18731 Spring 2018 Project Milestone
IoT Device Type Identification

Man Li, Qing Liu, Rongzhi Wan

March 29, 2018

Abstract

Quick 4-5 sentence abstract of the work: 1. Motivate problem 2. What is known 3. What you propose to do that is different 4. Expected outcomes

As Internet-of-things (IoT) gets more and more popularity, security issues of IoT device are gaining more attention. Device type identification is considered an important topic of IoT security for that device type can be exploited by many security approached to provide protection for IoT devices. Works have been done on extracting features from packet generated by the device and use those features for classification. This work is focused on improving classification accuracy by applying different feature engineering techniques on different traffic type. A general feature engineering strategy for IoT device type identification should be given as the contribution of the work.

1 Problem Statement

What is the problem that this project is going to address? Does it matter: why is the problem important? Who will benefit when the problem is solved

Has your problem statement been updated or pivoted from the proposal? Why? Did you address comments from the feedback. 0.5 page

Currently we are in the middle of the big trend of IoT. Connecting identifiable smart devices allowing them to exchange data has facilitated peoples life greatly. The number of IoT devices is estimated to reach 24 billion in 2020 [1]. However, an thriving area with such coverage and scale is not being paid sufficient attention with respect to security issues. The difficulty lies in the inconsistent nature among different vendors or even among different products of the same vendor. There are no standard security design guidelines that can be followed by all IoT developers. On the other hand, vendors may not value the security issue enough to invest in such development, which makes the problem more difficult to be solved at the design phase of IoT devices. Another concern is that IoT devices have been widely deployed for personal and public uses. Therefore a solution that does not require change in existing IoT networks is being expected. One of the possible solutions is device-type identification, which this work will focus on.

This project will address the problem of device-type identification. Solving this problem can help defend many attacks against IoT devices. Many IoT related attacks, including device compromising, data ex-filtration, and false information injection, can be characterized by unauthorized data transformation. These attacks can often be identified by noticing the difference between the device-type a device claims, and the device-type analyzed from its traffic.

Compared with proposal where our plan is to study type identification for wireless protocols, now we generalize our project by exploring a general feature engineering strategy – not limiting to wired or wireless protocols only. In addition, we expect to study different time period of traffic (both setup phase and afterwards), and to collect more traffic by applying active monitoring.

2 Related Work

What is the state-of-art? Have you done a reasonable

literature survey? How do you think your work will be different and/or better?

0.5 page

Many works on IoT devices identification have been done. One of them is "IoT SENTINEL"[2]. This system is able to automatically identify the types of devices being connected to an IoT network, and to enable corresponding enforcement rules. This system is based on passively observed network traffic during the setup phase of smart home devices. After fingerprinting, devices can be classified by comparing the edit distance of their fingerprints.

Another related work is an analysis of Home IoT network traffic and behaviour[3]. In the setting a smart home, this work aims to understand the overall IO behaviour of devices, including protocol used, data volumes, type, data rate and transmission frequency, etc. Different from IoT SENTINEL, this analysis is based on a relatively long term traffic, including both setup phase and further communications. Also, different from IoT SENTINEL who adopts machine learning techniques, this work uses statistics-based analysis.

These works provide valuable insights and outcomes, yet there are some improvements can be done to better identify devices. One of them is feature selection: in IoT SENTINEL, 23 features are chosen, including the type of protocols used across different layers, packet header information, packet size and so on. In the analysis of home IoT network, it has chosen many features in common such as protocols and data volumes. It also selects different features like data rate and transmission frequency. In our work, in the setting of smart home, we will select more representative and distinguishable features based on a series of experiments. This is expected to yield a higher classification rate. In addition, IoT SENTINEL only analyze header data, without touching any of payload information. Yet many useful features can be found in payload. For example, a camera using plain HTTP without encryption may have some keyword "camera" "video" in HTTP messages. In our work, we value such information in payload. This gives us more information than just looking at headers.

Another improvement can be collecting more traffic. IoT SENTINEL captures device traffic during setup phase, and the home IoT network analysis captures a 22-day period. Both of them use passive monitoring, either by Wireshark sniffing or monitoring tools like Bro. In our work, we expect to try actively sending messages to IoT devices and observe their responses. This may provide more desired information, compared with passive monitoring.

3 Approach and Project Description

What is the basic approach, method, idea or tool thats being suggested to solve the problem? What technical progress have you made in terms of the proposed approach? Have you had to rethink any of the original ideas? How have you addressed comments from the feedback? What tools and experimental platforms have you setup? 1 page

Different IoT device types may use different protocols, have different communication patterns, and send packets with different sizes. Such features is the key to differentiate them. In feature extraction phase, to re-implement the experiments in IoT SENTINEL[2], we refer to their selected features, as listed below:

1. Protocol features
   - Link layer protocol: ARP / LLC
   - Network layer protocol: IP / ICMP / ICMPv6 / EAPoL
   - Transport layer protocol: TCP / UDP
   - Application layer protocol: HTTP / HTTPS / DHCP / BOOTP / SSDP / DNS / MDNS / NTP
2. Network layer features
   - IP options: Padding / RouterAlert
   - Packet content: Size / Raw data
   - IP address: Destination IP counter
3. Transport layer features
   - Port class: Source / Destination

To extract features from Wireshark-captured raw data, we wrote a Python script using "dpkt" package. Output of this feature extraction phase is m files, each of them contains k lines whose format is presented below. Each

fingerprint is represented as a $23*n$ matrix **F**. To facilitate further classification phases, we define our training data as the following format:

type_number \t F_row_cnt \t F_col_cnt \t flattened_F

notations:
23: number of features for now
n: number of packets in each capture.
m: number of devices
k: number of captures for a device. In our case, most of k is 20.

However, we found these 23 selected features by IoT SENTINEL are not good enough for identifying device types. For example, port numbers are mapped into three categories, instead of keeping their original values. Also, some frequently used IoT protocols such as MQTT is not extracted. In addition, timing features are not recorded, losing communication frequency information. Therefore, one of our next steps will be developing better features selecting mechanisms. We expect to do a series of experiments to figure out what features will yield least classification error.

Random Forest Classifier:.....

Random Forest Classifier produces a candidate list for a device.The second layer of classifier - Edit distance classifier will select the final device type from candidate list. To calculate edit distance for fingerprints, we need to represent fingerprint as a word. Recall that each column in a fingerprint is representing a packet. By assigning a distinct ID for each packet(column), well be able to represent a fingerprint in a word-like manner. A fingerprint with n columns will be represented as a word with n characters.

After having word representation for each fingerprint, we can calculate a dissimilarity score for the testing fingerprint with all potential candidate device types. For each candidate device type, 5 fingerprints are randomly sampled as group truth. Edit distance is calculated using Damerau-Levenshtein edit distance algorithm between testing fingerprint and 5 sampled scores. The 5 scores are normalized by dividing the longest distance. Those 5 scores are summed up to calculate the final score for this testing fingerprint with current device type. After calculating a score for all candidates, the device type with smallest dissimilarity score will be chosen as final device type.

4 Preliminary results

What has been done to validate the early approach? (E.g. measurements, simulations, constructing code) 0.5–1 page Considering we only have limited number of IoT devices right now, and in order to get some insights and ideas before diving into the environment setup and real traffic capturing, we first did some experiments on the data captured by IoT SENTINEL[2]. Going further, we consider setting up our own environment to capture traffic.

Currently we have completed initial implementation of all three components – feature extraction, random forest classifier and edit distance classifier. Further tuning and modification is still in progress.

Also, we have successfully generated device fingerprints from the reference data – Wireshark captured raw traffic by IoT SENTINEL[2], and have transformed them to a ready-to-train format. The feature extractor, or data cleaner, employs a Python library "dpkt" that can resolve Wireshark ".pcap" files. By examining packet data, such as protocol used, IP addresses, port numbers and packet size, we are able to generate fingerprints from the raw data.

According to the reference data, there are 32 devices including various types such as camera, switch, and sensor. Since some devices share the same type, these total 32 devices fall into 27 types. To obtain adequate data, for each of those devices, there are multiple captures. Therefore, we have generated 32 files, each of which contains multiple fingerprints corresponding to each capture.

Our project repository, including captured data, cleaned data and implementation code can be found here: https://github.com/RongzhiWan/18731project

5 Stumbling blocks

Have you had any stumbling blocks in making progress in access to code/data/simulation tools etc? How can we help you make better progress? 1-2 paras

6 Next steps Timeline

- This week: Tune the module to achieve classification accuracy that is comparable to the result in referenced paper.
- Week 1: Set up experiment environment
- Week 2: Experiment on feature engineering: Active monitoring, Collecting packets for long term, Keyword extraction
- Week 3: Generalization for unauthorized devices
- Week 4: Work on report and poster

7 References

[1] Intelligence, B. (2018). There will be 24 billion IoT devices installed on Earth by 2020. Business Insider. Retrieved 28 March 2018, from http://www.businessinsider.com/there-will-be-34-billion-iot-devices-installed-on-earth-by-2020-2016-5

[2] Miettinen, M., Marchal, S., Hafeez, I., Asokan, N., Sadeghi, A. R., Tarkoma, S. (2017, June). IoT Sentinel: Automated device-type identification for security enforcement in IoT. In Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on (pp. 2177-2184). IEEE.

[3] Amar, Y., Haddadi, H., Mortier, R., Brown, A., Colley, J., Crabtree, A. (2018). An Analysis of Home IoT Network Traffic and Behaviour. arXiv preprint arXiv:1803.05368.