular architecture, as shown in Fig. 1, with the following components:

- **Feature Extractor:** This module extracts features from the aggregate traffic at the gateway. These features are then forwarded to the ML-based Bot Detector (MBD) for classification during the execution phase. The Feature Extractor (FE) also sends features extracted from the aggregate traffic to the *ML Model Constructor* (MC), during the training phase.
- **ML-based Bot Detector:** This is a 2-stage module with the first stage being a *coarse-grained* one that classifies the aggregate traffic samples using the features obtained from FE and the ML model trained and forwarded by the MC. Depending on the result of the classification, the second *fine-grained* stage attempts to identify the infected IoT device(s) from the set of devices connected to the gateway.
- **Traffic Parser:** The traffic parser (TP) sorts the combined gateway traffic into traffic sessions. During the bootstrap (training) phase of EDIMA, it also helps replay malware traffic samples along with normal traffic to generate malicious traffic samples.
- **Malware PCAP Database:** The database stores malware traffic *pcap* files captured from private and professional honeypots targeted at IoT malware.
- **ML Model Constructor:** The ML model used for classifying edge gateway traffic is trained by this module. We assume a publish-subscribe model where multiple gateways subscribe to a MC. A separate ML model is trained for each gateway for optimal performance. Whenever a gateway comes online, it registers with the MC. Malicious traffic samples from the *Malware PCAP Database* (mDB) are sent to the gateway to generate malicious aggregate traffic. The feature vectors extracted from benign (normal traffic with no

malicious scanning packets) and malicious aggregate traffic are subsequently sent by a gateway's FE to the MC. The extracted features are used to train a supervised ML classifier which is then published to the gateway's MBD.

– **Policy Engine:** The policy engine (PE) consists of a list of policies defined by the network administrator, which determine the course of actions to be taken once an IoT device connected to the edge gateway has been detected as a bot.

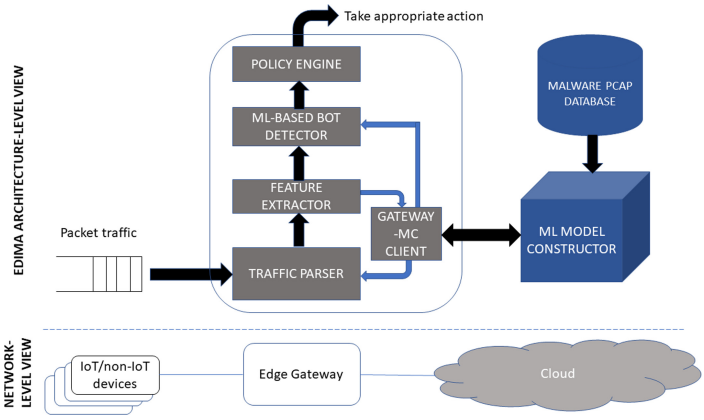

Fig. 1. EDIMA architecture

3 Description of EDIMA's Components

3.1 Detection of Scanning Activity in Aggregate Gateway Traffic

The first coarse-grained stage of the MBD performs classification on aggregate gateway traffic rather than per-device traffic. We define two classes of gateway traffic: *benign* and *malicious*. Benign traffic refers to the gateway traffic which does not include bot scanning packets while malicious traffic refers to gateway traffic that does. The gateway traffic is captured in the form of traffic sessions which are defined statically as the set of ingress/egress packets at a network interface over a fixed time interval. We apply the classification algorithm on these traffic sessions.

In a traffic session, we extract features from TCP packet headers only and not the payloads. The steps used for gateway-level traffic classification are given below:

1. Filter each gateway traffic session to include only TCP packets.
2. Extract the feature vectors for each traffic session.
3. Apply the trained ML classifier on the extracted feature vectors and classify the corresponding sessions.

We have carefully identified the following eight botnet-aware features for ML classification:

- Number of unique TCP SYN destination IP addresses
- Number of packets per unique destination IP address
  - maximum
  - minimum
  - mean
- Number of TCP half-open connections
- TCP packet length
  - maximum
  - minimum
  - mean.

3.2 Detection of Individual Bots Using Bot-CnC Communication Patterns

Once the aggregate traffic at an edge gateway has been classified as *malicious*, the second fine-grained stage of the ML-based bot detector attempts to detect the underlying bots by checking the ingress/egress traffic from each IoT device for the presence of bot-CnC communication patterns. In most existing IoT botnets, including the Mirai-variants, there is a periodic exchange of TCP messages ([PSH, ACK], [ACK]) or UDP messages between the bot and the CnC server. To detect the presence of bot-CnC communication, we propose the following approach: filter the traffic from a potential bot for UDP packets or TCP packets (with PSH and ACK flags *ON*) and exclude IoT application data packets from our analysis using appropriate packet capture filters. Subsequently, sample and encode the filtered packets to produce a uniformly sampled discrete-time signal. To detect periodicity in time series data obtained above, we use the autocorrelation function (ACF) [1].

4 Performance Evaluation

4.1 Testbed Description

To evaluate the performance of EDIMA on real devices, we built a testbed with IoT and non-IoT devices. The devices were used by 3 staff members in our lab over a period of 4 weeks, and thus the traffic data collected from those devices reflects real-world users' behaviour. The edge gateway where the traffic from all the above devices was aggregated was a Linksys WRT32X router running OpenWRT with a 1.8 GHz dual-core processor, 512 MB RAM, and 256 MB NAND flash memory. The testbed schematic is shown in Fig. 2.

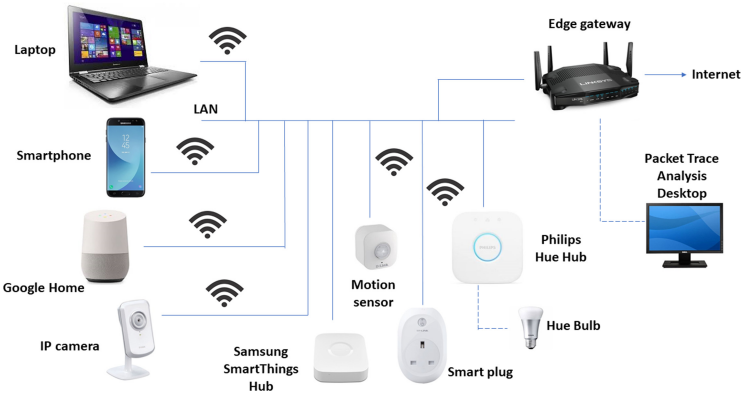

Fig. 2. Schematic of the testbed

4.2 Data Collection and Processing

As mentioned in Sect. 3.1, we classify the aggregate gateway traffic as *benign* or *malicious*. Therefore, we need training data samples to represent both classes. Benign traffic is not difficult to generate as it involves the normal operation of uninfected devices. However, malicious traffic contains both benign traffic as well as scanning/infection packets generated by malware. Towards this, we obtained 23 live IoT malware samples belonging to different malware categories over a period of 3 months (May–July, 2019) from APIs provided by New Sky Security [2] and malware hosting server links posted by Bad Packets Report Twitter account [3]. The FANTASM framework, provided by DeterLab team for safe experimentation with live malware based on their paper [4], was used to run the malware samples and collect the traffic generated by them.

Many of the malware samples were simple variants of each other, as revealed by analysing their traffic using Wireshark. We ended up with two malware samples, called *loligang* and *echobot* by their authors, which exploited TELNET and HTTP POST+GET vulnerabilities respectively. We ran both the malware binaries on the FANTASM testbed for 5 min each and captured the corresponding traffic *pcap* files. Malicious traffic was then generated by replaying the malware traffic collected from FANTASM on the edge gateway using the *tcpreplay* utility. This approach, in effect, emulates an IoT bot connected to the gateway.

We used a traffic session duration of 15 min for this study. 1000 traffic sessions were captured for benign traffic and a further 1000 sessions for malicious traffic through our testbed. The malicious traffic sessions consisted of 400 sessions corresponding to *loligang*, another 400 sessions corresponding to *echobot* and the remaining 200 sessions corresponding to both *loligang* and *echobot* traffic replayed at the OpenWRT router. The features mentioned in Sect. 3.1 were extracted from the captured sessions. Appropriate class labels were assigned to the extracted feature vectors.

The feature vectors were checked for missing values and handled appropriately. Next, all the values in a feature vector were scaled to lie within the range (0,1). Further, the feature vectors were randomly permuted. The combined benign and malicious feature vectors were randomly divided into *training* and *test* datasets using an 80:20 split. We used the χ^2 statistical test to compute the χ^2 test statistic for each feature from the sample data. Subsequently, we selected the best k= 6 features (having test statistic value more than zero) for training our ML classifiers.

4.3 Results

Scanning Activity Detection Performance. We trained the following ML models using the final feature vectors obtained in the previous section after completing all the data processing steps: Gaussian Naive Bayes' (GNB), Support Vector Machine (SVM) and Random Forest (RForest). Subsequently, the trained ML models are used to predict the class labels of the test dataset and thereby, the detection performance of the models is evaluated and compared. In this work, a 10-fold cross validation approach is used to tune the hyper-parameters of the ML classifiers for achieving the highest possible CV scores. The cross validation is based on training data only without using any information from the test dataset. Using the tuned hyper-parameters' values, the average classification accuracy, precision, recall and F-1 scores obtained for the final classifiers over 50 runs are shown in Table 1. It can be observed that the Random Forest classifier performs the best in terms of classification accuracy followed by SVM classifier and Gaussian Naive Bayes' classifier.

Table 1. Performance of ML classifiers for scanning activity detection

Dataset	Session duration	Method	AC	PR	RC	F1
Testbed	15 min	Rforest	1.0	1.0	1.0	1.0
		SVM	0.99	0.99	1.0	0.99
		GNB	0.97	0.97	1.0	0.97

5 Conclusion

We have proposed EDIMA, a solution for early detection of IoT botnets in home networks. It detects bots connected to an edge gateway in two stages- first by looking for scanning and subsequently bot-CnC server communication traffic patterns. EDIMA consists of a traffic parser, feature extractor, ML-based bot detector, policy engine, ML model constructor and a malware PCAP database. A performance evaluation of EDIMA using our testbed setup revealed that it has a close to 100% accuracy and very low false positive rate in detecting malicious aggregate gateway traffic with ML algorithms such as the Random Forest.

Acknowledgment. This research is supported by the National Research Foundation, Prime Minister's Office, Singapore under its Corporate Laboratory@University Scheme, National University of Singapore, and Singapore Telecommunications Ltd.

References

1. Martin, N., Mailhes, C.: About periodicity and signal to noise ratio - the strength of the autocorrelation function. In: Conference on Condition Monitoring and Machinery Failure Prevention Technologies (CM and MFPT 2010), Stratford-upon-Avon, United Kingdom (2010)
2. New Sky Security: New sky security IoT threat intelligence platform. https://iot.newskysecurity.com/
3. Bad Packets Report (Twitter): Mirai-like botnet hosts. https://tinyurl.com/y5y33omf
4. Alwabel, A., Shi, H., Bartlett, G., Mirkovic, J.: Safe and automated live malware experimentation on public testbeds. In: Proceedings of the 7th Workshop on Cyber Security Experimentation and Test (CSET 2014). USENIX Association, San Diego (2014)