Spying on Smart Home Devices

Jiajia Liang
University of Virginia
jl9pg@virginia.edu

Abstract

Internet of Things(IoT) and smart home devices are having increased popularity in everyday home, providing functionality such as plugs, light bulbs, speakers, and cameras. While it is convenient to use, the privacy issue is a critical aspect of concern. For example, an adversary can perform a passive attack, eavesdropping and potentially inferring sensitive information about the user's home and user's activities.

In this work, I demonstrate various ways to infer device identities as well as user activities using smart home traffic. Second, I develop an efficient visualization tool for device identification using DNS queries. Third, two case studies are analyzed to demonstrate how much information can be inferred for the selected smart home devices.

Keywords: Internet of Things, smart home, fingerprinting, privacy

1 Introduction

Smart home devices have rapidly increased in popularity and commercial availability within the past few years. These internet-connected devices often have always-on sensors that capture user's activity indoors, and send those data to cloud servers run by device manufactures. A variety of activities that can be captured with existing commercial products include sleeping patterns, exercise routines, children's behaviors, medical information, sexual activities, and much more[1].

While purchasing the device, users can expect device's manufacturer to collect and analysis their data to provide service, however, smart home devices are also susceptible to eavesdropping by other parties as well, including Internet service providers, Wi-Fi eavesdroppers, or state-level surveillance entities.

Most device manufacturers encrypt the data stream to protect against such passive adversary attacks. However, it is possible for an attack to use meta-data such as DNS queries, or device fingerprinting techniques to identify devices, and even further infer users' activities. This project explores to what extend smart-home device traffic can reveal private information to a passive adversary.

The paper will be organized as the following: Section 2 is a brief summary of state-of-the-art research work on IoT information exposure and smart home fingerprinting. Section 3 discusses methods to infer device identification and demonstrates the implementation and results of an efficient visualization tool to infer device identity. Inferring user's in-doors activities will be discussed in Section 4. Section 5 showcases two cases study using data collected on my own. Finally, a brief discussion is included in section 6.

2 Related Work

Device Identification: In prior research work on IoT information exposure and traffic attack, Apthorpe et al. show that using MAC addresses, DNS Queries, and traffic rates, it is feasible to identify some smart home devices[1]. Perdisci et al. demonstrate using a multi-label classifier to identify IoT devices via passive DNS traffic[6]. They also release a large-scale dataset contained DNS traffic, which will be used in this project as well.

Indoor activities inference: Wang et al. use the deep learning method to fingerprint encrypted voice traffic on smart speakers, and evaluation shows that the voice commend inference over encrypted traffic has an accuracy of 92.89%, compared with 1% random guess baseline[5]. Rahmadi et al. develop PINGPONG, a tool that can automatically extract packet-level signatures for device events[4]. The work demonstrated what they are able to detect devices or specific events with an average recall of more than 97%. Ren et al. collect large-scale controlled experiments using 81 smart home devices and find that a passive eavesdropper can reliably infer user and device behavior from the traffic of 30/81 devices they tested [4].

3 Device Identification

Knowing users have a TV doesn't provide much information to the adversary. However, in some other cases, the device itself carries some private information that is sensitive to users. For example, a baby monitor camera might imply users have a newborn at home, senior alert devices can imply users having elderly at home. In this section, I show two different ways to identify smart home devices, and then analyze how reliable and generalizable the approaches are.

3.1 Using DNS Queries

Previous work showed that DNS queries associated with each flow can often be associated with a particular device [1]. To understand what is the proportion of the devices that issued DNS queries for domains that are easily associated with the device, I create a visualization tool to easily extract DNS queries for specific IP addresses.

3.1.1 Dataset. The data used in this section is from IoTFinder Data [2]. The data contains DNS queries from 57 different devices and has a ground truth mapping from device IP with the device name, which can be used to determine whether the DNS queries reveal information about the manufacturer and the product itself.

3.1.2 Method. Existing tools such as Extract DNS utilizes tcpdump to extract DNS queries from PCAP files. The tool decodes each packet layer by layer and is therefore not efficient and can not be used for a large dataset like IOTFinder data. To address this problem, I observe that the queried DNS domain name always appears in the relatively same position in the DNS message. Therefore, the domain name can be easily extracted using regular expression. After extracting the domain name in the queried, the domain name is broken into word segment and a word cloud is being plot based on the frequency of each word being returned. The motivation of segmenting into words and utilize frequency is to emphasize words that have actual meaning (e.g. company name, device name, geolocation) while de-emphasizing meaningless words (e.g. ID).

3.1.3 Findings. The tool is extremely efficient, and work on large Pcap files. By examining the word cloud, we can easily determine whether the DNS queries can be mapped to a specific manufacturer, or mapped to a specific device, or not. For example, Fig 1 shows the word cloud extracted from ip=192.168.0.36 in the dataset. The device (doorbell) and the company name(August) are easily observed.

However, there are DNS queries for the domain that could only be associated with a manufacturer but not a particular device. An example of such a case is shown in Fig 2, extracted from ip=192.168.0.27 which is a Netgear Arlo Camera. From the word cloud, we observed that the device is associated with Arlo company. However, besides selling the camera, the company also sells smart doorbells. Therefore, the specific product type cannot be inferred from the DNS queries domain.

In addition, some products host their service on third-party cloud hosting services like Amazon AWS, therefore, the DNS queries for domain name do not produce useful association. association.

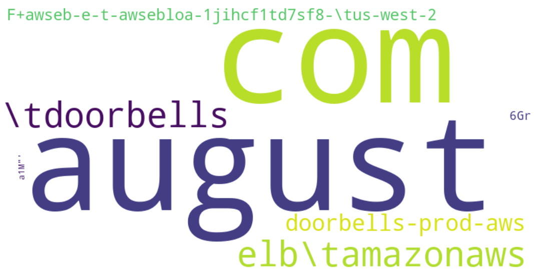

Figure 1. Word Cloud for DNS queries associated with August doorbell. The product is easily identified.

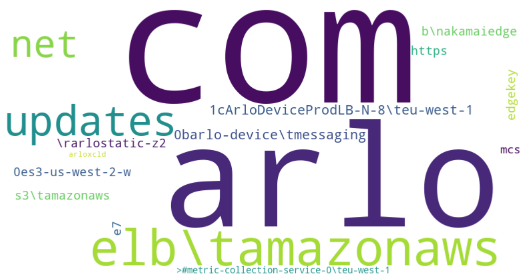

Figure 2. Word Cloud for DNS queries associated with Netgear Arlo Camera. Only the company name is inferable from the figure.

Among 57 devices in the dataset, I manually exam to what extend the DNS queries domain reveals device identity. The summarize results is shown in Table 1. From the table, we can see that, although the majority of the devices reveal the information about the manufacture, this information by itself is not very useful because majority company has multiple products. However, 20 out of 57 devices reveal device product type in the queried domain name. An adversary could use a laboratory setup to learn these mappings or perform reverse DNS lookups to pair service IPs with device-identifying domain names, and further tracking the packet coming from and to the device.

	Reveal Manufacture	Reveal Product
Number of Device	45	20

Table 1. Number of devices which DNS queries reveal the identity of the device.

3.2 Using k-clustering

Inspired by [3], I use the k-clustering algorithm to examine whether the different devices can be inferred from the traffic features. Experiment use Mon(IoT)r dataset. Mon(IoT)r dataset includes network traffic for IoT devices deployed on a local network, the data was collected in a controlled setting in US and UK testbeds and have the devices name and activities label. I select 5 devices from different categories, and the selected devices are all popular selling products in each category. Traffic associated with those selected devices in the US testbed were combined and perform a k-clustering using knn and random forest algorithm. The devices selected and the activities which the traffic captured are summarized in Table 2.

In total, 29 features are being used. *meanBytes, minBytes, maxBytes, medAbsDev, skewLength,kurtosisLength, q10, q20, q30, q40, q50, q60,q70, q80, q90, spanOfGroup, meanTBP, varTBP, medianTBP, kurtosisTBP, skewTBP, network to, network from, network both, network to external, network local, anonymous source destination, time start, time end.*

Device Category	Device Name	Activities Included
Camera	Amazon Cloud cam	Alexa stop, Alexa watch, Android lan watch, Android wan watch
Home Automation	Tplink bulb	Android wan color, Android wan dim, Android wan off, Android wan on
TV	Roku-tv	Android lan remote, Local menu
Audo	Google home mini	Local voice, Local volumn
Appliances	Samsung fridge	Android wan set, Android wan view inside

Table 2. Devices and activities selected for identifying devices using k-clustering

To reliably classify devices from features, the model uses 7/3 split cross-validation (Train on randomly selected 70% of the data and test on the 30% remaining data, and the process is repeated 10 times to get the average metrics).

The accuracy for using Knn is 0.989 and the accuracy for using random forest is 0.989 as well, showing that the k-clustering can accurately distinguish the traffic pattern of different devices using the features we defined. The visualization of the clustering produced by random forest algorithm is shown in Fig 3, using t-distributed Stochastic Neighbor Embedding.

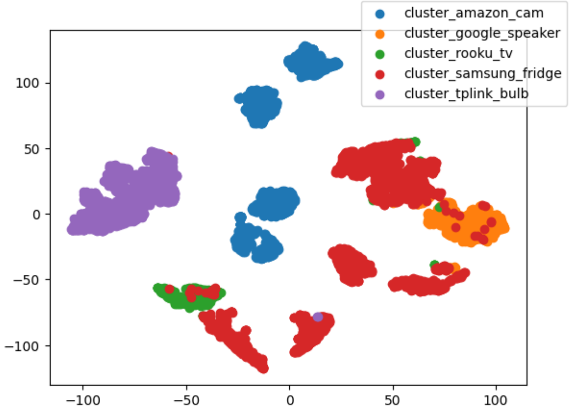

Figure 3. Device clustering with Random Forest projects in 2D.

The preliminary results show that different device has distinguishable traffic pattern. Future study should consider using devices from the same manufacturer to explore whether the differences comes from devices itself or manufacturer.

4 Activity Inference using k-clustering

In this section, I am interested in whether the user's activities for the same device can be accurately classified. As reported in [3], among 81 devices being tested in the US testbed, 30 of them are inferrable. I replicate their findings and verify that some devices can be accurately inferred, and some are not. The features and methods being used are identical to those described in Section 3.2. An example of an activity-inferrable device and its activity classification is shown in 4, which was tested on Yi camera. With the Knn model. the classifier is able to achieve an accuracy 0.948, and with random forest, the classification accuracy is 0.980. The takeaway for this section is that although the majority of the activities are non-inferrable from the features we extract, a small portion of the device activities can be accurately inferred.

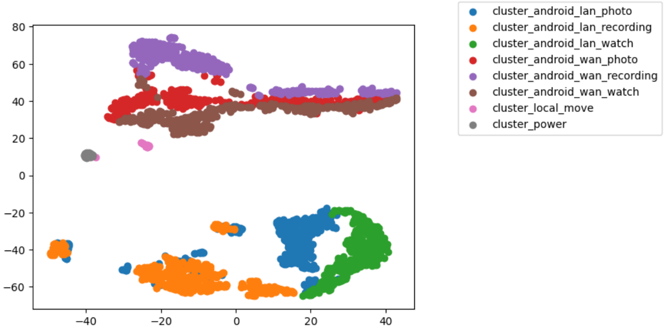

Figure 4. Yi camera activities clustering with Random Forest projects in 2D

5 Case Studies

The case studies analyze the network traffic pattern of two of the devices, namely (1) Google Nest Mini 2, and (2) Sengled Smart Color Light Bulb. Traffic data were collected with ground truth being labeled. The goal is to see how much information can be accurately extracted or inferred from the traffic data. The devices were connected to the laptop's hot spot, and I used Wireshark to capture traffic data go through the laptop. Based on the IP address of the two devices, we can separate the traffic between these two and analyze the traffic pattern for each of the devices.

5.1 Google Nest Mini 2

Google Nest Mini is a smart speaker with which users can interact using Google Assistant voice recognition or control directly from Google Home App. It can be used to control smart devices at home, such as light bulbs, plugs, and thermometers, and can be connected with thousands of apps and partners to play music, add a meeting, voice calls, etc.

5.1.1 DNS Queries.

Google Nest Mini's DNS queries are shown in Table 3. Similar to DNS queries keywords extracted using Section 3.1 methodology, the DNS reveals some information that the device is constantly visiting Google services. In addition, the DNS queries contain the addition keyword "home-devices", which was not seen in the Section3.1 dataset. This can reveal additional information that a google home device is being used in the house.

Queries name
home-devices.googleapis.com
geller-pa.googleapis.com
clients3.google.com
encrypted-tbn0.gstatic.com

Table 3. DNS queries associated with Google Nest Mini

5.1.2 Activities Type Inferring using k-clustering.

In the Mon(IoT)r dataset, data collected with Google Home Mini contains three activities, namely (1) local voice: using voice command (2) local volume: change device volume, and (3)power: turn on the device. Mon(IoT)r data collected in US testbed contains 40 traces of local voice category, 40 traces of local volume category, and only 3 traces of power category. Using the k-clustering algorithm describe in Section 4, the accuracy score for activity classification is 0.651, and 0.655 for knn and random forest, respectively. The classification of data points projects in the 2D plane is shown in Fig. 5. From the accuracy score and the figure, the activities cannot be accurately inferred from the network traffic. This might due to the imbalance of data, small training and testing set, or the features being used in our clustering cannot well capturing the charateristic of the activities.

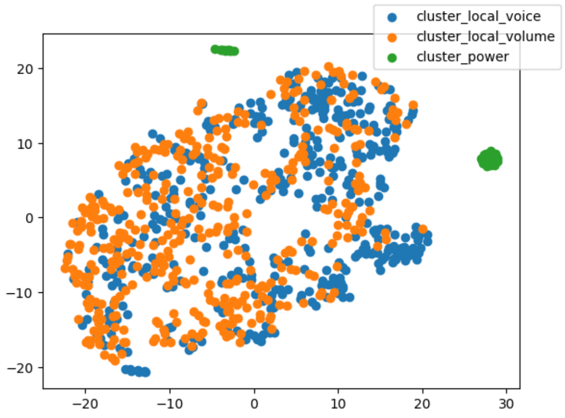

Figure 5. Google Nest Mini clustering with Random Forest projects in 2D

ID	Voice Command
1	Hey Google, what Do You Think of Alexa?
2	Hey Google, are you married?
3	Hey Google, self-destruct?
4	Hey Google, do the dishes.
5	Hey Google, sing me a song.
6	Hey Google, what's the weather today?
7	Hey Google, add a meeting to my calendar.
8	Hey Google, how do I say "Nice to meet you" in French?
9	Hey Google, give me a hug.
10	Hey Google, play some music on spotify.

Table 4. Voice command used in the experiment

5.1.3 Activities Inferring using packet rate.

While we cannot infer the type of activities from the traffic, it is worth exploring the correlation between activities and traffic size, as well as activities and traffic rate. Specifically, we want to measure whether an interaction between users and speaker is visible from traffic size and traffic rate. Traffic size is defined by the number of bytes of the sending or receiving package at the specific timestamp, while traffic rate is defined as the total number of bytes that are sent or received in 1 second time window.

To collect the data, I formulate 10 voice commands and then use a text-to-speech tool to play each command with a traditional speaker. There were 30 seconds gap between each command to make sure the traffic associated with the previous command has been captured. The voice command is shown in Table 4, and the commands are asked in the order listed. The correlation between activities and traffic length and traffic rate are shown in Fig 6 and Fig 7.

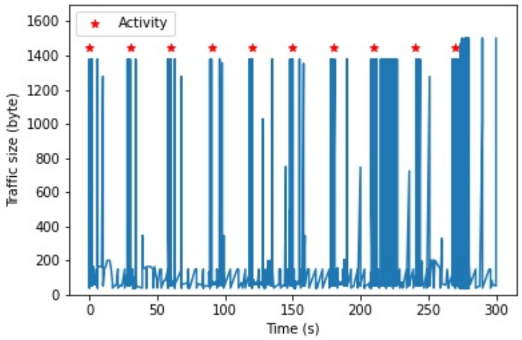

Figure 6. Google mini traffic size in bytes at given timestamp during the 10 voice command interaction.

From Fig 6, we can see that every time there is an activity, there is a sudden increase in the traffic being sent and being received. While activities 0 to 9 ask for some information or perform a single command, the last activity asks for music

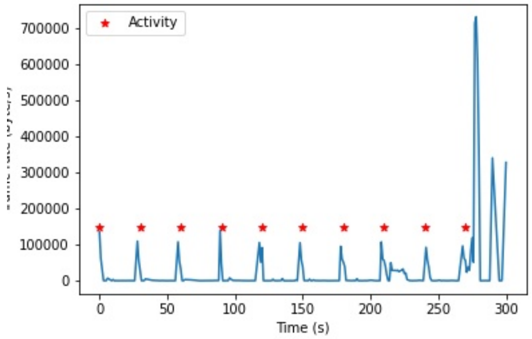

Figure 7. Google mini traffic rate in bytes/sec at given timestamp during the 10 voice command interaction.

streaming. This continuous data sending and receiving for music streaming can be observed in Fig 7, where the traffic rate is high and remain high.

5.2 Sengled Smart Light Bulb

Sengled smart light bulb uses Wi-Fi for smart lighting. The light bulb can be controlled using the Google Nest Mini, and also can be controlled on the smartphone with Sengled Home app. Experiment data was collected using the control from a smartphone. Two light bulbs are being studied. One is the colored light bulb, in which color temperature, color, and brightness can be modified. The other one is the traditional white light bulb, in which users can only change the brightness level.

5.2.1 DNS Queries. No DNS queries associated with the company or the product have been found. One possible explanation is that the server is hosted on some third-party cloud servers.

5.3 Activities Infer using packet rate

Similarly, I am interested in whether there is a correlation between activities and traffic rate. I define a series of activities, and the activities sequences are shown in Table 5. The light bulb was controlled using smartphone app, and each activity is scheduled with 20 seconds gap. The experiment used Sengled Color light bulb since there were more possible operations compared with the white light bulb.

The correlation between activities and traffic is shown in Fig 8 and Fig 9. From the figure, the correlation between activity and traffic size can be observed. Such a pattern can be used to infer when a user is interacting with the light bulb.

ID	Activity
1	Remain Off
2	Turn on
3	Turn off
4	Turn on
5	Change brightness
6	Change color temperature
7	change color
8	Turn off

Table 5. Sequence of activities tested on Sengled Color light bulb, controlled with Sengled Home App on smartphone

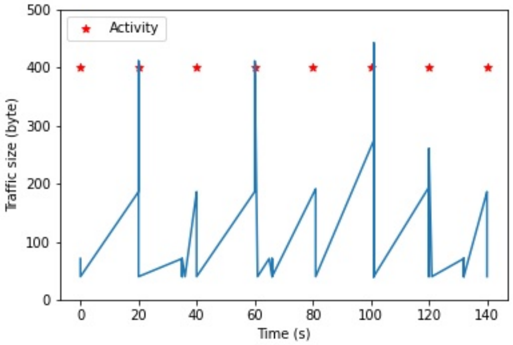

Figure 8. Sengled color bulb traffic size in bytes at given timestamp during the interaction

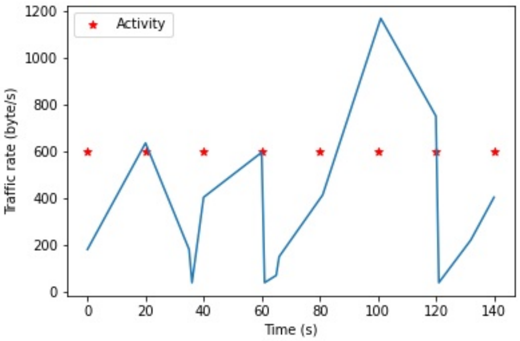

Figure 9. Sengled color bulb traffic rate in bytes/sec at given timestamp during the interaction

5.4 Activities Type Infer from traffic pattern

Since Mon(IoT)r dataset does not contain data for Sengled light bulbs, k-clustering cannot be used to infer user's activities. However, since light bulb operation should not contain complicated traffic, I manually inspect the traffic for each activity to explore patterns. Data sent between the light bulb and the server using BitTorrent protocol. The traffic patterns are consistent across time, and patterns are almost identical for a color light bulb and a white light bulb. In addition, the pattern for different activities is distinguishable. The pattern is summarized in 6. With such a strong association between

Activity	Color packet len	White packet len
Rest	85, 85	85, 85
Turn on	200, 426	200, 269
Turn off	200, 193	200, 193
Change brightness	205, 198	205, 198
Change temperature	299, [277, 426]	/
Change Color	207, 273 208, 274 209, 275	/

Table 6. Traffic length for each activity. First element represents traffic designated to the device, and second element represents traffic originated from the device

activity type and traffic pattern, it is easy for the attacker to infer the activity in the house. However, this pattern is only applied to the light bulb for Sengled company. Therefore, although there exists a strong association, this traffic pattern information is not generalizable.

6 Conclusion and Discussion

In this work, I show that although smart home devices widely adopt data encryption mechanisms to protect against privacy leaks, there are various ways that a passive adversary can identify devices and user activities using traffic meta-data or using traffic fingerprinting methods. In addition, two case studies show some interesting findings related to the two devices being analyzed.

Despite all the interesting findings, the analysis is done mostly on small-scale data, and data were collected in a controlled setting. In future work, it is worth exploring how feasible and reliable those methods work in an open-world setting. In addition, the work only focuses on a small set of devices and most data collection process remains manual work. In order to have a more complete view of the threats and protections on the current commercial smart home devices, the future study can develop efficient ways to collect smart home traffic data along with ground truths and utilize this tool to collect traffic data for a larger set of smart home devices.

References

[1] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. 2017. Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic. arXiv:1708.05044 [cs.CR]

[2] Roberto Perdisci, Thomas Papastergiou, Omar Alrawi, and Manos Antonakakis. 2020. IoTFinder: Efficient Large-Scale Identification of IoT Devices via Passive DNS Traffic Analysis. In *2020 IEEE European Symposium on Security and Privacy (EuroS P)*. 474–489. https://doi.org/10.1109/EuroSP48549.2020.00037

[3] Jingjing Ren, Daniel J. Dubois, David Choffnes, Anna Maria Mandalari, Roman Kolcun, and Hamed Haddadi. 2019. Information Exposure for Consumer IoT Devices: A Multidimensional, Network-Informed Measurement Approach. In *Proc. of the Internet Measurement Conference (IMC)*.

[4] Rahmadi Trimananda, Janus Varmarken, Athina Markopoulou, and Brian Demsky. 2020. PingPong: Packet-Level Signatures for Smart Home Device Events. arXiv:1907.11797 [cs.NI]

[5] Chenggang Wang, Sean Kennedy, Haipeng Li, King Hudson, Gowtham Atluri, Xuetao Wei, Wenhai Sun, and Boyang Wang. 2020. Fingerprinting Encrypted Voice Traffic on Smart Speakers with Deep Learning. arXiv:2005.09800 [cs.CR]

[6] Poonam Yadav, Angelo Feraudo, Budi Arief, Siamak F. Shahandashti, and Vassilios G. Vassilakis. 2020. Position paper: A systematic framework for categorising IoT device fingerprinting mechanisms. arXiv:2010.08466 [cs.NI]