IoTurva: Securing D2D Commmunications for IoT

Ibbad Hafeez
University of Helsinki
ibbad.hafeez@helsinki.fi

Aaron Yi Ding
Technical University of Munich
aaron.ding@tum.de

Markku Antikainen
University of Helsinki
Helsinki Institute for Information Technology (HIIT)
markku.antikainen@helsinki.fi

Sasu Tarkoma
University of Helsinki
Helsinki Institute for Information Technology (HIIT)
sasu.tarkoma@helsinki.fi

ABSTRACT

In this poster we present IoTurva, a platform for securing Device-to-Device (D2D) communication in IoT. Our solution takes a black-box approach to secure IoT edge-networks. We combine user and device-centric context-information together with network data to classify network communication as *normal* or *malicious*. We have designed a dual-layer traffic classification scheme based on fuzzy logic, where the classification model is trained remotely. The remotely trained model is then used by the edge gateway to classify the network traffic. We have implemented a proof-of-concept prototype and evaluate its performance in a real world environment. The evaluation shows that IoTurva causes very small overhead while it works with minimal hardware, and that our model training and classification approach can improve system efficiency and user privacy.

1 INTRODUCTION

With growing popularity of Internet-of-Things (IoT) and online services, edge networks are becoming even more densely populated. Latest generation of IoT devices are developed to cooperate with each other to improve automation in smart homes and industrial environments. Commonly seen examples of such connected IoT-systems are automated lighting systems and lock systems.

The IoT devices collect a lot of user related information in order to improve user experience. However, these devices are potentially vulnerable and could be compromised by a malicious actor [6, 7]. In a typical edge network, there are no security measures to detect a malicious or compromised device and to prevent it from talking to other devices in the network. External attackers can use these compromised devices as a foothold and attack other benign devices in the same network. These infected IoT devices can then be used, for example, as a part of a larger DoS campaign [8]. Therefore, it is crucial to monitor and control device-to-device (D2D) traffic in the edge networks to identify and isolate malicious devices, and prevent them from infecting larger parts of the network.

Unfortunately, due to their constrained nature, traditional security solutions do not help with IoT devices. For example, automated software patching, anti-virus solutions, and host-based firewalls are very rarely seen on constrained IoT devices. On the other hand, traditional network perimeter security systems, such as IDS, IPS, and firewalls, are deployed at the edge of the network and thus do not necessarily help in scenarios where an infected device connects and spreads malware to devices connected to same network [4].

We propose IoTurva to address the aforementioned security problems in IoT networks. Our solution is an easy to deploy, low cost, and scalable approach for securing one of the most vulnerable pieces in the of network security puzzle, such as the SOHO edge networks. Our proposal classifies the network activity of any user device, namely network host, as *normal* or *malicious*, and updates network configuration at runtime in order to prevent a malicious device from communicating with other devices in the same network. We use context information to improve our classification model for detecting misuse attacks where a seemingly *normal* D2D interaction between two hosts on the same network is, in fact, untrusted and a potential attack.

IoTurva prototype evaluation shows that our solution is easy to deploy and effective in securing edge networks. The fuzzy dual–layer classification model classifies traffic efficiently in real-time and is suitable for a small–medium sized edge networks ($\leq$ 20 connected to an AP) without significantly affecting user experience.

2 SYSTEM DESIGN

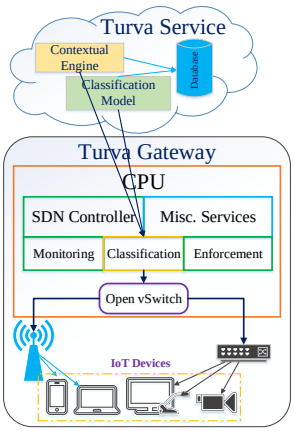

Figure 1: System design for IoTurva.

The high level system design is shown in Fig. 1. IoTurva consists of two primary components i.e. Turva Gateway and Turva Service. The Turva Gateway serves as an access point typically deployed in edge network and is responsible for traffic classification as well as enforcement of security policies in the IoT network. Turva Gateway offloads intensive tasks such as model training, training data collection, and mobility support to Turva Service. The Turva

Service is a logically centralized entity supporting one or more Turva Gateway installations across edge networks.

2.1 Turva Gateway

We designed Turva Gateway as an improvised gateway or access point typically used to setup SOHO edge networks. The lightweight design allows us to deploy it using minimal hardware resources e.g. Raspberry PI or legacy APs. The primary tasks of Turva Gateway include traffic monitoring and enforcement of security policies required to prevent any malicious devices from communicating to other devices in the network.

Figure 1 shows three major operations performed by Turva Gateway. First, it monitors all the traffic passing flowing through the network. Second, it classifies every network interaction using a set of rules (security policies) to be either malicious or normal traffic flow. Finally, once classified, it updates network rules to block any malicious traffic flows in the network.

The monitoring and enforcement mechanism are implemented using *software-defined networking* technologies, where Turva Gateway runs an OpenFlow-switch and a lightweight SDN-controller. The SDN-controller uses classification module to classify every unique network traffic flow it monitors. Based on classification results, it generates OF-rules necessary to allow or block (when classified as malicious) the given traffic flow(s). The rules used for traffic classification are obtained from Turva Service.

In IoT networks, Turva Gateway also acts as a sensor, which aggregates network statistics, traffic signatures and other network data to be shared with Turva Service, if required. Sharing this information, in aggregated and anonymous form, can protect user privacy and improve training process of traffic classification model.

2.2 Turva Service

The logically centralized Turva Service can be deployed as a single service or a set of services colluding together to generate classification model for classifying network traffic data. Two main components of Turva Service are 1) *classification model training engine*, which is responsible for feature extraction, feature analysis, and model training; and 2) the *contextual engine*, which is responsible for collecting context information from user devices.

In order to improve the classification accuracy and generalization of classification model, Turva Service should get data from multiple sources to develop a global view. Its design allows to easily plug-in data from 3rd party sources e.g. network middleboxes, perimeter security solutions, and device logs to improve model accuracy.

2.2.1 Training data collection. : A primary source of data collection can be Turva Gateway(s) as they provide insights of most commonly observed attacks, device usage patterns, and ratio of classification model hit/miss. Incentivized crowd-sourcing campaign can be used to collect signatures for normal and malicious device activity. Anonymized submissions, threshold-based acceptance criteria, user reputation, voting mechanism, etc., can also be used to overcome issues related to data quality and user privacy [3, 5]. Furthermore, public CVE [1], CWE [2] and malware databases [9] are reliable sources for obtaining information about latest vulnerabilities. Besides, device and service manuals also offer a lot of details about related services and features for any IoT devices. This information gives valuable information about cross device interactions and dependencies which are used in our training classification model.

In addition, the testbeds developed by researchers working on IoT security and communications can provide in-depth data about device vulnerability and cross-device dependencies. These testbeds can also be used to individually monitor device behavior and interaction mode in several states. IoTurva can use such data to model device behaviors and identify (signatures of) abnormal device interactions. A combination of these models can be further used to capture multi-level attacks using cross-device interactions.

2.2.2 Model training. We use Fuzzy C-means clustering technique to develop network traffic classification model. We obtain a set of clusters from the training data and each of these clusters can be represented as a security rule. The set of rules are transferred to Turva Gateway devices where they are used for traffic classification.

Feature extraction and selection: In model training, we first extract a set of features from training data. In this work, we mainly use network communication related features including network and transport protocol headers (port numbers, transport protocols, etc.). We also use features related to the context of the user and device including geo-location, user actions, and device state. We use Pearsons correlation coefficient to measure linear dependency among features and remove one of every two strongly linearly dependent features. All the features are normalized in range $[0, 1]$ to avoid any effects over clustering algorithm.

Classification engine: After feature selection, we assign each sample data point $X_j (j = 1, 2, ..., n)$ (in training data) to cluster $C_i (i = 1, 2, ..., c)$. Each of these clusters represents either *normal* or *malicious* traffic class. During training, we randomly assign each sample X_j to a cluster with random membership value μ_{ij} and iteratively update Eq. 2 and Eq. 3 to minimize objective function given in Eq. 1, where $1 \leq i \leq c$ and $1 \leq j \leq n$.

$$J_m = \sum_{i=1}^{c} \sum_{j=1}^{n} \mu_{ij}^{m} \left\| V_i - X_j \right\|^2 \tag{1}$$

$$\mu_{ij} = \left(\sum_{d=1}^{c} \left(\frac{\left\| V_i - X_j \right\|}{\left\| V_d - X_j \right\|} \right)^{\frac{2}{m-1}} \right)^{-1} \tag{2}$$

$$V_i = \frac{\sum_{j=1}^{n} (\mu_{ij})^m \times X_j}{\sum_{j=1}^{n} (\mu_{ij})^m} \tag{3}$$

For each cluster i, clustering is performed n times with goal to minimize *within-cluster-sum of distances* to best fit the data. At the end, each cluster can be represented as a rule e.g.

R_i: if $F_1 \in A_1, F_2 \in A_2, ..., F_k \in A_k, ..., F_h \in A_h \implies y \in B_i$

where A_k is subset of distribution for possible values of F. Set of these rules are sent to Turva Gateway for traffic classification.

3 EVALUATION

We have implemented a prototype for IoTurva to evaluate its performance in real world scenario. We setup Turva Gateway

using a Raspberry PI (R-PI) model B running Raspbian OS. We use `hostapd` to setup R-PI as WiFi access point. R-PI is connected to Internet and wired network using on board ethernet port. All wired and wireless interfaces are bridged to OVS running on R-PI and OVS is managed by our customized version of Floodlight controller performing network monitoring, classification and enforcement tasks. Turva Service runs on a HP Core i5 Laptop running Ubuntu 16.04LTS connected to a remote network. We implemented feature extraction and clustering using Python and scikit-fuzzy library and a RestAPI for communcations between Turva Gateway and Turva Service. The test network contains 15 different devices including smart phones, tablets, file server, PC, laptops and IoT devices.

For training data collection, we outlined different scenarios portraying normal and malicious network communication among devices connected to same network. We repeated same scenario for $n = 10$ times for data collection, to avoid any biases or artifacts.

During evaluation, we followed same scenario using different set of devices to check the robustness of classification model. We use 11 devices with 10 interactions for each pair of devices to get a total of $10 \times 10 \times 10$ test scenarios. Table 1 shows that we accurately classify malicious flows and normal flows with 96% and 93.4%, achieving overall accuracy of 93.7%. Our classification model achieves F1-score of 0.753% with 96% sensitivity and 93.4% specificity and classification precision of 0.62%.

Table 1: Confusion matrix

		Predicted class		
		Malicious	Normal	Total
Actual class	Malicious	96	4	100
	Normal	59	841	900
	Total	155	845	1000

With increasing popularity of streaming and video on demand services, minor deterioration in network performance can seriously affect user experience and cause business losses. Therefore, we have tested our system in real world setup to examine the impact on user experience in terms of latency and throughput. The results plotted in Figure 2 show that the traffic monitoring and classification procedures introduce minor overhead on the latency experienced by user.

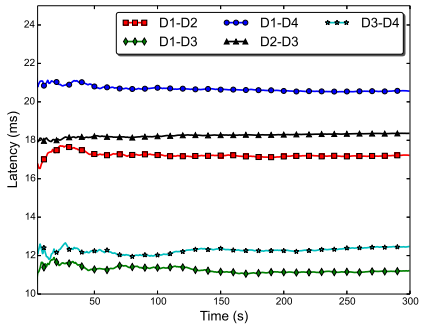

Figure 2: D2D communications latency using IoTurva

Table 2: Throughput (Mbps) achieved when using IoTurva.

Server	D1	D2	D3	D4
S1	25.1 (±0.3)	33.5 (±0.1)	33.1 (±0.1)	42.9 (±0.1)
S2	16.4 (±0.9)	14.4 (±0.9)	14.1 (±1.3)	32.7 (±0.8)
S3	9.6 (±10.7)	8.9 (±12.1)	9.5 (±11.1)	16.1 (±6.3)
S4	17.7 (±3.1)	15.8 (±3.9)	16.6 (±3.2)	30.9 (±2.1)

Table 2 shows the effect on network throughput with IoTurva in comparison with traditional network setup. We used local and remote servers (deployed in Amazon and FUNET cloud) for this test where *S1: localserver,S2: iperf.funet.fi, S3:iperf.scottlinux.com, S4:bouygues.testdebit.info.*

4 DISCUSSION

The evaluation shows that IoTurva can efficiently secure edge networks by identifying and blocking malicious D2D communications in real time with high accuracy. The seggregation of training and classification phases allows us to take the advantage of advanced anomaly detection mechainsm without requiring special hardware deployment in edge network. Currently, our model training focuses on minimizng *false-negatives* to remove any exploitable weak links in the network. However, our future work focuses on minimizing *false-positives* as they affect user experience by obstructing normal device operations (e.g. user is not able to use his smartphone to unlock the door because system thinks its an attack). To support incremental and easy deployment, Turva Gateway is designed to have minimal resource footprint to enable deployment using PC-on-a-card devices or legacy access points/gateways with OF support. We will extend this work with improved data collection, data processing schemes and improve our model to support classification of multi–level D2D interactions by incorporating state and other device–based features.

REFERENCES

[1] 2017. Common Vulnerabilities and Exposures. https://cve.mitre.org/. (2017). [Online; accessed 04-June-2017].

[2] 2017. Common Weakness Enumeration. https://cwe.mitre.org/. (2017). [Online; accessed 04-June-2017].

[3] Yves-Alexandre de Montjoye, Erez Shmueli, Samuel S. Wang, and Alex Sandy Pentland. 2014. openPDS: Protecting the Privacy of Metadata through SafeAnswers. *PLoS ONE* 9, 7 (07 2014), e98790. DOI:http://dx.doi.org/10.1371/journal.pone.0098790

[4] Ibbad Hafeez, Aaron Yi Ding, Lauri Suomalainen, Alexey Kirichenko, and Sasu Tarkoma. 2016. Securebox: Toward Safer and Smarter IoT Networks. In *Proceedings of the 2016 ACM Workshop on Cloud-Assisted Networking (CAN '16)*. ACM, New York, NY, USA, 55–60. DOI:http://dx.doi.org/10.1145/3010079.3012014

[5] Hiroshi Kajino, Hiromi Arai, and Hisashi Kashima. 2014. Preserving worker privacy in crowdsourcing. *Data Mining and Knowledge Discovery* 28, 5 (2014), 1314–1335. DOI:http://dx.doi.org/10.1007/s10618-014-0352-3

[6] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. 2016. IoT Sentinel: Automated Device-Type Identification for Security Enforcement in IoT. *CoRR* abs/1611.04880 (2016). http://arxiv.org/abs/1611.04880

[7] H. Ning, H. Liu, and L. T. Yang. 2013. Cyberentity Security in the Internet of Things. *Computer* 46, 4 (April 2013), 46–53. DOI:http://dx.doi.org/10.1109/MC.2013.74

[8] Nicky Woolf. 2017. DDoS attack that disrupted internet was largest of its kind in history, experts say. https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet. (2017). [Online; accessed 04-June-2017].

[9] Lenny Zelster. 2017. Malware Sample Sources for Researchers. https://zeltser.com/malware-sample-sources/. (2017). [Online; accessed 04-June-2017].