Senior Thesis

Christian Meinzen

November 10, 2021

Contents

Abstract

I do not have an abstract at the moment.

1 Introduction

As more and more people start to connect IP-enabled devices to a network, the network begins to become more and more crowded with packet information spawning from different devices. In order for a network to be properly managed, knowledge of devices attached to a network is necessary. Such devices, if vulnerable, could be utilized for insider attacks on the network. Hence, device identification is one of the main strategies for monitoring network traffic and the Internet-of-Things (IoT) in order to mitigate malicious attacks [5]. Device identification can have many different implementations. However, a notably prominent strategy is classification using packet feature extraction based upon some genetic algorithm (GA) [2]. Provided that this method is a strong candidate for device identification, it is necessary to root out any defects that may arise when using such classification.

1.1 Overview

1.1.1 Using Various Classification Features

Device identification is done mainly using the construction of *device fingerprints* (DFP), which is the packet header, session, and hardware information of a specific remote device connected to a network [2]. These three categories include a multitude of various features that can be used in the device identifier classification. Past research notes that an IoT device can be distinguished by the specific hardware and firmware information provided. This would include the IMSI number, IMEI number, device identifier, and the manufacturer. Most of these things are not sent as packet information over a network, but it could perhaps be detected by other software packages [3]. Most of the hardware specifications of the device could be either spoofed or encrypted so that the System Identifier will not have access to this specific information. Similarly, other features used in identification include session tuples containing *source IP*, *destination IP*, and *port numbers* (SYN to FIN). While this information is very powerful in device classification, they are easily spoofed over a network [4]. To combat this spoofing problem, some mechanisms have used session information (length, number of messages, minimal size, etc.) to boost the performance of the classification algorithm. This information will be hard to hide in my manipulation of the genetic algorithm input; however, it requires a mechanism to form groups by tying several packets to a specific device [4]. Adding this extra mechanism, while potentially helpful for device identification, creates a level of complexity to the research that is done in this paper that is not necessary for the goal. Because of this, the information found in packet headers will be the source of input for the System Identifier module. Session information is mentioned in Section 8 as an extension of the research done in this paper to handle grouping of packets.

1.1.2 Genetic Algorithms

The classification of devices requires a set of features that have strong weight on determining the device type to allow for the machine learning algorithm to output a significant model for prediction. This is the purpose of the GA: extracting a set of features from a network packet that optimizes prediction power of device-type [1]. The set of features that are extracted from the

GA form the foundation upon which the machine learning classification model trains. It is, then, reasonable to say that the complex task of converting packet information into a device identifier is held within the GA. The classification score of the device identifier is directly related to the output feature set provided by the GA.

1.1.3 SysID Framework

A unique framework for using a genetic algorithm called SysID has been tested and expanded upon looking to improve device identification classification. SysID uses three network data captures that runs a genetic algorithm for feature extraction from any single packet found in the captures to then be trained and tested in a ML classification algorithm. SysID uses the genetic algorithm specifically for generating a set of features that can be found in various network packets such as TCP, UDP, DNS, HTTP, and SSL [1]. This set of features will then be used for the device-identifying, classification algorithm as weighted features. The SysID framework also defines a specific metric to use for the genetic algorithm called fitness. This metric, defined in Section , provides the mathematical description for the weight that any feature has for identifying the device. Also, below provides details as to work related to the SysID framework as well as its performance and future extensions.

1.2 Past Research

1.2.1 Device Identification Using Machine Learning

Yair Meidan et al. proposed a look into using machine learning to identify specific devices on a network based upon traffic information by presenting the ProfilloT framework. They used supervised learning to classify and make a distinction between IoT devices and non-IoT devices interacting on a network. This provided a preliminary mark for device identification without rigorous device classification by simply determining if the devices are IoT or not. Yair Meiden et al. used feature extraction from TCP packets to retrieve source and destination IP addresses [4]. They used the information extracted as an input to two algorithms for classification. The first determines the optimal size of the sequence of sessions for which the classifier classifies correctly. The second is the actual IoT device classification to make predictions if a device is IoT or not. The results were promising with a 99.28% accuracy. This score provides necessary insight into how device identification works using machine learning.

Another paper that introduced advancement to automated device identification is Markus Miettinen et al. framework IoT SENTINEL. The research done in this paper advanced device identification from categories, like IoT devices, to specific device types, like Aria devices or Ednet-Cams [5]. Miettinen et al. also used feature extraction to form *device fingerprints*. Using these fingerprints to identify specific devices over a network was a large leap into helping detect specific device-types as opposed to general categories that were proposed in previous works. Fingerprinting was shown to be an effective proof-of-concept in this paper through the results having 95% accuracy of identification for 17 different devices. However, there were 10 devices that had a lower accuracy of 50%, which was the result of random type assignment [5]. The dropping of accuracy found in the 10 devices was due to a lack of diversity within the data, where the identification model overlooked

this assumption.

1.2.2 Device Identification Using Various Classification Features

When discussing the various frameworks used for classifying device identification, Narges Yousefnezhad et al. paper "Automated IoT Device Identification Based on Full Packet Information Using Real-time Network Traffic" provides context for the various different features that are applied to the learning model. They propose a combination of sensor measurements, statistically based feature sets, and header feature sets based upon their framework. The sets were based upon the different layers within a network such as the Network, Transport, and Application layers as well as other metrics such as flow duration and inter-arrival time [6]. Therefore, they were testing both the physical and software features given by a device. Yousefnezhad et al. proposes three different categories of feature information to base their results: Measurement, Header, and Statistical. Measurement is based upon sensor reading to gather information about the physical state of the device such as temperature or humidity. The header features were based upon packet information given across the network. The statistical features were taken from multiple packets given by the devices to give an understanding of the session such as flow duration and inter-arrival time. After running various ML classification models to these different categories, Yousefnezhad et al. found that using the measurement-only features provided a higher accuracy rating of 92.62%. This lowered to 88.47% when introducing header information on top of the measurement features, and the score lowered to 82.57% when it was simply the header information only. While there is significant evidence for measurement features leading to higher accuracy, the authors mention that gathering the measured features requires additional computation that may not be provided on the network [6]. Thus, I plan to exploit this by focusing on packet header information that can be found across a network.

1.2.3 Device Identification Using SysID Framework and Genetic Algorithms

The System IDentifier (SysID) Framework was first designed by Ahmet Aksoy and Mehmet Gunes outlined in their paper "Automated IoT Device Identification Using Network Traffic". Their goal was mainly to increase the performance of firewalls and improve device restriction among networks using a two-step, machine learning framework in the form of a genetic algorithm and classification. Aksoy and Gunes's framework proved to be effective once the results came out with an average classification performance of 82%. Furthermore, the SysID framework was able to fingerprint smart devices at an accuracy of 96% using very limited feature information provided by packet headers [1]. The authors continued to compare this framework with other device identifying systems. IoT SENTINEL (previously discussed) had a lower classification score, and it required sequences of packets to detect devices as opposed to SysID only needing a single packet from a specific device. While this was a definite leap forward in device identification, there may be a couple of areas within this research that could be improved.

The focus on the research was directed more to the classification algorithms used rather than the genetic algorithm. Thus, there was only one metric that was tested for feature extraction, which is the same feature defined in Section . This provides a heavy emphasis on the metric used as opposed to other optimal solutions that may be found. The SysID framework was also only used

on a single dataset that was used for testing IoT SENTINEL [1]. While this helps with comparing the two frameworks, having diverse sets of data for testing may reveal further issues that might leak into the framework. Rajarshi Chowdhurry et al. research on network analysis was built as an extension on the SysID framework. In this paper, it showed that SysID performed significantly better on UNSW's (University of Southern Wales) dataset compared to the dataset of IoT Sentinel. They mention that this is caused by a lack of diversity within the IoT SENTINEL dataset [2]. Hence, my goal is to further take advantage of the diversity issue when using genetic algorithms.

1.3 Goal of Research

Therefore, the goal of this paper is to create a mechanism for manipulating the quality of the GA's output through means of mutating the input data. It is important to note the difference between the original input data (OID) and the mutated input data (MID) for multiple reasons. Without taking into account any form of similarity between OID and MID, the task becomes oversimplified, resulting in a completely new dataset that does not represent the packets that were sent over the network. Therefore, there must be some threshold behind measuring the similarity between OID and MID. The threshold and metrics used for measuring this similarity are outlined in Section 2.

In addition to making sure that MID represents OID properly, there requires a strategy to mutate the data in such a way that it lowers the classification score of the system identifier. One common flaw that is found in GAs is the appearance of *local* optimum points rather than *global* optimal points [6]. This flaw can be exploited to lower the quality of features selected from the GA. Following this logic, how can the data be mutated in such a way to result in local optimum values? Chowdhury and Aneja found that there is an inverse correlation between data diversity and the effectiveness of the GA's output in classification [2]. In this paper, I propose a mechanism that mutates the OID such that the MID is both representative of the OID and diverse enough to weaken the output of the GA to lower the classification score of device identification, which highlights the weakness of device identification using genetic algorithms.

Section presents the detailed framework that will be tested where Section provides notations and device information along with preliminary metrics used for evaluation. The proposed mutation mechanism is illustrated in Section , while the procedure and experimental results are discussed in Section and Section , respectively. Section then presents the conclusion, which is followed by Section that discusses related work and future directions of this work.

2 Notation and Metrics

3 Framework for SysID

4 Framework for Mutation Module

5 Experimental Procedure

6 Experimental Results

7 Conclusion

8 Related and Future Works

References

[1] Ahmet Aksoy and Mehmet Hadi Gunes. "Automated iot device identification using network traffic". In: *ICC 2019-2019 IEEE International Conference on Communications (ICC)*. IEEE. 2019, pp. 1–7.

[2] Rajarshi Roy Chowdhury et al. "Network Traffic Analysis based IoT Device Identification". In: *Proceedings of the 2020 the 4th International Conference on Big Data and Internet of Things*. 2020, pp. 79–89.

[3] Shuodi Hui et al. "Systematically Quantifying IoT Privacy Leakage in Mobile Networks". In: *IEEE Internet of Things Journal* 8.9 (2020), pp. 7115–7125.

[4] Yair Meidan et al. "ProfilIoT: A machine learning approach for IoT device identification based on network traffic analysis". In: *Proceedings of the symposium on applied computing*. 2017, pp. 506–509.

[5] IoT SENTINEL. "Automated Device-Type Identification for Security Enforcement in IoT,"" in: *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA*. 2017.

[6] Narges Yousefnezhad, Avleen Malhi, and Kary Främling. "Automated IoT device identification based on full packet information using real-time network traffic". In: *Sensors* 21.8 (2021), p. 2660.