Poster: Fingerprinting IoT Devices in Open-world Setting

Dilawer Ahmed
North Carolina State University
dahmed2@ncsu.edu

Benjamin Zhang
North Carolina School of Science and Math
zhang22benjamin@ncssm.edu

Anupam Das
North Carolina State University
anupam.das@ncsu.edu

***Abstract*—Existing fingerprinting attacks on IoT devices have primarily focused on closed-world settings and lack comprehensive open-world analysis. In this poster, we try to understand how effectively an attacker can fingerprint unseen targeted IoT devices when building a classifier using either devices manufactured by the same company or devices with similar functionality. We find that an adversary benefits when the training set contains at least one device type per company, enabling it to predict the other devices manufactured by the same company even when the device functionality might be different.**

I. Introduction

Recent years have seen a surge in popularity in smart home IoT products due to the convenience they provide through the automation of appliances. However, such convenience often comes with unforeseen security and privacy risks. For example, simply knowing which devices are present in a household can be sensitive, e.g., knowing the existence of a heart rate monitor within a home can reveal the health condition of the inhabitant. Similarly, knowing the exact make and model of a smart lock can potentially help an adversary launch targeted attacks, e.g., exploiting known vulnerabilities that remain unpatched.

Researchers have recently exploited network traffic generated by IoT devices to uniquely fingerprint such devices [3], [7] and infer their device-level activities [4], [6], [9]. However, existing fingerprinting attacks on IoT devices have focused on closed-world settings where the devices can be identified with high accuracy and lack any comprehensive open-world analysis — something that an adversary is bound to face in any real-world setting. In this paper, we focus on determining to what extent an adversary can successfully fingerprint IoT devices in the open-world setting; more specifically, we focus on targeted attacks where the attacker is focusing on inferring a specific device manufactured by a specific vendor or simply a certain category of IoT devices. To answer this question, we first collect network traffic generated by a significant number of IoT devices. Next, we build our own device fingerprinting technique using well-known features [1] and perform open-world evaluations to quantify how well an attacker can launch targeted attacks.

II. Methodology

Datasets. We used six different datasets containing a total of 188 IoT devices (out of which 120 were unique make and model). Some of the datasets also contained traffic from non-IoT devices such as printers, phones, laptops and tablets. These datasets contain popular public datasets such as YourThings [2], UNSW IoT traces [8]. Other datasets from recent works were made available upon request such as Mon(IoT)r datasets [6], HomeSnitch [4] and PingPong [9]. We also collect our own dataset. These datasets contain traffic from multiple geographical regions such as the US, Europe, Australia. They also contain different types of traffic, such as continuous and event-based traffic, and vary in time from when they were collected from 2016 to 2021. We combined all these datasets and their devices in one combined dataset for our analysis in an open-world setting. We extracted 154 features using 5-minute windows out of which 21 features are multi-valued bag-of-words representation which are serialized and one-hot encoded during model training

Threat model. Our threat model is defined as any passive on-path adversary sniffing network traffic from the home router. This definition can include an ISP or any other upstream sniffer. The adversary is also able to use the datasets available publicly and, if needed, can also collect its own traffic. However, the adversary cannot access the local area traffic (e.g., ARP) and does not modify any traffic from the devices. The adversary is also constrained by the fact that they cannot collect or otherwise obtain a dataset that contains data from all IoT devices in the wild.

Open-world Perspective. Traditional work on IoT device fingerprinting has focused on closed-world fingerprinting where, given a labeled dataset, they train and test on data from the same dataset. In the real world, however, many devices are not seen by the classifier. So a useful insight, in this case, is to determine if a given device is among the devices previously seen by the classifier or not. The next step is to gain additional information about the unseen devices. For example, if the classifier has not seen a Nest Camera but has seen other cameras, it can classify the device as a camera.

Classifier Design. In terms of machine learning models, we found Random Forests [5] to be most effective (we also tested SVM and decision trees). We set the number of trees in the forest to 100 (i.e., *n_estimators*=100). We did not use deep-learning techniques for better explainability, like understanding why certain features rank at the top. To evaluate the effectiveness of our model, we use well-known metrics, like accuracy, precision and recall. We perform 5-fold cross-validation to account for the randomness inherent in the training data and repeat the whole process 10 times to account for any randomness inherent to the Random Forest model. We report the mean and 95% confidence interval (CI) over 10 runs unless mentioned otherwise.

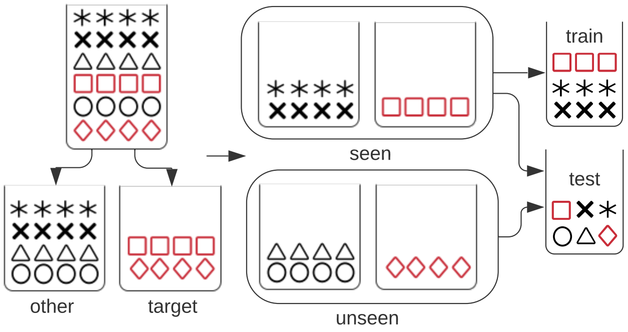


Fig. 1. Process of creating train and test set for the targeted attack scenario. Each shape is a device sample and same shapes denote samples from the same device. Red and black color denote target and other devices respectively.

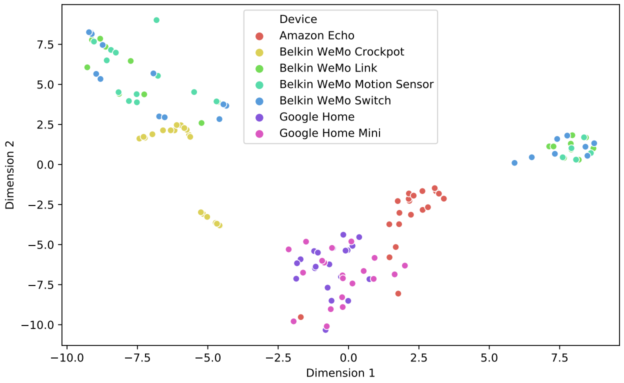


Fig. 2. t-SNE scatter plot showing different devices with similar functionality and vendor have similar fingerprints

Targeted Attacks. The aim is to identify *similar* devices. If we have devices from the same vendor, e.g., Amazon Echo Dot, we want to see if we can then identify, using network traffic only, other devices from that vendor, e.g., Amazon Echo Look. The first step is to identify the target group of devices. We label all the target device samples as ”target” and all the other devices, which are a mix of IoT and non-IoT devices, as ”other”. These are colored as red and black in Figure 1, respectively. We then randomly split devices equally from both sets to create the corresponding *seen* and *unseen* datasets. We then perform a standard 80:20 train-test split on the datasets and use only the *seen* dataset to create the *train* set while both *unseen* and *seen* samples are used to create the *test* set. We then train the classifier using only the devices in the *train* set. We then measure the performance of the classifier using the *test* dataset. The overall process is explained in Figure 1.

III. Results

We evaluated our approach on different possible target groups, including groups such as similar series of devices, devices from the same vendor, devices with similar functionality, and a combination of these possible groups. The results are shown in Table I. We see high accuracy across the board, and precision is on the higher side as well. In certain cases, we observe low recall scores. This occurs for devices/groups which are more generic, e.g., different models of Roku TV provide Amazon Alexa and Google Assistant support. Some groups also contain devices from different vendors, which might be different in their functionality as well, e.g., Smart plugs and switches are made by a lot of different vendors. Conversly, Belkin WeMo and Google Home have less variability in their network fingerprints, as shown through the scatter plot in Figure 2.

TABLE I. Effectiveness of classifier under targeted device settings

Group	Accuracy	Precision	Recall
Geeni Cameras	98.03 (0.44)	89.39 (4.61)	69.26 (10.09)
Google Home Devices	99.74 (0.12)	99.66 (0.64)	88.98 (5.5)
Roku TV	98.6 (0.38)	91.37 (4.98)	61.42 (12.62)
Ring Doorbells	98.46 (0.47)	99.94 (0.05)	65.9 (10.45)
Amazon Echo Devices	97.78 (0.53)	88.47 (3.9)	69.72 (9.59)
Belkin WeMo Devices	99.73 (0.21)	99.68 (0.37)	96.9 (2.48)
Smart Switches	95.29 (0.59)	77.24 (4.86)	60.67 (6.34)

IV. Conclusion and Discussion

We show that it is possible to fingerprint previously unseen devices which are similar to other devices with high accuracy and precision. We also observe that these IoT device fingerprints are relatively stable over time (our datasets cover data over multiple years) because of minimal updates and changes in traffic patterns. Furthermore, we observe that some devices have similar fingerprints, e.g., Belkin WeMo devices and potentially share the same network stack and infrastructure. Overall, the network traffic of an IoT device can be easily fingerprinted, and in the future, we plan to explore feasible countermeasures.

References

[1] D. Ahmed, A. Das, and F. Zaffar, “Analyzing the feasibility and generalizability of fingerprinting internet of things devices,” *Proceedings on Privacy Enhancing Technologies (PoPETs)*, vol. 2022, no. 2, 2022.

[2] O. Alrawi, C. Lever, M. Antonakakis, and F. Monrose, “Sok: Security evaluation of home-based IoT deployments,” in *Proceedings of the 40th IEEE Symposium on Security and Privacy (SP)*, 2019, pp. 1362–1380.

[3] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, “Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic,” *CoRR*, vol. abs/1708.05044, 2017. [Online]. Available: http://arxiv.org/abs/1708.05044

[4] T. OConnor, R. Mohamed, M. Miettinen, W. Enck, B. Reaves, and A.-R. Sadeghi, “Homesnitch: behavior transparency and control for smart home IoT devices,” in *Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks (WiSec)*, 2019, pp. 128–138.

[5] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[6] J. Ren, D. J. Dubois, D. Choffnes, A. M. Mandalari, R. Kolcun, and H. Haddadi, “Information exposure from consumer IoT devices: A multidimensional, network-informed measurement approach,” in *Proceedings of the 19th Internet Measurement Conference (IMC)*, 2019, pp. 267–279.

[7] A. Sivanathan, H. H. Gharakheili, F. Loi, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, “Classifying IoT devices in smart environments using network traffic characteristics,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 8, pp. 1745–1759, 2018.

[8] A. Sivanathan, D. Sherratt, H. H. Gharakheili, A. Radford, C. Wijenayake, A. Vishwanath, and V. Sivaraman, “Characterizing and classifying IoT traffic in smart cities and campuses,” in *IEEE Conference on Computer Communications Workshops*, 2017, pp. 559–564.

[9] R. Trimananda, J. Varmarken, A. Markopoulou, and B. Demsky, “Packet-level signatures for smart home devices,” in *Proceedings of the 27th Annual Network and Distributed System Security Symposium (NDSS)*, 2020.

We show attackers can fingerprint previously unseen devices in an open-world setting with an average accuracy of more than 97%

Fingerprinting IoT Devices in Open-world Setting

Dilawer Ahmed, Benjamin Zhang, Anupam Das

INTRODUCTION

- There has been a rapid increase in prevalence of IoT devices, which can lead to unforeseen security and privacy risks.
- We aim to see if, in an open-world setting, an adversary can successfully fingerprint previously unseen devices with a high degree of precision.
- To that end, we aim to identify devices which have a *similar* functionality, such as identifying an Amazon Echo Dot device from network traffic from other Amazon devices.

METHODOLOGY

- We used data from 6 datasets including 188 IoT devices, of which 120 were unique make and models.
- We split *target* group devices and other devices randomly into to *seen* and *unseen* sets as shown in the Figure below.
- We use a Random Forest classifier (100 trees).
- We use 154 features out of which 21 were bag-of-word representation for features such as ports, ips, hostnames.
- We also vary the ratio of known-to-unknown devices and show even with decent ratios results are similar

RESULTS

- We show we can achieve good AUC scores for open-world settings and distinguish effectively between previously known and unknown devices.
- We show, compared to extended open-world containing all devices to gain additional information, we achieve much better results when targeting a particular device group.
- We observed high accuracy values across the board, and relatively high precision values as well.
- We observe that certain recall values were low, which occur in groups that have more generic devices.
- We observe better results on target groups with similar fingerprints as seen is t-SNE scatter plot.
- Overall, we observe that it is fairly easy to fingerprint the network traffic of an IoT device.

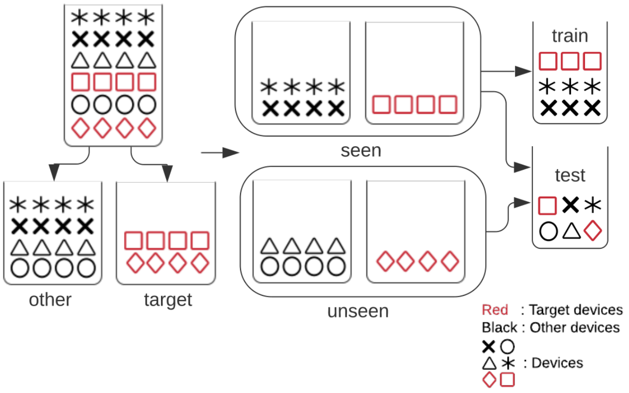
FUTURE DIRECTIONS

- Generalizability of classifiers across time and space.
- Feasible countermeasures.

Dataset information

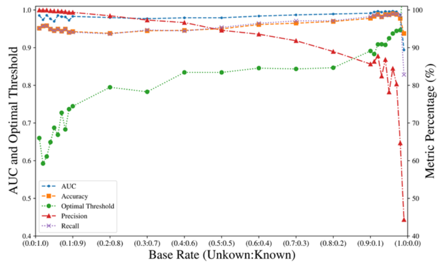
Name (Country)	Capture Period (Duration in days)	IoT Devices (Unique)
YourThings (US)	Early 2018 (11)	45 (45)
HomeSnitch(US)	Early 2020 (12)	28 (24)
PingPong (US)	Late 2019 (51)	18 (17)
Mon(IoT)r (US)	Early 2019 (14)	41 (41)
Mon(IoT)r (UK)	Early 2019 (17)	29 (29)
UNSW (AU)	Late 2016 (21)	19 (19)
Our (US)	Early 2020 (11)	8 (8)

Splitting devices for targeted attacks

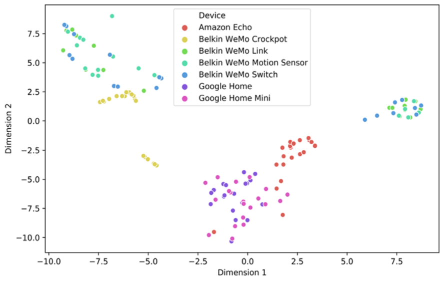


Device Group	Accuracy	Precision	Recall
Geeni Cameras	98.03	89.39	69.26
Google Home	99.74	99.66	88.98
Roku TV	98.6	91.37	61.42
Ring Doorbells	98.46	99.94	65.9
Amazon Echo	97.78	88.47	69.72
Belkin WeMo	99.73	99.68	96.9
Smart Switches	95.29	77.24	60.67

Varying known-to-unknown ratio against metrics



t-SNE Scatter plot of device samples



Contact:

- Dilawer Ahmed (dahmed2@ncsu.edu)