Evaluating compliance of the actual behaviour of IoT devices with their Privacy Policy Agreement

A thesis submitted in partial fulfilment

of the requirement for the degree of Doctor of Philosophy

Alanoud T. Subahi

September 2020

Cardiff University
School of Computer Science & Informatics

Declaration

This work has not previously been accepted in substance for any degree and is not concurrently submitted in candidature for any degree.

Signed ..... (candidate)

Date .....

Statement 1

This thesis is being submitted in partial fulfillment of the requirements for the degree of PhD.

Signed ..... (candidate)

Date .....

Statement 2

This thesis is the result of my own independent work/investigation, except where otherwise stated. Other sources are acknowledged by explicit references.

Signed ..... (candidate)

Date .....

Statement 3

I hereby give consent for my thesis, if accepted, to be available for photocopying and for inter-library loan, and for the title and summary to be made available to outside organisations.

Signed ..... (candidate)

Date .....

To my soulmate and the source of my success, my beloved Suhail.
To my children, the source of my happiness and delight.
To my beloved father, mother, and my siblings, I love you all and I hope you are proud of me.

Abstract

In the past few years, Internet of Things (IoT) devices have emerged and spread everywhere. IoT has the potential to make people's lives more comfortable and more efficient. Many people use smart home devices, and such devices can communicate with each other without user intervention. To control, configure, and interface with the IoT device, a companion mobile application comes with each IoT device, which needs to be installed on the user's smartphone or tablet.

IoT devices send information in three different ways. The first way is from the IoT Device to the Cloud (D-C). Through this way, the device can send the user's data to the IoT device's cloud. The second way is from the IoT app to the IoT Device (A-D). In this way, the IoT app sends a command(s) to the IoT device to work based on a specific command. The third way is from the IoT app to the IoT Cloud (A-C). Through this way, the device can also send user's data to the IoT device's cloud. Despite the importance of the privacy risk, the majority of IoT users don't understand what kind of information is being collected about them or their environment. Privacy is not only limited to encryption and access authorization, but also related to the kind of transmitted information, how it's being used, and with whom it will be shared. Accordingly, many researchers have been motivated to study the security and privacy issues of those devices due to the sensitive information they carry about their owners. Thus The limitation of existing methods are:

1. They only study the security and privacy issues by analyzing the traffic that goes

directly from the IoT device to the IoT cloud (i.e. D-C).

2. They never study the privacy violations between the IoT traffic with its PPA, i.e., compliance violations.

In contrast, this research aims to study the privacy violations through analyzing the alternate path, i.e. (A-C). In particular, we consider the compliance issues between the data sent from the IoT mobile app to the IoT cloud and what the manufacturer of this IoT device states about the data that they collect about its users. IoT manufacturers are compelled to issue Privacy Policy Agreements (PPA) for their respective devices as well as ensure that the actual behavior of such devices complies with the issued PPA. To evaluate this compliance, we make the following contributions:

The first contribution is investigating issues around IoT privacy in general and the compliance violations between the IoT devices with their PPA. To do so, we need to implement two stages. The first stage is to read and study, manually, the PPA of eleven IoT manufacturers. The results reveal that half of those IoT manufacturers do not have an adequate privacy policy specifically for their IoT devices. Consequently, we create eight main criteria, based on the GDPR, that any IoT manufacturer should implement when designing its PPA. Also, we argue that the IoT manufacturer should apply these criteria as well as adhere to them when they issue their new IoT products. While the second stage is to design a testbed to capture the traffic of two IoT devices (i.e., Tp-link smart plug and Belkin NetCam). Then, we analyze the collected traffic to find out the type of data transferred from the devices to their manufacturer's cloud. Finally, we evaluate the compliance of the actual behavior of the IoT devices (Tp-link smart plug and Belkin NetCam) with their PPAs as well as with our eight criteria. The results prove that the data sent from the two IoT devices to their clouds does not comply with what they stated in their PPA.

The second contribution is a tool that automatically infers the actual behavior (i.e. the type of the transmitted data) of an IoT device from its encrypted network traffic. In particular, the tool infers three critical things; first of all, the tool reveals from the traffic

the interaction type between the user and his/her IoT device through the IoT device's app (e.g., the user login to the IoT app to control the device). Second, it reveals whether the IoT device sends sensitive Personal Identifiable Information (PII) about the user to its cloud. Finally, the tool reveals the content type of such sensitive information (e.g., user's location detail). This information helps IoT users to make rational decisions regarding their privacy risks. We implement this tool using supervised machine learning algorithm, we obtain the following classification accuracy values of inferring the three types of information, as mentioned above, respectively: 99.4%, 99.8%, and 99.8%. This high accuracy proves the reliability of our proposed method.

The third contribution is a method to analyze the text of IoT PPAs. In this method, we aim to imitate the way that an ordinary person, with an average education level, reads and understands such long policies. To do so, we implement a text-mining tool to read and extract specific type of information using a supervised machine learning algorithm. Our goal is to determine the types of personal information that the PPA mentions are collected about the IoT device users. Furthermore, we categorize such information according to its sensitivity level to either sensitive personal information or non-sensitive personal information. Using our tool, we analyze and label 31,661 sentences from 50 IoT PPAs. The high accuracy achieved by the classifier (i.e. 98.8%) proves the validity and reliability of our proposed method.

Finally, we combine the second and the third contributions to investigate whether there is a mismatch between the actual data sent to the IoT manufacturer cloud with what the manufacturer states in its PPA.

The experimental results demonstrated in this thesis confirm our hypothesis that most IoT manufacturers don't provide sufficient information in their PPA or they don't comply with what they state in their PPA.

Acknowledgements

First of all, thanks and praise be to Allah, the Most Compassionate, and the Most Merciful. I express my great appreciation to Allah who gave me the power to fulfill and deliver this dissertation.

I can't believe that my journey has finally come to the end. Having a PhD has been one of the most challenging and rewarding experiences in my life. I went through different moments of disappointment and failure, often thinking of giving up. I also lost some people I love without even saying goodbye. However, there was always a glimmer of hope that penetrated inside me to make me feel strong and confident and pushing me to continue this journey despite all these difficulties. This sparkle of hope came from my husband and my children, my parents, my siblings, and my best friends. How much I complained to them and how much they motivated me for patience and success.

For his insightful guidance and great support, I would like to express my big gratitude to Dr. George Theodorakopoulos, the main supervisor of this dissertation. Despite his many duties, he was very generous in spending his time with me. I really appreciate his enthusiastic supervision, enlightening inspiration, continuing encouragement, and invaluable technical suggestions. I am much blessed at being under the supervision of such a person.

I want to thank all the members of the School of Computer Science and Informatics at Cardiff University for their kind assistance. Also, thanks to all my friends and colleagues in Cardiff city who have been positive and supportive through my PhD journey.

I want to extend my thanks to my beloved country Saudi Arabia, for supporting me financially and allows me to pursue my dream of studying the PhD. I am also thankful to my friends and colleagues in the Department of Computer Science at King Abdul-Aziz University in general, and those in Rabigh branch in particular.

Finally, I am proud of myself and proud of my accomplishments. I admitted that I made lots of mistakes, but in turn, I learned a lot from these mistakes to reach where I am now.

Contents

List of Publications

The work introduced in this thesis is based on the following publications.

- Alanoud Subahi and George Theodorakopoulos. Ensuring compliance of IoT devices with their privacy policy agreement. In 2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud), pages 100-107. IEEE, 2018 [94].
- Alanoud Subahi and George Theodorakopoulos. Detecting IoT user behavior and sensitive information in encrypted IoT-app traffic. Sensors, 19(21):4777, 2019 [95].
- Alanoud Subahi and George Theodorakopoulos. Automated Approach to Analyze IoT Privacy Policies, under review.

List of Figures

List of Tables

List of Acronyms

IoT Internet of Thing

ICO Information Commissioner Office

GDPR General Data Protection Regulation

PII Personal Identifiable Information

sensitive PII sensitive Personal Identifiable Information

non-sensitive PII non-sensitive Personal Identifiable Information

IoT-app PIT IoT-app Privacy Inspector Tool

PPA Privacy Policy Agreement

D-C IoT device to the IoT cloud

A-D IoT mobile application to the IoT device

A-C IoT mobile application to the IoT cloud

ML Machine Learning

TP True Positive

TN True Negative

FP False Positive

FN False Negative

c-K collect keyword

neg-K negative keyword

s-K sensitive keyword

wc-K wrong collect keyword

thirdParty-K third-party keyword

share-K share keywords

cookie-K cookie keyword

Chapter 1

Introduction

1.1 Background

The Internet of Things (IoT) refers to the tens of billions of low-cost devices that communicate with each other and with remote servers on the Internet autonomously. It comprises everyday objects such as lights, cameras, motion sensors, door locks, thermostats, power switches, and household appliances, which facilitate our daily lives in almost every aspect [25, 96, 106, 113].

IoT technology has become one of the fastest developing and growing technologies today due to its ability to provide a new platform for services and decision making. In November 2019, Statista Research [93] projected the number of connected IoT devices to be 75.44 billion worldwide by 2025. According to McKinsey Global Institute, the financial impact of the IoT market on the global economy may reach as much as $11.1 trillion by 2025 [54, 55].

It is important to emphasize that most of these smart devices are manufactured for personal use; therefore, they deal with a user's Personal Identifiable Information (PII) [45, 76] all the time. IoT devices can monitor, collect, and store a massive amount of sensitive data and information about their users [43]. The popularity of wearable tech is one trend that is currently supporting much more extensive data capturing processes. For example, many users wear smartwatches most of the time, and thus their personal information, habits, and behavior are collected and sent to the smartwatch manufac-

turer's cloud [84]. However, this proliferation creates essential security and privacy problems. When such sensitive personal data is released to third parties, the possibility of an unintentional or malicious privacy breach, such as detection of user activity, is very high [43]. Thus, IoT users need to know what kind of personal information will be collected by the IoT device and why.

In the Information Commissioner Office (ICO) report [8], the General Data Protection Regulation (GDPR) sets the criteria for manufacturers' data collection processes. The report emphasizes that companies are required to protect the privacy of their EU customers by keeping their PII secure. Companies whose business practices are found to be inconsistent with their privacy policies will face regulatory enforcement actions [2]. PII can be categorized to either sensitive PII (e.g. login information) or non-sensitive PII (e.g. email address); see chapter 2.2.4 for more details. Hence, as PII can be sensitive, it is essential to notify IoT users with respect to their personal data and help them make rational decisions about their privacy risks.

IoT manufacturers need to clearly specify in their Privacy Policy Agreement (PPA) what data type they collect from the users of their IoT products. In fact, it is essential for IoT manufacturers not only to have a sufficient PPA for their respective devices but also to comply with what they state in this PPA. It should be noted that privacy is not only about access authorization and encryption; rather, it also emphasizes on the type of transmitted information and on how it will be used and shared by the legitimate recipient (e.g., IoT manufacturer) [37].

To the best of our knowledge, most academic research focuses only on:

- Analyzing IoT devices' security and privacy issues,
- Discover IoT attacks and violations,
- Perform different attacks targeting various types of IoT devices; related to user data disclosure.

In contrast, we are the first who highlight the importance of enforcing IoT manufacturers to issue a sufficient PPA as well as monitoring the behavior of such devices to ensure their compliance to the PPA.

1.2 Main Problem

In this thesis, when we talk about the IoT compliance issues, we mean the mismatch between the actual behavior of the IoT device and what the PPA of this particular IoT device states. It is difficult to ensure privacy in IoT devices because they are capable of transmitting substantial amounts of data, including the user's personal information and his life pattern.

IoT devices talk to each other as well as to their manufacturer's cloud. Consequently, they transmit sensitive information about their users most of the time, resulting in potential security and privacy issues, see Figure 1.1.

IoT users have no control over their IoT devices' communication. On the other hand, the IoT manufacturer's use of obtained data can go beyond the reasons for which it was initially collected, or may exceed what is permissible within the PPA. Therefore, risks can be identified regarding the collected data's security and privacy.

Despite the importance of the privacy risk, most IoT users are not completely conscious of the kind of information being collected about them or their environment, even being uncertain of whether the information is being shared with others. Instead, they choose convenience over privacy as sharing their data is not a big deal for them. However, lots of people do have a concern about their data privacy [82, 69].

According to our published work in [94], IoT devices are often not compliant with their PPA requirements. Also, an examination of 121 shopping apps revealed that many PPA are vague and fail to convey how apps handle consumers' data [74]. Consequently, any conflict can have real consequences as they may lead to enforcement actions by the

Figure 1.1: Overview of the IoT data privacy problem

GDPR and other regulators. In fact, without the presence of effective mechanisms to ensure the compliance between the IoT device and its privacy policy, managing data flow can be particularly problematic.

1.3 Motivation

Compared to computer or smartphone traffic, the characteristics and features of IoT traffic are very different, as explained in detail in chapter 2.2.3.

Previously, Internet user activity was mostly user-initiated web browsing. Nevertheless, this phenomenon has changed with the emerge of IoT devices. The contents, patterns, and metadata of IoT network traffic can all reveal sensitive information about a user's physical activities. In parallel, IoT manufacturers run cloud and other services externally to a domestic IoT network. They capture and store data that is conveyed to them by IoT sensors that are constantly on and monitoring even clients' offline actions; this data capture differs from traditional internet browsers, as it happens surreptitiously.

As with website publishers, IoT manufacturers are accountable for issuing PPA to de-

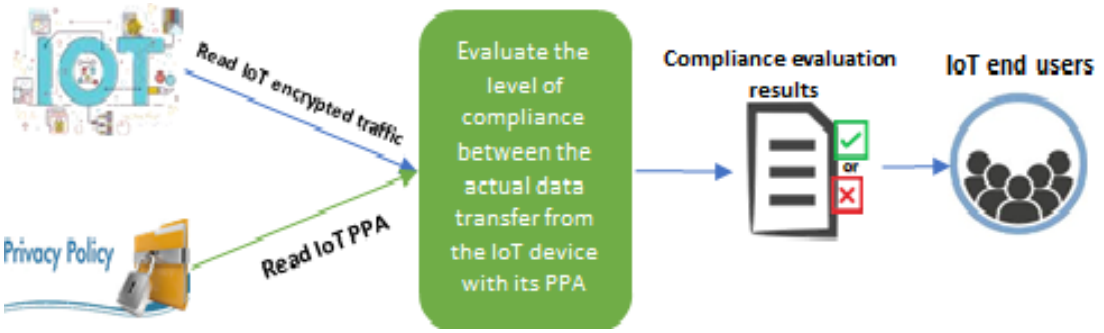

Figure 1.2: General overview of our suggested tool to evaluate the compliance of the IoT device with its PPA and present the results to the IoT end users.

tail the type of such collected information when the user interacts with their respective devices. Despite the importance of reading such PPAs before using any IoT device, many users still ignore them because they are too long and complex.

Based on the above observations, we believe that IoT end users need a tool that automatically detects and describes their data collection practices. Such a tool can evaluate the level of compliance of the actual data flows from the IoT device with the data type that its PPA collect. Figure 1.2 illustrates the general overview of our suggested tool, that we aim to implement through this PhD research. We are sure that this tool will be useful not only IoT end users but also IoT manufacturers for the following reasons:

1. Researchers found that privacy violations often appear to be based on developers' difficulties in understanding privacy requirements rather than on malicious intentions [27]. Therefore, such a tool will help them ensure compliance between their IoT devices and the legal PPA before they launch the new IoT devices in the market.

2. Using this tool will help preserve the privacy of IoT users. IoT end users will be aware of whether their privacy has been violated or not as well as give them a free choice to decide what IoT device they want to use.

1.4 Hypothesis and Research Questions

Our main hypothesis in this thesis is: *IoT devices send sensitive PII about their users to their manufacturer's cloud. In turn, most IoT manufacturers either don't provide sufficient information with regards to the type of such information on their PPA or don't comply with what they state in their PPA with regards to the actual behavior of the IoT device.*

Due to this conflict, we want to automatically evaluate the level of compliance between the actual behavior of the IoT device with its PPA presented in the IoT manufacturer website. In order to verify this hypothesis, we address the following set of research questions:

Research Question 1: *Is the data sent from the IoT device limited to an identified purpose of their PPA? If so, do the IoT end users know what type of information is being sent about them?*

Research Question 2: *Can the encrypted traffic of the IoT device expose sensitive PII about their end users? If so, can we know the type of such information sent from the IoT device to its cloud?*

Research Question 3: *Can an automated text mining mechanism help IoT end users avoid reading long and complicated IoT PPA text to know whether such PPA collects sensitive PII about them, and knowing the type of such information?*

Research Question 4: *Can we automatically inform the IoT end users whether the data sent from an IoT device complies with its PPA?*

1.5 Contributions

The main contribution of this thesis, which has not been in the focus of IoT research yet, is the development of a method for evaluating the compliance of the actual beha-

vior of the IoT device with its privacy policy, see Figure 1.2. To achieve this contribution, we made several contributions during this PhD research as follows:

1. We provide a theoretical overview of issues around IoT PPAs and argue that there is an urgent need to update the privacy law of the IoT devices. Moreover, we focus on the language used within PPA by merging and analyzing the existing privacy principles. As a result, we establish eight privacy criteria based on the GDPR. We argue that any IoT manufacturer should adhere to those criteria when they issue their PPA for their IoT products. The main objective of this work is to find out whether the data transferred from the IoT app to the IoT cloud (A-C) comply with what stated in its PPA. To do so, we design and implement a practical testbed to evaluate, manually, the compliance of the actual behavior of two different IoT devices (i.e. Tp-link smart plug and Belkin NetCam) with their PPAs as well as with our eight criteria. The results prove that the two IoT devices don't fully comply with what they state in their PPA, nor they comply with the eight criteria. This work was published in [94].

2. We show how passive packet-level analysis can be done to infer the behavior of an IoT device through its encrypted network traffic emit from its apps (i.e. A-C). Furthermore, we show how an attacker can violate user's privacy through monitoring such traffic. Consequently, we develop a novel method that automatically analyze the collected encrypted traffic from IoT app in order to infer critical information regarding user's data privacy. Theses information are the following:

   - Whether the traffic of an IoT device sends sensitive information about the end user to its respective cloud,
   - The type of such sensitive information,
   - The type of user's interaction(s) with the IoT device.

   We named this innovative tool IoT-app privacy inspector tool (IoT-app PIT), which combines three different multi-class classifiers. Each classifier used to

infer one type of the information above. The objective of such a tool is to involve IoT end user to take an active role in protecting their privacy. This work was published in [95].

3. We propose a novel method for analyzing the PPA text of any IoT manufacturer. This method aims to extract only the type(s) of personal information collected by such a manufacturer. Then, it classifies the collected data to either sensitive PII or non-sensitive PII. Finally, it presents such data to the IoT end-user. In our method, we don't ask the users to read the whole PPA text, nor we shorten the length of the text nor we highlight the paragraphs that refer to the data collection practices, then ask the users to read such paragraphs. In contrast, our innovative method focuses only on informing the IoT end users about the types of their collected PII. The objective of this method is to help end users make rational decisions before using any IoT device based on a prior understanding of the type of collected data from such device(s). To do so, we develop a text mining tool, called IoT-PPA reading tool, that automatically reads and analyzes long and complicated IoT PPA text. As a results, it only informs the end user about the data collection types as well as the category of such data types i.e. sensitive PII or non-sensitive PII.

4. We propose a new method to compare the outputs of the two main tools to evaluate the level of compliance between the actual behavior of the IoT device with its PPA. Each of these tools has its own input and output data types. The first tool is the IoT-app PIT, which automatically detects the encrypted traffic of the IoT device. In comparison, the second one is the IoT-PPA reading tool that reads the IoT PPA text of this specific IoT device. To evaluate the compliance of an IoT device, the user must run IoT Behavior Compliance Tool, then select the IoT device that he wants to evaluate. This tool will execute and produce the results from the two tools mentioned above. After that, it will analyze and compare the results to present to the end user the compliance level of this particular IoT

device. This holds especially true for the analysis of the IoT devices and what data they send to their cloud.

1.6 Thesis Structure

The rest of the thesis is organized as follows:

- Chapter 2- Background and Related Work- provides general background regarding the IoT. In particular, the chapter introduces the concept of IoT network technologies, IoT network traffic, as well as defines the fundamental terminologies used throughout this thesis. Also, the chapter reviews the related work in the area of IoT privacy testbeds, monitoring the IoT traffic, and issues around the PPA in general, and the ones related to IoT PPA in specific.

- Chapter 3- Data Collection Methodology- introduces the datasets used in this work, covering both IoT traffic data and IoT PPA data. First, it describes the testbeds and the controlled experiments used to obtain ground-truth information about the IoT network traffic generated by IoT-device and its IoT-apps. This data used to identify sensitive PII in IoT network flows. Then, the chapter describes the methodology used for collecting and analyzing the IoT PPAs to extract the type of collected PII from such PPA.

- Chapter 4- Ensuring compliance of IoT devices with their Privacy Policy Agreement- first demonstrates the issues around IoT PPA by focusing on the language used within such policies. This chapter also introduces eight data privacy criteria that must be applied by any IoT manufacturers as well as comply with these criteria. Second, the chapter explain a practical testbed that carried out with the aim of proving whether there is a compliance issue between the actual behavior of the IoT device and its PPA. Surprisingly, the results of this experiment show that there is a compliance issue, that need to be addressed.

- Chapter 5- Detecting IoT User Behavior and Sensitive Information in Encrypted IoT-App Traffic- presents how a passive network observer can infer the interaction type between the IoT end user and the the IoT device through analyzing the encrypted traffic of its apps. In addition he can infer whether the IoT app sends sensitive PII to the IoT manufacturer cloud as well as the exact type of such PII data. A novel tool called the IoT-app PIT has been developed in this chapter to inform the IoT user about the previous information in order to help him preserve his privacy information.

- Chapter 6- Automated Approach to Analyze IoT Privacy Policies- presents the proposed method for analyzing the IoT PPA by developing a text mining tool called IoT-PPA reading. The tool focuses on reading long and complicated texts, then present to the end users the types of PII that the IoT manufacturer's PPA collects about them.

- Chapter 7- IoT Behavior Compliance- discusses a novel tool that combines and executes two different tools, each of which serves different purposes; then, it compares the results of these tools. Based on this comparison, the tool will evaluate the level of compliance between the actual behavior of the IoT device with its privacy policy agreement and presents the final results to the IoT user.

- Chapter 8- Conclusion and Future Work- concludes the thesis by summarizing our contributions, findings, as well as highlighting proposals for future work.

1.7 Summary

To sum up, this chapter introduces the background, the main problem, and the motivation behind evaluating the compliance between the IoT device and its PPA. Also, we discuss the hypothesis and the main research questions of this thesis. Finally, we highlight the thesis contributions and structure of the current thesis.

Before we explain our main contributions to this thesis, first we need, in the next chapter, to provide a more detailed background and highlight previous studies to put the thesis in the context of existing work.

Chapter 2

Background and Literature Review

2.1 Introduction

In order to better clarify the innovative contribution of this thesis, this chapter discusses related works regarding IoT in depth. In particular, we discuss the existing literature about security and privacy i.e. IoT testbed for privacy violations and monitor and analyze the IoT traffic. Also, the chapter provides a general background or several key concepts used in this thesis. First, Section 2.2 of this chapter presents the required background knowledge about the concept of the IoT; the network technologies used by the IoT; the main differences between the IoT network traffic and the non-IoT network traffic; what do we mean by the term Personal Identifiable Information as well as how we use it in this thesis; finally, we explain the meaning of PPA and data privacy. Section 2.3 discusses the IoT literature relevant to this thesis. Literature focusing on IoT security and privacy testbeds, as well as different attacks and vulnerabilities targeting various types of IoT devices are disussed in Section 2.3.1. This literature serves chapter 4 which ensure the compliance of IoT devices with their PPA. While Section 2.3.2 and 2.3.3 discuss tow closely related researches which serve chapter 5, which is: privacy research that monitors IoT network traffic to infer sensitive information contained in the traffic and research that monitors and classifies IoT traffic. Section 2.3.4 presents a summary of various research that has been proposed to solve some of the PPA issues such as: evaluating the readability of PPA documents and assessing the language used, evaluating the content, and transparency of PPAs. Also, a

particular discussion recording the IoT PPA of systems and devices analysis has been provided in the same section. This literature is related to chapter 6. To serve chapter 7, Section 2.3.5 presents the current researches that address different compliance issues. Finally, Section 2.4 summarizes the main topics discussed in this chapter.

2.2 Background

2.2.1 The concept of Internet of Things (IoT)

The concept of the IoT was first proposed in 1999 by Kevin Ashton, noting that the IoT is interoperable, uniquely identifiable things with Radio Frequency Identification (RFID) technology [24]. As Figure 2-1 shows, the IoT is a multi-domain (physical and digital) environment. It is made up of multiple services and devices, which are linked up and used to gather to exchange data. IoT devices are connected to the Internet so that they make the shift from functionality to connectivity and data-driven decision making, meaning that a device can produce and share information to become more useful. However, the IoT is not just a collection of devices and sensors connected in a wired or wireless network; it is an intense condensation of virtual and the real world, where people and devices communicate. It can be considered an interlocking medium of networks of different sizes [50], which form a large global network. The diversity of IoT application domains is wide, including smart cities, smart homes, logistics and transportation, environmental monitoring, smart enterprise environments [101].

In this thesis, we mainly focus on addressing smart home devices. When talking about smart homes, people may ask what makes a smart home different from the traditional home? Four main characteristics differentiate a smart home from a conventional home, as mentioned in detail by Edwards andGrinter [41]. A smart home environment uses sensor data to evaluate the current state and make a decision (for example, if someone walks around, the motion sensor picks up this movement, and therefore a decision is

made to open the light in the room). The recent rapid development of the IoT and its ability to offer a new platform for services and decision-making have made it one of the fastest-growing technologies today. This new disruptive paradigm of a pervasive, physically connected world has a significant impact on social interactions, business, and industrial activities [43].

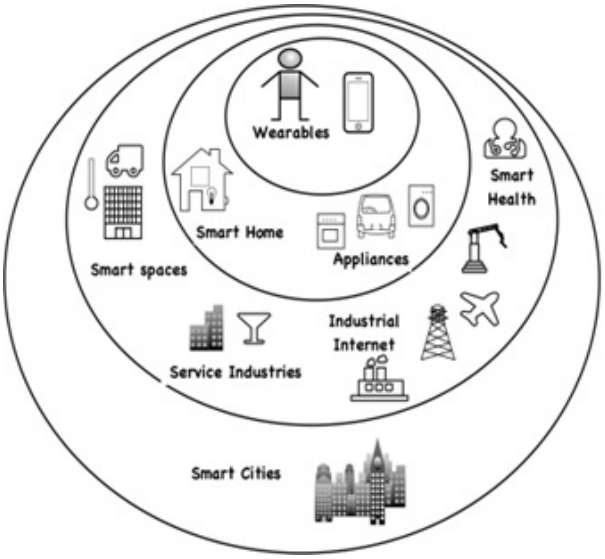

Figure 2.1: The deployment map of IoT

[92]

2.2.2 IoT Network Technologies

This section briefly describes the network technologies of IoT devices to give the readers a better understanding of how do the IoT devices communicate and exchange data with each other.

In the context of a smart home environment, IoT devices must be interconnected to exchange information. The ways in which these devices and sensors communicate are determined by communication protocols, which classified into three groups based on the propagation medium: 1) wired, 2) wireless, 3) hybrid. Noting that choosing the right technology for use depends on the use case and the size of the network [64].

When we look at the smart home networks,in particular, we find that IoT application domains use wireless communication protocols to connect them to the Internet, due to the ease of use and lower costs of setting up the network and installing new devices [63]. The most commonly used wireless protocols used in smart homes are Wi-Fi, Bluetooth, Z-wave, Zigbee, and 6LowPAN [63], [64], [99]. Due to the heterogeneity of IoT devices present in a smart home, the problem of interoperability between devices using different communication protocols arises.

In this thesis, we only study IoT devices that use Wi-Fi protocol to connect to the Internet.

2.2.3 Differences between IoT traffic and non-IoT traffic

This section highlights the different characteristics of the network traffic of IoT devices and non-IoT devices. Also, we briefly mention the previous works that been done to discover such differences.

Before we analyze the behavior of any IoT device, it is important to understand the nature of their network traffic, why it's different than non-IoT traffic, and what are the key attributes that distinguish IoT traffic from the non-IoT traffic. According to [87], there are many reasons why it is important to classify IoT traffic from other traffic. First of all, from a security perspective, the most important reason for distinguishing IoT traffic is to detect and mitigate cyber-security attacks. For example, knowing that a particular IoT device from a specific manufacturer is connecting to the network (e.g. security camera) can help the network administrator to apply specific security rules i.e. limit the camera only to do specific behavior [81]. Secondly, a network administrator will be able to control unnecessary multicast/broadcast traffic as well as limit their impact on other applications. Finally, the network administrators of smart cities and enterprises will be able to define their networks to measure appropriate levels of performance in terms of reliability, loss, and access time needed for environmental,

health, or safety applications. However, the process of classifying IoT traffic (e.g. smart bulbs, smart camera) from non-IoT traffic (e.g. computers, mobile phones, tablets) within a specific LAN network consider a big challenge. Due to the heterogeneity of IoT devices, researchers motivated to propose network-level security mechanisms that analyze traffic patterns to identify attacks [112], [90]. It should be noted that the success of these approaches relied on understanding the nature of IoT traffic.

IoT devices have been manufactured to perform specific tasks, unlike non-IoT devices such as laptops or smartphones. For example, a smart plug or smart lamp can be turned on or off, or even the brightness level of a smart lamp can be adjusted. In fact, we don't expect IoT devices to perform like laptops i.e. browsing YouTube or send emails to others. Due to such limited functions of an IoT device, it generates a stable pattern of network traffic, which makes it predictable and easy to distinguish from the network traffic of the non-IoT device.

Meidan et al. [60], [59] emphasize that locating and detecting IoT traffic in a network become clearly evident. They prove with high accuracy (99.281%) that by analyzing network traffic, one can differentiate between IPs that belong to IoT devices, PCs, and smartphones based on their single session.

Sivanathan et al. [88] have demonstrated that there are eight critical attributes based on the basic characteristics of network tracking, by which IoT devices' behavior can be distinguished from the non-IoT device. These attributes are flow volume, flow duration, average flow rate, device sleep time, server port numbers, DNS queries, NTP queries, and cipher suites. However, some IoT devices can be distinguished by considering one or two attributes, such as DNS, port numbers, or cipher suite [88], [89].

Also, the work done by [44] found that unlike non-IoT devices, IoT devices have small buffer size for TCP stack and therefore commonly has a smaller TCP window size. Once a device has been identified, techniques such as the one presented in [13] can be used to further determine the current state of the IoT device.

In our research, in terms of a smart home, we leverage how to distinguish IoT traffic from non-IoT traffic and identify its type to study their behavior in more depth. Such information can help to infer the life pattern of a specific house as well as infer the user data that the IoT device sends to its manufacturer. Consequently, knowing this type of information by an illegitimate person (i.e. hacker) raises different concerns related to violating the user's data privacy.

2.2.4 Personal Identifiable Information

In this section, we aim to define the term of Personal Identifiable Information (PII), and how we employ it through this thesis.

PII is a generic term referring to "information which can be used to distinguish or trace an individual's identity" [34]. While the GDPR defined Personal Data as follows: "Article 4(1): 'personal data' means any information relating to an identified or identifiable natural person ('data subject'); an identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person" [45], [76].

The meaning of PII and Personal Data is similar. However, the difference between PII and Personal Data is mostly a difference between US and EU legal definitions. Sweeney [98] writes that "Personal Data is considered to be the European equivalent of PII."

In this thesis, we use the term PII, not Personal Data, to refer to the user personal data. According to [98], [49], we classify the type of user PII as follows:

1. sensitive PII- which comprises information related to the user that is not for public use or may violate the individual privacy and security by being made

publicly available, e.g., Credentials (username, password), telephone number, date of birth, and Location (GPS latitude and longitude, zip code).

2. non-sensitive PII- which is information that can identify the user but will not affect his privacy or security, such as email address, first name, nickname, social media profile, website, device, or OS installation.

2.2.5 Privacy Policy Agreement and Data privacy

This section describes the meaning and the objective of using a Privacy Policy Agreement (PPA) and why IoT manufacturers need to issue and adhere to their PPA.

PPA aim to answer questions related to the users' privacy simply and understandably, such as: what information is collected by the manufacturer? Who collects such information? How is the information collected, used, and protected? Who can access my information, and what information is being shared and with whom? Thus, the importance of having a PPA is to safeguard individual privacy. While most privacy policies look similar, the details vary depending on the scenario and why they seek such information. It should be noted that the definition of 'Personal Identifiable Information' can also change, depending on who is collecting it. Nevertheless, they all include general information like names, email addresses, and details like IP addresses and browsing history.

The Internet Security Glossary has described Data Privacy as "the right of an entity (normally a person), acting in its own behalf, to determine the degree to which it interacts with its environment, including the degree to which the entity is willing to share information about itself with others" [83]. Recently, the majority of governments do treat data privacy as an essential human right [4]. Most have created laws designed to protect citizens and prevent manufacturers from taking their information without consent. It is now the norm for businesses to be obligated to state precisely why they want the information and why they plan to do with it [8]. Privacy is not only about access

authorization and encryption; rather, it also emphasizes on the type of transmitted information [37], and on how it will be used and shared by the legitimate recipient (e.g. IoT manufacturer).

Based on the EU Commission report on the IoT, The Cluster of European Research Projects on the IoT [96], privacy and security continue to be the biggest challenge for IoT research that contains privacy-preserving technology for heterogeneous device sets. They suggest that a lot more research is needed to uncover better forms of security and find more effective ways to protect the privacy of user's data. The FTC (Federal Trade Commission) [2], which advises businesses and manufacturers as to their responsibilities regarding data privacy in US, provides information on the best security techniques and works closely with manufacturers to try and create stronger, safer devices. The head of the FTC, Edith Ramirez, is keen to point out that, if businesses don't take the right steps, their relationships with consumers could be damaged, he said "The only way for the Internet of things to reach its full potential for innovation is with the trust of American Consumers. We believe that by adopting the best practices we've laid out, businesses will be better able to provide consumers the protections they want and allow the benefits of the Internet of things to be fully realized".

2.3 Literature Review

2.3.1 IoT Testbeds for security and privacy violations

In this section, we examine the available IoT literature focusing on IoT security and privacy testbeds as well as different experiments of attacks and vulnerabilities targeting various types of IoT devices that are related to user data disclosure, as shown below.

Most of the security and privacy research regarding IoT devices has focused on security issues [40]. Other security research monitors IoT traffic to detect intrusion attempts [19] or has discovered various IoT vulnerabilities [107].

Secu Wear [46] is a testbed designed for wearable IoT devices proposed by Hale et al. this testbed aimed to assess both software and hardware vulnerabilities. Its platform consisted of several open-sourced technologies such as Django, MetaWear, Ubertooth One, and Apache Cordova. The shortcomings of this testbed were; first, it only tests the security of BLE (Bluetooth low energy). Secondly, it tested security using basic intrusion attacks only.

Another state-of-the-art testbed targeted wearable IoT device was proposed by Siboni et al. [84]. Its main goal is to apply a set of security requirements against wearable IoT devices in order to test their security level. Also, it tested the behavior of those wearable IoT devices under several conditions, for example, when different applications are running. While the testbed of this thesis aims to examine the behavior of the IoT device in order to detect the type of personal information being sent from IoT devices.

Other researches have been carried out, which were intended to discover different vulnerabilities in smart IP cameras [100, 1, 52]. The testbed used in this thesis includes the same Netcam device used in [100]. However, such testbed aims to collect IoT traffic in order to prove the level of compliance between the actual data transferred from the Netcam and whether its PPA stated this particular data practices, which is different than the objectives of the researches mentioned above.

To conclude this section, the previous literature is limited to either unauthorized access to personal data (e.g. anticipating the users' behavioral patterns by sniffing wireless traffic exclusively) or applying different experiments of attacks and vulnerabilities targeting various types of IoT devices that are related to user data disclosure. In contrast, the first IoT testbed of this thesis (Section 4.3) proves the existence of the compliance issue between the actual behavior of IoT devices with their respective PPA, which has not been in focus in the field of IoT devices before.

2.3.2 IoT privacy concerns

In this section, we present the most relevant and state-of-the-art researches that aim to discover different privacy violations in the field of IoT devices. Motivated by privacy issues, Apthorpe, Reisman, and Feamster [22] showed how a passive network observer, e.g., an Internet Service Provider (ISP), can analyze traffic data to infer sensitive information about consumers as well as the type of connected IoT device even when the traffic is encrypted. They examine four commercially available IoT smart-home devices and find that an IoT device's particular activity and its type can be revealed through network traffic rates by anybody passively monitoring the traffic rate pattern. For example,

1. An Amazon Echo's traffic[1] can indicate when the intelligent personal assistant is being engaged with by a user.

2. Motion detection by a camera, as well as a user is observing its live images, can all be determined from a Nest Cam[2] Indoor CCTV's traffic levels.

3. Whether a Belkin WeMo switch[3] device is on or off in a smart house can be inferred from its traffic.

4. The sleep pattern of a user can be understood from a Sense sleep[4] device's traffic levels, by someone monitoring such traffic.

Our work is similar to the above in that we also study and analyze the encrypted traffic of the IoT devices, see chapter 5. However, in their research, the focus was on: (1) examining the traffic that goes directly from the IoT device to the IoT cloud (D-C), i.e. path A, see Figure 3.2, (2) studying only the traffic rate pattern to infer the type and the activity of the IoT device. In contrast, our focus is on the traffic that goes directly

[1] https://www.amazon.com/Amazon-Echo-Bluetooth-Speaker-with-WiFi-Alexa/dp/B00X4WHP5E
[2] https://nest.com/cameras/nest-cam-indoor/overview/
[3] http://www.belkin.com/us/
[4] https://sleeptrackers.io/sense/

from the IoT app that controls the IoT device to the IoT cloud (A-C), i.e. path C, see Figure 3.2. Also, we do a more in-depth analysis by examining the size and sequence of the packets, and because of this we are able to infer the user interaction with the IoT device (e.g., login to the IoT device), the user sensitive data, and finally the type of such sensitive data (e.g., password).

Siby et al. [85] developed a system called IoTScanner to analyze the IoT environment. This system can scan traffic in the Wi-Fi, Zigbee, and Bluetooth Low Energy frequencies. Furthermore, IoTScanner gives an overview of active IoT devices in a particular environment as well as the communication taking place between them. As a result, they find that it is possible to violate user privacy by classifying active Wi-Fi IoT devices, via the ratio of the send and receive traffic. In contrast, in this thesis, we prove that user privacy can be violated by monitoring the IoT traffic to determine user behavior with the IoT device via its app (e.g. login to the IoT app to control the IoT device). Also, we infer the type of such sensitive information revealing from such behavior (e.g. login credentials).

Torre et al. [102] discover a new kind of privacy risk related to personal data leakage when users share their data with third parties while using IoT applications. They define several algorithms in order to conduct inference attacks as well as offer strategies to avoid such attacks. An Adaptive Inference Discovery Service has been proposed by them, which helps users configure their permissions to share personal data and to allow them to identify any risks related to this shared information. Notice that the proposed system works as an add-on to personal data managers PDMs as a recommended system.

Wang et al. [105] present another contextual attack system called MoLe (Motion Leaks through Smartwatch Sensors) using the smartwatch device. They were able to prove that the user's sensitive and personal information has been leaked by using smartwatch devices.

The work done by [23] and [13] used the traffic levels to determine the behavior of the IoT device. They find that encryption processes would still enable packet headers and

smart home traffic levels to be used by a passive network attacker to determine local activities by profiling network traffic using machine learning algorithms.

In contrast, our research is different than the previous studies [102, 105, 23, 13] in the following:

1. The proposed system by [102] works as an add-on to personal data managers PDMs as a recommended system. While in our work, we invent an interactive tool to notify the IoT user about his sensitive PII data or non-sensitive PII data that the IoT device sends to its cloud via its app of a specific interaction.

2. The proposed system by [105] found that it is possible to recognize and identify the keyed words with reasonable accuracy. This indicates that the user's privacy could be violated by an attacker using such an attack within the context of keyboard keying. In contrast, our study proves that user interaction with the IoT device (e.g. user login to the IoT app to control the IoT device) can be also inferred with high accuracy of 99.8%.

3. Similar to [23, 13], the privacy tool invented in this research can detect with high accuracy (99.8%) the encrypted packets of a smart home traffic. However, our tool differs from them in that it only detects the packets that carry sensitive information (e.g. login credentials). It can also determine the behavior of the IoT user by observing the pattern and the sequence of the IoT traffic.

A comprehensive security test has been applied in [48, 38] on the fitness device, which is a popular tracker device. They mainly examine the Bluetooth connection between the tracker device and its paired Android smartphone device, which includes the Fitbit application. They analyze the communication between the Fitbit application and its web service. Interestingly, they find that sensitive information such as the BLE credential is sent in plaintext from the Fitbit web server to the smartphone application. This means that an attacker could obtain this information with a Man-in-the-Middle-Attack (MITM). Also, they point out that smartphones could eavesdrop on any close

Fitbit devices and send their MAC addresses to the Fitbit server; notice that these security issues will allow anyone to track other Fitbit users. While Symantec analysis of self-tracking devices investigate [29] shows that lots of self-tracking devices and its applications have security and privacy threats.

The work of Obermaier et al. [66] on cloud-based cameras found that although the device had what appeared to be a strong password (36 characters of alphanumeric and symbols), the password was the MAC address of the camera reversed and Base64 encoded.

To conclude this section, the second contribution of this thesis (section 5.3) is to emphasize the privacy risks and vulnerabilities associated with the type of personal information (e.g. user location) being transferred from the IoT device via its app to the IoT cloud. In addition, we aim to inform IoT users whether such information could reveal his activity with the IoT device (e.g. user login, or log out from the IoT app), which has not been addressed before.

2.3.3 IoT Traffic Classification

In this section, we first examine the researches that classifies IoT traffic. Second, we considered the most relevant research to ours, which is privacy research that monitors IoT network traffic to infer sensitive information contained in the traffic.

Even though there is a huge body of work characterizing general Internet traffic, research focusing on characterizing IoT traffic (also called machine-to-machine-M2M-traffic) is still in its infancy. One of the first huge-scale studies to investigate the nature of M2M traffic has been done by Shafiq et al. [80]. They want to understand whether IoT traffic imposes new challenges for cellular networks in terms of their design and management. The work done by [51] has suggested that vast quantities of IoT device information reflecting common behavior, and a sole IoT device's communication behavior can be determined through a Coupled Markov Modulated Poisson Processes

model.

Sivanathan et al. [89] characterize, classify, and analyze 21 distinct IoT devices in smart cities and campuses. Three weeks of data from traffic traces were obtained during the research, which was then put in the public domain. Subsequently, the protocols, signaling, activity trends, and other features of the traffic were statistically assessed. Ultimately, a classification method was devised with a greater than 95% precision rate for determining specific IoT devices, in addition to being able to ascertain whether the device was IoT-enabled or otherwise.

A logical IoT device classification model was developed in [53]. However, their model was limited to only classify the IoT devices into two categories, namely high vs. low energy consumption. Hence, as the authors state, it is still at a primary stage.

Furthermore, Y. Meidan et al. [59] propose a machine learning algorithm to categorize the IoT devices and the non-IoT devices based on network traffic analysis. They use features extracted from full TCP sessions (from SYN to FIN) of two smartphones, two computers, and ten IoT devices with a 99.281% precision rate in the classification.

Contrary to the other mentioned approaches, M. Miettinen et al. [62] uses a variety of features extracted during the device setup phase to develop a method for determining the type of IoT device connected to the network. They train one classifier per device type with the aim of restricting the communications of vulnerable IoT devices.

All previous research deals with IoT traffic classification with the aim of (1) classifying the IoT traffic from the non-IoT traffic, (2) classifying the IoT traffic to determine specific IoT device, (3) classifying the IoT traffic into low energy or high energy in order to understand the current behavior of the IoT device. In contrast, in this thesis, we leverage the advantage of the correlation between traffic patterns of IoT device and sensitive activities to apply machine learning in order to classify IoT traffic aiming to infer accurately:

1. Packets that reveal the interaction type between the IoT device and its corres-

ponding IoT app.

2. Packets that reveal sensitive Personal Identifiable Information (PII) about their user.

3. Packets reveal the content type of such sensitive information

2.3.4 Privacy Policy Analysis

A growing body of literature has examined the privacy policies of websites and mobile apps in different fields. Section 2.3.4.1 presents several studies that focus on evaluating the readability of PPA documents of Internet websites and mobile apps as well as assessing their language. While in section 2.3.4.2, we discuss various approaches that focus on annotating and categorizing the text of privacy policies. A few works have recently emerged, focusing on analyzing the IoT privacy policies of systems and devices, which we discuss in section 2.3.4.3.

2.3.4.1 Difficulties in Reading Privacy Policies Analysis

One strand of research [79, 28, 58, 109] examines the reasons why most users ignore the PPA, what is the best time to display privacy notices to users, and why privacy policies are full of jargon and not understandable to users.

While other research [56, 36] suggests solutions to help users not to read the full PPA but to read only the paragraphs that belong to the categories that interest them. The previous methods aim to shorten the privacy policies, so users read few paragraphs as possible. However, the problems of understanding complicated, ambiguous, and hidden information [35] have not been solved.

Another strand of research has studied the readability of PPA documents within mobile environments [86, 97].

In our approach, we aim to solve the previous problems in IoT privacy policies by only informing the users with the type of PII information that has been collected by the IoT manufacturer without asking them to read the full PPA or specific paragraphs. Also, we do our analysis automatically, avoiding problems with manual analysis.

2.3.4.2 Privacy Policy Annotation and Text Categorization Analysis

Baalous et al. [26] relied on manual testing and review to analyze the type of information collected, collection mechanisms, the purpose for collection, sharing of information, user controls, and the information period of privacy policies of cloud storage mobile applications which claim zero knowledge. However, manual testing is time-consuming despite the correct results.

The work of [47] proposed an automated framework for PPA analysis (Polisis), which automatically annotates, with high accuracy, each segment with a set of labels describing its data practices. They compared their automatic annotation with the manual annotation done by [108] to prove the accuracy of their results. Although in their approach, the users will read only a few paragraphs, the problem of the complexity and the difficulty in understanding the hidden meanings in such paragraphs still does not solve [35].

Massey et al. apply an automated text mining analysis to analyze PPA documents. They perform a large-scale analysis of 2,061 policies providing the most extensive evaluation. However, they didn't focus on their legal analysis but rather their readability and suitability for identifying privacy protections and vulnerabilities from a requirement engineering perspective [57].

[16, 72] used machine learning techniques for text categorization on privacy policies to determine whether the company has access to personal data as well as if the users can cancel, terminate, or delete their accounts. Whereas Sathyendra et al. [78, 77] aimed to detect the provision of choices in the PPA as they focused on extracting opt-out

instances.
In contrast, our research is different from the previous researches in the following:

1. We focus on the PPA text of the IoT manufacturers.
2. We propose an automated tool to read and extract the text that only collect PII about the users when they use and interact with the IoT devices.
3. We categorize the sensitivity level of the collected PII by the IoT manufacturer into sensitive-PII and non-sensitive PII according to the GDPR [49].
4. Our classifier works at the level of sentences instead of segments or word level as we analyze 31661 sentences from 50 IoT privacy policies.

Reidenberg et al. [73] propose a method to score parts of privacy policies based on their ambiguity. Hence, in their study, they develop a theory of vague and ambiguous terms that could address privacy policies' ambiguity. They used machine learning techniques to classify ambiguity in "share", "collect", "retain" and "use".
Our work is similar to the previous study in that we also study and analyze ambiguous language but in IoT privacy policies. However, their method does not take any further steps in solving these ambiguities within privacy policies. In contrast, in our research, we propose a method to solve such ambiguity. Consequently, we come up with ten different corner cases that may affect the way of understanding the correct meaning of a PPA. In each case, we apply different sets of rules to solve different types of ambiguity in order to understand the true meaning of such privacy policies, see chapter 6.3.2

2.3.4.3 IoT Privacy Policy Analysis

We find that all previous studies have focused either on; making the privacy policies of the websites and the mobile apps either:

- more readable by shortening their duration,

- determining whether personal information can be collected,
- determining whether personal information disclosed to advertisers,
- or determining whether personal information kept indefinitely.

While a few works have emerged focusing on analyzing the IoT PPA. IoT users understand that their PII is used for some purposes. For example, smartwatch users expect their data to be transferred to the company's servers to calculate their burned calories. However, they do not know the type of personal information that was transferred, nor if this information might violate their privacy[70].

Recently, the following studies are the only ones that focused on addressing issues around IoT PPAs. However, none of them solved the problem of understanding such long and complicated text as well as informing the IoT users with the PII type that the IoT manufacturers collect about them stated in their PPA.

Shayegh and Ghanavati [82] analyzed 25 IoT privacy policies and proposed a set of new annotations. They used these new annotations to manually classify IoT PPA in order to present short notices on the IoT device's screen. As a result, they generated a graph-based view and showed data practices in a better way to users. However, lots of IoT devices do not have a screen like smart switches or smart labs. Their goal was to propose a method for software designers and developers to create more effective privacy policies. In contrast, our work is different from them in that; first, we analyze twice as many as their policies (50 policies). Second, we propose a new set of annotations for:

- specifying the type of information collected, i.e., whether the IoT manufacturer PPA collects user login information,
- categorizing the collected data to either sensitive PII or non-sensitive PII according to the general data protection regulation to the GDPR [49]. For example, if

the IoT PPA states that it collects user login credentials, then we categorize it as collecting sensitive information.

The work of Perez et al. [71] is different from Shayegh et al. [82] in terms that they provide an analysis of the privacy practices instead of proposing a model for the analysis of privacy practices for six IoT devices and systems. They presented a review of issues related to privacy policies about the practices that manufacturers provide related to data collection, data ownership, data modification, data security, external data sharing, policy change, and policies for specific audiences. In contrast, our study is the only one that analyzes the largest dataset of IoT PPAs, among other IoT PPA researches. Furthermore, we propose a novel method to inform the user about the type of PII that has been collected by the IoT manufacturer's PPA as well as categorizing the sensitivity level of such information;see chapter 6 for more detail.

2.3.5 Compliance to Data Protection Regulation

The European Union affirms that all individuals who communicate online have the right to privacy and accordingly have drafted data protection legislation. The Federal Trade Commission (FTC) has sent complaints to Microsoft, Google, and Facebook because of posting privacy policies that do not conform to the company's actual practices, which led to the deception and misleading of users. In 2011 Apple Inc. has also been accused of collecting and tracking user's locations without their knowledge [12]. Apple's defense regarding this tracking was to save battery power in the device by using the location caching algorithms [67].

Recently, the research community has shown interest in the area of compliance with requirements and policy documents, which has led researchers to discuss such requirements. For example, an integrated strategy has been developed by Anton and Earp, focusing on the initial identification of security and privacy policies and their activation in system-compliant system requirements [20]. In addition, they have technologies

to identify early conflicts as well as prevent incompatible behavior, and unsafe imbalances and requirements, so that security and privacy are built instead of being added at a later stage [21].

The Theory of obligations, privileges, and rights has been used by Young et al. to determine software requirements based on the obligations organizations express in their policy documents (such as privacy notices, terms of use, etc.) [111, 110].

Allison et al aimed to seek compliance with the FIPPs by creating a PPA element model for information systems in Service-Oriented Architectures [14].

While a framework called REQMON has been developed by Robinson which monitors requirements compliance with policy documents at run-time [75].

In contrast to the above researches, the objective of this thesis is to evaluate the level of compliance between the actual behavior of the IoT device with the requirements of its PPA.

On the other hand, there are several analyses have been performed to address the compliance issues between android applications and their privacy policies. For example, Sivain et al. [91] proposed a method to check whether an Android application complies with its PPA by linking the PPA statements with Application Programming Interface (API) that produce sensitive information.

While Zimmeck et al. [114] develop a method to collect and analyze free Android apps, their research has two parts: (1) PPA analysis and (2) mobile application analysis, then evaluate the compliance between them.

Our research differs from previous studies in the following:

1. The proposed method done by [91] seeks compliance between the android application and its PPA. While in our research, we develop a tool to evaluate the compliance of the IoT device with its PPA by linking the PPA statements with the actual behavior of such a device.

2. Similar to [114], our invented tool evaluates the level of compliance by combining the results of two parts. However, we differ from them in that we analyze the encrypted traffic of the IoT device in the first part; then, we analyze the texts of the IoT PPA in the second part. Finally, we link the results from both parts to evaluate the compliance level, as described in chapter 7.

Recently, few works have emerged from analyzing and evaluating the compliance of the IoT systems and devices with their privacy policies. For example, Neisse et al. [65] have developed a security toolkit called SecKit that specifically designed to cope with the unique security issues posed by the IoT and to facilitate adherence to data protection legislation. SecKit integrates with the Message Queue Telemetry Transport (MQTT) protocol layer. MQTT is extensively used to manage information flow among devices connected via the IoT.

While Perez et al. [71] developed a testbed aimed at investigating traffic generated by two IA assistive devices, Amazon Echo Dot 2.0 and Google Home, when they actively and passively listen to sounds. They found that the two devices behaved as described in their privacy policies: the sound only recorded when the "wake word" is used. However, their work is limited because:

1. They only used voice-activated IAs, while in our work, we use a various range of IoT devices e.g., smart plug, smart cameras, and smart lamb.

2. They did not consider encryption traffic generated by the IoT device, and they suggested a tool to collect the IoT packets to analyze it.

In contrast, our work solves the previous issues by developing a tool that automatically reads collected encrypted IoT traffic and infers, with high accuracy (99.8%), the type of data transferred to the IoT cloud (Chapter 5). Also, we develop another text mining tool that automatically reads IoT privacy policies. Finally, we link the two tools together to evaluate compliance according to the actual behavior of the IoT device with what the manufacturer states in its PPA.

2.4 Summary

In this chapter, we discuss the important background and definitions of the main terminologies we used in this thesis. To gain a full understanding of the topic, we review the recent literature in four main arias that are related to our study. First, we review the current IoT testbeds techniques and the objectives of each one. Second, we discuss the latest studies and experiments related to the IoT data privacy concerns and the type of data leak from the IoT device. Also, we have briefly discussed some of the most recent literature in collecting and analyzing IoT traffic as well as review the existing work in analyzing different issues related to reading, understanding the complicated meanings, and annotating privacy policies. Thirdly, we pay particular attention to the literature that focuses on analyzing IoT privacy policies. Finally, we discuss the recent studies targeting compliance issues between the Android apps and their privacy policies as well as the studies targeting compliance issues between the IoT device and their privacy policies.

With the acquired information from this chapter, we can proceed to the next chapter in order to discuss the methods we used to collect the dataset that will be used in this thesis.

Chapter 3

Data Collection Methodology

3.1 Introduction

In this chapter, we explain how the data were collected and extracted to serve this research. In particular, we cover the methodology used to collect the required data from two different data sources to conduct the first contribution in chapter 4, the second contribution in chapter 5, and the third contribution in chapter 6. To collect the first data source, we study the IoT device's network behavior and analyze its pattern by using the traffic properties obtained at the network level. Using this, we establish the ground truth in order to develop a tool to classify the encrypted traffic that emerged from the IoT device. For the second data source, we study in-depth the language used within IoT PPAs. Then we annotate the texts to create an automated tool that can read long and complicated privacy policies to extract relevant information.

In Section 3.2, we describe the IoT devices used in this thesis. While, Section 3.3 describes the methods used to collect the data to achieve the first contribution. In Section 3.4, we describe the methods used to collect the data to perform the second contribution. Section 3.5 describes the methods used to collect the data to conduct the third contribution. Finally, the summary of this chapter is presented in Section 3.6

3.2 IoT Devices

Our analysis covers four IoT devices. We selected these devices based on the following factors: First, we expect an average consumer can afford the price to buy such IoT devices. Second, we choose IoT devices based on their popularity and customer ratings.

The devices included in this thesis, see Table 3.1 for more details, are two smart cameras and two smart home automation devices i.e. smart plug and smart lamp. Also, we use an Android smartphone to install the recommended apps of each IoT device in order to control its functions.

Type of Devices	Model Type	IoT Device Manufacturer	Type of IoT-app (iOS, Android)
Smart Plug	HS110	TP-link	KASA version 2.11.0
Smart Camera	NC200	TP-link	TpCamera version 3.1.12
NetCam HD	F7D7601fc	Belkin	NetCam version 2.0.4
Smart Lamp	B22	Lifx	LIFX version 3.13.0

Table 3.1: IoT devices used in this thesis

3.3 Data Collection Experiments for the first contribution

The objective of this experiment is to determine to what extent IoT manufacturers are adhering to their PPA presented on their website. To achieve this, we implement two different stages, i.e. theoretical and practical stage, each of which has its own data source. Then, we compare, manually, the results from both stages to evaluate the level of compliance between the data transfer from the IoT device with what stated in its PPA. The results of our experiment are presented in chapter 4.4. Now we explain each stage separately.

3.3.1 Stage one (Theoretical)

The objectives of this stage are: First, determine whether IoT manufacturers have a PPA that is appropriate for their IoT products? Second, create eight main criteria, based on the GDPR, that must be applied by all IoT manufacturers when they design their PPA. The aim of implementing such criteria is to preserve the privacy of the IoT end users. To do this, we analyze eleven popular IoT manufacturers with the aim of finding out if such manufacturers offer appropriate PPA for their devices. The second aim is, to investigate whether the IoT manufacturers provide sufficient information in their PPA, such as what kind of personal data they collect from their IoT device, whether they interact with a third party or not, etc. The eleven IoT manufacturers that we analyze are the following:

1. LIFX.
2. AWAIR (Bitfinder).
3. Google Home.
4. Tp-link.
5. Samsung smart home.
6. Belkin.
7. Nest Labs.
8. Hive.
9. Toymail.
10. Philips Lighting.
11. Honeywell.

The results of this analysis proved the there are critical issues related to the IoT PPA that has not been addressed before. Accordingly, we establish eight main criteria in which any IoT manufacturer should apply them when creating a PPA for their respective IoT devices. These criteria can be found in chapter 4.4.1

3.3.2 Stage two (Practical)

The objective of the second stage is to investigate the actual behavior of the IoT device. To do this, we conduct a testbed to collect the traffic from two different IoT devices, i.e. Belkin NetCam and Tp-Link Smart Plug (see Table 3.1). In particular, we need to find out precisely from such traffic what kind of information is being transferred from the IoT device, and whether these information are sufficiently detailed in the IoT PPA. We explain in the following subsections the network configuration, the data collection, and the interaction experiment used to collect the traffic.

3.3.2.1 Network configuration

Figure 3.1 illustrates that, in this context, traffic is transmitted (and therefore needs to be monitored) among three points: IoT device, IoT app installed in a smartphone, and IoT manufacturer's cloud. In this testbed, we use two IoT devices as we mentioned earlier. Also, we use the Kali Linux laptop for traffic sniffing/monitoring, Ethernet cable, and home router. For IoT devices that require a companion app, we use an Android smartphone (Samsung S8 edge) to install the recommended app of each device, see Table 3.1. The methods used to analyze the traffic as well as the results of the analyses will be discussed in more detail in chapter 4.4.2.1

First, we need to configure the Kali Linux laptop to work as a Wi-Fi hotspot to connect the IoT devices and the Android smartphone to the Internet. To do this, we connect the Kali laptop to the router via the Ethernet cable to access the Internet. Then, we activate its Wi-Fi hotspot, as described in [7]. Second, we install the recommended

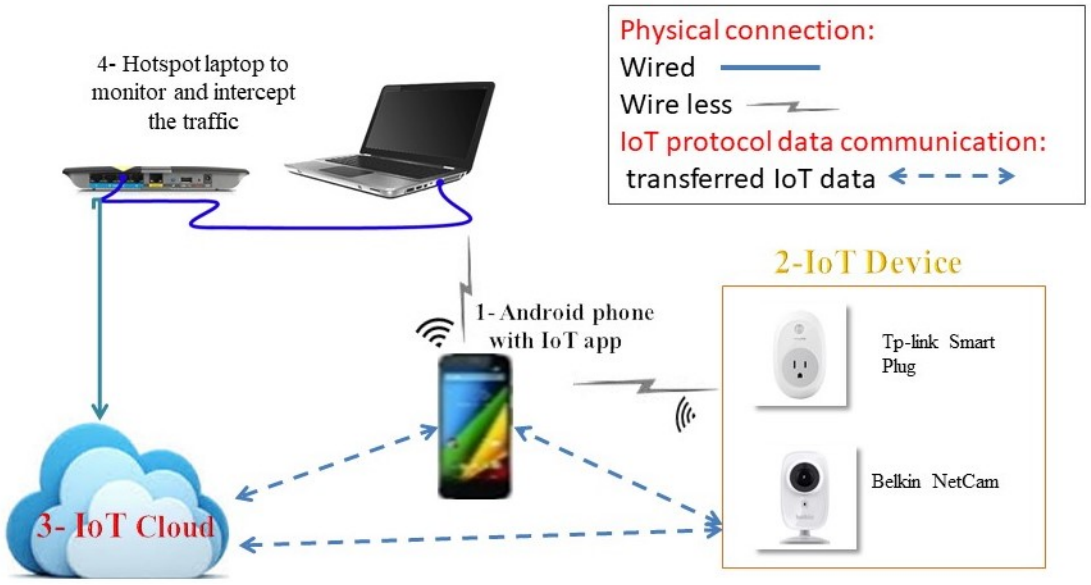

Figure 3.1: IoT Compliance Testbed

app of each IoT device in the Android smartphone. Next, we connect the smartphone to the Internet using the Kali hotspot. Finally, we configure the two IoT devices through their IoT apps to connect them to the Internet through the Kali hotspot. Using this configuration, we were able to sniff and monitor the local traffic moving between the IoT devices and the android application to the IoT cloud (and vice versa).

3.3.2.2 Data Collection

All the traffic of the testbed's network was automatically collected using the Wireshark tool [11] running on the Kali Linux laptop. We used the MAC address of each device as an identifier to separate its traffic from the traffic mix that includes other devices in the network. The resulting traces were stored as pcap files on an external storage device. We started logging the network traffic in our smart home environment from 1-April-2017 to 20-Jul-2017; each raw trace data contains packet headers and payload information.

3.3.2.3 Interaction Experiments

To analyze the behavior of the IoT device and thus find out whether its actual behavior complies with its PPA, we conduct several active interactions with each IoT device. These include manual and automated interactions as following:

1. Control the functionality of the Tp-Link plug through its app:
    - Switch on/off the plug manually.
    - Set up a timer to switch on/off the plug automatically.
    - Define a schedule with specific times within one day to switch the plug on/off automatically.
2. Control the functionality of the Belkin Netcam through its app:
    - Record live videos and capture pictures manually.
    - Set up a timer to record live videos automatically.
    - Define a schedule with specific times to record videos and capture pictures automatically.

3.4 Data Collection Experiments for the second contribution

The objective of this testbed is to study in-depth data disclosure from path C in Figure 3.2. The information sends to the IoT cloud from the IoT app i.e. path C, is much more sensitive than the information sends to the IoT cloud from the IoT device itself i.e. path A. This information not only reveals the type or the traffic rate of the IoT device, but also it reveal user's PII, such as credentials, location, as well as the interaction between the IoT end user and the IoT device. The latter type of information

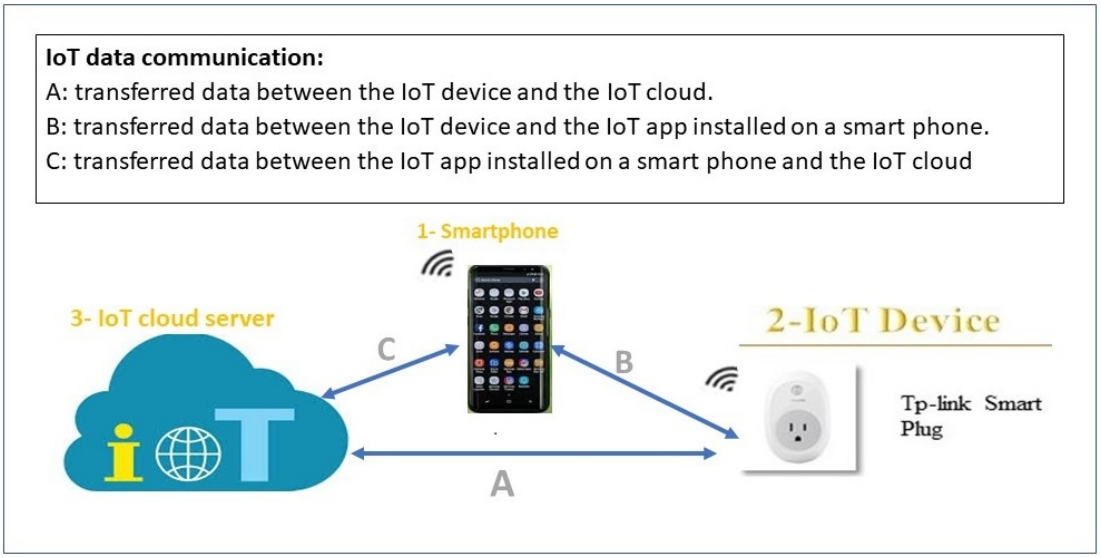

Figure 3.2: Methods of IoT communication with its cloud to transfer data

is not evident from the traffic on path A. In the following subsections, we explain the network configuration, the data collection, and the interaction experiment to establish the ground truth for the second contribution.

3.4.1 Network configuration

We set up our smart home testbed with four well-known and commercially available IoT devices as a representative example of a smart home. The devices include in this testbed are TP-link smart plug, TP-link smart camera, Belkin NetCam, Lifx smart bulb. Also, we use an Android smartphone and connect it to the network. We install the recommended apps on the smartphone to control the functions of each IoT device in our testbed; see Table 3.1 for more details. Additionally, we use Kali Linux laptop to perform two tasks: (1) monitor and continually collect the network traffic between the IoT device and the smartphone app, and also between the smartphone app and the cloud, and (2) perform a Man in the Middle attack (MITM) as we explain in the next section. Figure 3.3 displays the architecture of the smart-home testbed.

Figure 3.3: Detecting the behavior of the IoT user testbed network architecture.

3.4.2 Data Collection

We conduct our experiments to establish ground truth from November 2018 until April 2019. We use Wireshark [11] running on the Kali Linux laptop to passively capture and collect the traffic data of the IoT devices and their relevant IoT apps: First, we determine the IP address of each IoT device within the smart home network; then, we identify the IP address of the smartphone that has the installed IoT apps. The second and third steps are performed in parallel: In the second step, we intercept and therefore collect the traffic by conducting a MITM attack with ARP spoofing between the smartphone and the IoT cloud, the steps of performing this attack are detailed in [5]. This attack allows us to record all network traffic between the IoT cloud and the IoT app in both directions. Figures 3.4 and 3.5 illustrate the redirection that ARP spoofing causes in the traffic between the IoT app and the IoT cloud. Before ARP spoofing, the traffic goes via the router, Figure 3.4; after the ARP spoofing, the traffic

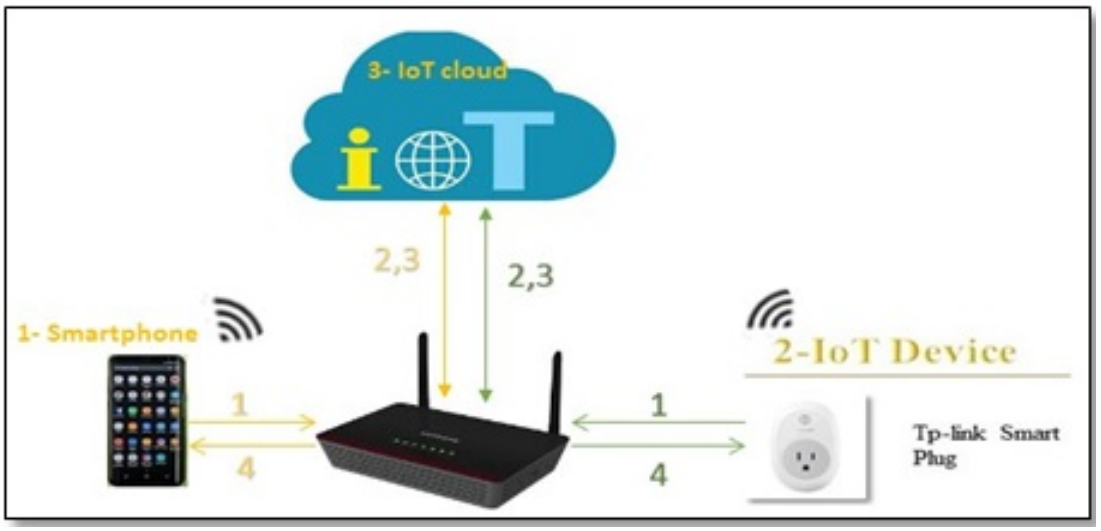

Figure 3.4: All traffic goes through the router.

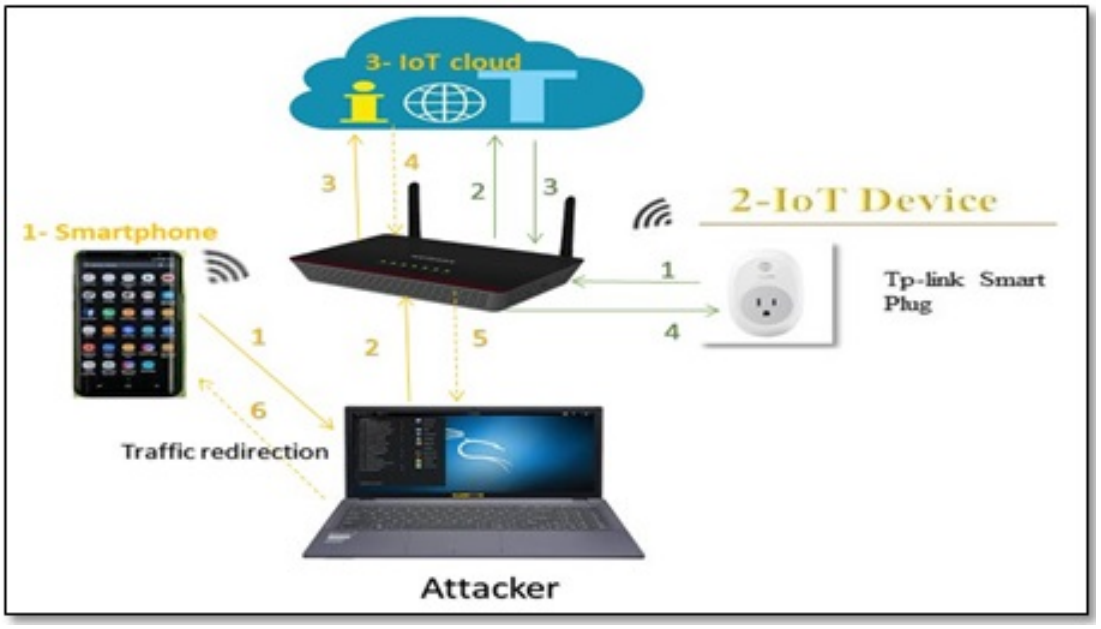

Figure 3.5: IoT-app traffic is redirected through the Kali laptop (Attacker).

goes via the attacker device (in this case via our Kali laptop), then Kali sends it to the router as Figure 3.5 shown. While the MITM attacks are active, we interact with each IoT device mentioned in Table 3.1 separately. We perform several interactions because they are common among IoT apps (see the next subsection). This second step collects encrypted TLS traffic that we need to decrypt to establish the ground truth

about the packets that the IoT app sends to the IoT manufacturer's cloud. We do this decryption in the third step, while we are collecting the traffic. First, we used the Burp Suite tool [6] on our Kali laptop. In Burp Suite, we set up the proxy server port to 8080 to listen to the network traffic of the smartphone and the IoT device. Second, we configure the Wi-Fi setting of the smartphone to use the same proxy server port. Finally, we install the Burp Suite certificate onto the smartphone User Trust Store.

It should be noted that these steps only work if the IoT app does not employ certificate pinning [61]. In our case, KASA, TpCamera, and NetCam do not employ certificate pinning, but Lifx does. One of the solutions to solve the certificate pinning problem is to reverse engineer the IoT app. Then, install the fake certificate from the Burp Suite [6]. Finally, recompile the new version of the IoT app and re-install it on the smartphone. We make the collected traffic of some IoT devices publicly available at (`https://github.com/Alanoud-Subahi/IoT-app_PrivacyInspector`).

3.4.3 Interaction Experiments

To understand what information does the IoT device sends to its cloud, we conduct several interaction experiments with each device. First, we wait for each device to start up, then we perform one interaction at a time. Next, we start capturing the traffic and label it with the name of this interaction. The time specified for each interaction varies depends on the type of device. The type of interactions we consider for these experiments are the following:

1. Login to the IoT application: permitting the user to control the IoT device functions.

2. Alter settings including changing the password: permitting the user to modify the IoT device settings or the password.

3. Delete the IoT device: allowing the user to cease use of the IoT device by deleting it from the application, and consequently deleting it from the IoT cloud/server database.

4. Logout from the IoT application: which ends the user's access or control of the IoT device functions.

3.5 Data Collection Experiments for the third contribution

The goal of this experiment is to simulate how an average person can read and assimilate the information in IoT PPA, especially if the text is long and complicated.
Therefore, we study in-depth the complicated and ambiguous sentences that average end users won't understand with the aim of informing them about the data collection practices as well as the type of personal information that IoT manufacturer's PPA collects when they use its products. The results of this analysis can be found in chapter 6.

3.5.1 Data collection

To perform our analysis and apply our annotation scheme, we need to collect a range of IoT PPAs. We select our policies based on the popularity of the IoT manufacturers. In total, we come up with 50 different IoT PPAs, covering smart home appliances, smart kitchen appliances, smart security devices, smart wearable devices, and smart health and fitness devices. However, the tool we create in chapter 6.4 trained to accept and analyze the URL of any IoT manufacturer PPA.

3.5.2 Data Pre-processing

To prepare the collected IoT PPAs for the analysis conducted in chapter 6, we need first to pre-process the collected data. The methodology that we use includes the following steps:

1. We use Urllib.request module for fetching the URLs of the IoT PPA; the result of this module is a text contained HTML and XML tags.
2. To extract the HTML text only and remove all unwanted tags, we use the Beautiful-Soup2 library.
3. We use Regular Expressions to remove non-ASCII characters such as punctuation and special characters.
4. The final text has been tokenized into sentences using Natural Language Toolkit, and lower case them.

In contrast with other approaches, i.e. [82], we do not remove English stop words such as "you", "we", "they" etc. because, in our analysis, we consider the role of the party who performs the action. In total, we process 31,661 sentences from 50 IoT privacy polices.
Once the IoT PPAs are ready, we start applying our annotations scheme to each sentence. After that, we use these sentences as instances to extract the relevant features for the classification algorithm. More details are explained in chapter 6.2.3.

3.6 Summary

In this chapter, we describe the environment of two different IoT testbeds used to collect and synthesize the network traffic from various IoT devices, each of which serves

different contribution. In addition, we explain the methods we use to collect and pre-process 50 different IoT PPA to prepare them for the feature extraction and labeling process to serve the third contribution.

Now that we have described the data necessary for this research, we can move on to address our hypotheses and research questions discussed in chapter 1. In particular, in the next chapter, we will prove the existence of a compliance issue between the actual behavior of the IoT device and its corresponding PPA. In addition, we analyze eleven IoT manufacturer and establish the eight criteria that any IoT PPA should apply to preserve the privacy of the IoT users.

Chapter 4

Ensuring compliance of IoT devices with their Privacy Policy Agreement

4.1 Introduction

As we discuss in chapter 2, most of the existing IoT literature is limited to either analyzing IoT device's security and privacy issues, discovering IoT devices vulnerabilities, or performing different attacks targeting various types of IoT devices, which related to user data disclosure. However, no attention has been given to risks and vulnerabilities associated with the type of personal information being collected from IoT devices, nor to the level of compliance to the corresponding PPA. In fact, a significant proportion of users do not understand what kind of information does the IoT device collects about them, or that they are sharing information in the first place [70].

In this chapter, we consider reading and analyzing, manually, the PPA of eleven popular IoT manufacturers. The results reveal that half of those IoT manufacturers do not have an adequate PPA specifically for their IoT devices. Also, we capture the traffic of two IoT devices (i.e., Tp-link smart plug and Belkin NetCam) to prove that the data leaked from the two devices don't comply with what they stated in their PPA.

The rest of the chapter is organized as follows: In Section 4.2, we explain why it is essential for IoT devices to have separate PPA, while Section 4.3 discusses some significant differences between website PPAs and IoT PPAs. In Section 4.4, we present

our methodology in proposing the main eight privacy criteria that should be applied to any IoT device; also, our IoT compliance testbed design and results are explained in detail in this section. Finally, the conclusion is presented in Section 4.5.

4.2 The importance for IoT devices to have separate PPA

In this section, we explain why it is important to have a separate PPA for IoT devices. As we mention in chapter 2.2.5, if a manufacturer intent to collect personal information about the user, it should state that in the form of a PPA to safeguard individual privileges. Therefore, IoT devices need special PPA because they collect personal information about their users. However, existing privacy laws and regulations are not focused on IoT devices specifically. We argue that they are insufficient to capture important differences between general data protection scenarios and IoT-specific scenarios. We now provide some arguments to support our claim:

1. IoT devices are being manufactured for close, personal use. For example, a smartwatch could be worn for most of the day, which would collect a huge amount of information about the personal habits and behavior of the wearer [84],[33]. Therefore, the user has the right to have prior knowledge of what kind of sensitive information is being transmitted.

2. The financial value of IoT users' data is connected to the ability of this data to help manufacturers sell more products (e.g. by knowing the user behavior or the user preferences). It could be argued that IoT manufacturers have a vested interest in collecting user data without informing users about it [84]. In this scenario, to prevent IoT manufacturers from using user's data for their interest, they should issue a sufficient PPA and comply with it.

Therefore, consumers need to be made aware in advance that their information is not completely secure and private. They should also know that outside entities may be able to eavesdrop on their information. This prior knowledge is typically encoded in a PPA, which covers the whole data lifecycle, from the exact point in time when the IoT device's sensors capture data packets until the phase where raw data is effectively deleted, specifically for sensitive data gathering devices [70]. Furthermore, it is vital that users recognize their own rights, even if they agree to a PPA and the use of their data, they still have a right to take it back at any time [68].

The FTC [2] stated that putting PPAs on websites only does not do the job of informing users about its data practices implementation. The correct alternative, according to them, was to clearly state the PPA through its device setup or upon purchase. A study by the ICO [8] reveals that six in ten IoT devices do not come with sufficiently comprehensive PPAs. These agreements fail to explain why and how IoT devices utilize personal data fully. The study also reveals that 59% of IoT device PPAs did not clearly explain to the users how their information was going to be collected, used, and disclosed. In comparison, 68% failed to specify how they store the information adequately. Also, a high percentage (72%) of IoT devices did not mention how users could edit their information (delete, update), and finally only 38% adequately explain how users could contact the manufacturer if they have any privacy concerns.

4.3 Difference between Website PPA and IoT PPA

It should be acknowledged that the actual creation of PPA is not always an easy task. For website designers and developers, the focus is on transparency and clarity. The goal must be helping users get a comprehensive picture of why their data is being collected and what to do if they wish to prevent this. Consequently, the words and concepts should be expressed as simply as possible so that the user has a chance to fully understand everything on the PPA. There must be no ambiguities or grey areas, and it

must be clear how the device manufacturer plans to treat their PII. When it comes to websites anyway, there is much evidence to suggest that PPA do work, but are changes to the traditional method needed for IoT devices? This is the big question. Is the IoT so fundamentally different from older technologies that it requires a completely new approach?

According to Perez et al. [71], there are three different ways in which the IoT PPA is organized. The first one is "All included," which refers to the PPA that covers all the privacy practices of the manufacturer in addition to the IoT system and devices e.g. Ecobee, Rachio, and Fitbit. The second way is "Referencing," which refers to a separate IoT system/device PPA on a different web page but still linked to the manufacturer's general PPA, such as Google Home and Amazon Echo devices. While the last way is "Isolated," which refers to a totally separated IoT system/device PPA from the manufacturers' privacy practices e.g. Nest smart devices, **and this is the most suitable approach.**

Moreover, it is important to highlight that there are some significant differences between IoT PPAs and traditional PPAs for websites due to the following reasons:

1. IoT privacy has changed the concept of previous website PPA content due to the sensitivity of personal data transferred from the IoT devices to the cloud and vice versa.

2. The data captured by, e.g., a wearable device, which reveals the pattern of the user's life, is transferred from the IoT device to its cloud. This information is much more sensitive than what happens when data is collected and transferred while a user is browsing, searching, or even emailing through websites.

3. IoT devices create data while they are actively connected to the internet. With wearable tech and other IoT devices, for example, it is not always necessary to manually connect to the web, so there is the potential for data capture and transfer at times when the user is not aware.

Therefore, IoT manufacturers need to be thinking about these issues when designing and implementing PPAs for their IoT devices. In fact, Governments, along with industry stakeholders, have already established several regulations and policies to standardize and ensure IoT privacy [42], such as:

- Before using an IoT device, users must be fully informed of the ways in which their data will be used and must give their consent to these terms. Users must be clearly told, in the form of a PPA, about what information will be stored and shared, why this information is being collected, and who will be able to access it.
- IoT users should have the freedom to decide whether to participate or not in any exchange of information.
- IoT users should always be allowed to remain anonymous (not share identifying details) on domains that require identity proof.

4.4 Methodology

The goal of this section is to prove that most IoT manufacturers are not adhering to what they state in their PPA. To do this, we split our methodology into two phases. Section 4.4.1 explains phase one, which is the theoretical analysis in order to establish the eight privacy criteria. While section 4.4.2 describes phase two, which is the practical testbed in order to prove whether there is a compliance issue between the actual behavior of IoT devices with their respective PPA or not.

4.4.1 Theoretical Phase

In this stage, we focus on the language used within the IoT PPAs. The goal of this phase is to determine the following:

1. How many IoT manufacturers have a PPA that is appropriate for their IoT products?
2. To what extent do these IoT manufacturers adhere to the eight criteria outlined in this section?
3. Which criteria are most and least likely to be sufficiently met?

To achieve these goals, we conduct two separate studies. The first study is to read and analyze the PPA of eleven popular IoT manufacturers, as we mention in Chapter 3.3.1. The objectives of this analysis are:

1. find out if these companies offer appropriate PPA for their devices;
2. for those who provide PPA, investigate whether they provide sufficient information in their PPA.

As a result of this study, we discover some crucial issues:

1. According to [71], we found that most of the eleven IoT manufacturers have no separate PPA for their IoT products i.e. "Isolated". Instead, the vast majority are categorized either as "All included", or "Referencing".
2. In terms of the manufacturers who have separate PPA, most of them did not apply the Government standard regulations [42]. On the contrary, we found that there are no standard criteria to cover and explain all the information that the user needs to know before using such IoT devise, i.e., what kind of personal data they collect from their IoT device, whether they interact with a third party or not, etc.

based on the above mentioned results, we conduct the second study, which focuses on setting standard criteria based on the GDPR that each IoT manufacturer must implement in their PPA. To create these key standards, we conduct research on the responsibilities of modern IoT manufacturers. Consequently, we propose eight main privacy policy criteria in the form of the following obligations for any IoT manufacturers:

Criterion.1) Explain what kind of personal and non-personal information the manufacturer will collect from their IoT device and explain why they need it.

Criterion.2) Clearly specify to IoT users what specific information will be provided by IoT users themselves, once they create their IoT account.

Criterion.3) Explain to IoT users what information will be collected from them automatically when they perform specific action with their IoT devices and why the manufacturer needs to collect that information.

Criterion.4) Explain to IoT users how their information will be used and treated by the IoT manufacturer.

Criterion.5) The rights of IoT users to control (edit, delete) their data saved in IoT cloud.

Criterion.6) Clearly specify to IoT users how long they will store their PII on the IoT manufacturer's cloud.

Criterion.7) Clearly ask for the IoT user's consent in order to collect/share extra information and explain the reason for this request.

Criterion.8) Clearly inform the IoT users of the geographical location of the IoT servers where the manufacturer keeps/stores the IoT user's data.

These criteria have been supported by the ICO report [8] based on the following considerations:

1. The standards set in place by the GDPR clearly state that any personal data should be processed in highly secured environment and guarantee total privacy of personal data, for instance protecting any type of unauthorized access by using standard security methods. The GDPR has set the criteria for manufacturers on what data needs to be collected about the users through a table created by them.

Categories of personal data represent one such information. This point support criteria number 1,2, and 3.

2. The GDPR underlines the importance of telling users how their data is being used. This point covers criterion number 4.

3. The GDPR is critical on the fact that users have the right to remove their personal data at any time with no restrictions as be totally forgotten. This point covers criterion number 5.

4. The GDPR states that users have the right to know the period of keeping their personal data under the manufacturer's possession. In addition, they have the right to withdraw their consent at any time. This point covers criterion number 6,7.

5. Special restrictions have been imposed by the GDPR on the transfer of personal data outside the European Union, to third countries, or to any international organizations without prior user knowledge and approval, to ensure that the level of individual protection is not undermined. This point covers criterion number 8.

4.4.1.1 Analyze the level of compliance of the eleven IoT manufacturer to the eight criteria

After we analyzed the PPAs of eleven IoT manufacturers, as mentioned earlier, we manually apply the eight key criteria to each IoT manufacturer. Then, we identify the respective levels of adherence of each manufacturer as well as identify which criteria are most likely to be sufficiently met according to this analysis. Tables 4.1(a) and 4.1(b) illustrate each individual company's compliance (eleven IoT manufacturers) to the mentioned eight requirements. We establish the level of compliance by studying the PPA for each IoT manufacturer.

As we can see from Tables 4.1(a) and (b), the most likely criteria to be fulfilled are criteria no 1,2,4 and no 5 with (82%), in other words, nine out of eleven IoT manufacturers comply to these four criteria, while eight out of eleven IoT manufacturers comply to only criterion no.3 (73%), followed by criterion 7 which achieved compliance by seven out of eleven IoT manufacturers (64%). Furthermore, only six of the IoT manufacturers comply to criterion no.6 (55%). Finally, there is one criterion which are poorly explained or consistently overlooked, criterion no 8, this criterion achieved compliance by only four IoT manufacturers (36%). Figure 4.1 demonstrates a comparison of levels of compliance to the eight IoT privacy criteria among the eleven IoT manufacturers. Firstly, the graph shows that only one of the eleven IoT manufacturers (Awair) comply to all eight privacy policy criteria. While four out of eleven manufacturers (88%) comply to seven criteria. Secondly, 63% which represent three out of eleven IoT manufacturers comply only to five criteria, whereas just two IoT manufacturers comply to half of the criteria. Finally, it should be noted that the lowest level of compliance is for one IoT manufacturer (LIFX) which comply to only 2 criteria.

Based on our results, we could argue that the eleven IoT manufacturers did not achieve full compliance with the eight criteria. However, it is crucial for any IoT manufacturer to comply with the list of criteria because it could be considered as a definitive breakdown of the things that IoT manufacturers must tell users both before and after they activate their IoT devices. In addition, according to Edith Ramirez statement [2], by adhering to these criteria, IoT manufacturers will gain transparency, honesty, and trustworthy relationship between them and their IoT users/consumers, which will have a great impact on the IoT manufacturers' profits.

4.4.2 Practical Phase:

In this phase, we design and implement a practical testbed to investigate whether there is a compliance issue between the actual behavior of IoT devices with their respective PPA or not. To do so, we need to identify if the two IoT devices are adhering to

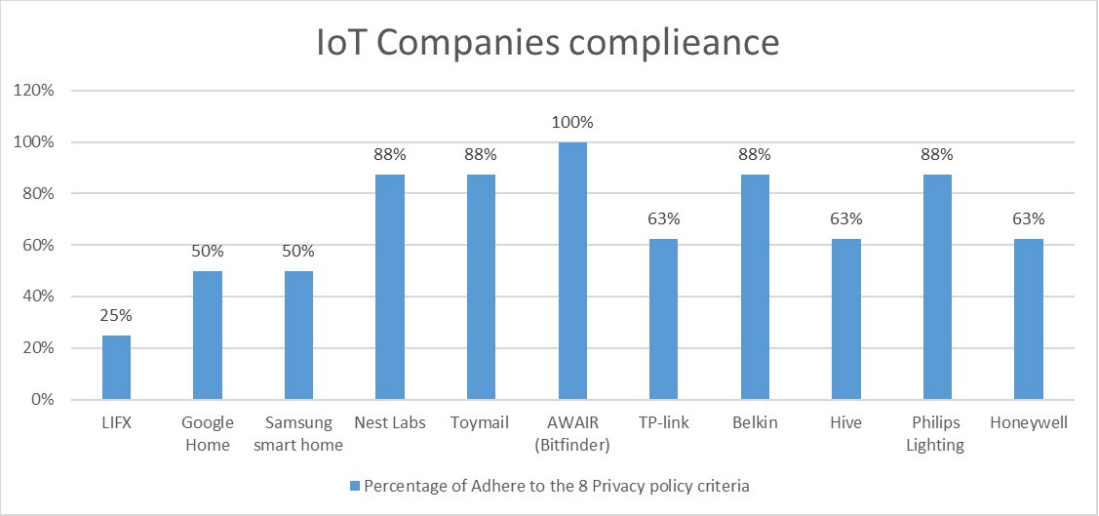

Figure 4.1: How many of the 8 privacy criteria does each IoT manufacturer adhere to.

their own PPA presented in their website as well as to our eight privacy criteria. The network configuration of this testbed and the process of collecting the IoT traffic, has been explained in detail in chapter 3.3.2.

4.4.2.1 Results of IoT Compliance Testbed Experiments

Belkin NetCam The NetCam smart camera relies on a digital video recording cloud service called Seedonk (cloud.seedonk.com) for storing images and video streams. NetCam could be controlled remotely via its app called "NetCam.", see Table 3.1. This app allows remote access to the cam as well as view live video either from a smartphone, a computer, or any other device. Besides, the app supports two-way audio and allows users to communicate with each other.

A) Packet analysis using Wireshark

Using the IoT architecture illustrated in Figure 3.1, we managed to sniff the data packets moving between the NetCam and its cloud, as well as between the NetCam app and the cloud. Using Wireshark installed on Kali Linux, we were able to monitor the traffics and observe the following:

IoT company and Privacy policy main criteria	LIFX	Google Home	Samsung smart home	Nest Labs	Toymail
Criteria no.1	X	X	✓	✓	✓
Criteria no.2	✓	✓	✓	✓	✓
Criteria no.3	X	X	✓	✓	✓
Criteria no.4	X	X	✓	✓	✓
Criteria no.5	X	✓	X	✓	✓
Criteria no.6	X	✓	X	X	✓
Criteria no.7	✓	✓	X	✓	X
Criteria no.8	X	X	X	✓	✓

(a) apply the 8 criteria to the first 5 IoT manufacturers

IoT company& PPA main criteria	AWAIR	TP-link	Belkin	Hive	Philips Lighting	Honeywell	The % of compliance to each criterion
Criteria no.1	✓	✓	✓	✓	✓	✓	82%
Criteria no.2	✓	✓	✓	X	✓	X	82%
Criteria no.3	✓	✓	✓	✓	✓	X	73%
Criteria no.4	✓	✓	✓	✓	✓	✓	82%
Criteria no.5	✓	✓	✓	✓	✓	✓	82%
Criteria no.6	✓	X	✓	✓	✓	X	55%
Criteria no.7	✓	X	✓	X	✓	✓	64%
Criteria no.8	✓	X	X	X	X	✓	36%

(b) apply the 8 criteria to the last 6 IoT manufacturers

Table 4.1: The level of compliance between 11 IoT manufacturers against 8 criteria.

a) SSL/TLS traffic as well as unencrypted traffic. It was clear from Wireshark that video files aren't transferred using encrypted methods.

b) Two distinctive communication patterns that give details about the NetCam network. The first one relates to how the TCP connection is maintained via

the Seedonk cloud. While the second exposes the consistent DNS queries designed to find the IP address of the Seedonk server.

c) After the TCP handshake, a packet is delivered from the camera to the cloud and significant amounts of data can be inferred from this packet such as the username of the device owner, the MAC address of the IP camera, and the local IP address.

B) Mobile app analysis using Burp suit tool

As we mentioned previously, some traffic was encrypted. In this section, we use the Burp Suite tool to intercept the SSL/TLS encrypted traffic between the NetCam app and the Seedonk cloud using man in the middle (MITM) attack. We set up the Burp Suite by following [6], and then we download the Burp Suite certificate in the Android smartphone (Samsung S8 edge) trust store. It should be noted that this kind of attack only works if the application does not employ certificate pinning [61]. By accessing the burp suite interface, the SSL/TLS traffics were displayed in plain text form, see Figure 4.2. It's worth saying that we could not uncover any user credentials via the NetCam application. Consequently, we attempted to do so in another way. We navigated to the NetCam website `https://NetCam.Belkin.com` from the Android smartphone. Accordingly, we manage to break the SSL/TLS connection between the smartphone web browser and between the NetCam web servers and uncover the credentials in plain text form, see Figure 4.3.

C) Belkin NetCam Compliance to its PPA

As regards information which complies with the NetCam PPA:

a) Netcam application does not transmit information about the exact location of the device. In this case, we did not give consent for this data to be captured. This demonstrates a high level of compliance because the privacy agreement states that no such information can be collected without permission from the user.

b) NetCam appears to transmit only data that has been expressly permitted and described in the agreement. This includes technical information about the NetCam device (model, version, H.W, S.W, firmware, etc.) and utility settings (resolution, status, size, mode, notifications, etc.).

c) We could not capture any information related to the smartphone, such as (O.S, H.W, manufacturer, model number, etc.). This demonstrates a high level of compliance because the privacy agreement states that no such information can be collected.

As regards information which does not comply with the NetCam PPA:

a) We discover that the Belkin NetCam uses encryption technology to protect PII data as it moves between the application to the cloud (and vice versa). While this encryption is a good way to ensure that personal data is secure, there is no proper mention of this in their PPA. Therefore, the manufacturer needs to think about providing more details about its encryption process. If it does not, customers might feel deceived, and it could reflect badly on the IoT manufacturer and even damage its sales. On the other hand, most users are aware of the importance of employing data encryption methods.

b) Even though the NetCam PPA does not include the name of the cloud server used by them, we are able to discover this information. Also, attempting to uncover the geographical location of the cloud server we find two locations, one server located in Ireland/Dublin and the other located in the United States/Virginia, this finding violates criterion number 8. According to GDPR, the user has the right to know the geographical area containing the servers/clouds where their personal data is kept.

c) We found that, although NetCam collects user's images and videos and sent them to the cloud server, there is no explicit mention of this process in the NetCam PPA. This critical finding violates two main criteria, which are number 1 and number 3. According to FTC [2] and ICO [49, 8], it is

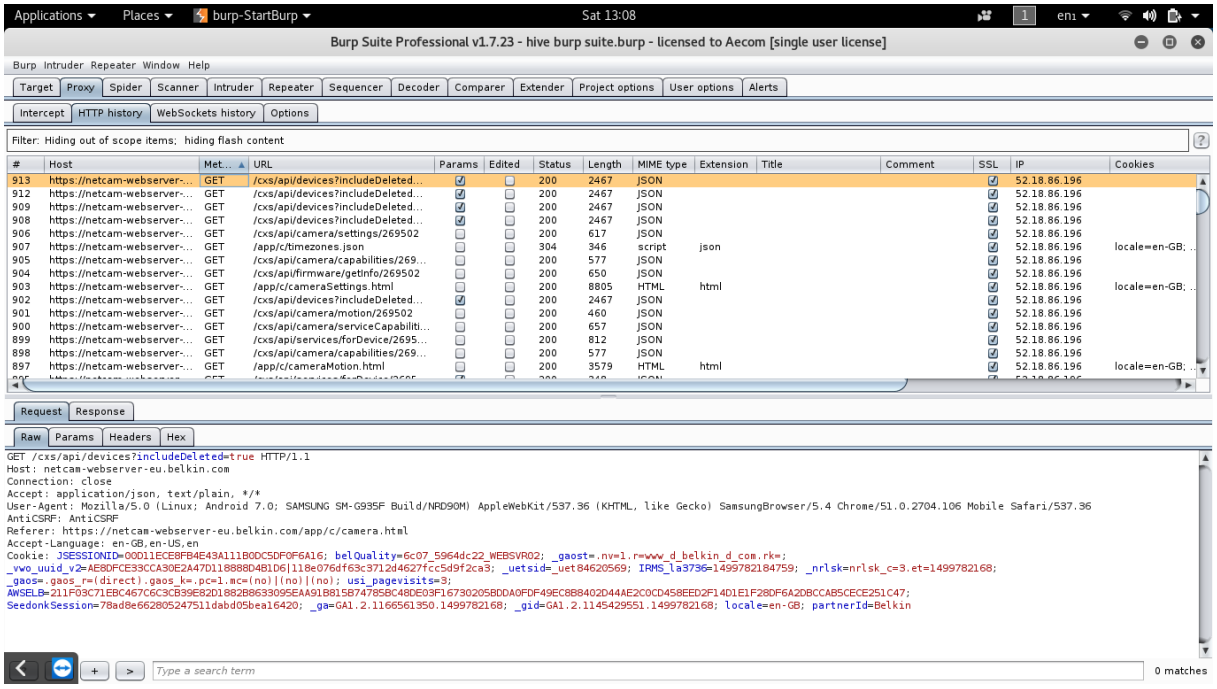

Figure 4.2: Decrypted SSL traffics of NetCam application, as seen in Burp Suite after a Man-in-the-Middle attack.

highly important to inform the users of what kind of information is being collected about them.

Tp-link Smart Plug The Tp-link smart plug contains two physically operated buttons. The first one is an on/off switch, while the second one is a device reset button. Also, Tp-link smart plug uses "Tplinkra" cloud servers to store any information about the user and his devices. The smart plug could be controlled remotely via the KASA app, see Table 3.1. This allows the user to switch the smart plug on and off without touching the physical buttons. The device (and its remote app) provides energy monitoring and scheduling capabilities.

A) Packet analysis using Wireshark

As we mentioned, the KASA app controls the Tp-link smart plug. Hence, we attempt to sniff the traffic moving between the smart plug and its app, which

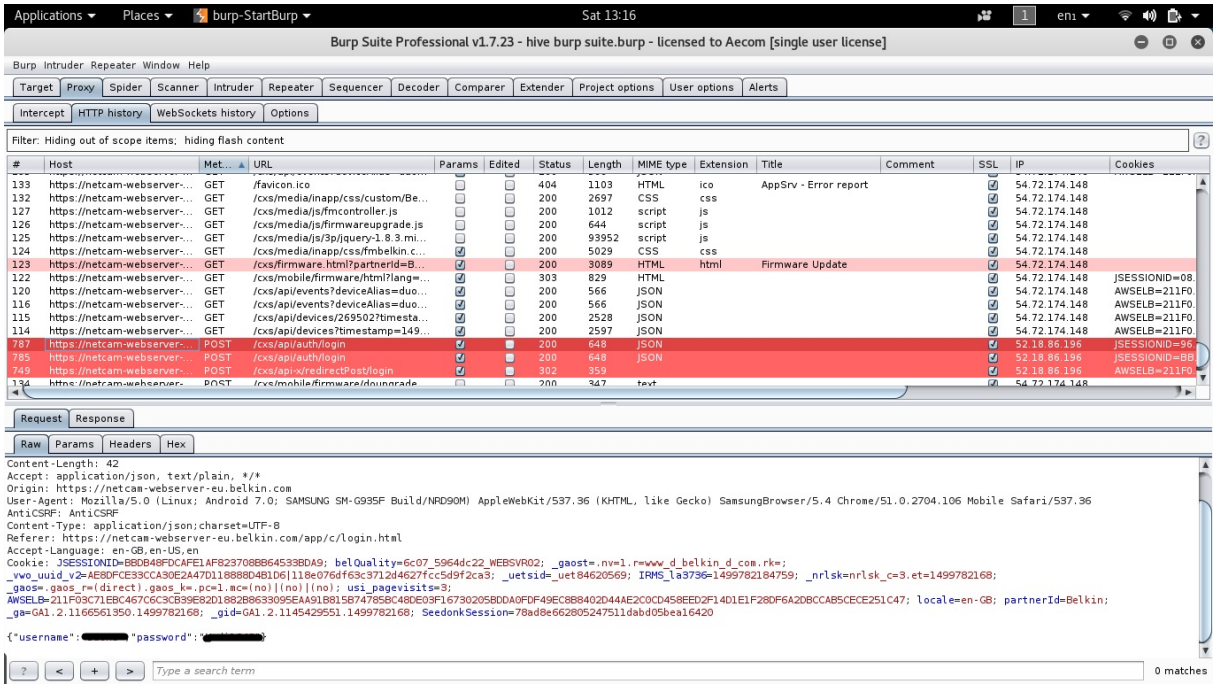

Figure 4.3: Decrypted SSL traffics of NetCam application, as seen in Burp Suite after a Man-in-the-Middle attack.

installed in the Android smartphone (Samsung S8 edge), and between the app and the smart plug "Tplinkra" cloud as illustrated in Figure 3.1. After observing the Wireshark network traffic, we detect encrypted traffic during the interaction between the KASA app and the smart plug. Next, we successfully decompile (reverse engineer) the KASA app and find the encryption function that is used to encrypt the traffic between the KASA app and the Smart Plug server. We use this encryption file to apply the Wireshark dissector in the LUA code. By plugging in the new LUA file, the traffic will automatically decrypt [9]. As a result, we are able to monitor the communications between the KASA app and the Smart Plug on their local Wi-Fi in plain text.

B) Mobile app analysis using Burp suite tool

In order to intercept the SSL/TLS traffic between the KASA app and the cloud via the burp suite tool, we follow the same steps described in Belkin NetCam Section *B)*. We find that when we launched the KASA app at the first time, login

method is triggered and therefore sends user's credentials to the cloud. However, every time we open the application to perform any action (i.e. switch Plug on/off or schedule an event), the helloIoTCloud method triggers and again sends the user's credentials to the cloud, see Figure 4.4 and 4.5. Lastly, we uncover eight main methods of requesting/sending personal data to/from the TP-Link cloud, which are: login method, helloIoTCloud method, list scenes method, isLinked method, retrieve location method, list Rules method, pass through method, and get device list method. The following types of information are transferred using these methods:

a) Application such as appName, appType, appVersion.

b) Client such as clientId, geolocation, locale time-zoneId, mobileType, user-Device manufacturer, userDevice model, device osVersion, ownerEmail.

c) Smart Plug information such as: sw ver, hw ver, type, model, mac address, hwId, dev name, alias, location, fwVer, deviceName, status, deviceType, appServerUrl, deviceModel, deviceMac, isSameRegion.

C) Tp-link Smart Plug Compliance to its PPA

As regards information which comply with the Smart Plug PPA:

The information collected from the Smart Plug and the Kasa application mentioned earlier appears to be in full compliance with the PPA as they stated in detail what type of information the smart plug will collect.

As regards information which does not comply with the Smart Plug PPA:

a) As with the NetCam, it was discovered that the Smart Plug does utilize encryption technologies, even though there is no mention of this in the PPA.

b) There was no information provided about the name of their cloud server, but we could find out that the manufacturer uses a TPLinkra cloud server. Besides, we could determine the geographical location of the cloud serv-

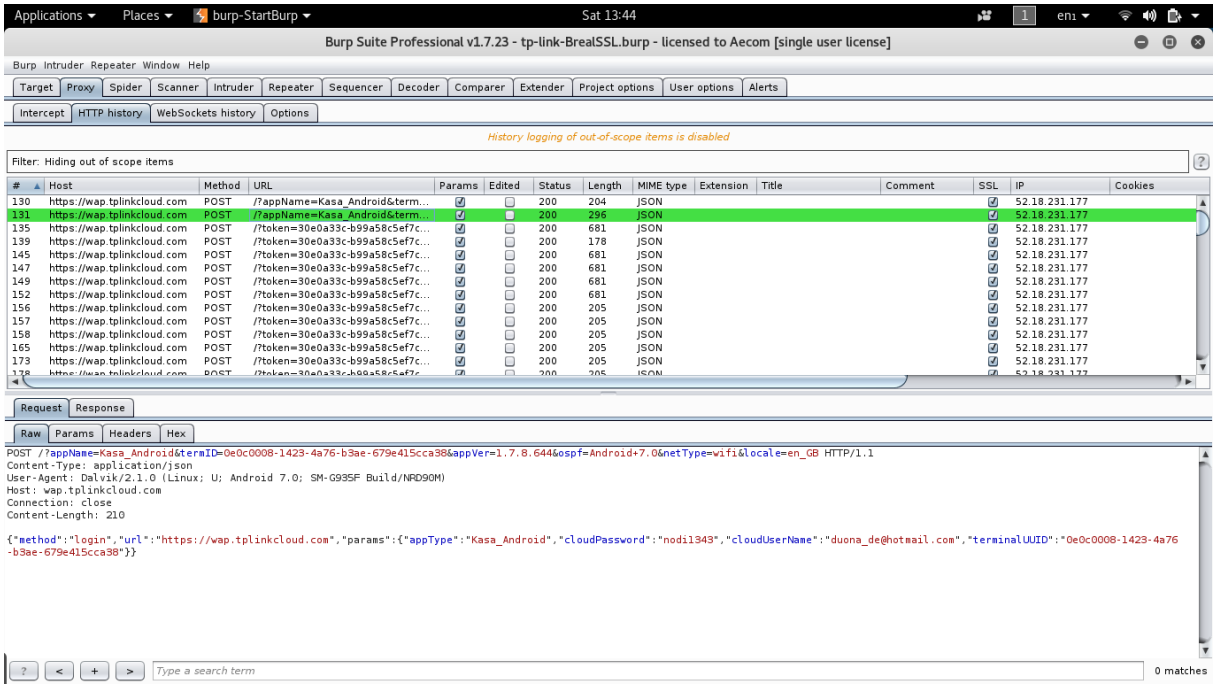

Figure 4.4: Login method with user's credential

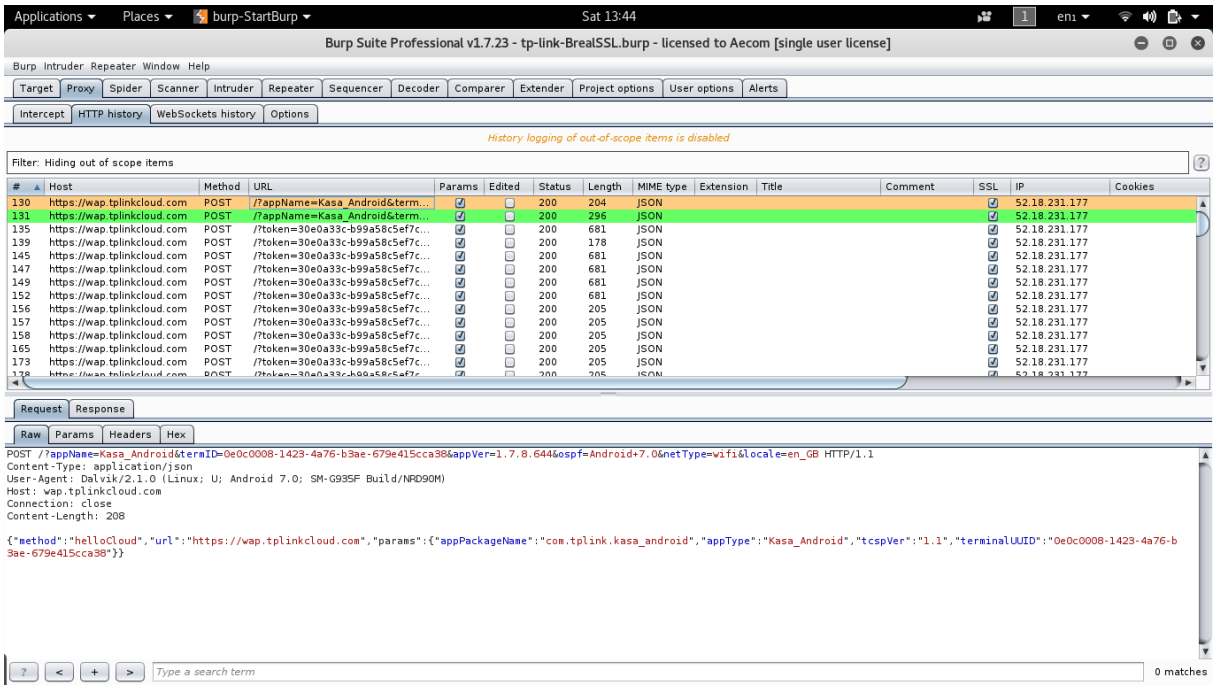

Figure 4.5: Hello IoT Cloud method with user's credential

ers, which was located in the United States/Virginia. This finding violates criterion number 8. According to GDPR, the user has the right to know the geographical area containing the servers/clouds where their data is kept.

To conclude this section, our findings prove that there is a critical violation in terms of the IoT companies' levels of compliance with their PPA. We find that the actual data we obtained from capturing Belkin NetCam and Tp-link smart plug traffic did not comply with what they stated in their PPA. Interestingly, we conclude that Belkin NetCam shows a quite high level of compliance with our eight criteria (88%), whereas from our experiment we prove that the level of compliance of Belkin NetCam with what they stated in their PPA is low as they violate 3 statements within their PPA which are similar to criteria (no.1, no.3, and no.8). In contrast, we find that the Tp-link smart plug shows a quite high level of compliance to what they stated in their PPA as they only did not comply to one statement which is similar to criterion no. 8 whereas it shows only 63% of compliance to the eight criteria see Figure 4.1. Unless IoT companies issue an appropriate PPA that comply with the eight privacy policy criteria and, more importantly, comply with what they state in their own PPA, user's privacy issues will always be compromised.

4.5 Summary

In this chapter, we discuss the importance of having a separate PPA for IoT devices as it differs from website PPA, and we implement IoT privacy compliance testbed. The main objective of such a testbed is to determine the level of compliance of IoT manufacturers with their respective PPA. We posit eight key criteria and compare them with the actual PPA carried out by each IoT device.

First, we investigate the PPAs of eleven IoT devices. Then we manually compare their respective PPA with the eight privacy criteria. The results show that only one criterion

out of the eight criteria has been fulfilled by eleven IoT manufacturers. In contrast, only four out of eleven IoT manufacturers comply with 88% of the eight criteria.

The next step is to construct and execute a testbed procedure for two selected IoT devices; the Belkin NetCam and the Tp-Link Smart Plug. We sniff the data packets being moved between the IoT device and the cloud, between the IoT device and the smartphone, and between the smartphone and the cloud. Surprisingly, we find that the Smart Plug adheres to 63% of the established eight criteria, but as for the terms of their PPA, they show a high level of compliance because they only did not comply to one statement which is similar to criterion (no.8) of the promises contained in its own PPA. Similarly, although we find that the NetCam show quite a high level of adherence to 88% of the established eight criteria, they failed to adhere to their own PPA because they violate three statements, which are similar to criteria (no.1, no.3, and no.8).

Yet, it could still be argued that the percentages of the adherence to the eight criteria are not high enough, particularly in the case of adherence to key data privacy targets. There is a clear need for IoT manufacturers to continue evolving and developing their PPA by either changing the behavior of the device to comply with their PPA or by modifying the PPA to reflect the actual behavior of the IoT device.

We proved in this chapter that there are compliance violations in between the IoT manufacturers with their PPA. Consequently, we seek to overcome these issues by inventing a method that can automatically check the actual behavior of the IoT device as well as read its PPA at the same time. To do so, we have to work into two phases, as we explain in Figure 1.2. We implement the first phase in chapter 5, which aims to analyze the encrypted traffic of the IoT devices in order to understand the behavior of each one from its traffic pattern. While chapter 6 describes the implementation of phase two, which aims to analyze the text of the IoT PPAs. Finally, in chapter 7, we combine the results from the two phases in order to build our compliance tool. Such a tool will present to the IoT end user whether the actual behavior of his IoT device(s) comply with its PPA or not.

Chapter 5

Detecting IoT User Behavior and Sensitive Information in Encrypted IoT-App Traffic

5.1 Introduction

In this chapter, we start with the observation that there are two different ways of sending information about the IoT user to the IoT cloud i.e., D-C and A-C. To the best of our knowledge, no research has been conducted on the second way i.e. A-C. In addition, we prove in this chapter that an adversary who observe and collect smart home traffic can reveal sensitive information about the IoT user throughout the packet sizes and the packet sequences. For example, the adversary can infer, in real time, that a specific interaction (e.g. login to the IoT app) is occurring between the user and a smart plug via its respective IoT app. Also, the adversary can infer which packets carry sensitive PII about their user, as well as the type of this information (e.g. user location or user credential).

Our contribution in this chapter is a multi-class classification tool, called IoT-app PIT, using supervised machine learning to raise the awareness of the IoT users through informing them whether there is a compliance violation of their IoT devices. The objective of this tool is to:

1. Classify the interaction(s) of the user with every IoT-app (e.g. login to/logout from the IoT-app).

2. Classify the packets generated by the user interaction(s) according to their sensitivity level (e.g. sensitive PII, non-sensitive PII, non-PII).

3. Classify the content of the sensitive PII (into e.g. user credentials, user location) and the content of the non-sensitive PII (into e.g. user email, username).

This tool can be continuously applied to classify newly collected (unlabeled) IoT device traffic data. The results of applying such a tool show that 99.4% of the user interactions with the IoT app are correctly detected, whereas 99.8% of the packets the carry sensitive PII caused by this interaction are correctly detected. Finally, 99.8% of the content type of this sensitive PII packets are correctly detected. We leverage the observation that the traffic generated by IoT apps follows a limited set of patterns, which allows us to perform the three classifications above. Finally, we clarify that if an attacker identifies a user's interaction type (e.g. login to the IoT app), he can infer sensitive PII packets caused by this particular interaction. Thus, he can infer the content type of such sensitive PII packet (e.g. login credentials or geographical location). According to Wang et al. [103, 104], 77.38% of users reuse one of their existing passwords. Also, Das et al. [39] estimate that 56% of users change their password at least once every six months because they tend to have the same passwords. This means that if an attacker manages to find the packet that contains the user's password, he could mount an offline password attack to crack the password, which is impossible to detect and faster than an online attack. Therefore, he can gain access to every account the user has.

The rest of the chapter is organized as follows: Section 5.2 discusses the communication methods between the IoT device and its cloud. While Section 5.3 outline how an attacker could attack and collect smart home traffic using our attacker model, followed by a detailed description of the method we use to establish the ground truth in

Section 5.4. In Section 5.5, we present our attack design and implementation, while in Section 5.6 we develop our inspector tool with three multi-class classification methods, each one used to infer a different goal; we also evaluate our tool in the same section. We present the results and discussions in Section 5.7, followed by a summary and conclusion in Section 5.8.

5.2 Methods of Communication between the IoT device and its Cloud

According to [15], there are three different methods of communication between the IoT device and its cloud:

1. IoT device to IoT cloud (D-C);
2. IoT mobile application to IoT device (A-D);
3. IoT mobile application to IoT cloud (A-C).

In fact, there are ample research efforts to uncover IoT security vulnerabilities and exploits, such as [25, 113, 17, 90]. However, researchers that address the privacy risks of IoT devices have focused on the traffic that goes directly from the IoT device to the IoT cloud (D-C) Path A in Figure 3.2. Nevertheless, a significant number of home-based IoT devices come with a companion mobile application. Each IoT manufacturer creates its own mobile application to control, configure, and interface with the device. Therefore, data from the IoT device can also reach the IoT cloud via the IoT app installed on the smartphone (Paths B and C in Figure 3.2).

To the best of our knowledge, no research studies this alternative path. Based on our analysis, the information that is being sent to the IoT cloud from the IoT app (path C) is much more sensitive than the information sent to the IoT cloud from the IoT device

itself (path A) because this information not only reveals the type or the traffic rate of the IoT device, but also it could expose users' credentials, users' location, or users' current interaction with the IoT device via the app. The latter type of information is not evident from the traffic on path A.
We now demonstrate the two different ways that IoT devices use to communicate with their manufacturer's cloud:

1. Device-to-cloud (D-C): as Figure 3.2 illustrates, path A represents the direct data transfer from the IoT device to the IoT cloud. To the best of our knowledge, most research, i.e. [51, 22, 23], focuses only on this path to study the contents, patterns, and metadata of IoT network traffic that reveals sensitive information about user activity. However, this type of information does not violate user privacy as the second way (A-C) does.

2. App-to-cloud (A-C): all IoT devices are controlled and configured via their mobile apps [89]; no two IoT devices from two different manufacturers are sharing the same app. For example, a TP-link smart plug is controlled by a mobile application called KASA, while a WeMo smart plug is controlled by a different mobile app called Wemo, See Table 3.1. These mobile apps are recommended by the IoT device manufacturers and installed on the smartphone or tablet to control the IoT device. In a typical scenario, as in Figure 3.2 paths B and C, when a user wants to switch on/off a smart plug, he first needs to log in to the IoT app and then press the switch on/off button. In this case, a command is sent to the smart plug via its app to switch on/off, i.e path A. In parallel, traffic with sensitive PII is sent to the smart plug cloud from the IoT app to inform that the user has logged in to the app and switch on/off the smart plug, i.e. path C.

In this thesis, we focus on collecting and analyzing the data transferred from the IoT app to the IoT cloud (A-C). It is important to highlight that many IoT devices use TLS/SSL when communicating with cloud servers, so the traffic we collect is encryp-

ted. Given the increasing focus on security in the IoT community, we expect that encrypted communications will become standard for smart home devices.

5.3 Attacker Model

We consider a passive network observer who accesses smart home traffic. We assume that our adversary can collect the transport layer traffic of a smart home. Also, we assume that packet contents are encrypted using TLS. This adversary can be the ISP provider, who can collect and store traffic regularly, or, in general, it can be any adversary who knows the SSID and the WPA2 password of the smart home router. Finally, the adversary can get a database of labeled traffic from smart-home devices for training machine learning algorithms. The adversary's goals are the following:

1. Infer the user's interaction(s) with IoT devices in a smart home (e.g., logging into the smart-plug app).
2. Determine whether the transmitted data carries sensitive PII, non-sensitive PII, or non-PII about the user.
3. Determine the type of sensitive PII (e.g., the password for the IoT device app) or non-sensitive PII (e.g., user email) that is being transmitted.

5.4 Methodology

In this section, we give an overview of the IoT-App Privacy Inspector tool (IoT-app PIT) in section 5.4.1. Then, we describe the smart home environment testbed in section 5.4.2.

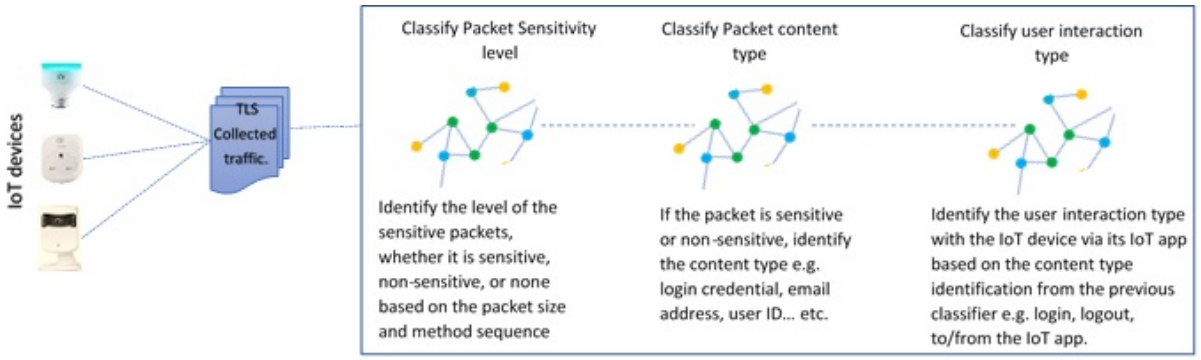

Figure 5.1: IoT-app PIT overview

5.4.1 Overview of the IoT-App Privacy Inspector tool

The IoT-app PIT takes as an input, encrypted traffic collected from different IoT devices. The first classifier classifies the packets according to whether they contain sensitive PII or non-sensitive PII, or none. While, the second classifier classifies the content type of such sensitive PII packets (e.g. carry user location information) or the content type of non-sensitive PII packets (e.g. carry username information). Finally, the third classifier classifies the packets based on the user interaction type with the IoT device (i.e. login, logout, delete a device, change password). Figure 5.1 gives an overview of the proposed tool. We make our tool publicly available at (`https://github.com/Alanoud-Subahi/IoT-app_PrivacyInspector`).

5.4.2 IoT Smart-Home Testbed

The main objective of this testbed is to study the alternative data disclosure, i.e. path C in depth. As we explained in chapter 3.4, we set up a laboratory smart home environment with several commercially available IoT devices to establish the ground truth; see Table 3.1. Details of the network configuration of the testbed and the interaction experiments with the IoT devices are given in section 3.4.1. An overview of our experiments can be seen in Figure 5.2

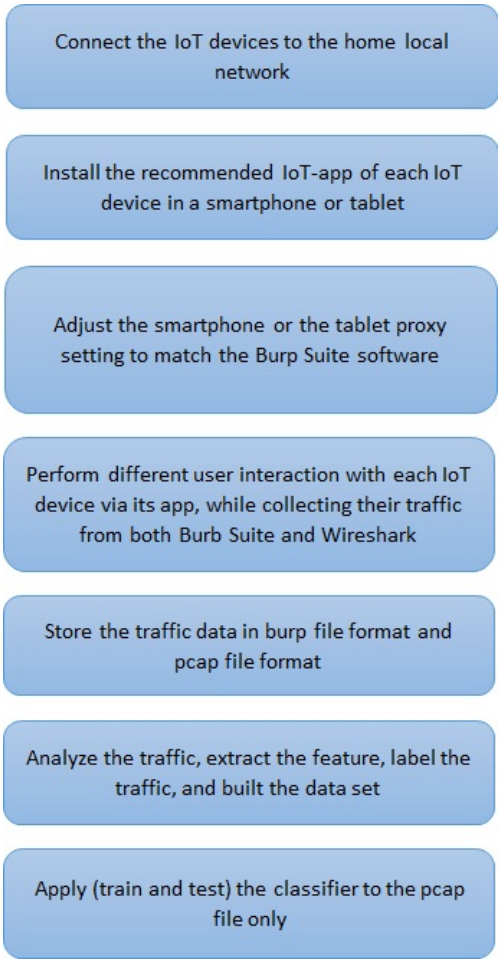

Figure 5.2: Overview of the steps used to collect the encrypted TLS traffic and the encrypted one of the IoT device to establish the ground truth of the IoT-app PIT.

5.5 Attack Design and Implementation

The steps below give a high-level description of our implementation methods for collecting the IoT traffic, which is what the attacker would do:

1. Select IoT devices whose traffic should be classified by the tool.
2. Establish ground truth about user interactions with the IoT devices by doing the following steps:

   A) Collect IoT traffic while performing various interactions with each device

to generate traffic.

B) Analyze the IoT traffic in order to identify the interaction type, the packets containing sensitive PII, non-sensitive PII, and non-PII, and within the PII traffic (both sensitive and non-sensitive) identify the content type (e.g., user credentials or username).

C) Annotate the traffic by labeling each packet with the interaction type that created it.

3. Use the labeled traffic as training data for a classifier to infer the three goals stated above (Section 5.3) from unlabeled/unseen traffic. This point will be explained in detail in Section 5.6.

5.5.1 Activity Inference from Collected Traffic and Identification of Packets Comprising User Interaction, Sensitive PII, and the Content Type of the Sensitive PII

In this section, we present our observations from a passive packet-level analysis of collected traffic from the IoT devices installed on the smart home testbed. As we explain in the Data Collection section 3.4.2, for each of the four interactions with each of the four IoT devices, we collect one encrypted pcap file from the Wireshark and one corresponding decrypted burb file from the Burp Suite. To analyze and therefore identify the type of user interaction with the IoT app, the packet sensitivity level, and the packets that contain personal information along with the type of personal information that they contain. We analyze each pair of burb file and pcap file for each interaction with each IoT device separately.

5.5.1.1 Analyzing the Burp Suite Files

We establish the ground truth about the user's interactions with the IoT devices by analyzing the decrypted traffic we obtain from the Burp Suite file of each IoT app. In particular, we correlate the actions that the user invokes on the IoT device with the packet sizes and sequences that result from these actions.

We find that each IoT app communicates with several domain names associated with the IoT device manufacturer. Interestingly, we realize that each domain name is responsible for certain types of interaction. For example, the KASA and TpCam apps from TP-link communicate with two different domain names, while the NetCam app from Belkin communicates with three different domain names, and finally the LIFX app from Lifx communicates with five different domain names.

Figure 5.3 illustrates an example of the two domain names that the KASA app communicates with, which are api.tplinkra.com and eu-wap.tplinkcloud.com, both owned by TP-link. Each domain name is responsible for a particular set of methods. See Appendix A for the rest of the IoT-apps domain names. For example, each time the user logs in to KASA app, the methods listed in Table 5.1 are executed, always in the same sequence. Each method always generates a request packet from the KASA app to the domain name responsible for this method. It is followed by a response packet from that domain name to the KASA app, with the indicated packet sizes and sequences. In Table 5.2, we observe the sequence and the packet sizes of the methods that are executed when the user logs out from the KASA app, and we see that they are different from Table 5.1. We observe similar differences for the other actions of this and the other IoT devices; see Appendix B. Because these sizes and sequences are unique to each action, an attacker can use them to identify the invoked actions. Also, because each packet in a sequence always contains the same type of information, the attacker can detect the packets that contain sensitive information.

Based on these findings, we conclude that we can rely on the packet sizes, and se-

	Domain Name	Methods	Request packets size in bytes	Response packet size in bytes
Login Action	eu-wap.tplinkcloud.com	login	548	318
	api.tplinkra.com	auth token	315	278
	eu-wap.tplinkcloud.com	postPushInfo	692	178
	api.tplinkra.com	helloIotCloud	1031	435
	api.tplinkra.com	listRules	700	566
	eu-wap.tplinkcloud.com	getDeviceList	415	1143
	api.tplinkra.com	listScenes	768	568
	api.tplinkra.com	isLinked	662	817
	eu-wap.tplinkcloud.com	passthrough	520	873
	api.tplinkra.com	retriveLocation	662	574

Table 5.1: User login interaction with KASA app that controls TP-link smart plug. Methods are always invoked by the app in the order shown – top to bottom ("retrivelocation" is mis-spelled like this in the packet contents). The sizes are of decrypted packets..

	Domain Name	Methods	Request packets size in bytes	Response packet size in bytes
Logout Action	eu-wap.tplinkcloud.com	logout	521	178
	api.tplinkra.com	helloIotCloud	888	427
	api.tplinkra.com	isLinked	542	772
	api.tplinkra.com	retriveLocation	542	380
	eu-wap.tplinkcloud.com	helloCloud	546	204

Table 5.2: User logout interaction with KASA app that controls TP-link smart plug. Methods are always invoked by the app in the order shown – top to bottom. The sizes are of decrypted packets..

quences to infer whether the user interaction with the IoT app is login, logout, and so forth. Furthermore, we manage to identify the length of every packet that sends to or receives from the IoT cloud any PII (e.g. user location, username and password). For example, we can confirm that any packet sent by the KASA app with a packet size of 520 bytes and a received size of 873 bytes from the TP-link domain name

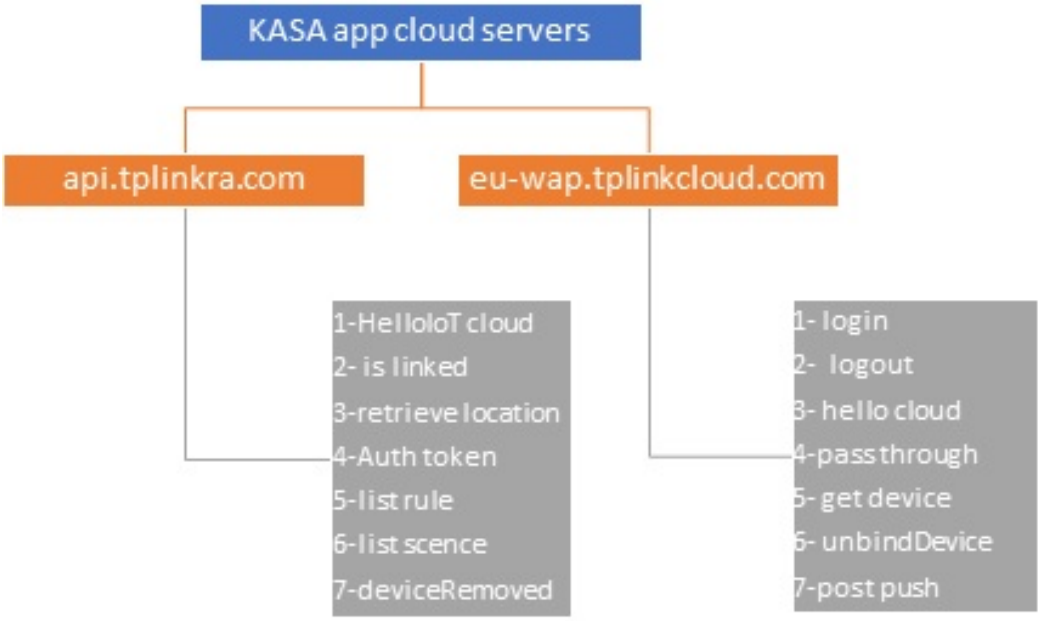

Figure 5.3: TP-link smart plug domain names that KASA app communicates with. Each domain responsible for specific methods.

eu-wap.tplinkcloud.com, is the passthrough method. This method is always triggered when the user logs in to the KASA application, and it carries information regarding the user's geographical location, see Figure 5.4; similarly for the remaining methods.

In some cases, we notice that the packet sizes do vary across executions of a method. This variation is small and thus does not affect our classification negatively, but it can reveal additional information. For example, the size of the request packet for the login method, see Table 5.1, is always 542 bytes plus the length of the user's password. This means that the password length is only 6 bytes in this example. From a security perspective, this is an important finding because the attacker can determine the password length, and therefore determine whether a brute force attack is feasible to obtain the password. Note that this attack can be done offline, so any measures on the IoT cloud side to block repeated failed password submissions would not help.

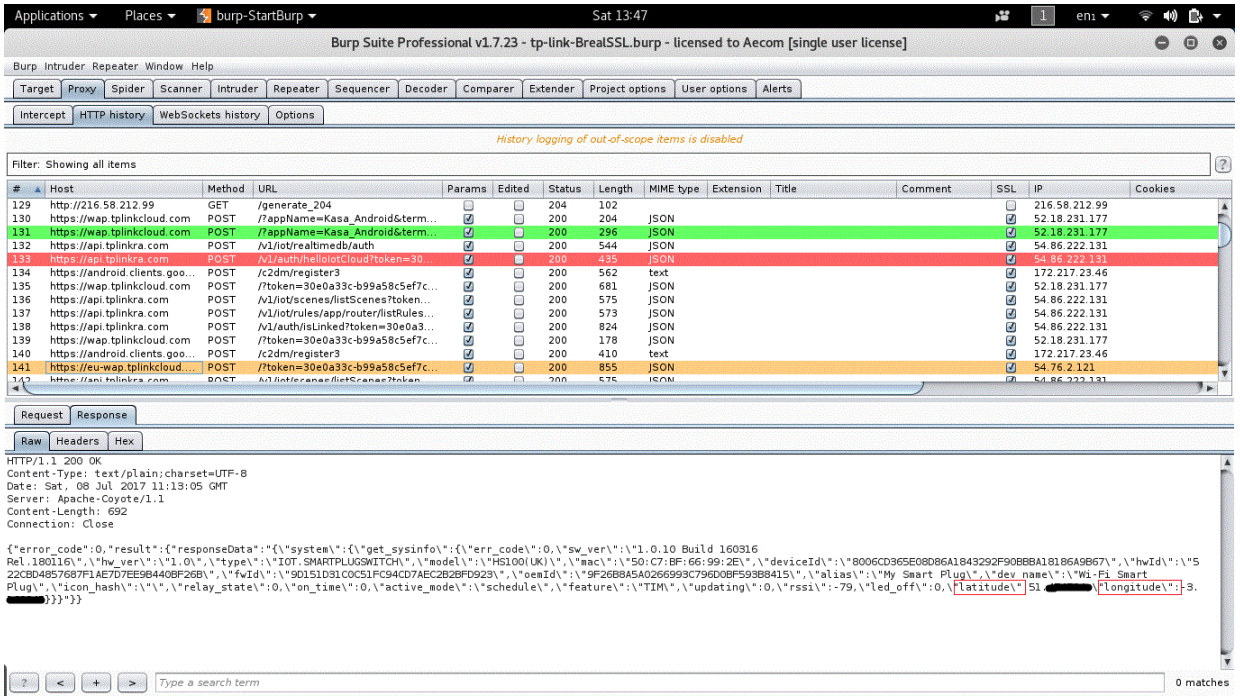

Figure 5.4: Screen shot from Burp Suite showing user's exact location (latitude and longitude).

5.5.1.2 Analyzing the Wireshark file

We now aim to match the encrypted packets from the Wireshark file to the equivalent decrypted packets from the Burp Suite file. We can then label each packet of the encrypted traffic and use this labeled traffic to train our machine learning classifier. The most straightforward way to do this match would be to match encrypted packets to decrypted packets of the same size. However, the sizes of encrypted and decrypted packets are not similar, so we design a new method to find this match. We believe that this new method will be the cornerstone of helping other researchers on how to analyze the IoT payload data.

We apply our method to all actions of the IoT apps. We describe this method in the steps below, in which we aim to match encrypted-decrypted packets for the logout action in KASA app as an example.

1. First, we filter the pcap file to keep only the packets whose source IP address belongs to the smartphone that has the IoT app, and whose destination IP address belongs to one of the two IoT domains of the KASA app, see Table 5.2.

2. Then, in the pcap file, we look for a sequence of encrypted packets whose source and destination IP addresses match the corresponding sequence in the methods of the logout action in the decrypted packets from the burp file. For example, the user logout action from the KASA app triggers five methods. Therefore, in the pcap file, we expect to find the same five methods in the same order. As we can see in Table 5.2, the first method in the logout action is the logout method, which communicates with the eu-wap.tplinkcloud.com server, followed by the second method helloIotCloud, which in turn communicates with the pi.tplinkra.com server and so on for the rest of the methods. Therefore, we should find in the pcap file the same domain names in the same order. As we mentioned earlier, each domain name is responsible for specific methods. By finding the same sequence of the domain names, we can prove that we have found the correct expected method.

3. After identifying the correct method, we now want to match the actual packets. We compare the request and the response packet size of the logout methods from the pcap file with the response and the request packet size of the equivalent logout methods from the burp file. We find that encryption always adds a constant number of bytes to the plain packet size:

    - The size of the encrypted packet for the logout method request is equal to the decrypted packet size plus 148 bytes (decrypted: 521 bytes; encrypted: 669 bytes).

    - Similarly, for the response traffic, the encrypted packet size is equal to the decrypted packet size plus 95 bytes (decrypted: 178 bytes; encrypted: 273 bytes).

We observe the same constants (148 bytes for request packets and 95 bytes for the response packets) for all packets of the KASA app. We link this constant to the type of cipher suite that KASA app use, which is TLS-ECDHE-RSA-WITH-AES-128-GCM-SHA256. Other apps also exhibit the same behavior, only with different additive constants for their request and response packet sizes, because they have different cipher suite. For example, the netcam app uses the TLS-RSA-WITH-AES-128-CBC-SHA cipher suite.

4. Finally, as a visual verification step that we match the correct packets, we create a plot per decrypted action and a corresponding plot per encrypted action. By comparing the two plots, we find that they are equivalent. Figure 5.5 illustrates the logout action and the method sizes and sequences from the burp file from the KASA app. After applying our method, we find the same methods with the same order in the pcap file, as you can see in Figure 5.6. Note the packet sizes are for encrypted packets. The plots for the rest of the actions can be found in Appendix C.

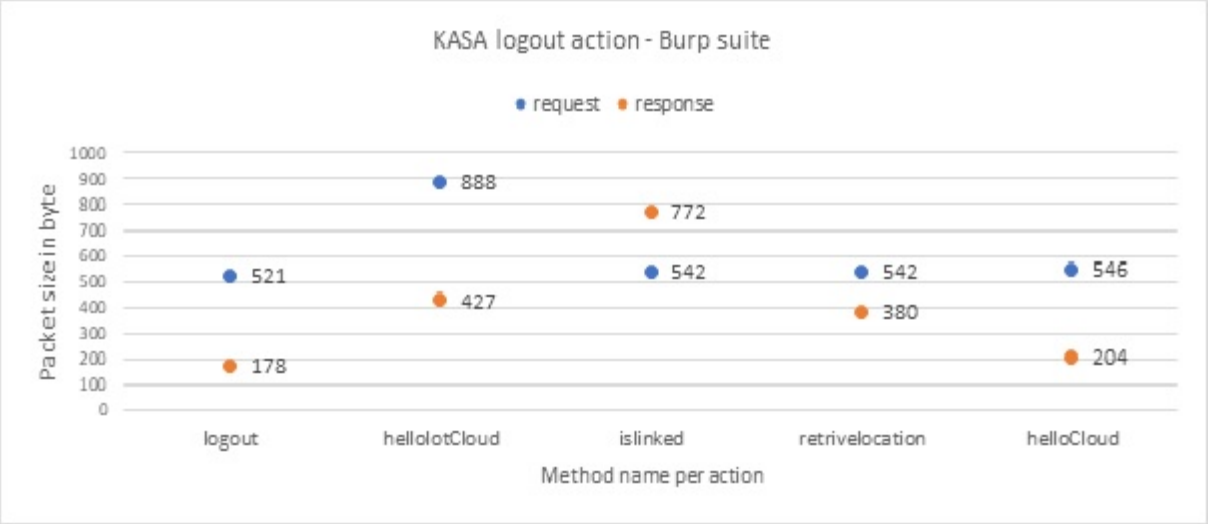

Figure 5.5: User logout interaction from KASA in decrypted format

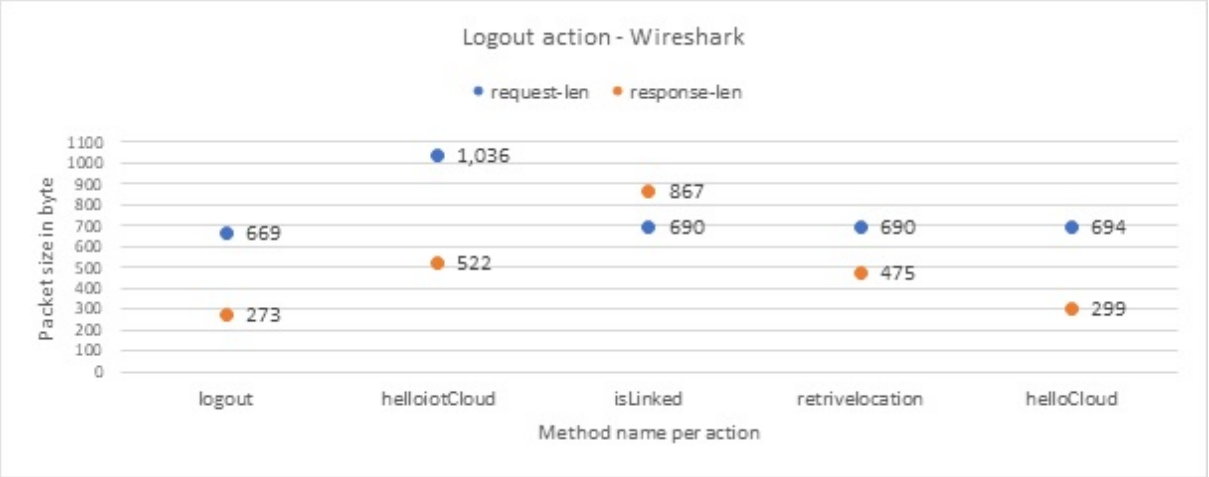

Figure 5.6: Equivalent user logout interaction from KASA in encrypted format

5.5.1.3 Feature selection and Data labeling

During this stage, we compose all packets that are transmitted between the same pair (IP-src, IP-dst) to a group of sessions. Next, we select the most important features that help us manually label all the encrypted session according to the following categories:

1. the user interaction with the IoT device that the packet is part of;
2. whether they contain sensitive information;
3. the content type of the packets that contain sensitive information.

These features are the following:

1. IP-src: refers to the IP address of the smartphone running the IoT app;
2. IP-dst: refers to the IP address of the IoT app domain.
3. Comm-type: refer to which domain name the IoT app communicates with (e.g. KASA app communicate with two domains, so if the IP-src belongs to the smartphone and the IP-dst belongs to the second domain name, then the comm-type set to 1.2);

4. Req-len: refers to the length of the sending packet (from the IP-src to the IP-dst);

5. Resp-len: refers to the length of the receiving packet (from the IP-dst to the IP-src).

We label the sessions in three different ways, thus creating three different datasets. Each one is used to train and test one classifier, see Figure 5.7. For the first dataset, named IoT-interactionType, we label the packets according to the interaction type between the user and the IoT app with either "Login," "Logout," "Change Password," "Delete," or "None." For the second dataset, named IoT-PII, we label the packets according to their sensitivity level with either "Sensitive PII," "Non-sensitive PII," or "None." For the third dataset called IoT-user-PIItype, we label the sensitive packets (sensitive PII or non-sensitive PII) according to their content type with either "User credentials," "User location," "username," or "None."

Once an adversary creates or obtains such labeled traffic for the IoT devices of his choice, he can create a classifier to identify packet streams pertaining to a specific IoT device. Then, he can infer a specific user interaction in unlabeled traffic. Therefore, he will be able to infer the packets that carry sensitive information and the content type of this sensitive information. In the next section, we describe the design of the classifiers.

5.6 Machine Learning-Based Classification

We treat the tasks of identifying user interaction type, packet sensitivity level, and sensitive data type as a multi-class classification problem. Accordingly, six classifiers were selected based on their ability to support multi-class classification.

To evaluate the performance of the selected algorithms and hence choose the best classifier for our problem, we apply several measures. The most common measures are

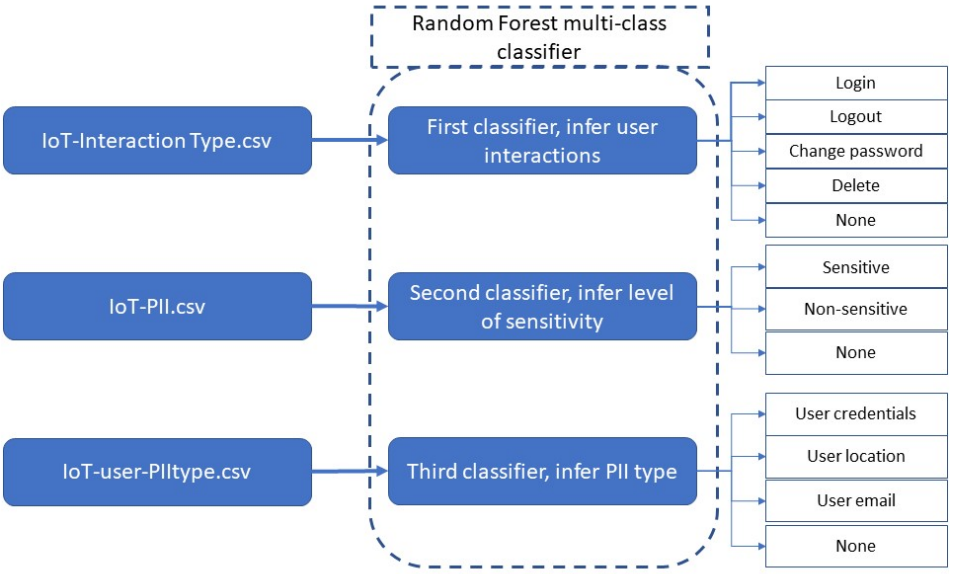

Figure 5.7: Overview architecture of the multi-class classifier

precision, recall, F-mean, and accuracy. As an example, the first multi-class classification problem is evaluated relative to the training dataset, producing the following four outputs:

- **True positive (TP)**– packets are predicted as a sensitive PII, when they are truly sensitive PII.
- **True negative (TN)**– packets are predicted as a None when they are truly None.
- **False positive (FP)**– packets are predicted as sensitive PII, when they are truly None.
- **False negative (FP)**– packets are predicted as None when they are truly sensitive PII.

Precision (P) measures the ratio of the packets that were correctly labeled as sensitive PII to the total packets that are truly sensitive PII *[Precision = TP/(TP+FP)]*. Recall

(R) measures the ratio of the packets that were correctly labeled as sensitive PII to the total of all packets *[Recall = TP/(TP+FN)]*. F-measure (F) takes both false positives and false negatives into account by calculates precision and recall. Then, it provides a single weighted metric to evaluate the overall classification performance *[F1 Score = 2*(Recall * Precision) / (Recall + Precision)]*. Accuracy measures the ratio of the packets that were correctly predicted to the total packets number of the packets *[Accuracy = (TP+TN)/(TP+FP+FN+TN)]*. However, using accuracy to measure the performance of a classifier is a problem. This is because if the classifier always infers a particular class, it will achieve high accuracy, which makes it useless when it comes to building such a classifier.

The goal is to maximize all measures, which range from 0 to 1, to achieve better classification performance. Table 5.3 illustrates the overall results based on previous measurements. As we can see, the Random forest exhibits the best performance across all six classifiers. Therefore, we develop our classification tool based on the Random Forest classifiers. To support our choice, a recent survey on ML methods for security [31] discusses the advantages of using Random Forest. Their study is related to our research as it combines decision-tree induction with ensemble learning; these advantages are:

1. Very fast when classifying input data
2. Resilient to over-fitting.
3. It takes a few input parameters.
4. The variance decreases as per the increment of tree numbers, excluding any biased results.

	Packet Sensitivity type Classifier				Packet Content type Classifier				Interaction type Classifier			
Classifier	P	R	F	Time	P	R	F	Time	P	R	F	Time
Decision Tree	97.1	97.1	97.1	0.093	97.1	97.1	97.1	0.088	97.1	97.1	97.1	0.072
Naive Bayes	74.0	42.1	38.8	0.043	65.3	42.05	37.6	0.035	61	51.1	49.1	0.041
K Nearest Neighbor	98.5	98.5	98.5	0.161	98.5	98.5	98.5	0.159	97.7	97.7	97.7	0.189
Multi-Layer Perception	54.2	73.6	62.4	0.873	1	71.4	83.3	1.206	52.7	72.6	84.1	1.501
Support Vector Machine	96.1	95.4	95.6	125.165	95.7	94.8	95	179.739	93.5	92.9	93.5	166.316
Random Forest	**99.8**	**99.8**	**99.8**	**0.35**	**99.8**	**99.8**	**99.8**	**0.35**	**99.4**	**99.4**	**99.4**	**0.35**

Table 5.3: The results of all selected classifiers based on the most common measurement; precision, recall, and F-mean .

5.6.1 Multi-class Classifier Training

In order to perform our classification experiments, we randomly split each dataset described in section 3.4.2 into 80% for training, and the remaining 20% for testing. Notice that each classifier applies to one dataset; see Figure 5.7. Each classifier is responsible for inferring the possible label of one category. As we can see in Table 5.3, the Random Forest classifier achieves the best performance resulting in 99.8%, 99.8%, and 99.8% in the first and the second classifier, while it achieves 99.4%, 99.4%, and 99.4% in the third classifier for the measurements of precision, recall, and F-mean score, respectively. Additionally, the time taken to classify the tasks for each classifier is 0.35 seconds.

To validate that the classifier does not over-fit, we perform several experiments:

5.6.1.1 10-fold cross-validation

To determine the optimal hyperparameters of the Random Forest algorithm [30], [3], we try many different combinations using GridSearch algorithm optimization. Based on the results, we set our hyperparameters as follows: the number of n-estimator is

10, min-samples-leaf is 3, bootstrap is "False", min-samples-split is 8, criterion is "entropy", max-features is "auto", and the max depth is 90.

5.6.1.2 Confusion Matrix

To get a better understanding of the performance of the classifier across the experiments, the confusion matrices of the three classifiers in Tables 5.4, 5.5, and 5.6 consecutively show the predicted classes for individual packets compare against the actual ones. Every confusion matrix is a synopsis of inferring the outcome of one multi-classification problem, which demonstrates the process in which our classification model is confused upon making an inference. Then correct and incorrect inference numbers are summarized through count values and decoded to each class. The individual confusion matrix gives us an in-depth look into errors being made by a classifier and mainly focuses on the sort of errors being made. For example, in Table 5.4, the confusion matrix, which is related to inferring the user interaction, shows that the actual number of the Delete interaction sessions is 284. However, the classifier infers correctly 281 sessions as a Delete interaction, while it infers incorrectly two packets as Logout interaction and one packet as No-action. These results confirm the high accuracy and reliability of our classifiers.

		Predicted Labels				
		Delete	Login	Logout	Modify Password	No-action
True Labels	Delete	281	0	2	0	1
	Login	0	655	7	0	2
	Logout	0	0	207	0	6
	Modify Password	0	0	1	233	2
	No-action	9	0	3	0	3694

Table 5.4: Confusion matrix of the first classifier which is responsible to infer the user interaction. Rows show the actual class of a repetition and columns show the classifier's prediction.

		Predicted Labels		
		Non	Non-sensitive	Sensitive
True Labels	Non	3693	6	4
	Non-sensitive	5	699	0
	Sensitive	0	0	696

Table 5.5: Confusion matrix of the second classifier which is responsible to infer the sensitivity level of the packet. Rows show the actual class of a repetition and columns show the classifier's prediction.

		Predicted Labels				
		Non	Credential	Location	Location+Credential	User name
True Labels	Non	3643	1	0	0	6
	Credential	0	457	0	0	1
	Location	1	0	126	0	0
	Location+Credential	0	0	0	92	0
	User name	6	0	1	0	769

Table 5.6: Confusion matrix of the third classifier which is responsible to infer the type of the sensitive packet. Rows show the actual class of a repetition and columns show the classifier's prediction.

5.6.1.3 Compare the accuracy of the training dataset with the accuracy of the testing dataset

The training accuracy is the accuracy of the classifier on the training dataset, while the testing accuracy is the accuracy of the classifier on the testing dataset. If the accuracy of the training data is almost similar to the accuracy of the testing dataset, then there is no over-fitting issue; otherwise, we have an over-fitting issue. Table 5.7 shows that the accuracy of the training dataset and the accuracy of the testing dataset are very similar in all of the three classifiers.

As a result of the previous experiments, we conclude that the IoT-app PIT does not fall into the over-fitting problem.

	Packet Sensitivity Type Classifier	Packet Content Type Classifier	Interaction Type Classifier
Train accuracy	99.9%	99.9%	99.7%
Test accuracy	99.8%	99.8%	99.4%

Table 5.7: The accuracy of the training data and the testing data among the three classifiers.

5.7 Results and Discussion

In this section, we first give an overview of the steps of the IoT-app PIT, as described in section 5.7.1. Then the results of the performance of our tool are discussed in detail in section 5.7.2.

5.7.1 Overview of the steps of the IoT-app Privacy Inspector

The steps of the IoT-app PIT are outlined in Figure 5.8. At first, the tool receives collected unseen IoT traffic in a pcap file format. Next, it extracts the relevant features from the pcap file, as mentioned earlier (section 5.5.1.3). Three different classifiers will be applied to this dataset. Each one is used for different inferences (Figure 5.7).

5.7.2 Evaluate the performance of the IoT-app PIT

To evaluate the performance of our tool, we apply the trained classifiers to unseen datasets. We collect such datasets in section 3.4.2 in order to validate the classifiers. Notice that we did not include the validation dataset in the original dataset used to train our classifiers. Accordingly, we conduct two types of evaluations to evaluate the accuracy and reliability of the IoT-app PIT as following:

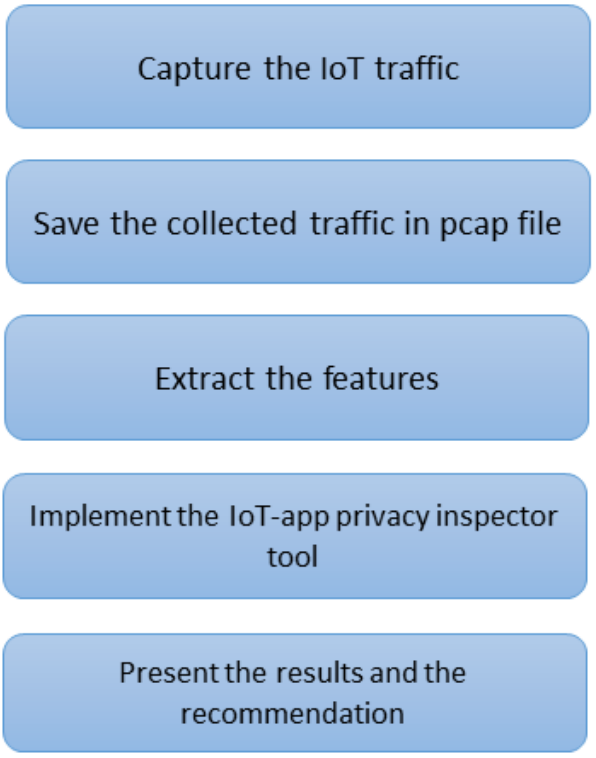

Figure 5.8: An overview of IoT-app PIT for IoT app user interaction type identification; identification of sensitive packet, and content type of sensitive packet identification..

5.7.2.1 Classification accuracy for each IoT app interaction separately

In the first evaluation experiment, we test the tool on each IoT device individually (one IoT device each time). For each IoT device, we apply the tool four times, on a collected dataset for each interaction Login, Logout, Delete, and Change Password. Thus, we apply the tool 16 times in total.

The results show that in every experiment, the tool infers the correct class. We summarize and group the results from the sixteen experiments according to each IoT app in Table 5.8. Each row represents one user interaction and the output of the IoT-app inspector tool (the three classifiers). For example, in the first row, the IoT-app inspector tool accurately infers that when the user logs into to KASA app, only sensitive PII

packets are sent to the IoT cloud. The type of these sensitive packets is user credentials and user location.

In Table 5.9, we compare the results of all user interactions with all IoT devices. Our findings show that most interactions are similar in terms of sending sensitive PII or non-sensitive PII packets to their IoT cloud. However, we highlight three important things. First, the change-password interaction and the login interaction send both sensitive PII and non-sensitive PII packets to the IoT cloud from the Lifx app. This means that the Lifx app excessively sends sensitive PII packets about their user to the Lifx cloud through these two interactions. Second, logout interaction from the netcam app doesn't send any type of sensitive packets to its IoT cloud, which makes it the safest interaction among the others. Finally, the delete interaction and the logout interaction of KASA, TpCam, and Lifx send only non-sensitive PII packets to its IoT cloud. Hence, these two interactions are seen to be the interactions that least send sensitive PII packets about the user to the IoT cloud.

5.7.2.2 Classification accuracy with mixed IoT interactions in the same file

In the second evaluation experiment, we test the tool four times on each IoT device individually (one IoT device each time). For each IoT device, we apply the tool on mixed user interactions between the IoT app and its IoT device in order to validate the classification accuracy by inferring the previously mentioned aims. The results presented in Table 5.10 demonstrate very high classification accuracy of our three classifiers:

- the average accuracy (number of correctly inferred user interactions divided by the total number of interactions) is 99.4% with F1 score 0.994;
- the average accuracy (number of packets for which the level of sensitivity is correctly inferred divided by the total number of packets) is 99.8% with F1 score 0.998;

	Sensitivity level of the packet			Content type of the sensitive packet		
	User Interaction	Sensitive PII	Non-Sensitive PII	User Credentials	User Location	Username or Email Address
KASA app	Login	✓	x	✓	✓	x
	Logout	x	✓	x	x	✓
	Delete	x	✓	x	x	✓
	Change Password	✓	✓	✓	x	✓
TpCam app	Login	✓	x	✓	✓	x
	Logout	x	✓	x	x	✓
	Delete	x	✓	x	✓	✓
	Change Password	✓	✓	✓	x	✓
Netcam app	Login	✓	x	✓	x	x
	Logout	x	x	x	x	x
	Delete	x	✓	x	x	✓
	Change Password	✓	x	✓	x	x
Lifx app	Login	✓	✓	✓	x	✓
	Logout	x	✓	x	x	✓
	Delete	x	✓	x	x	✓
	Change Password	✓	✓	✓	x	✓

Table 5.8: Summary of the IoT-app PIT results on the IoT apps interactions

- the average accuracy (number of packets for which the content of the sessions correctly inferred divided by the total number of packets) is 99.8% with F1 score 0.998.

As a result of the previous experiments, we prove the validity and reliability of such a tool. We achieve high accuracy for inferring the correct type of sensitive information, as well as for inferring the user interaction type that occurs between the IoT device and the user.

IoT apps	User Interactions	Sensitive PII	Non-Sensitive PII
KASA app	Login	✓	x
	Logout	x	✓
	Delete	x	✓
	Change Password	✓	✓
TpCam app	Login	✓	x
	Logout	x	✓
	Delete	x	✓
	Change Password	✓	✓
NetCam app	Login	✓	x
	Logout	x	x
	Delete	x	✓
	Change Password	✓	x
Lifx app	Login	✓	✓
	Logout	x	✓
	Delete	x	✓
	Change Password	✓	✓

Table 5.9: Comparison between the IoT apps user interactions to find out which IoT app send excessive sensitive PII about their user .

5.8 Summary

In this chapter, we have invented a tool called IoT-app PIT that can automatically infer the following from the IoT encrypted network traffic:

1. The packet(s) that reveals user interaction type with the IoT device via its app (e.g. login).
2. The packet(s) that carry sensitive Personal Identifiable Information (PII).
3. The content type of such sensitive information (e.g. user's location).

IoT-app privacy inspector		Accuracy	F1 score
User Interaction Classifier	Login	99.4	0.994
	Logout		
	Delete		
	Change Password		
Packet Level of Sensitivity Classifier	Sensitive PII	99.8	0.998
	Non-Sensitive PII		
Packet Content Type Classifier	User Credential	99.8	0.998
	User Location		
	User name or Password		

Table 5.10: The Accuracy results of IoT-app privacy inspector of inferring user interaction, packet level of sensitivity, and packet content type.

We use the Random Forest classifier as a supervised machine learning algorithm to extract features from network traffic. To train and test the three different multi-class classifiers, we collect and label network traffic from various IoT devices via their apps. We obtain the following classification accuracy values for the types of information, as mentioned above: 99.4%, 99.8%, and 99.8%. This tool can help IoT users take an active role in protecting their privacy.

Our tool aims to help IoT users by notifying them of any interactions that send excessive personal data to the IoT cloud e.g. when they login to the IoT app. The high accuracy results achieved by our tool prove the reliability of such a tool. Finally, we point out a security problem: It is possible for an attacker to identify the packet(s) that contains the user's password, and thus to launch an offline password cracking attack.

By the end of this chapter, we will have finished the first phase of building the IoT behavior compliance tool. Now, we move onto implementing the second phase, which aims to develop an automated method to read IoT PPA texts and only extract the type

of PII that the IoT manufacturer collects about its users.

Chapter 6

Automated Approach to Analyze IoT Privacy Policies

6.1 Introduction

The goal of this chapter is to introduce a new method of analyzing IoT PPA texts. In particular, we are focusing on determining whether the IoT manufacturers collect PII about the end users, without asking them to read the whole PPA nor highlighting the paragraphs that refer to the data collection practices and then ask to read such paragraphs. In contrast, we aim in our method to mimic how an ordinary person reads and understands such policies sentence by sentence.

Our contribution in this chapter is a tool called IoT-PPA reading, that automatically extracts the type of the user's information that the IoT manufacturer collects while using its IoT devices. The objective of this tool is to save time spent on reading long PPA text as well as reduce the effort on understanding complex and ambiguous meanings hidden in such a text. Such a tool will help end users make rational decisions before using or buying any IoT device based on a prior understanding of the type of collected data. To implement our methods, we use MultinomialNB supervised machine learning algorithm. The high accuracy achieved by the classifier (98.8%) proves its validity and reliability.

The rest of the chapter is organized as follows: Section 6.2 discusses how we collect,

annotate, and select the features from fifty IoT PPAs. While section 6.3 gives an overview of the IoT PPA reading tool, as well as a detailed description of the ten cases used to extract the features from IoT PPA. In section 6.4, we develop our multi-class classifiers to classify the sentences of IoT PPAs based on their sensitivity level. Then, we discuss the results and evaluate performance of the tool in section 6.5. Finally, we provide a summary and conclusion for the chapter in section 6.6.

6.2 Collecting, Annotating, and Extracting the Features from IoT PPA

6.2.1 Collecting IoT PPAs

To perform our analysis, we collect fifty IoT PPA based on the popularity of the IoT manufacturers as well as the popularity of their IoT devices among the end users. As we explained in chapter 3.5.1, we pre process the collected data and remove the unwanted texts to make them ready for annotation and feature extraction process.

6.2.2 Annotation Scheme

To annotate the texts of fifty IoT PPA, we apply two sages. Section 6.2.2.1 describes the first stage, which is manual annotation scheme. While section 6.2.2.2 describes the second stage, which is automated annotation scheme.

6.2.2.1 Manual Annotation Scheme

In this phase, we manually annotate ten out of fifty IoT PPAs. We create four main annotation labels, which are **"Collect", "Sensitive", "Non-sensitive", and "Not-include"**. In addition, we create extra sub-annotations for the last three main annota-

tions. These sub-annotations help us to be more accurate regarding the type of the collected data by the IoT PPA, as per the following explanation:

1. Collect: we label any phrase or word that means "collect user information by the first party", as "Collect". Notice that we only care about the first party collection, which represents the IoT manufacturer.

2. Sensitive: we label any phrase or word that means "user sensitive PII information", such as user location, user login details, or user password information as "Sensitive". Under this annotation, we create three sub-annotations # Location, # Login, # Password. For example, the sentence "we collect user location" is labeled as Collect, Sensitive PII-Location.

3. Non-sensitive: we label any phrase or word that means "user non-sensitive PII information", such as user email address, username, or device information as "non-sensitive". Under this annotation, we create three sub-annotations # Email, # Username, # Device. For example, the sentence "you provide us with your first name" is labeled as Collect, Non-sensitive PII-username.

4. Not-include: under this annotation, we create nine sub-annotations # Negative-words, # Wrong-words, # Share-words, # Third-party, # Cookie-words, # Wrong-credentials, # Wrong-location, # Wrong-email, and # Wrong-name.

Based on the previous annotation scheme, we are ready to label the rest of the IoT PPA automatically, as we explain in the next phase.

6.2.2.2 Automatic Annotation

It is time-consuming if we continue to annotate the rest of the forty IoT PPAs manually; hence, we need to automate the annotation process. To do that, we use a web-based annotating tool called Tagtog [10]. Cejuela et al. [32] illustrated how Tagtog-assisted

annotation could benefit manual and automatic annotation and shows a successful annotation with high accuracy.

To better use this tool, we need first to implement the annotation manually on a few documents, as we explain in section 6.2.2.1. Second, based on such manual annotation scheme, Tagtog will generate a model to automatically annotate the new documents by creating a custom ML model. Figure 6.1 shows the automated annotation process in the Tagtog tool. It is important to emphasize that we manually verified the annotations that Tagtog produced.

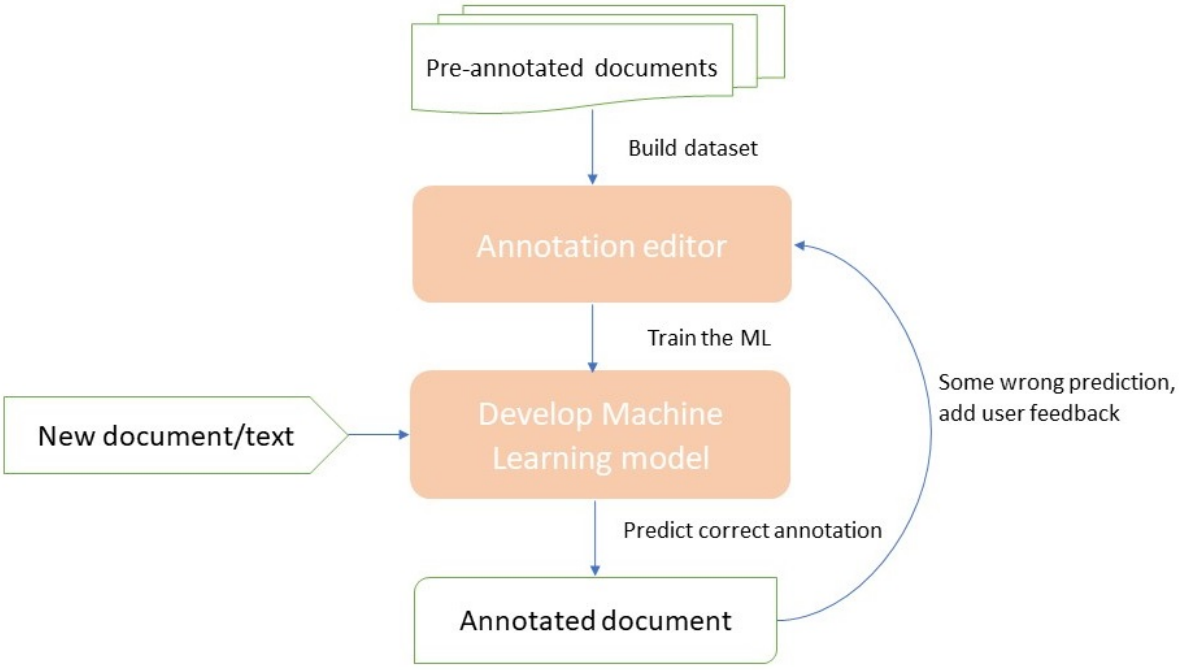

Figure 6.1: The process of how to use Tagtog custom ML to automate the annotation scheme.

6.2.3 Feature selection

After annotating the fifty IoT PPA texts, we extract only the labeled phrases and remove the unlabeled one. As a result, we get 31,661 labeled phrases. For example, the phrase "providing us location" is labeled as "CollectLocation-sensitive", while the phrase "you

may supply us your e-mail" is labeled as "CollectEmail-nonSensitive", and so forth for the rest of the phrases. We use this dataset for training and testing our classifier, as we explain in section 6.4. Moreover, we create five different assistant datasets for our feature extraction rules,i.e. the ten corner cases, as follows:

Dataset#1) includes phrases or keywords that represent negative meaning (neg-K), e.g. "not collect", "we don't collect", "we won't collect".

Dataset#2) includes phrases or keywords that mention a "collect" keyword without implying that any user data is being collected, i.e. wrong collect (wc-K), e.g. "When you access your location", "to provide you with latest update".

Dataset#3) includes phrases or keywords that mention data sharing (share-K), e.g., "when you choose to share your location", "we share your personal information".

Dataset#4) includes phrases or keywords that mention third-party involvement (thirdParty-K), e.g. "we collect your third-party account information".

Dataset#5) includes phrases or keywords that mention cookies collection (cookie-K), e.g. "our cookies store your log in details".

6.3 Methodology

In this section, we give a brief overview of the IoT PPA reading tool in the subsection 6.3.1. Then, we explain in detail how we create and apply ten different cases to help us extract the correct features. Also, we explain how such cases can adversely affect the validity of extracting the results in subsection 6.3.2.

6.3.1 Overview of the IoT-PPA reading tool

Initially, the tool asks the user to provide the URL of the IoT PPA as an input. After that, the tool processes the document to prepare it for features extraction, as explained

in 3.5.2. The results are saved in a CSV file for later prediction. After that, the classifier classifies the sentences of the IoT PPA into one or more of six classes according to whether it collects sensitive PII or non-sensitive PII information, as follows:

1. "CollectLocation-sensitive",
2. "CollectPassword-sensitive",
3. "CollectLogin-sensitive",
4. "CollectEmail-nonSensitive",
5. "CollectUsername-nonSensitive",
6. "CollectDevice-nonSebsitive".

Figure 6.2 gives an overview of the proposed method. We make our tool publicly available at (`https://github.com/Alanoud-Subahi/IoT-PPA_Reading_Tool`).

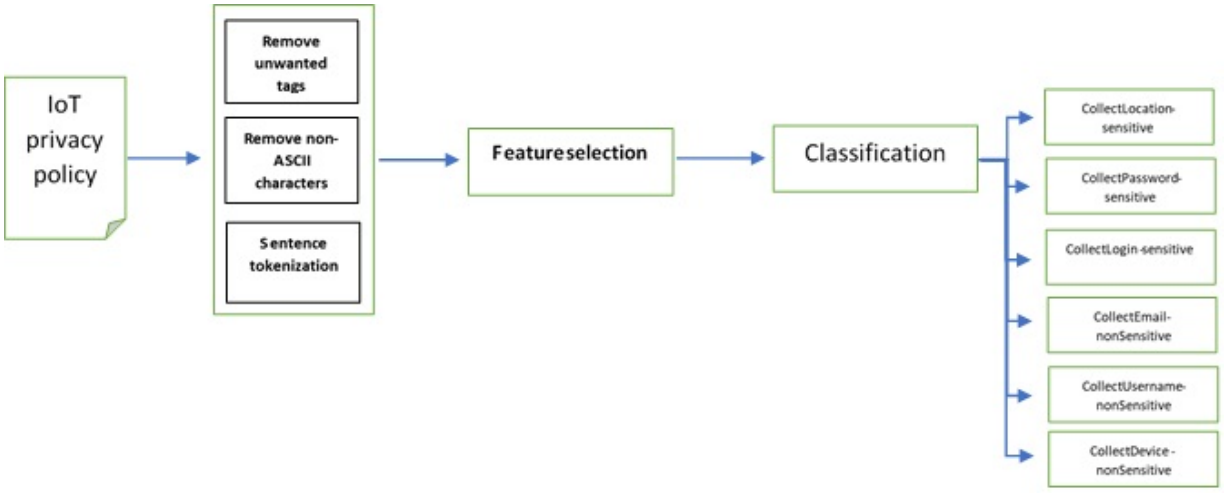

Figure 6.2: Overview of the proposed method of analyzing the IoT privacy policy documents.

6.3.2 Extracting Relevant Features

In this section, we aim to extract from the IoT PPA sentences, whether it contains one or more of the previously mentioned six features. Accordingly, we build six different functions, each of which is responsible for extracting one feature. In our approach, we aim to imitate how a person could understand the meaning of a sentence, i.e. knows whether the sentence collects sensitive PII or non-sensitive PII.

Before we explain our method, we must first clarify that the previous approach to finding out whether a PPA collects personal information or not is **keyword matching**. This method checks whether the text contains any word from the collection keywords such as "collect", "collected", "provide", "provided", ...etc. Also, it checks whether the text contains any word from the PII keywords such as "location", "password", "username", ...etc. Hence, if the **keyword matching** method finds both keywords in the text, then the PPA collects PII about the users. Otherwise, it does not collect any PII about the user. To prove whether such a method is reliable or not, we will test it using three different examples as follows:

Example.1) if we have the sentence, "We collect your personal information such as your geographic location, email address and your device software information." The **keyword match** method will conclude that the sentence collects your location, email address, and device information because it matches the keywords. This is a positive result.

Example.2) if we have the sentence, "We collect your personal information to improve our services", the **keyword match** method will conclude that the sentence dose not collect PII about the user because it only match the "collect" keyword, and there is no word matches the PII keywords. This is a positive result.

Example.3) if we have the sentence, "We will not collect your geographic location", the **keyword match** method will conclude that the sentence collects geographic location. This is false results because the sentence does not collect any PII about

the user. The reason behind this false result is that **keyword matching** method does not consider the impact of the negation words within the sentence.

Consequently, the main objective of our method is to overcome the previous false results and any similar ones due to the ambiguity of the meaning. Thus, we study in-depth all the possible cases that might affect understanding the correct meaning of such a sentence. As a result, we come up with ten different cases, each of which has its own set of rules. These rules depend on two main conditions:

1. The role of the party (i.e if its the manufacturer as a first party or the end user as a second party).

2. The position of the keywords in the sentence (i.e the collect keyword, the sensitive keyword, the negative keyword...etc).

To guarantee that we collect the correct feature(s), We should apply these cases onto each sentence in order. In Figure 6.3, we applied ,in order, the ten cases with its rules to illustrate how we extract the location feature from one sentence. The first case explains how we deal with the negative keyword if its appear in the sentence. While, the second case until the sixth case, we explain how we address the problem of long, ambiguous, and complicated sentences. Finally, from the seventh case until the end, we explain how we treat four different type of ambiguous sentences, which imply hidden meaning of collecting information. We will now discuss each case separately:

Case 1- Negative sentences: The first checkup is to ensure that the sentence does collect the user's information, if so, we continue until we extract the feature(s). Otherwise, we delete the sentence from the list because no information has been collected by the manufacturer. To do so, we need to check whether the sentence contains any negative words (neg-K); if so, we have to identify the position of such keywords in the sentence. The objective of applying this case first is to is to ensure that the sentence

mentioned that it collects user's information, which we need to extract. First, the tool loops through the negative Dataset #1, section 6.2.3, and checks whether any of the phrases exist in the sentence. If so, we apply three different rules to ensure that we extract the correct results; these rules are based on the position of the negative word within the sentence as follows.

- First rule: if the position of the neg-K comes before the position of the sensitive keyword (s-K) and the collect keyword (c-K), then we ignore the sentence. For example, in the sentence, "If you do not wish to have your location recorded while taking a photo, you can turn this off at any time within the camera settings of the device". The negative phrase "you do not" comes first, then the sensitive phrase "your location", then the collect keyword "recorded". Hence, if the rule is (neg-K + s-K + c-K) or (neg-K + c-K + s-K), then we ignore the sentence.

- Second rule: if the position of the neg-K comes in between the s-K or the c-K, then we also ignore the sentence (c-K + neg-K + s-K) or (s-K + neg-K + c-K). For example, in the sentence, "We may ask you not to turn on your location". The negative phrase "not to" comes between the collect keyword "we may ask" and sensitive keyword "your location". If we apply the straightforward method, that we explained earlier, on the previous sentences, the results will give us that the sentences collect the user's location, although the sentence doesn't mean collect your location due to the negation meaning.

- Third rule: if the position of the neg-K comes after the s-K and the c-K, then we are sure that we extract the correct feature. For example, in this sentence, "This location data is collected anonymously in a form that does not personally identify you", the sensitive keyword "location" comes first, then the collect phrase "is collected", then the negative phrase "does not". Thus, if we apply the rule (s-K + c-K + neg-K) or (c-K+ s-K + neg-K), we are sure that we extract the correct feature. Notice that it doesn't matter if the c-K comes before or after the s-K.

Case 2- Long and complicated sentences (combination of wrong collect keywords, third-party keywords, and share keywords): In this case, we study the first type of complicated sentences, which include a combination of, wrong collect keyword (wc-K), third-party keyword (thirdParty-K), and share keyword (share-K). For example, we have this long and complicated sentence, after processing the privacy policy of Ring manufacturer for smart doorbell[1], *"The types of personal information we obtain include: Contact information, such as name, phone number, and email; Account information, such as online password and other log-in details used to access Ring products and services; Payment information, such as name, card number, expiration date and security code, which is collected and stored by our third-party payment processor on our behalf; Information we obtain from third-party social media services (e.g., Facebook) or payment services (e.g., PayPal) if you choose to link to, create or log into your Ring account through these services (including when you share Ring videos or content via your social media account); Information we obtain from third-party business partners if you choose to use our Ring+ Service, such as your account ID, account name and email address."*

Initially, the average reader can be confused in understanding the type of information that the sentence collects and who is responsible for collecting it. In fact, a sentence like this is too long and complicated, so the user cannot immediately understand it. However, by careful reading, we can infer the following information:

1. The manufacturer of Ring obtains personal information such as password and log in details, which consider sensitive PII, as well as information such as name, phone, and email, which consider non-sensitive from the user.

2. On behalf of Ring, a third-party payment processor collects payment information from the user, such as username, card number and expiration date, and security code.

[1]https://en-uk.ring.com/pages/privacy-notice

3. Only if the user chooses to log in to her Ring account through third-party social services such as Facebook Ring will obtain her personal information, such as login details.

4. If the user chooses to share her video information via social media such as Facebook, Ring will obtain this video information from the user.

The user is only concerned about the type of personal information the IoT manufacturer collects about him, i.e. the first point only. Hence, we build our tool to handle these long and complicated sentences in order to help users understand the meaning of such complicated sentences. First, the tool checks if any word from the wc-K and any word from the thirdParty-K and any word from the share-K exists in the sentence. We have already built our datasets during the analysis stage (Dataset #2, #3, #4 in section 6.2.3). If we find all the words, we create a list that contains the index of each word within the sentence. After that, we divide the sentence into partitions based on these indices. For the example of the sentence above, the keywords that we find are **"to access"**, **"third-party"**, and **"you share"**. Hence, the new sub sentences of the previous sentence are the following:

1. "The types of personal information we obtain include: Contact information, such as name, phone number, and email; Account information, such as online password and other log-in details used to access."

2. "Ring products and services; Payment information, such as name, card number, expiration date and security code, which is collected and stored by our third-party."

3. "payment processor on our behalf; Information we obtain from third-party."

4. "social media services (e.g., Facebook) or payment services (e.g., PayPal) if you choose to link to, create or log into your Ring account through these services (including when you share."

5. "Ring videos or content via your social media account); Information we obtain from third-party."

6. "business partners if you choose to use our Ring+ Service, such as your account ID, account name, and email address."

To guarantee that our tool extracts the correct features, we apply the following rules on each partition.

- The first rule is related to the wrong collect keyword. If any of the sub-sentences include either this rule (c-K + s-K + wc-K) or this rule (s-K + c-K + wc-K), then we collect the feature. Otherwise we ignore the sentence.

- The second rule is related to the third-party keyword. If any of the sub-sentences include either this rule (c-K + s-K + thirdParty-K) or this rule (s-K +c-K + thirdParty-K), then we collect the feature. Otherwise we ignore the sentence.

- The third rule is related to the share keywords. If any of the sub-sentence include either this rule (c-K + s-K + share-K) or (s-K + c-K + share-K),then we collect the feature. Otherwise we ignore the sentence.

By applying these three rules, we come up with the same results we previously inferred from the sentence, i.e. the first point. The results: "we obtain name", "we obtain email", "we obtain password", and "we obtain login".

Case 3- Long and complicated sentences (combination of wrong collect keywords, Cookies keywords, and share keywords): Case 3 is similar to Case 2. The only difference is that we search for a cookie keyword (cookie-K) instead of a third-party keyword (thirdParty-K). For example, we have this long and complicated sentence, after processing the privacy policy of Google home manufacturer[2], *"Examples of how*

[2]https://policies.google.com/privacy

we use your information to deliver our services include: We use the IP address assigned to your device to send you the data you requested, such as loading a YouTube video; We use unique identifiers stored in cookies on your device to help us authenticate you as the person who should have access to your Google Account; Photos and videos you upload to Google Photos are used to help you create albums, animations, and other creations that you can share."

By careful reading, we infer from the sentence that Google home manufacturer doesn't collect any personal information. Hence, the purpose of our tool is to give us the same result. Therefore, we apply the same rules related to the wrong collect keyword and share keyword as before (6.3.2). Moreover, we apply further rules related to the cookie keywords, which are either (c-K + s-K + cookie-K) or (s-K + c-K + cookie-K). As a result, we conclude that the previous sentence does not collect any personal information, which is similar to what we infer manually.

Case 4- Long and complicated sentences (a combination of wrong collect keywords, and share keywords): In this case, we study the second type of complicated sentence, which only includes a combination of wrong collect keyword and share keyword. For example, we have this sentence, after processing the privacy policy of Ezviz manufacturer[3], *"When you save and share content through EZVIZ Services, like video clips, live video streams, images, captions, and comments ("Your Content"), for other individuals to access, we will collect information to allow you to save and share your content, such as your email address and email address of the person you elect to share your content with."*

By careful reading, we infer from the sentence that EZVIZ manufacturer doesn't collect any personal information. We address this case just like **Case 2 and Case 3**. We divide the sentence into partitions based on the index of the wc-K and share-K. By applying the same rules related to the wc-K and share-K mentioned in (6.3.2), we con-

[3]https://www.ezvizlife.com/uk/legal/privacy-policy

clude that the sentence does not collect any personal information from the user.

Cases 5 -Long and complicated sentences (a combination of wrong collect keywords, and third-party keywords): Case 5 is similar to Case 4, however, the sentences include only a combination of wrong collect keywords and third-party keywords. For example, *" Information we collect 1.1 Information We obtain About You Contact information, such as name, phone number, and email and postal address; Account information, such as online password and other log-in details used to access Neos products and services; Payment information, such as name, billing address and payment card details, including card number, expiration date and security code, which is collected and stored by our third- party payment processor on our behalf"*.

By careful reading, we infer from the sentence that manufacturer collects contact information, name, email, and user address. We address this case just like **Case 2 and Case 3**. We divide the sentence into partitions based on the index of the wc-K and thirdParty-K. By applying the related rules mentioned in (6.3.2), we conclude that the sentence does not collect any personal information from the user.

Case 6 -Long and complicated sentences (a combination of wrong collect keywords, and cookie keywords): Similar to case 4 and 5, the sentences in this case include only a combination of wrong collect keywords and cookie keywords. For example, *"When you access our Sites, you automatically provide certain information from and about your computer or mobile device, including the activities you perform on our Sites, the type of hardware and software you are using (for example, your operating system or browser), information stored in cookies, IP address, access times, the web pages from which you came, the regions from which you navigate the web page, and the web page(s) you access (as applicable)."*

By careful reading, we infer from the sentence that manufacturer collects device information and the user address. We address this case just like **Case 2 and Case 3**. We

divide the sentence into partitions based on the index of the wc-K and cookie-K. By applying the related rules mentioned in (6.3.2), we conclude that the sentence does not collect any personal information from the user.

Cases 7, 8, 9, and 10 with single keyword These cases are about ambiguous sentences which contain at least one keyword. As mentioned earlier, we have already built a dataset of all possible phrases that include third-party keywords, share keyword, wrong collect keywords, and cookie keywords, during the analysis stage (Dataset #2, #3, #4, #5 in section 6.2.3). We now explain each case separately:

Case 7) In this case,the tool checks whether the meaning of the sentence implies collecting personal information by third-party. Hence, we apply three different rules as follows to ensure that we extract the correct results.

- The first rule: if the position of the third-party-K comes between s-K and the c-K, then we collect the feature i.e. (s-K + thirdParty-K + c-K) or (c-K + thirdParty-K + s-K). For example, "we collect and use information obtained from Facebook, Google, Amazon, and other accounts you use to log in to the Services ("third-party Accounts"), such as your name, birth date, picture, and other details you may have on your account profile".
- The second rule: if the position of the thirdParty-K comes after the s-K and the c-K, then we collect the feature i.e. (c-K + s-K + thirdParty-K) or (s-K + c-K + thirdParty-K). For example, "we collect your email, or log in for a third-party account (like Facebook), to create an online or application account ("Account") or subscribe to our communications".
- The third rule: if the position of the thirdParty-K comes first then the c-K then the s-K, we ignore the sentence i.e. (thirdParty-K + c-K + s-K) or (thirdParty-K + s-K + c-K). For example, "When you purchase LIFX Products through the LIFX Website, our third-party provider will collect,

your first and last name, email address, shipping and billing address, and complete credit card information or bank account information".

Case 8) In this case, the tool checks whether the meaning of the sentence implies collecting personal information for share purposes. In this case, we apply two different rules:

- The first rule: if the position of the share-K comes after the s-K and the c-K, then we collect the feature i.e. (c-K + s-K + share-K) or (s-K + c-K + share-K). For example, "we will collect information about your exact location when you choose to share that with us and motion information from the motion sensors in your Hive products that detect movement in your home."
- The second rule: if the position of the share-K comes before the s-K and the c-K, then we ignore the sentence i.e. (share-K + s-K + c-K) or (share-K + c-K + s-K). For example, "The share information also includes the information related to you shared by other users who use the services of Mobvoi including collect location data and log information".

Case 9) In this case, the tool checks whether the meaning of the sentence implies collecting personal information when it actually didn't collect any personal information. Hence, we apply three different rules:

- The first rule: if the position of the wc-K comes after the s-K and the c-K, then we collect the feature i.e. (c-K + s-K + wc-K) or (s-K + c-K + wc-K). For example, "We collect information that your Device sends out or receives to tailor the Services to our users in different regions, such as: geo-location, IP addresses, and external hardware information from your Device about surrounding Wi-Fi access points, beacons, and cell towers".
- The second rule: if the position of the wc-K comes between the s-K and the c-K, then we ignore the sentence i.e. (c-K + wc-K + s-K) or (s-K +

wc-K + c-K). For example, "Include fulfilling orders for products or services, delivering packages, sending postal mail and e-mail, removing repetitive information from customer lists, analyzing data, providing marketing assistance, providing search results and links (including paid listings and links), processing payments, transmitting content, and providing customer service."

- The third rule: if the position of the wc-K comes before the s-K and the c-K, then we ignore the sentence i.e. (wc-K + c-K + s-K) or (wc-K + s-K + c-K). For example, "You can access your information, including your name, address, payment options, profile information, and order history in the "Account" section of the website."

Case 10) In this case, the tool checks whether the meaning of the sentence implies collecting personal information by cookie. Hence, we apply three different rules:

- The first rule: if the position of the cookie-K comes after the s-K and the c-K, then we collect the feature i.e. (c-K + s-K + cookie-K) or (s-K + c-K + cookie-K). For example, "Other information collected automatically through the foregoing means may include your IP address, location details, cookie information, mobile device, operating system, the type of browser, demographic information, application and/or device(s) you're using to access our Services, and other indicators of how you are interacting with the Services."
- The second rule: if the position of the cookie-K comes between the s-K and the c-K, then we ignore the sentence, i.e. (c-K + cookie-K + s-K) or (s-K + cookie-K + c-K). For example, "We treat information collected by cookies and other technologies as non personal information, except where Internet Protocol (IP) addresses or similar identifiers are considered personal information by local laws."
- The third rule: if the position of the cookie-K comes before the s-K and the

c-K, then we ignore the sentence, i.e. (cookie-k + c-K + s-K) or (cookie-K + s-K + c-K). For example, "We use cookies, small text files which, for example, are stored temporarily on your computer system for a shopping basket or for the OSRAM log in and which your browser stores."

After applying all the ten corner cases, in order, onto each sentence, we are sure that our tool extracts the correct features.

6.4 Machine Learning-Based Classification

To solve our classification problem, we compare several popular classification algorithms from different literature. The work done by [18] supports the popularity rank of our selected algorithms to solve similar problems like ours. Accordingly, we train five machine learning models, i.e. Decision Tree, Linear Support Vector Machines, Random Forest, Multinomial Naive Bayes, and Multi-Layer Perception to classify IoT PPA texts based on (a) whether it collects sensitive PII or non-sensitive PII as well as (b) the type of such PII. To do this, we use the dataset that we have already created during the analysis stage (section 6.2.3). We randomly split the dataset into 60% for training, 20% for validation, and 20% for testing and evaluating the performance of our tool, see section 6.5.

We train each of these classification algorithms using the training dataset, and we evaluate them with the following four metrics:

- **True positive (TP)** – the number of sentences that are sensitive and are correctly predicted as sensitive.
- **False positive (FP)** – the number of sentences that are non-sensitive but are falsely predicted as sensitive.

- **True negative (TN)** – the number of sentences that are non-sensitive and are correctly predicted as non-sensitive.
- **False negative (FP)** – the number of sentences that are sensitive but are falsely predicted as non-sensitive.

As is standard in the literature from these four metrics, we calculate three more: precision, recall, and F-measure. Precision (P) is the fraction of the sentences that are correctly labeled as sensitive among all sentences that are labeled sensitive by the classifier *[Precision = TP/ (TP + FP)]*. Recall (R) is the fraction of the sentences that are correctly labeled as sensitive among all sentences *[Recall = TP/ (TP + FN)]*. F-measure (F) calculates precision and recall; it takes both false positives and false negatives into consideration to evaluate the overall classification performance *[F1Score = 2 *(Recall * Precision)/ (Recall + Precision)]*. Accuracy calculates the fraction of the sentences that are predicted correctly to the total number of sentences *[Accuracy = (TP + TN)/ (TP + FP + FN + TN)]*.

Based on the results of the previous measurements, shown in Table 6.1, we find that all the classifiers achieve high accuracy. However, to select the best classifier, we compare the time efficiency to accomplish the task of each classifier. Hence, Multinomial Naive Bayes classifier achieves the best performance resulting in 97.4%, 97.4%, and 97.5%, respectively. Besides, it achieves the shortest time in performing the task with 0.16 seconds for 18997 sentences.

To evaluate the classifier and to ensure that it avoids over-fitting problems, we perform the following experiments:

Confusion matrix experiments To better understand the performance of the selected classifier, we create confusion matrices of the classifier in Table 6.2. The predicted label of the individual sentence appears in the columns while the actual label appears in the rows. The goal of using a confusion matrix is to look deeply into errors made

	Common Measures			
Classifier	P	R	F	time
Decision Tree	98.1%	98.1%	98.1%	0.70
Multi-Layer Perception	98.9%	98.9%	98.9%	5.5
Support Vector Machine	98.2%	98%	98%	68.8
Random Forest	98.4%	98.4%	98.4%	1.07
MultinomialNB	97.5%	97.4%	97.4%	0.16

Table 6.1: The results of all selected classifiers based on the most common measurement; precision, recall, and F1-score.

by a classifier as it focuses mainly on the sort of errors being made. For example, we can see from the confusion matrix that the actual number of the sentences that collect password information (the fifth row) is 542. However, the classifier correctly predicts 498 sentences as collectPassword-sensitive; in contrast, it predicts incorrectly that 44 sentences are collectLogin-sensitive. The overall results confirm that our classifier achieves high accuracy, and we can rely on such a classifier to classify the IoT privacy policy.

Compare the accuracy of the training dataset with the accuracy of the validation dataset One of the methods that we use to ensure whether we have an over-fitting issue or not is comparing the accuracy of the validating dataset with the accuracy of the training dataset. As we can see in Table 6.3, both results are very similar; hence we conclude that there is no over-fitting.

10-fold cross-validation The best way to determine optimal values of hyperparameters is through GridSearchCV over possible parameter values using k-fold cross-validation on different random subsets of our labeled dataset. We use k = 10 where a random (k-1)/k fraction of the dataset is used to train the classifier, and the remaining $1/k$ are tested for accuracy. Based on the results we set our hyperparameters as

	Predicted labels					
	cDevice-nS	cEmail-nS	cLocation-s	cLogin-s	cPassword-s	cUsername-nS
cDevice-nS	1533	0	0	0	0	0
cEmail-nS	0	453	0	0	0	0
cLocation-s	0	0	2019	0	1	0
cLogin-s	0	0	0	572	117	0
cPassword-s	0	0	0	44	498	0
cUsername-nS	0	0	0	1	0	1094

Table 6.2: Confusion matrix of the Multinomial classifier. Rows show the actual class of repetition and columns show the classifier's prediction. Row and column titles have been abbreviated using "c" for "collect," "s" for "sensitive," and "nS" for "nonSensitive.".

	Multinomial classifier
Train accuracy	97.57%
Validation accuracy	97.42%

Table 6.3: The accuracy of the training data and the validating data

follows: alpha = 1.0, fit-prior = True, and class-prior = None.

The results of the previous experiments prove that our classifier doesn't fall in over-fitting problems.

6.5 Results and Discussions

To evaluate the performance of our tool, we apply the trained classifier to 20% of the test dataset (i.e. 6,332 of unseen sentences). The results show that the classifier classifies the unseen sentences correctly with high accuracy equal to 98.8%. As a result, we prove the validity of our tool to infer whether the IoT PPA collects sensitive or non-sensitive information about the end user.

6.6 Summary

In this chapter, we describe our methods in analyzing and extracting PII information from the IoT PPA texts to inform the end users of the type of collected information.

We build a multi-class classifier tool to read IoT PPAs, sentence by sentence. Then, the tool extracts from the sentences the correct feature(s) with high accuracy (98.8%), and high speed in accomplish such tasks i.e. 0.16 sec to classify18997 unseen sentences. We study in-depth, long, complicated, and ambiguous sentences that average users won't be able to understand. As a result of this study, we come up with the most ten corner cases that affect the way of understanding the correct meaning of the sentence, which hasn't been addressed in the literature before. We study each case separately and create a set of rules for this particular case. The main goal is to extract from each case the correct type of sensitive PII and non-sensitive PII that the IoT manufacturer collects wile using their IoT devices.

In contrast to other research, we save the IoT user's time and effort by only give him the relevant information without highlighting the paragraphs or shorten the length of such long text. The limitation of the previous methods is that they leave it to the user to try understanding the hidden and ambiguous meaning of such paragraphs, which we overcome in our method.

By the end of this chapter, we will have finished the second and the last phase of building the IoT behavior compliance tool. In the next chapter, we will accomplish building the tool by creating an environment that combines phases one and two to finally evaluate the level of compliance between the actual behavior of the IoT device with its PPA.

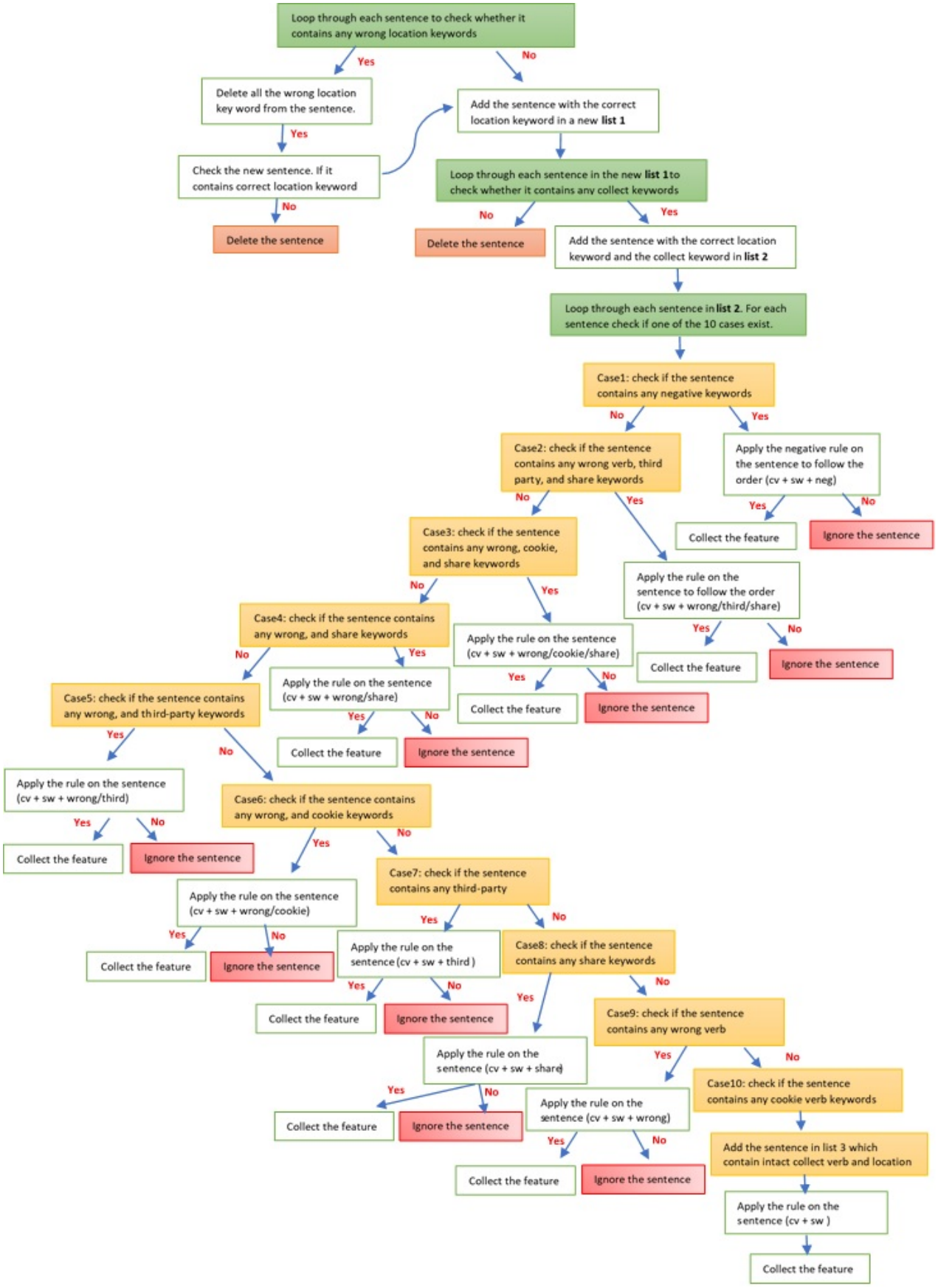

Figure 6.3: An Example of how we apply the ten corner cases to extract location feature.

Chapter 7

IoT Behavior Compliance

7.1 Introduction

In this chapter, we introduce our innovative tool, which is the IoT behavior compliance tool. This tool combines and executes two different tools, i.e., phase one and phase two, respectively. Each of these tools has its own type of inputs and outputs data. Next, The final results from both tools will be compared and evaluated to conclude the compliance level of the IoT devices.

The rest of the chapter is organized as follows: Section 7.2 provides an overview of the IoT behavior compliance tool. While Section 7.3 demonstrates a case study scenario to evaluate the compliance of an IoT device with its PPA. Finally, we summarize the chapter in Section 7.4.

7.2 Overview of the IoT behavior compliance tool

As we mention in chapter 1.5, the main contribution of this thesis is, to evaluate the compliance level of the actual behavior of IoT devices with their PPA. To do that, we need do the following steps, as Figure 1.2 shows:

1. Monitor the IoT traffic to analyze its behavior to infer the data type transferred from the IoT device via its app to its manufacturer's cloud server, see chapter 6

2. Analyze the PPA text of the IoT manufacturer presented on their website in order to extract the exact types of PII that been collected about the users, see chapter 6

3. Compare both results from the second point and the third point, which we will do in the current chapter.

In order to run the IoT Behavior Compliance tool, the user should already have a pcap file(s) contains collected traffic of his interaction with IoT device(s), saved somewhere in his smartphone or tablet device.

As we mentioned above, the IoT behavior compliance tool consists of two tools working, respectively. When the user run the IoT behavior compliance tool, it starts by executing the IoT-app PIT to infer the behavior a selected IoT device, i.e. the type of the transferred data, see chapter 5 for more detail. Next, the IoT-PPA reading tool executes by getting the PPA URL of such an IoT device's manufacturer to read and extract the type(s) of data collected by such an IoT manufacturer, see chapter 6 for more detail. Finally, the IoT behavior compliance tool will process the results from both tools to investigate whether the data type transferred to the IoT manufacturer's cloud stated in its respective PPA or not. If so, then the actual behavior of the IoT device complies with its PPA. Otherwise, there are compliance issues related to this particular IoT device with its manufacturer. In both cases, we present to the IoT end user the final results, and he/she can act accordingly. See Figure 7.1. Our tool is publicly available at (`https://github.com/Alanoud-Subahi/Evaluating_IoT_behavior_compliance`).

7.3 Case study: Evaluate the Tp-link smart plug

In this case study, we have already collected the encrypted traffic between the end user and the tp-link smart plug device in section 3.4.2. Hence, we apply the IoT behavior compliance tool to evaluate the compliance of the Tp-link smart plug with its PPA.

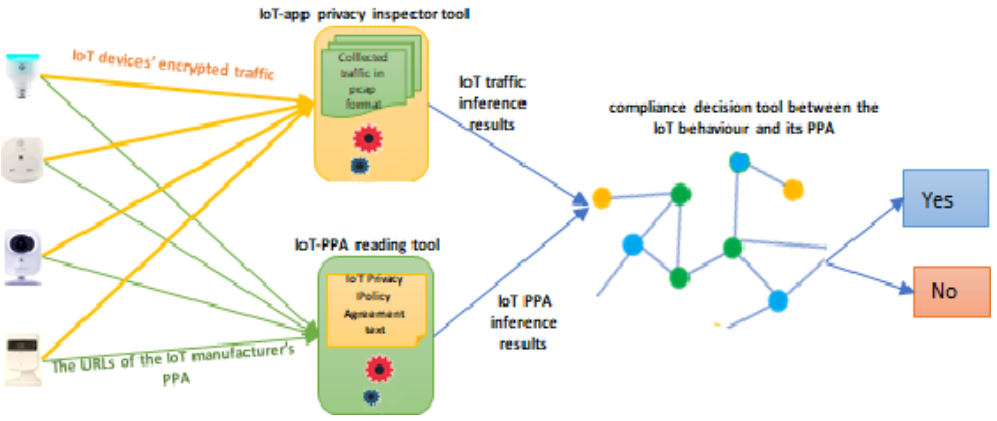

Figure 7.1: Overview of the IoT behavior compliance tool

At first, when running the the IoT behavior compliance tool, a welcome screen appears to the end user and briefly explains the purpose of using the IoT behavior compliance tool. See Figure 7.2

After that, the user is asked to specify three things, as following:

1. Select the IoT device that he wants to evaluate its compliance,
2. The path in which the collected encrypted traffic file (.pcap file) is stored,
3. The path where the results will be saved.

Once the user assigns everything, see Figure 7.3, the IoT-app PIT and the IoT-PPA reading tool will execute in the background simultaneously. As a result, two main information will appear to the user on the screen. First, the type of interaction(s) that the IoT end user made with the smart plug. In this particular scenario, the IoT user

```
Welcome to the IoT compliance evaluation tool.
This tool aims to evaluate the level of the compliance of the actual behavior of your IoT device with its privacy policy agreement.
First, the tool infers the data type that transferred from the IoT device to the IoT cloud. Then, it reads the privacy policy agreement of such IoT device.
Finaly, it compares the results to check wheather the actual behavior of the IoT device match with what stated in its privacy policy.
To proceed please type (Y/y), otherwise press (N/n) to exit.
your answer is: y
```

Figure 7.2: A welcome screen appears when running the IoT behaviour compliance tool.

```
Welcome to the IoT compliance evaluation tool.
This tool aims to evaluate the level of the compliance of the actual behavior of your IoT device with its privacy policy agreement.
First, the tool infers the data type that transferred from the IoT device to the IoT cloud. Then, it reads the privacy policy agreement of such IoT device
Finaly, it compares the results to check wheather the actual behavior of the IoT device match with what stated in its privacy policy.
To proceed please type (Y/y), otherwise press (N/n) to exit.
your answer is: y

Please select the IoT device that you want to evaluate its compliance:
1- Tp-link SmartCam.
2- Tp-link Smart Plug.
3- Belkin NetCam.
4- LIFX smart lamp.

your answer is: 2

Please, provide the full path where your IoT file (.pcapng) locate (folder_name1\..\folder_nameN\name.pcapng):
.pcapng file: ~/Downloads/IoT-Devices/Tp-link/Smart_plug/tpLink_1.pcapng
Please, provide the full path where your want to store your database and the final results: (folder_name1\..\folder_nameN):
dataset destintation: ~/Downloads/IoT-Devices/Tp-link/Smart_plug/
```

Figure 7.3: The user selections to specify the IoT devices and the encrypted pcap file for the evaluation.

```
Welcome to the IoT compliance evaluation tool.
This tool aims to evaluate the level of the compliance of the actual behavior of your IoT device with its privacy policy agreement.
First, the tool infers the data type that transferred from the IoT device to the IoT cloud. Then, it reads the privacy policy agreement of such IoT device.
Finaly, it compares the results to check wheather the actual behavior of the IoT device match with what stated in its privacy policy.
To proceed please type (Y/y), otherwise press (N/n) to exit.
your answer is: y

Please select the IoT device that you want to evaluate its compliance:
1- Tp-link SmartCam.
2- Tp-link Smart Plug.
3- Belkin NetCam.
4- LIFX smart lamp.

your answer is: 2

Please, provide the full path where your IoT file (.pcapng) locate (folder_name1\..\folder_nameN\name.pcapng):
.pcapng file: ~/Downloads/IoT-Devices/Tp-link/Smart_plug/tpLink_1.pcapng
Please, provide the full path where your want to store your database and the final results: (folder_name1\..\folder_nameN):
dataset destintation: ~/Downloads/IoT-Devices/Tp-link/Smart_plug/
Running as user "root" and group "root". This could be dangerous.
Your dataset was created and ready for the evaluation.....


The user interaction type with the IoT-app is/are :  ['user Logout from the IoT-app', 'user Login to the IoT-app']
```

Figure 7.4: The first result of applying the IoT behaviour compliance tool

made two interactions: he logged in, then he logged out from the KASA app. See Figure 7.4

The second result, is a table, demonstrating the following, see Figure 7.5:

1. the first column of the table shows various data types that might be transferred to the IoT cloud from the IoT device. The first three types are sensitive PII (i.e. location, login, password), while the last three types are non-sensitive PII (i.e. username, email, device information).

2. the second column of the table presents the results of executing the IoT-app PIT. The tool will assign 1 in front of the data type that has been sent to the IoT cloud; otherwise, it will assign 0.

The evaluation results are:

	IoT-app Inspector	IoT Privacy Plicy Agreement	Compliance decision
Collect Location	1	1	Yes
Collect Login	1	1	Yes
Collect Password	1	1	Yes
Collect Username	1	1	Yes
Collect Email	1	1	Yes
Collect Device Information	1	1	Yes

Figure 7.5: The final results of applying the IoT behaviour compliance on the tp-link smart plug.

In our scenario and for the two interactions mentioned above, particular types of PII have been sent to the IoT cloud, i.e., user location, user login information, user password details, user username, user email, and the user's device information.

3. the third column of the table presents the results of executing the IoT-PPA reading tool. The tool works similarly to the previous one. If the IoT manufacturer's PPA collects the data type, then it will assign 1; otherwise, it will assign 0.
   In this typical scenario, the Tp-link smart plug PPA states that it collects all the data types mentioned in the second point, i.e., the IoT-app PIT.

4. the last column of the table is the compliance decision of evaluating the actual behavior of the IoT device with its PPA. For example, if the result for a particular data type is **yes**, then the data sent to the IoT cloud matches what is stated in the IoT PPA. Otherwise, if the result assigns in this column for a particular data type is **no**, then the user has a compliance issue between the behavior of the IoT device with its PPA. Figure 7.5, illustrates that in this scenario, we have no compliance issues.

However, it is important to highlight that the results vary not only among the IoT devices, but also among the user interaction(s) with the same IoT device. For example, in a different scenario, we have collected encrypted traffic between a user who deleted the Tp-link smart plug from its app, i.e. the KASA app. After applying the IoT behaviour compliance tool on the collected traffic, we found that there is a compliance

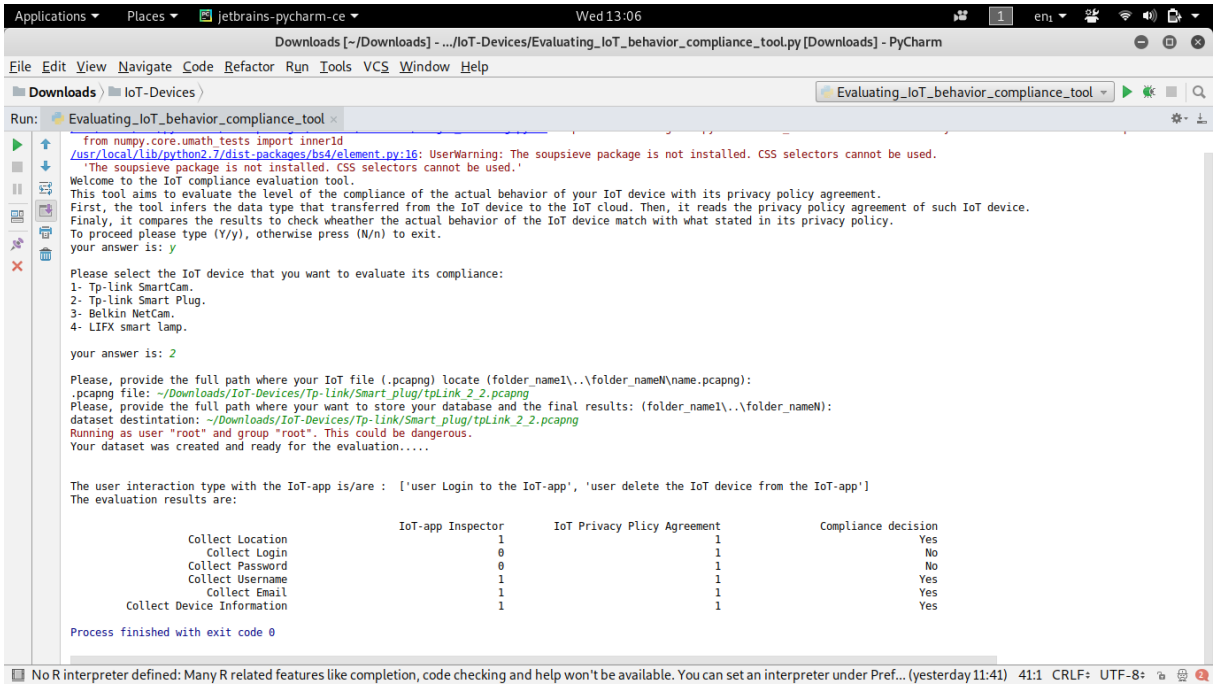

Figure 7.6: Evaluate the level of compliance of the Tp-link smart plug with its PPA- "Delete the IoT device" interaction.

issue between the data sent to the Tp-link cloud and what is stated in the its PPA. See Figure 7.6.

We apply the IoT behavior compliance tool to the rest of the four IoT devices. For each device, we have a several collected files for various interactions. See the remaining results in Appendix D.

7.4 Summary

By the end of this chapter, we fulfill our goal in building our novel tool, which aims to evaluate the level of compliance between the actual data sent to the IoT cloud with what is stated in its PPA. The tool consists of two parts, each of which is responsible for different tasks. The first part is for inferring, from encrypted network traffic, the behavior of an IoT device, while the second part is to extract the PII collected by the

IoT manufacturer's PPA.

The tool works by combining and comparing the results coming from both parts to investigate whether there is compliance between them. If the evaluation result of a particular data type is yes, this means that there is no compliance issue. Otherwise, we have a compliance issue.

Chapter 8

Conclusions and Future Work

8.1 Introduction

This chapter provides a summary of our work conducted through this thesis. First, it recaps the contributions of this thesis. Then, it summarizes the main findings and the work done in the previous chapters. Next, it answers the research questions by linking them to the chapters that aim to solve and address such questions. Finally, it highlights some suggestions for possible future work.

8.2 Thesis Summary and Contributions

Recently, IoT has become an extension of the physical world. It provides an opportunity to transform traditional devices into smart devices capable of communicating with other smart devices or with the cloud server, affecting every aspect of our daily life. Although these devices have spread rapidly with unlimited possibilities to facilitate our lives, they are highly vulnerable to security and privacy breaches. Therefore, these attacks and security and privacy threats need to be addressed in depth.

In this thesis, we attempted to explore and discuss a different type of security and privacy breaches, which has not been addressed before. These types of violations are related to compliance issues associated with the actual data transferred from the IoT device to its cloud and with what is stated in the IoT PPA. Consequently, we developed

a novel IoT tool to automatically evaluate the level of compliance between the actual behavior of the IoT device with its PPA by using network analytic and text mining methodologies.

We began our thesis by introducing a general background about the IoT devices; then, we highlighted the main problem that this thesis attempted to solve as well as the motivation behind this study. Moreover, we presented the research hypothesis, the research questions, and the contributions of this thesis in chapter 1. We developed several methods to test and support this hypothesis, which presented mainly in chapters 4,5,6, and 7.

In Chapter 2, we defined the main terminologies that the readers need to understand in this thesis. Also, we provided a broad literature review focusing on IoT privacy and security testbeds, and monitoring and analyzing IoT traffic. Moreover, we summarized the previous work related to the difficulties of reading and understanding the complex meaning of PPAs in general, and the one related to the IoT PPA devices in particular. While in chapter 3, we discussed the methods applied to collect and process the datasets used in this thesis.

Chapter 4 explained in detail the issues related to the current IoT PPA and why it's essential to update the current IoT privacy law. After analyzing the language used in several IoT PPAs, we have established the eight privacy criteria that any IoT manufacturer must apply to preserve the IoT end user's privacy. Also, we investigated whether there is a compliance issue between the actual behavior of the IoT device and its PPA presented in its manufacturer website. To do that, we set up a smart home testbed to collect the IoT traffic, as explained in this chapter. After analyzing the payload of the collected traffic, we compared, manually, the data sent to the IoT cloud with what the IoT manufacturer stated in its PPA about the type of data they collect from the IoT device. Interestingly, the results prove that most of the IoT manufacturers don't comply with what they stated in their PPA, as explained in chapter 4.4.1.1.

In Chapter 5, we illustrated the different methods in which the IoT device can com-

municate and send the user's data to its cloud server. Moreover, we proved that any passive observer could infer critical information by analyzing the pattern of IoT traffic, which could violate the end user's security and privacy. Consequently, we developed a novel tool called the IoT-app PIT to read and interpret the encrypted IoT traffic. After that, the tool will inform the user whether the traffic transferred to the IoT cloud carries sensitive PII or non-sensitive PII. Also, it will tell the IoT end user with the type of such PII (e.g. user credentials or user email), as well as the type of interaction(s) that the user made with such IoT device(s).

In Chapter 6, we proposed a novel method in text mining in order to read long and complicated PPA texts. In this method, we only ask the user to provide the URL link of the PPA that he/she want to analyze. Then, the tool will read and extract only the types of sensitive PII and non-sensitive PII that such PPA collects about their users automatically, we called this tool the IoT-PPA reading tool. Our tool differ from others in that it is not asking the user to read the whole text, nor it highlights several paragraphs and asks the user to read them. In contrast, our novel tool read and understand the ambiguous texts and the hidden meanings to present to the end user the information that he needs to know.

Following that, in chapter 7, we developed the IoT behavior compliance tool by combining two different tools i.e. the IoT-app PIT (Chapter 5) and the IoT-PPA reading tool (Chapter 6). The objective of this novel combination is, to compare the results coming from two different sources, which are the IoT traffic and the IoT PPA text. As a result, the tool will present to the IoT users the final decision whether the transferred data from an IoT device complies with what is stated in the manufacturer's PPA of such a device or not.

8.3 Research Questions Answered

In this section, we repeat the research questions identified previously in Chapter 1.4 and answer them as follows:

- **Research Question 1:** *Is the data sent from the IoT device limited to an identified purpose of their PPA? If so, do the IoT end users know what type of information is being sent about them?*

  To answer this question, chapter 4 proved that most of the IoT devices send information about their users without specifying the reasons behind such a process, i.e. explain the reason in their PPA. In addition, most IoT end users don't know that they are sharing their information in the first place, nor what type of data is being sent about them. Moreover, this chapter pointed out the existence of a compliance issue between the actual behavior of two different IoT devices (Tp-link smart plug and Belkin NetCam) with its PPA, which has not been addressed before.

- **Research Question 2:** *Can the encrypted traffic of the IoT device expose sensitive PII about their end users? If so, can we know the type of such information sent from the IoT device to its cloud?*

  In chapter 5, we explained that analyzing the pattern of the IoT traffic, even if it's encrypted, as well as investigating the plain text protocols (e.g., TCP/IP headers, TLS handshakes) of an IoT device might violate end user's privacy. Also, we proved that any eavesdropper could infer the activity type of an IoT device, as well as the type of user's sensitive PII being sent to the IoT cloud (e.g., user location) by passively monitoring such traffic.

- **Research Question 3:** *Can an automated text mining mechanism help IoT end users avoid reading long and complicated IoT PPA text to know whether such PPA collects sensitive PII about them, and knowing the type of such information?*

In chapter 6, we answer this question by proposing a text mining tool called IoT-PPA reading tool. This tool reads and extracts the type of information that is collected by the IoT manufacturers while using their IoT devices, without asking the end users to read or understand such long and vague texts.

- **Research Question 4:** *Can we automatically inform the IoT end users whether the data send from an IoT device complies with its PPA?*

  We seek to assist IoT end users in maintaining their privacy and let them make the right decision in terms of using IoT devices. Therefore we attempt to inform the IoT end user to what extent their IoT device(s) complies with its PPA. To do this, we need to proceed in two stages: first, read the IoT traffic to determine the actual data type sent to its cloud. Second, read the IoT PPA text to decide which data type the IoT manufacturer collects about its IoT end users. Then compare the results to inform the end users about the compliance decision regarding their IoT device(s).
  We implemented the first tool in chapter 5, while we implemented the second one in chapter 6. To combine the two tools, in chapter 7, we developed a method to utilize the previously mentioned tools to work as one tool. The outcome of this combination aims to evaluate the compliance level of the actual behavior of the IoT device with its PPA. Then it presents the results to the IoT end user to inform him if there is any compliance issues or privacy violations.

8.4 Future Directions

In this section, we describe some techniques in which the research of this thesis can be extended further in the future.

The novel mechanisms for evaluating the level of compliance between the IoT actual behavior with its PPA outlined in this thesis can be improved and refined based on real-world deployment scenarios. Some of the key refinements are summarized below:

- Our tool aims to detect whether there are any compliance issues; then, it informs the IoT end users with the mismatched data types, i.e. **compliance issue detection method**. However, in this thesis, we have not implemented any mechanism to prevent such compliance issues, i.e. **prevention compliance issue method.** Thus, in the future, it will be useful to improve the tool to not only **detect** compliance issues but also to implement methods to **prevent** these issues. For example, if the tool detects that an IoT device's traffic sends user location while its IoT PPA doesn't state such a process. The tool can prevent this by either dropping that packet or by adding extra padding. By doing this, the attacker won't recognize the traffic that carries sensitive PII about the user. Furthermore, we can notify the IoT manufacture to work out such issues from their side.

- In this thesis, we analyzed the packet length and the traffic sequence as well as interpreted the payload of the IoT traffic using two steps running in parallel. First, we collected encrypted IoT traffic using Wireshark. Then we used Burp Suite to decrypt such traffic.

  In the future, we can shorten and automate this process by using a certificate designed explicitly for our tool. Thus, the tool will be able to collect encrypted traffic, decrypt the traffic, analyze the payload of the traffic, and identify the traffic that carry sensitive PII to the IoT cloud, automatically in one step. For example, if any IoT user wants to download our tool to evaluate the compliance of his/her IoT device(s), he will be notified that our certificate will be automatically downloaded in his smartphone or tablet as part of the downloading steps.

- We developed a text-mining tool to solve issues around reading long and ambiguous IoT PPA texts. This method focused on identifying and extracting only the type of PII that the IoT PPA collects about its users and informing them of such information.

  In the future, this tool can be improved to inform the users whether such collected information is being shared with third parties. Also, we can design a graph

to illustrate how much PII the IoT manufacturer collects about its users. For instance, we can set up three different color coding, each of which represents the amount of data that the IoT manufacturer accumulates from its end users. If the graph of a particular IoT PPA, e.g. is red color, then the IoT manufacturer collects too much PII about its users, which they don't need to. Similarly, if the color of the graph, e.g. is yellow, then the IoT manufacturer collects lots of PII about its users, which they need it. In contrast, if the color is green, then the manufacturer collects reasonable PII about its users. We can identify the allowed amount of PII that any manufacturer can collect based on the GDPR [45].

8.5 Summary

In general, the research conducted, and the results obtained throughout this thesis demonstrate the importance of preserving the privacy of IoT end users not only through data encryption but also through adherence to the GDPR law regarding user data protection regulations. Users' sensitive data should not be sent in any case if there is no reason for this. In addition, it is important to adhere to what is stated in the IoT manufacturer's PPA with regard to the type of data collected by their IoT device(s).

The bottom line and the take-away message is that this thesis, in general, and the IoT behavior compliance tool, in particular, contribute to helping two main stakeholders:

1. IoT end users to obtain the decision regarding the privacy compliance of their IoT device(s).

2. IoT manufacturers to investigate the compliance of their IoT device(s) before launching new IoT products. Also, they can design their IoT PPA based on the eight privacy criteria. As a result, the developers will be able to create IoT devices for a smart home in a compliance and compatible way with their PPA.

Appendix A

IoT-app cloud server names

This appendix shows the domain names related to each IoT device used in this thesis, notice that each domain name is responsible for particular interactions.

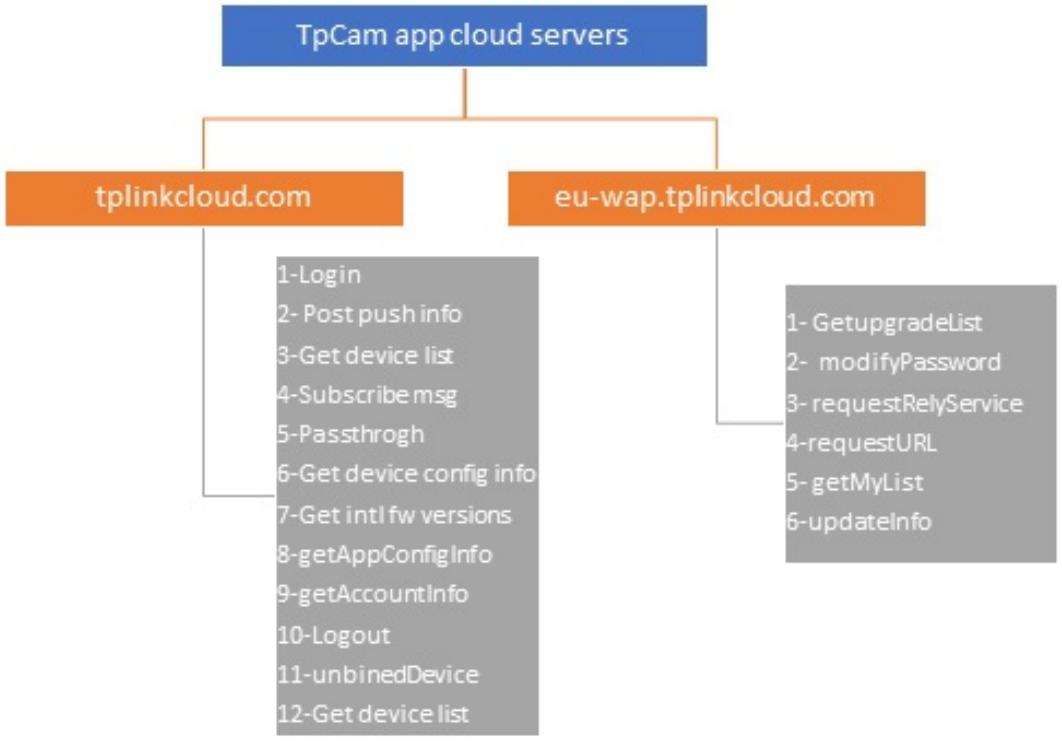

Figure A.1: TP-link smart camera domain names that TpCam app communicates with. Each domain responsible for specific methods..

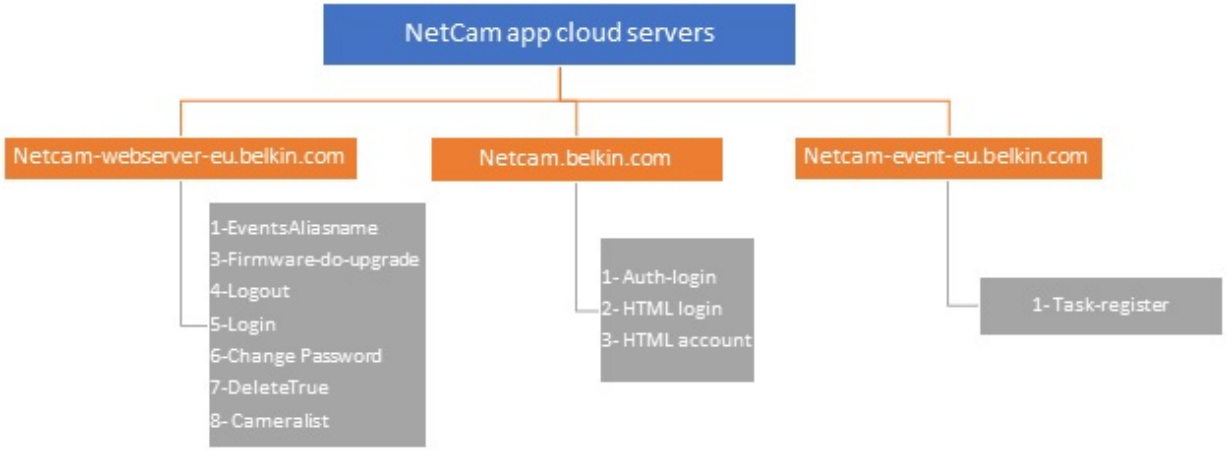

Figure A.2: Belkin Netcam smart camera domain names that NetCam app communicates with. Each domain responsible for specific methods..

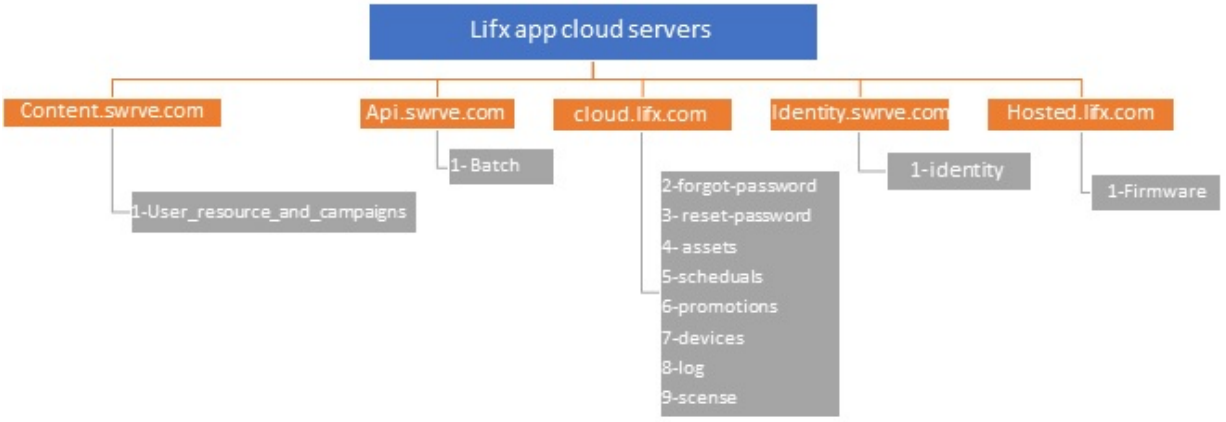

Figure A.3: LIFX smart bulb domain names that Lifx app communicates with. Each domain responsible for specific methods..

Appendix B

Methods of different user Interactions

B.1 Methods of different user Interactions

This appendix explain the packet sequences and the sizes of the methods that are executed when the user interact with the IoT app.

B.1.1 TP-link smart Plug app KASA user interactions packet sizes and sequences

	Methods	Request length in byte	Response length in byte
change password action	modifyCloudPassword	600	171
	getDeviceList	415	205
	listScenes	768	381
	listRules	700	381
	isLinked	662	381
	login	517	330
	listScenes	768	381
	authenticate token	315	278
	postPushInfo	692	178
	getDeviceList	415	1143
	helloIotCloud	1031	435
	passthrough	520	873

Table B.1: User change password interaction with KASA app that controls TP-link smart plug. Methods are always invoked by the app in the order shown - top to bottom. The sizes are of decrypted packets.

	Methods	Request length in byte	Response length in byte
Delete action	unbindDevice	513	171
	deviceRemoved	716	419
	getDevice	415	646
	listScenes	769	629
	isLinked	663	889
	retrieveLocation	663	655
	listRules	701	642

Table B.2: User delete interaction with KASA app that control TP-link smart plug. Methods are always invoked by the app in the order shown - top to bottom ("retrivelocation" is misspelled like this in the packet contents). The sizes are of decrypted packets.

B.1.2 User interactions with TP-link smart cam appTpCam, methods are always invoked by the app in the order shown - top to bottom. The sizes are of decrypted packets

	Methods	Request length in byte	Response length in byte
Login action	Login	508	318
	Post push info	713	178
	Get device	434	653
	Subscribe msg	431	178
	Passthrogh	565	464
	Get device config info	497	267
	Get intl fw versions	642	190

Table B.3: Packet sizes and sequence of User login interaction with TpCam app

	Methods	Request length in byte	Response length in byte
Logout action	logout	442	178
	postPushInfo	713	178
	subscribeMsg	431	178
	getAppConfigInfo	474	190
	getAccountInfo	450	252
	login	508	318
	getDeviceCofigInfo	497	370
	passthrough	565	462
	Get device list	434	653

Table B.4: Packet sizes and sequence of User logout interaction with TpCam app

	Method	Request length in byte	Response length in byte
Change password action	GetupgradeList	1007	227
	modifyPassword	1379	227
	login	1108	235
	HTML	812	257
	requestRelyService	927	290
	options	464	333
	requestURL	589	346
	login	1112	428
	login	1114	428
	isRelyReday	627	466
	passthrough-changepass	682	572
	getMyList	1012	809
	Admin	821	35375
	Cloud	812	61668
	updateInfo	953	63854

Table B.5: Packet sizes and sequence of User change password interaction with TpCam app.

	Method	Request length in byte	Response length in byte
Delete action	unbinedDevice	506	178
	Get device list	434	205
	getDeviceCofigInfo	497	267
	passthrough	565	464
	Get device list	434	653

Table B.6: Packet sizes and sequence of user deletes interaction with TpCam app..

B.1.3 User interactions with Belkin NetCam cam app netcam ,methods are always invoked by the app in the order shown - top to bottom. The sizes are of decrypted packets

	Method	Request length in byte	Response length in byte
Login action	Login	1190	324
		1092	541
		1043	454
		993	459
		528	451
	HTML login	1020	16228
		1028	16228
		970	13350
		1102	3098
		1002	3089
		1102	3098
		1002	3089

Table B.7: Packet sizes and sequence of User login interaction with Netcam app

Logout action	Method	Request length in byte	Response length in byte
	Logout	1013	338

Table B.8: Packet sizes and sequence of User logout interaction with Netcam app

Change-password action	Method	Request length in byte	Response length in byte
	Change password	1130	338

Table B.9: Packet sizes and sequence of User change password interaction with Netcam app.

Delete action	Method	Request length in byte	Response length in byte
	Camera delete	1022	338

Table B.10: Packet sizes and sequence of User delete interaction with Netcam app.

B.1.4 User interactions with LIFX smart lamb applifx, methods are always invoked by the app in the order shown - top to bottom. The sizes are of decrypted packets

	Method	Request length in byte	Response length in byte
Login action	Sign in	302	721
		307	446
		409	541
		414	531
		458	555
	Log	472	592
	Batch.login	680	114

Table B.11: Packet sizes and sequence of User login interaction with lifx app

	Method	Request length in byte	Response length in byte
Logout action	Batch.logout	716	114

Table B.12: Packet sizes and sequence of User logout interaction with lifx app

	Method	Request length in byte	Response length in byte
Change password action	forgot-password	292	350
	reset-password	540	4698
		1295	4751
	assets	839	276599
		862	32134

Table B.13: Packet sizes and sequence of User change password interaction with lifx app.

	Method	Request length in byte	Response length in byte
Delete action	Batch.delete	682	114
	Device-delete	207	696
		402	534
	Schedule-delete	368	510
	Promotion-delete	396	640

Table B.14: Packet sizes and sequence of User deletes interaction with lifx app

Appendix C

Visual plots of the encrypted and decrypted traffic for various actions from the Tp-link smart plug

C.1 Login interaction Plot

The following plots illustrate the packet sizes and sequences of the login interaction between the user and the KASA app in encrypted and decrypted format.

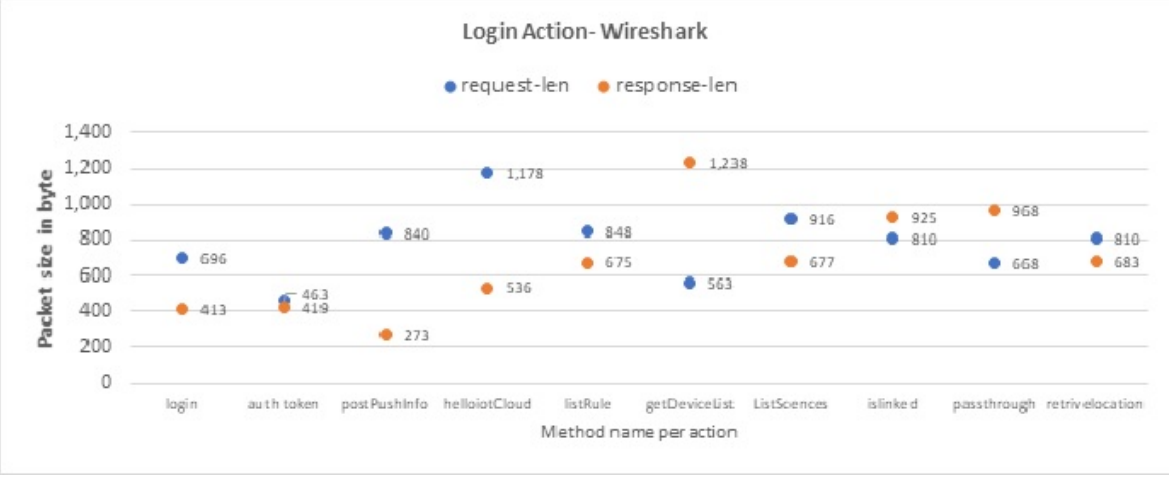

Figure C.1: User login interaction from the KASA in encrypted format

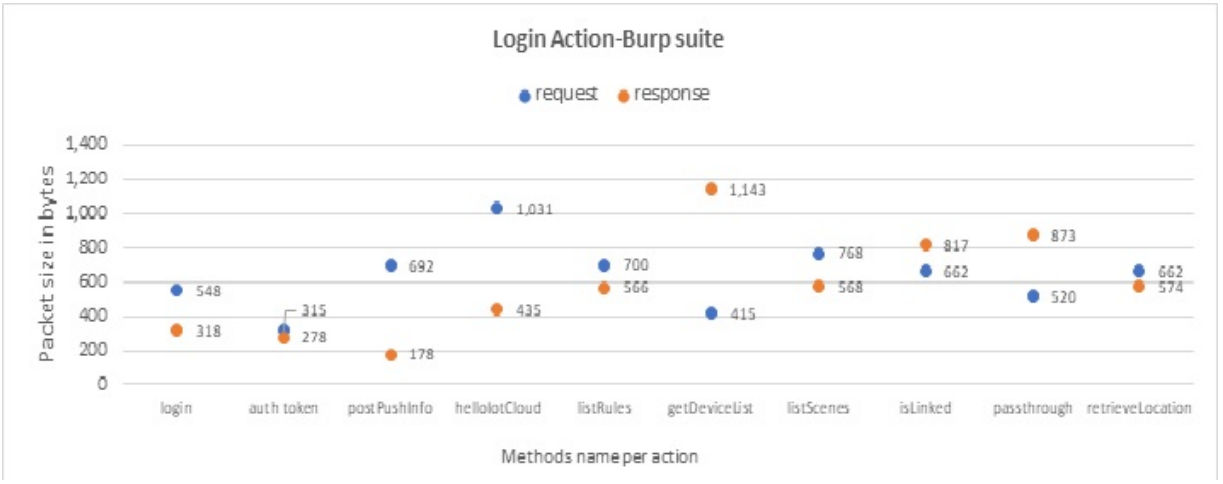

Figure C.2: Equivalent user login interaction from the KASA in decrypted format.

C.2 Change Password interaction Plot

The following plots illustrate the packet sizes and sequences of the change password interaction between the user and the KASA app in encrypted and decrypted format.

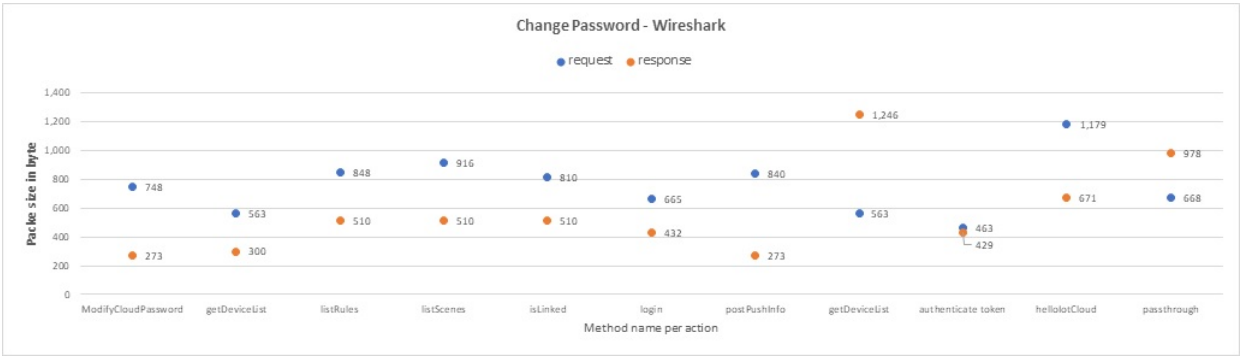

Figure C.3: User change password interaction from the KASA in encrypted format.

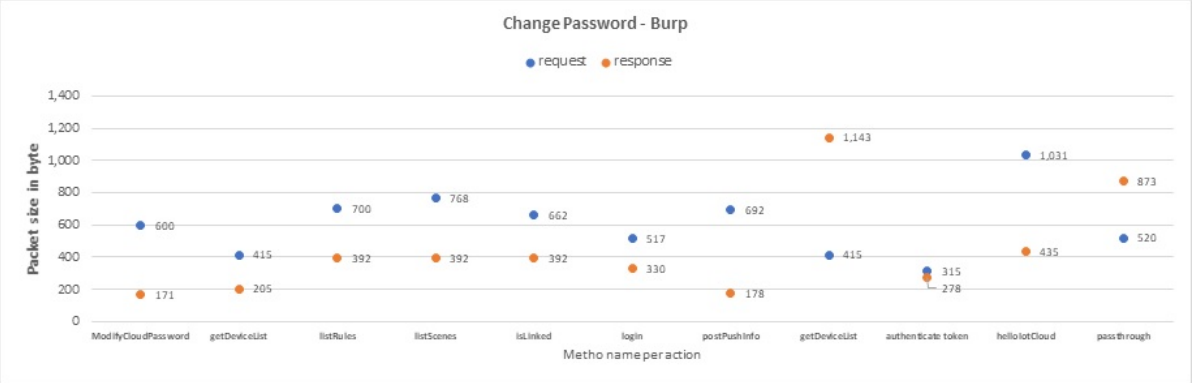

Figure C.4: Equivalent user change password interaction from the KASA in decrypted format.

C.3 Delete interaction Plot

The following plots illustrate the packet sizes and sequences of the delete interaction between the user and the KASA app in encrypted and decrypted format.

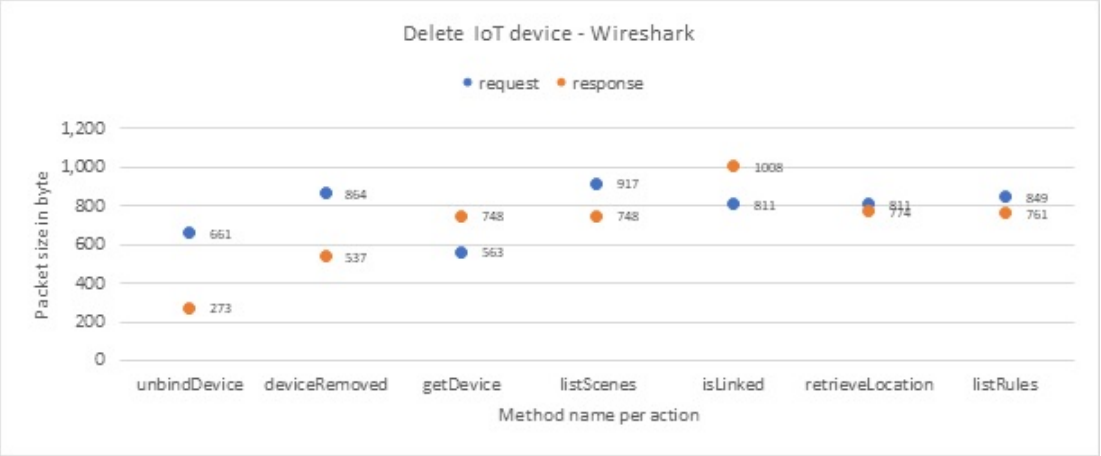

Figure C.5: User delete interaction from the KASA in encrypted format

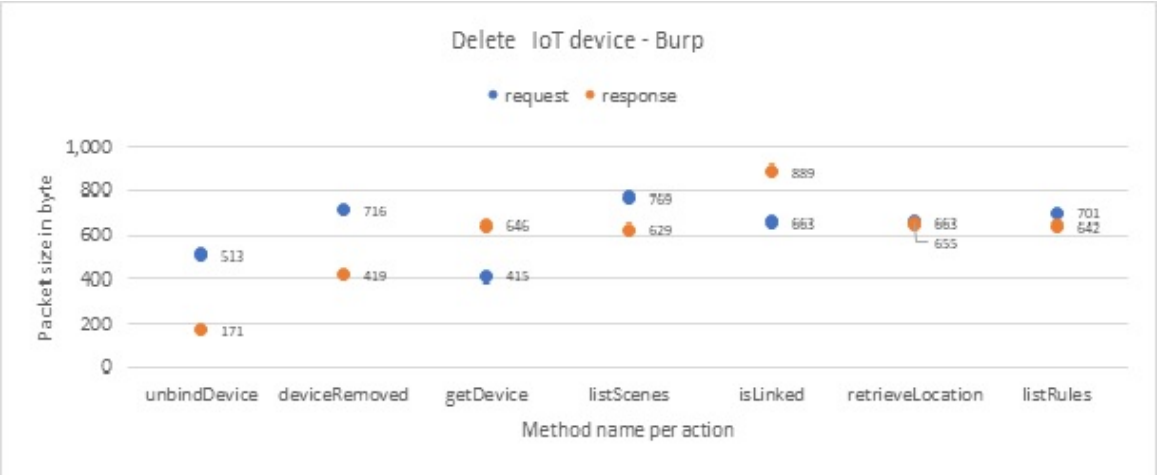

Figure C.6: Equivalent user delete interaction from the KASA in decrypted format.

Appendix D

The results of applying the Evaluating the IoT behavior compliance tool with its PPA on the IoT devices

D.1 Evaluate the compliance of Tp-link smart plug

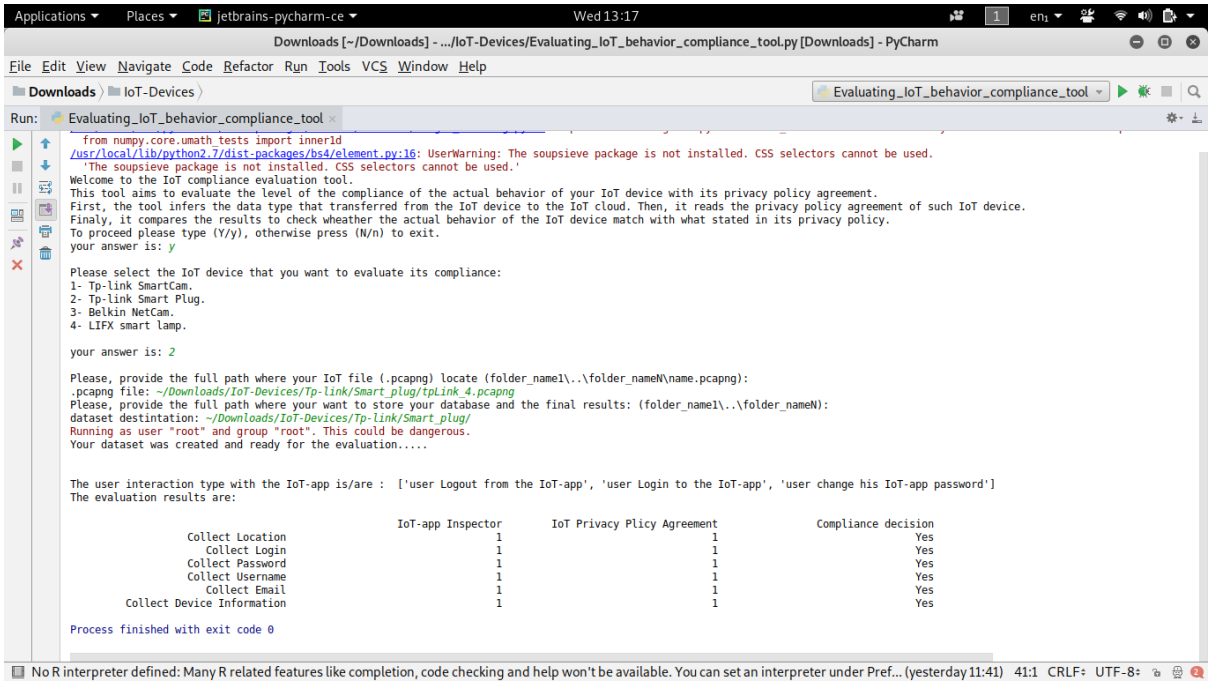

Figure D.1: Evaluate the level of compliance of the Tp-link smart plug with its PPA-1.

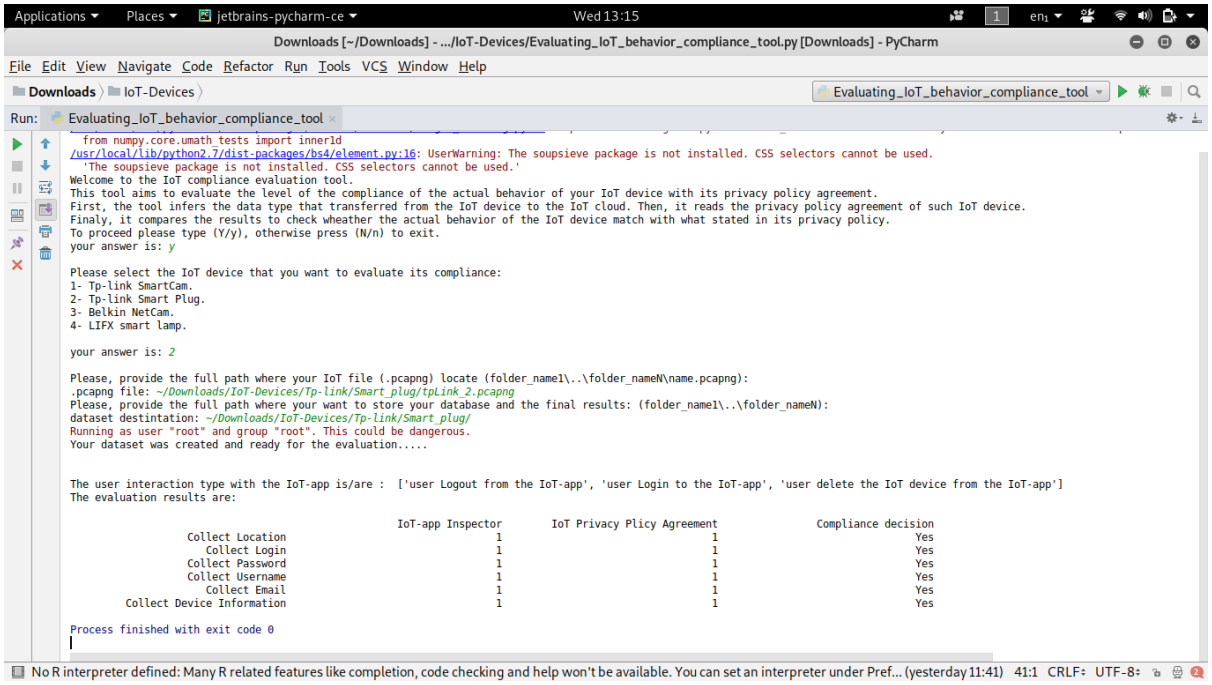

Figure D.2: Evaluate the level of compliance of the Tp-link smart plug with its PPA-2.

D.2 Evaluate the compliance of Tp-link smart cam

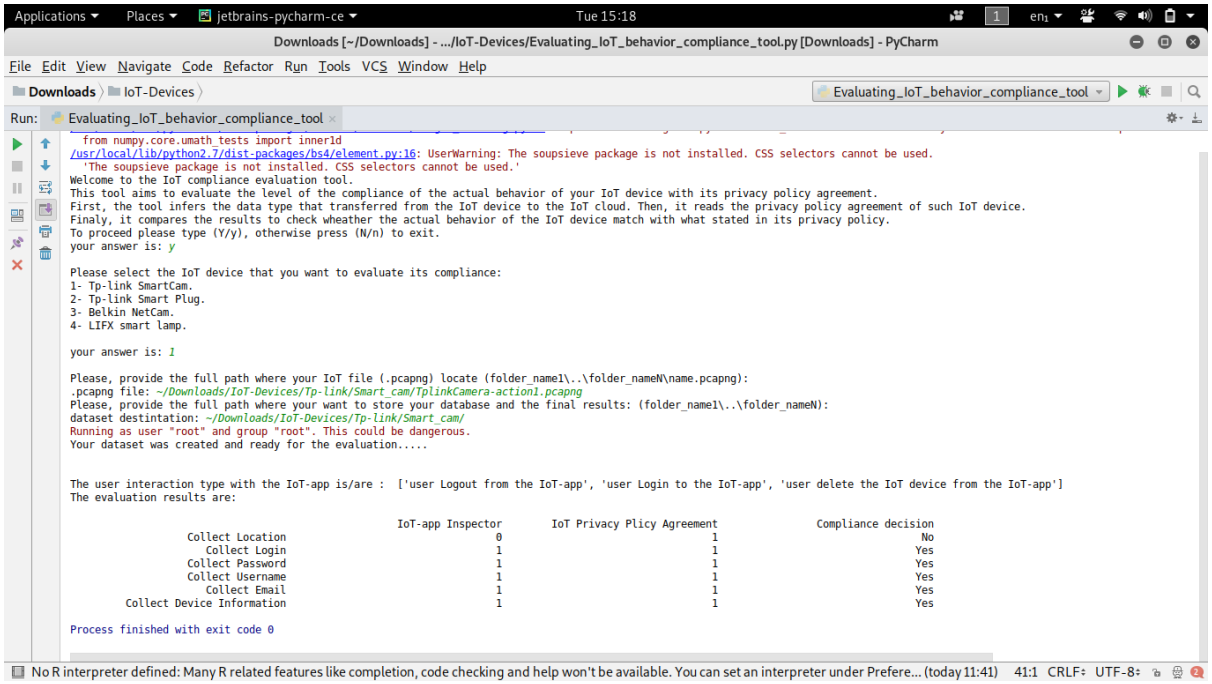

Figure D.3: Evaluate the level of compliance of the Tp-link smart camera with its PPA-1.

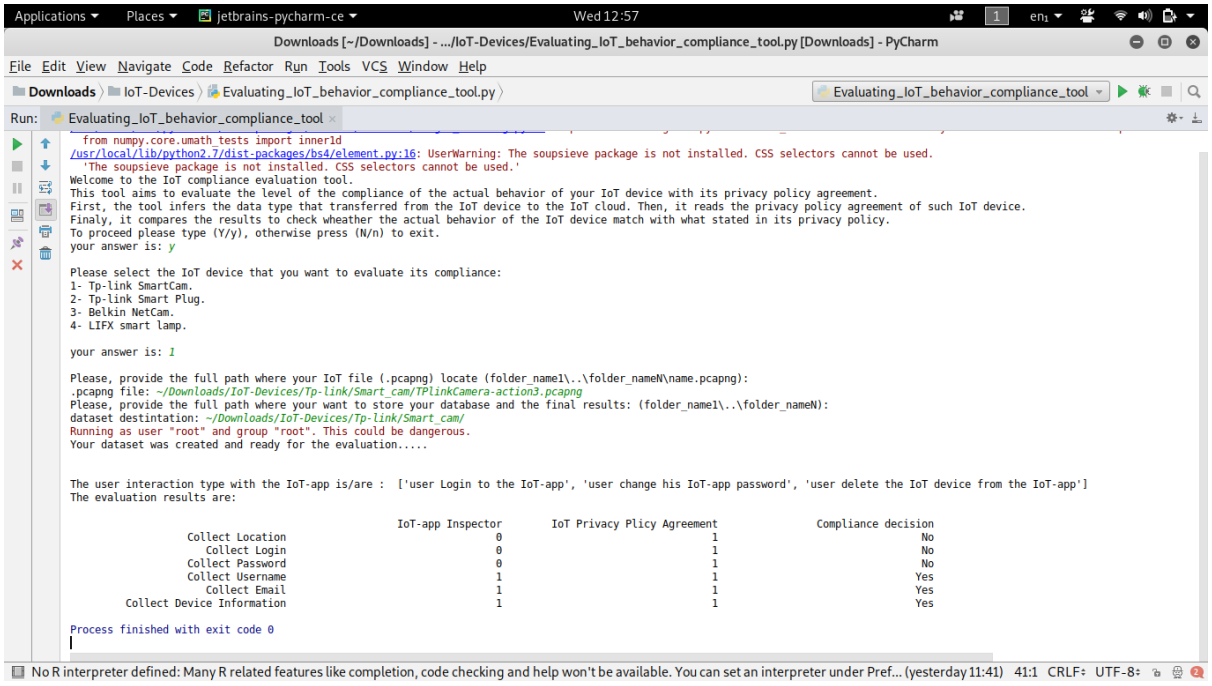

Figure D.4: Evaluate the level of compliance of the Tp-link smart camera with its PPA-2.

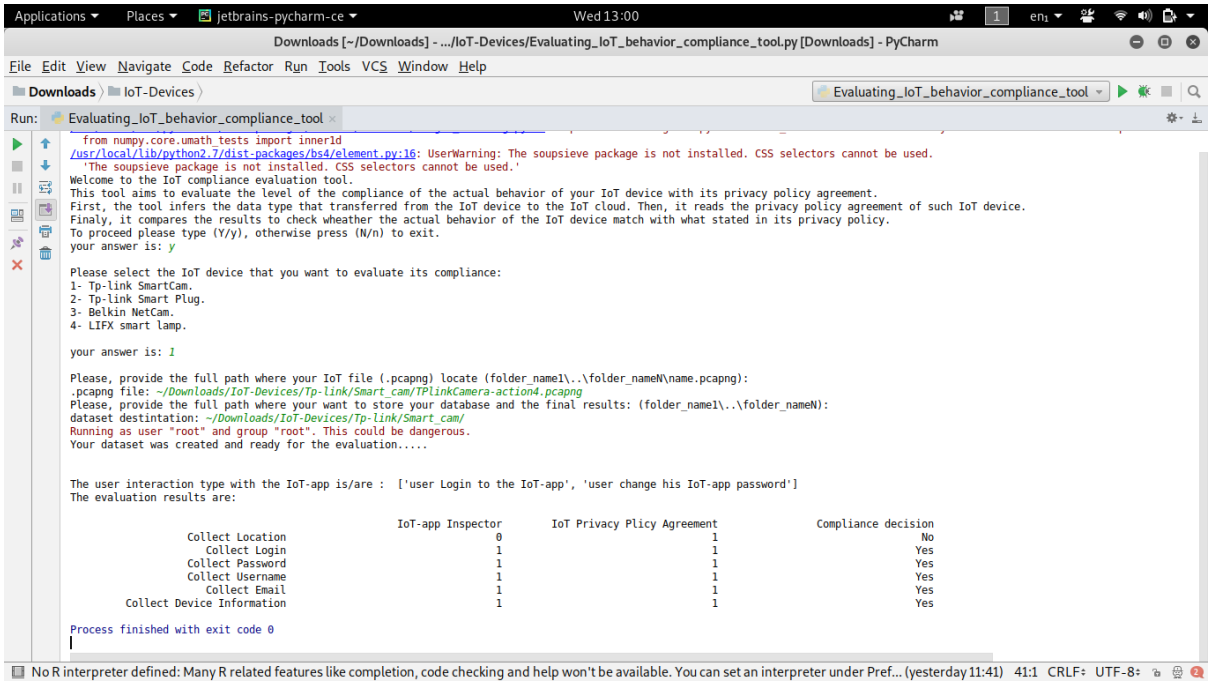

Figure D.5: Evaluate the level of compliance of the Tp-link smart camera with its PPA-3.

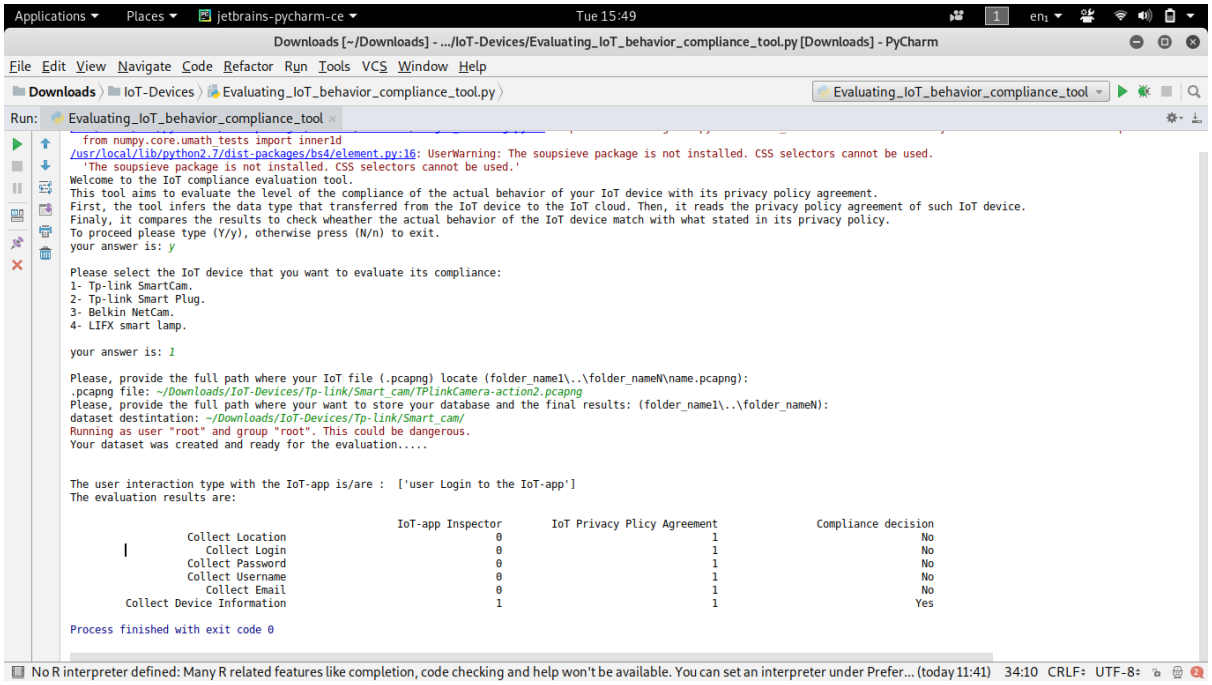

Figure D.6: Evaluate the level of compliance of the Tp-link smart camera with its PPA-4.

D.3 Evaluate the compliance of Belkin NetCam

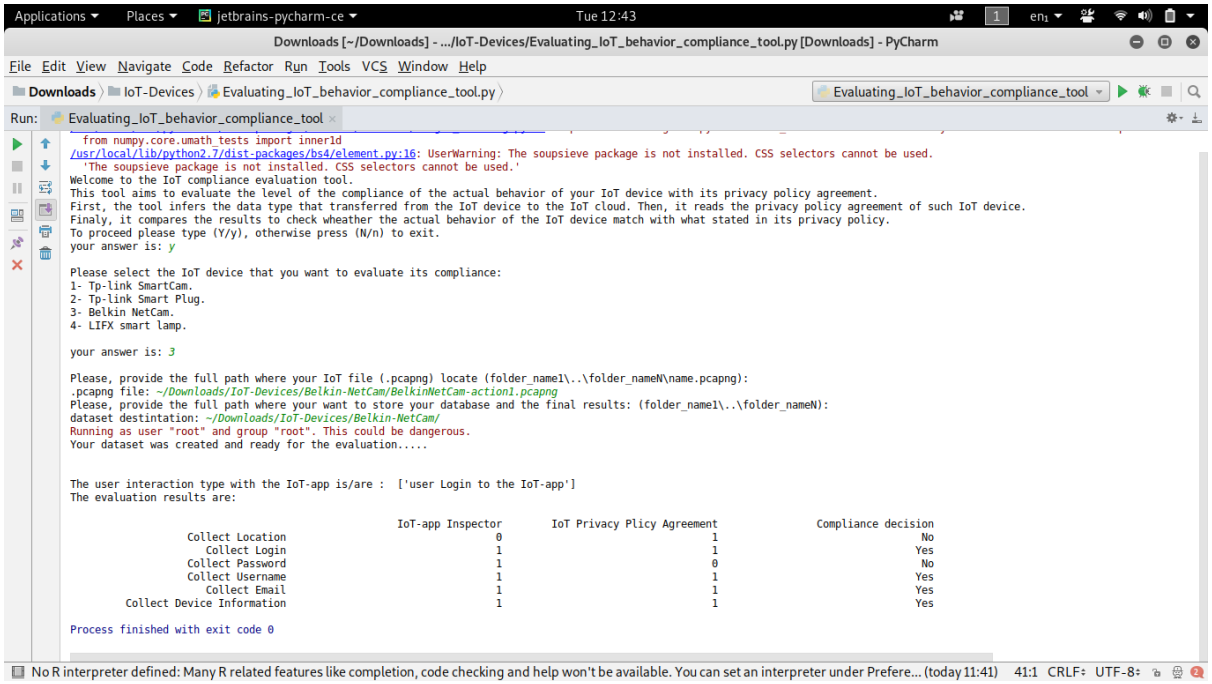

Figure D.7: Evaluate the level of compliance of the Belkin NetCam with its PPA-1.

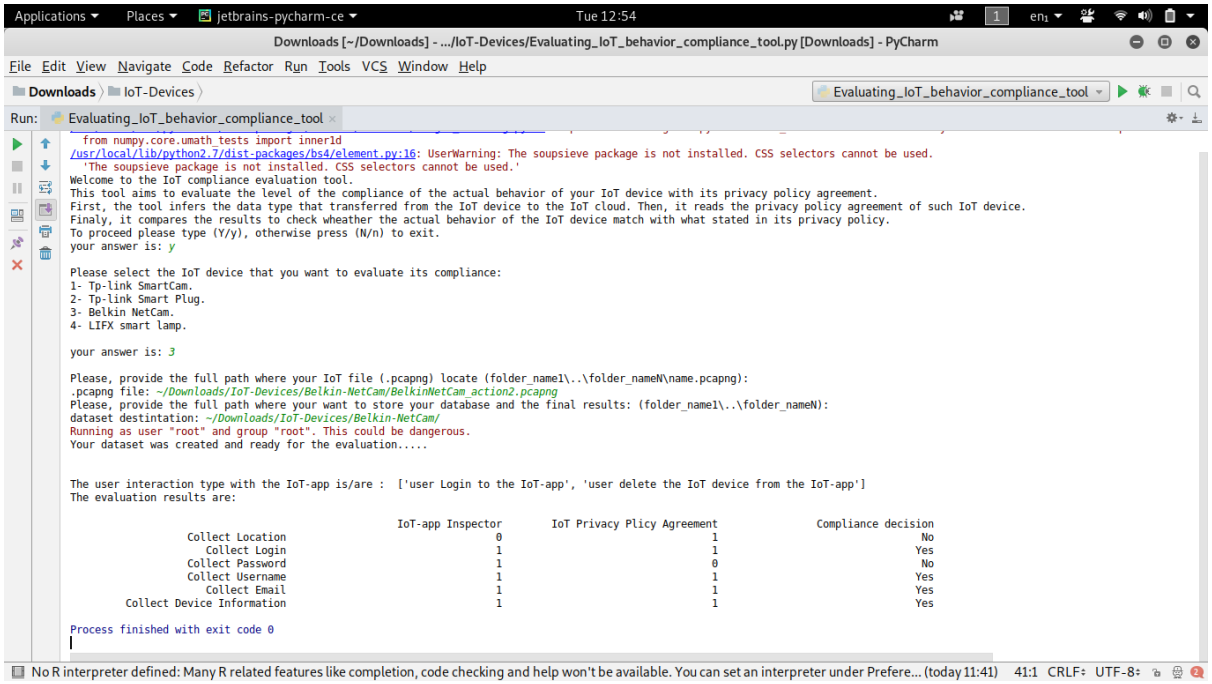

Figure D.8: Evaluate the level of compliance of the Belkin NetCam with its PPA-2.

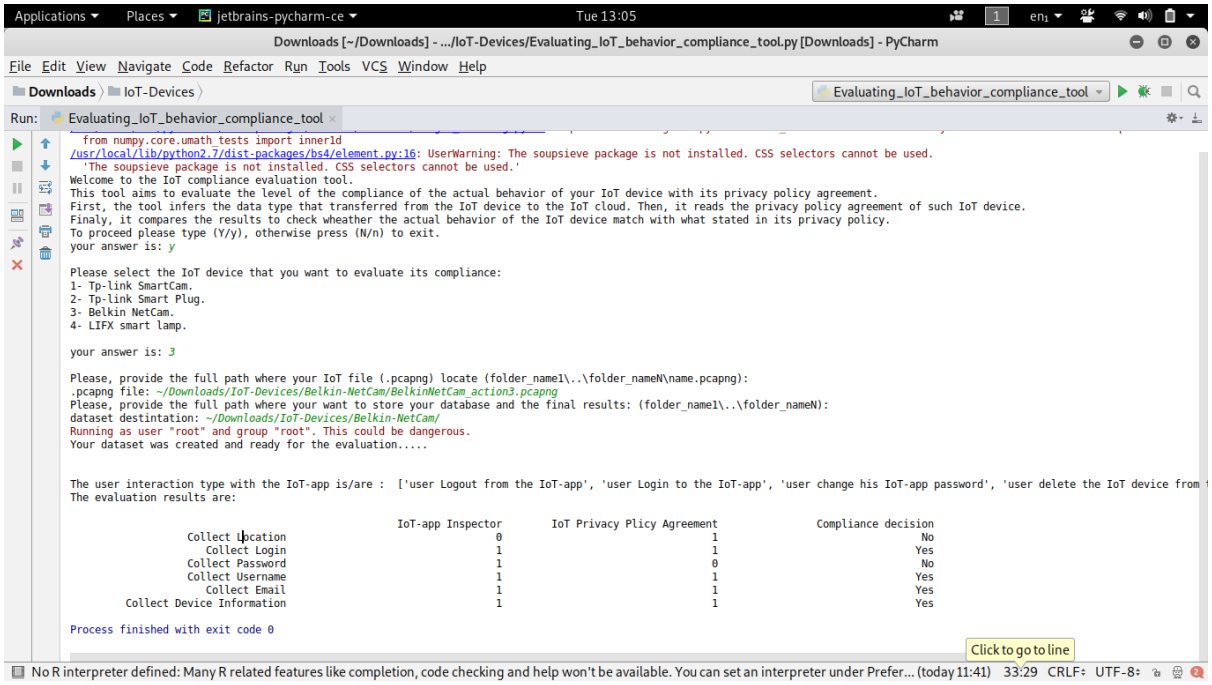

Figure D.9: Evaluate the level of compliance of the Belkin NetCam with its PPA-3.

D.4 Evaluate the compliance of Lifx smart bulb

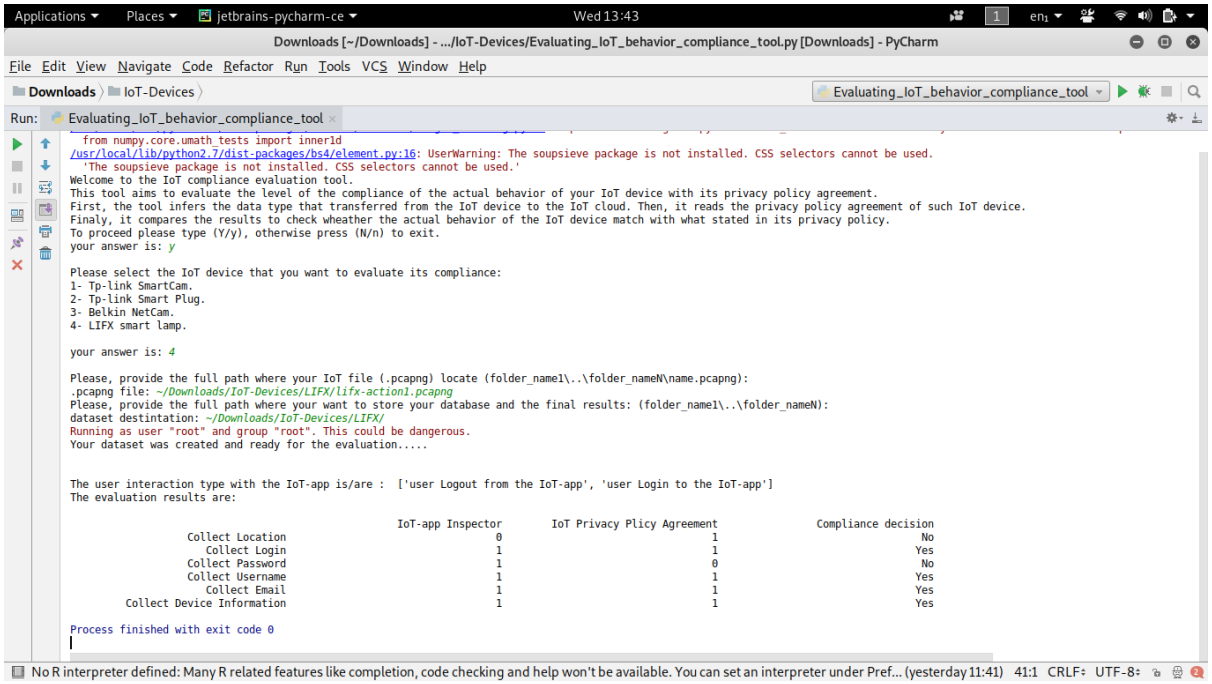

Figure D.10: Evaluate the level of compliance of the Lifx smart bulb with its PPA-1.

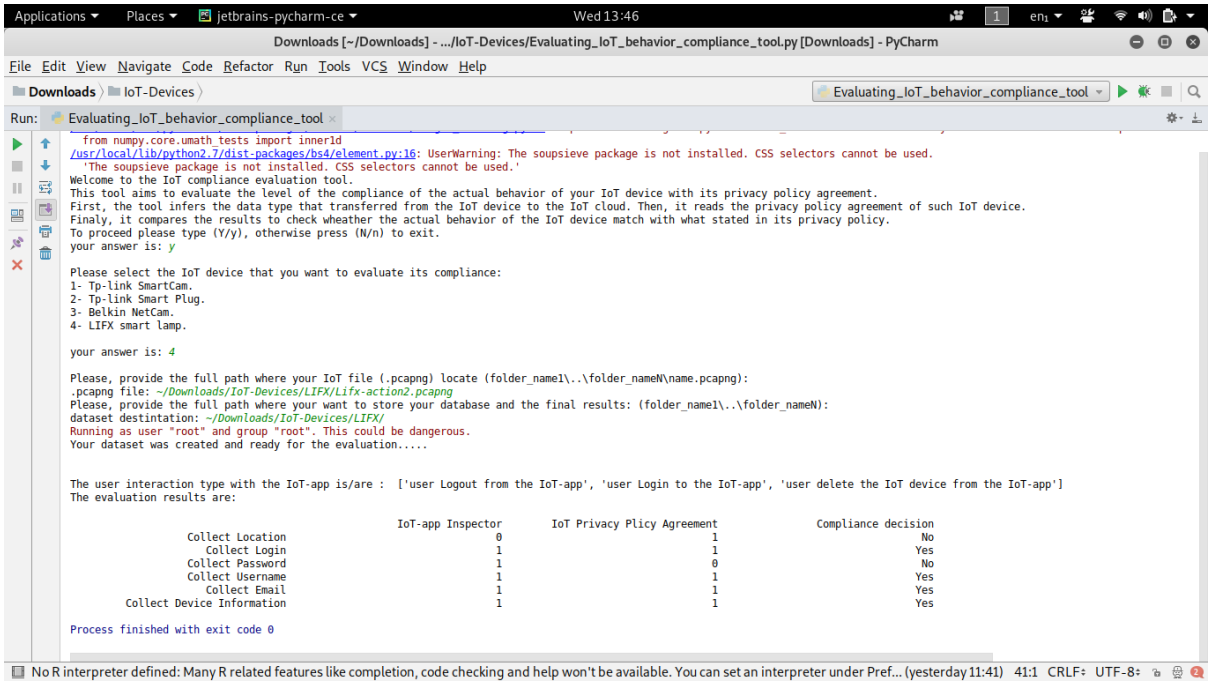

Figure D.11: Evaluate the level of compliance of the Lifx smart bulb with its PPA-2.

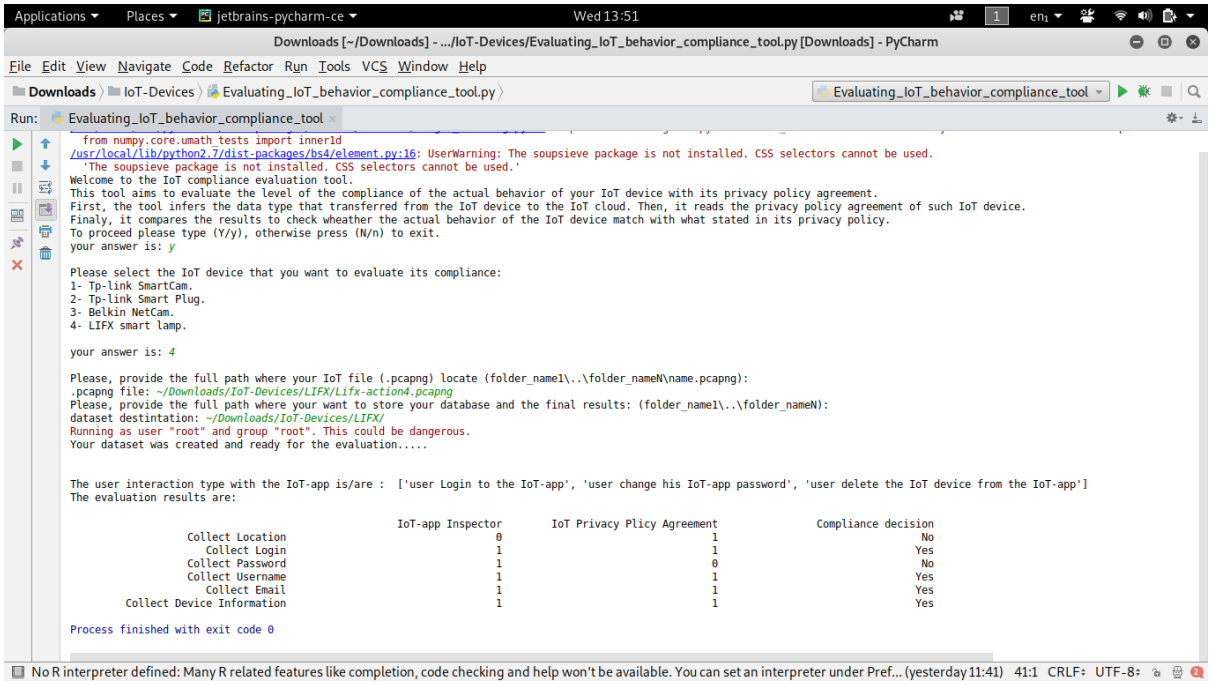

Figure D.12: Evaluate the level of compliance of the Lifx smart bulb with its PPA-3.

Bibliography

[1] Trendnet cameras - i always feel like somebody's watching me. http://console-cowboys.blogspot.co.uk/2012/01/trendnet-cameras-i-always-feel-like.html, 2017.

[2] Federal Trade Commission. https://www.ftc.gov/, 2019.

[3] sklearn.ensemble.randomforestclassifier – scikit-learn 0.21.1 documentation. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html, 2019.

[4] You need a privacy policy. here's why. https://www.webhostingsecretrevealed.net/blog/blogging-tips/have-a-website-you-need-a-privacy-policy-heres-why/, 2019.

[5] Arp spoofing. https://www.veracode.com/security/arp-spoofing, 2020.

[6] Configuring an android device to work, portswigger web security. https://support.portswigger.net/customer/portal/articles/1841101-configuring-an-android-device-to-work-with-burp, 2020.

[7] create wifi hotspot in linux kali linux. http://techsarjan.com/2014/10/how-to-create-wi-fi-hotspot-in-linux.html, 2020.

[8] Information commissioner office. https://ico.org.uk/, 2020.

[9] Reverse engineering the tp-link hs110. https://www.softscheck.com/en/reverse-engineering-tp-link-hs110/, 2020.

[10] The text annotation tool to train ai. https://www.tagtog.net/, 2020.

[11] Wireshark chapter 1. introduction. `https://www.wireshark.org/docs/wsug_html_chunked/ChapterIntroduction.html`, 2020.

[12] Allan A. Got an iphone or 3g ipad? apple is recording your moves. `http://radar.oreilly.com/2011/04/apple-location-tracking.html`, 2020.

[13] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and A Selcuk Uluagac. Peek-a-boo: I see your smart home activities, even encrypted! *arXiv preprint arXiv:1808.02741*, 2018.

[14] David S Allison, Hany F EL Yamany, and Miriam AM Capretz. Metamodel for privacy policies within soa. In *2009 ICSE Workshop on Software Engineering for Secure Systems*, pages 40–46. IEEE, 2009.

[15] Omar Alrawi, Chaz Lever, Manos Antonakakis, and Fabian Monrose. Sok: Security evaluation of home-based iot deployments. In *IEEE S&P*, pages 208–226, 2019.

[16] Waleed Ammar, Shomir Wilson, Norman Sadeh, and Noah A Smith. Automatic categorization of privacy policies: A pilot study. *School of Computer Science, Language Technology Institute, Technical Report CMU-LTI-12-019*, 2012.

[17] Ioannis Andrea, Chrysostomos Chrysostomou, and George Hadjichristofi. Internet of things: Security vulnerabilities and challenges. In *2015 IEEE Symposium on Computers and Communication (ISCC)*, pages 180–187. IEEE, 2015.

[18] Darko Andročec and Neven Vrček. Machine learning for the internet of things security: A systematic review. In *The 13th International Conference on Software Technologies*, 2018.

[19] Eirini Anthi, Shazaib Ahmad, Omer Rana, George Theodorakopoulos, and Pete Burnap. Eclipseiot: A secure and adaptive hub for the internet of things. *Computers & Security*, 78:477–490, 2018.

[20] Annie I Antón and Julia B Earp. Strategies for developing policies and requirements for secure and private electronic commerce. In *E-commerce security and privacy*, pages 67–86. Springer, 2001.

[21] Annie I Antón, Julia B Earp, and Ryan A Carter. Precluding incongruous behavior by aligning software requirements with security and privacy policies. *Information and Software Technology*, 45(14):967–977, 2003.

[22] Noah Apthorpe, Dillon Reisman, and Nick Feamster. A smart home is no castle: Privacy vulnerabilities of encrypted iot traffic. *arXiv preprint arXiv:1705.06805*, 2017.

[23] Noah Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. Spying on the smart home: Privacy attacks and defenses on encrypted iot traffic. *arXiv preprint arXiv:1708.05044*, 2017.

[24] Kevin Ashton et al. That'internet of things' thing. *RFID journal*, 22(7):97–114, 2009.

[25] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer networks*, 54(15):2787–2805, 2010.

[26] Rawan Baalous, Ronald Poet, and Timothy Storer. Analyzing privacy policies of zero knowledge cloud storage applications on mobile devices. In *2018 IEEE International Conference on Cloud Engineering (IC2E)*, pages 218–224. IEEE, 2018.

[27] Rebecca Balebako, Abigail Marsh, Jialiu Lin, Jason I Hong, and Lorrie Faith Cranor. The privacy and security behaviors of smartphone app developers. 2014.

[28] Rebecca Balebako, Florian Schaub, Idris Adjerid, Alessandro Acquisti, and Lorrie Cranor. The impact of timing on the salience of smartphone app privacy notices. In *Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices*, pages 63–74, 2015.

[29] Mario Ballano Barcena, Candid Wueest, and Hon Lau. How safe is your quantified self. *Symantech: Mountain View, CA, USA*, 16, 2014.

[30] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[31] Anna L Buczak and Erhan Guven. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, 18(2):1153–1176, 2015.

[32] Juan Miguel Cejuela, Peter McQuilton, Laura Ponting, Steven J Marygold, Raymund Stefancsik, Gillian H Millburn, and Burkhard Rost. tagtog: interactive and text-mining-assisted annotation of gene mentions in plos full-text articles. *Database*, 2014, 2014.

[33] Yi Cheng, Mats Naslund, Göran Selander, and Eva Fogelstrom. Privacy in machine-to-machine communications a state-of-the-art survey. In *Communication Systems (ICCS), 2012 IEEE International Conference on*, pages 75–79. IEEE, 2012.

[34] III C.Johnson. Us office of management and budget memorandum m-07-16. `https://www.whitehouse.gov/sites/whitehouse.gov/files/omb/memoranda/2007/m07-16.pdf`, 2020.

[35] Elisa Costante, Jerry Den Hartog, and Milan Petkovic. On-line trust perception: What really matters. In *2011 1st Workshop on Socio-Technical Aspects in Security and Trust (STAST)*, pages 52–59. IEEE, 2011.

[36] Elisa Costante, Yuanhao Sun, Milan Petković, and Jerry den Hartog. A machine learning solution to assess privacy policy completeness: (short paper). In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, pages 91–96, 2012.

[37] Andy Crabtree. Enabling the new economic actor: Personal data regulation and the digital economy. In *2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW)*, pages 124–129. IEEE, 2016.

[38] Britt Cyr, Webb Horn, Daniela Miao, and Michael Specter. Security analysis of wearable fitness devices (fitbit). *Massachusetts Institute of Technology*, 1, 2014.

[39] Anupam Das, Joseph Bonneau, Matthew Caesar, Nikita Borisov, and Xiaofeng Wang. The tangled web of password reuse. 2014. *Cited on*, page 7, 2014.

[40] Aveek K Das, Parth H Pathak, Chen-Nee Chuah, and Prasant Mohapatra. Uncovering privacy leakage in ble network traffic of wearable fitness trackers. In *Proceedings of the 17th International Workshop on Mobile Computing Systems and Applications*, pages 99–104, 2016.

[41] W Keith Edwards and Rebecca E Grinter. At home with ubiquitous computing: Seven challenges. In *International conference on ubiquitous computing*, pages 256–272. Springer, 2001.

[42] Mahmoud Elkhodr, Seyed Shahrestani, and Hon Cheung. The internet of things: new interoperability, management and security challenges. *arXiv preprint arXiv:1604.04824*, 2016.

[43] Roman Ferrando and Paul Stacey. Classification of device behaviour in internet of things infrastructures: towards distinguishing the abnormal from security threats. In *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, page 57. ACM, 2017.

[44] C Gomez, J CROWCROFT, and M SCHARF. Tcp usage guidance in the internet of things (iot). Technical report, Internet Draft. Internet Engineering Task Force (IETF), 2018.

[45] Roger A. Grimes. What is personally identifiable information (pii)? how to protect it under gdpr. `https://www.csoonline.com/article/3215864/how-to-protect-personally-identifiable-information-pii-under-gdpr.html`, 2020.

[46] Matthew L Hale, Dalton Ellis, Rose Gamble, Charles Waler, and Jessica Lin. Secu wear: An open source, multi-component hardware/software platform for exploring wearable security. In *2015 IEEE International Conference on Mobile Services*, pages 97–104. IEEE, 2015.

[47] Hamza Harkous, Kassem Fawaz, Rémi Lebret, Florian Schaub, Kang G Shin, and Karl Aberer. Polisis: Automated analysis and presentation of privacy policies using deep learning. In *27th {USENIX} Security Symposium ({USENIX} Security 18)*, pages 531–548, 2018.

[48] Andrew Hilts, Christopher Parsons, and Jeffrey Knockel. Every step you fake: A comparative analysis of fitness tracker privacy and security. *Open Effect Report*, 76(24):31–33, 2016.

[49] ICO. What is personal data? `https://ico.org.uk/media/for-organisations/documents/1549/determining_what_is_personal_data_quick_reference_guide.pdf`, 2020.

[50] Thorsten Kramp, Rob Van Kranenburg, and Sebastian Lange. Introduction to the internet of things. In *Enabling Things to Talk*, pages 1–10. Springer, Berlin, Heidelberg, 2013.

[51] Markus Laner, Philipp Svoboda, Navid Nikaein, and Markus Rupp. Traffic models for machine type communications. In *ISWCS 2013; The Tenth International Symposium on Wireless Communication Systems*, pages 1–5. VDE, 2013.

[52] L.Constantin. Widely used wireless ip cameras open to hijacking over the internet, researchers say. http://www.pcworld.com/article/2033821/security/widely-used-wireless-ip-cameras-open-to-hijacking-over-the-internet-researc html., 2017.

[53] Parikshit N Mahalle, Neeli Rashmi Prasad, and Ramjee Prasad. Object classification based context management for identity management in internet of things. *International Journal of Computer Applications*, 63(12), 2013.

[54] James Manyika and Michael Chui. By 2025, internet of things applications could have $11 trillion impact. https://www.mckinsey.com/mgi/overview/in-the-news/by-2025-internet-of-things-applications-could-have-11-trillion-impact, 2020.

[55] James Manyika and Michael Chui. The internet of things: Mapping the value beyond the hype. https://www.mckinsey.com/~/media/McKinsey/Industries/Technology%20Media%20and%20Telecommunications/High%20Tech/Our%20Insights/The%20Internet%20of%20Things%20The%20value%20of%20digitizing%20the%20physical%20world/Unlocking_the_potential_of_the_Internet_of_Things_Executive_summary.ashx, 2020.

[56] Massimo Marchiori, Lorrie Cranor, Marc Langheinrich, Martin Presler-Marshall, and Joseph Reagle. The platform for privacy preferences 1.0 (p3p1. 0) specification. *World Wide Web Consortium Recommendation REC-P3P-20020416*, 2002.

[57] Aaron K Massey, Jacob Eisenstein, Annie I Antón, and Peter P Swire. Automated text mining for requirements analysis of policy documents. In *2013 21st*

IEEE International Requirements Engineering Conference (RE), pages 4–13. IEEE, 2013.

[58] Aleecia M McDonald and Lorrie Faith Cranor. The cost of reading privacy policies. *Isjlp*, 4:543, 2008.

[59] Yair Meidan, Michael Bohadana, Asaf Shabtai, Juan David Guarnizo, Martín Ochoa, Nils Ole Tippenhauer, and Yuval Elovici. Profiliot: a machine learning approach for iot device identification based on network traffic analysis. In *Proceedings of the symposium on applied computing*, pages 506–509, 2017.

[60] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martin Ochoa, Nils Ole Tippenhauer, Juan Davis Guarnizo, and Yuval Elovici. Detection of unauthorized iot devices using machine learning techniques. *arXiv preprint arXiv:1709.04647*, 2017.

[61] Georg Merzdovnik, Damjan Buhov, Artemios G Voyiatzis, and Edgar R Weippl. Notary-assisted certificate pinning for improved security of android apps. In *2016 11th International Conference on Availability, Reliability and Security (ARES)*, pages 365–371. IEEE, 2016.

[62] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. Iot sentinel: Automated device-type identification for security enforcement in iot. In *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, pages 2177–2184. IEEE, 2017.

[63] Jozef Mocnej, Adrian Pekar, Winston KG Seah, and Iveta Zolotova. Network traffic characteristics of the iot application use cases.

[64] Dragos Mocrii, Yuxiang Chen, and Petr Musilek. Iot-based smart homes: A review of system architecture, software, communications, privacy and security. *Internet of Things*, 1:81–98, 2018.

[65] Ricardo Neisse, Gary Steri, and Gianmarco Baldini. Enforcement of security policy rules for the internet of things. In *2014 IEEE 10th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 165–172. IEEE, 2014.

[66] Johannes Obermaier and Martin Hutle. Analyzing the security and privacy of cloud-based video surveillance systems. In *Proceedings of the 2nd ACM international workshop on IoT privacy, trust, and security*, pages 22–28, 2016.

[67] U.S. Senate Committee on the Judiciary. Protecting mobile privacy: Your smartphones, tablets, cell phones and your privacy. https://www.judiciary.senate.gov/meetings/protecting-mobile-privacy-your-smartphones-tablets-cell-phones-and-your-pri 2020.

[68] Charith Perera, Chi Harold Liu, and Srimal Jayawardena. The emerging internet of things marketplace from an industrial perspective: A survey. *IEEE Transactions on Emerging Topics in Computing*, 3(4):585–598, 2015.

[69] Charith Perera, Rajiv Ranjan, Lizhe Wang, Samee Khan, and Albert Zomaya. Privacy of big data in the internet of things era. *IEEE IT Special Issue Internet of Anything*, 6, 2015.

[70] Charith Perera, Rajiv Ranjan, Lizhe Wang, Samee U Khan, and Albert Y Zomaya. Big data privacy in the internet of things era. *IT Professional*, 17(3):32–39, 2015.

[71] Alfredo J Perez, Sherali Zeadally, and Jonathan Cochran. A review and an empirical analysis of privacy policy and notices for consumer internet of things. *Security and Privacy*, 1(3):e15, 2018.

[72] Rohan Ramanath, Fei Liu, Norman Sadeh, and Noah A Smith. Unsupervised alignment of privacy policies using hidden markov models. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 605–610, 2014.

[73] Joel R Reidenberg, Jaspreet Bhatia, Travis D Breaux, and Thomas B Norton. Ambiguity in privacy policies and the impact of regulation. *The Journal of Legal Studies*, 45(S2):S163–S190, 2016.

[74] Federal Trade Commission Staff Report. What's the deal? https://www.ftc.gov/reports/whats-deal-federal-trade-commission-study-mobile-shopping-apps-august-2014, 2020.

[75] William N Robinson. Implementing rule-based monitors within a framework for continuous requirements monitoring. In *Proceedings of the 38th Annual Hawaii International Conference on System Sciences*, pages 188a–188a. IEEE, 2005.

[76] Margaret Rouse. personally identifiable information (pii). `https://searchfinancialsecurity.techtarget.com/definition/personally-identifiable-information`, 2020.

[77] Kanthashree Mysore Sathyendra, Florian Schaub, Shomir Wilson, and Norman Sadeh. Automatic extraction of opt-out choices from privacy policies. In *2016 AAAI Fall Symposium Series*, 2016.

[78] Kanthashree Mysore Sathyendra, Shomir Wilson, Florian Schaub, Sebastian Zimmeck, and Norman Sadeh. Identifying the provision of choices in privacy policy text. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2774–2779, 2017.

[79] Florian Schaub, Rebecca Balebako, Adam L Durity, and Lorrie Faith Cranor. A design space for effective privacy notices. In *Eleventh Symposium On Usable Privacy and Security ({SOUPS} 2015)*, pages 1–17, 2015.

[80] Muhammad Zubair Shafiq, Lusheng Ji, Alex X Liu, Jeffrey Pang, and Jia Wang. A first look at cellular machine-to-machine traffic: large scale measurement and characterization. *ACM SIGMETRICS performance evaluation review*, 40(1):65–76, 2012.

[81] Mustafizur R Shahid, Gregory Blanc, Zonghua Zhang, and Hervé Debar. Iot devices recognition through network traffic analysis. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5187–5192. IEEE, 2018.

[82] Parvaneh Shayegh and Sepideh Ghanavati. Toward an approach to privacy notices in iot. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 104–110. IEEE, 2017.

[83] R. Shirey. Internet security glossary. `https://www.ietf.org/rfc/rfc2828.txt`, 2020.

[84] Shachar Siboni, Asaf Shabtai, Nils O Tippenhauer, Jemin Lee, and Yuval Elovici. Advanced security testbed framework for wearable iot devices. *ACM Transactions on Internet Technology (TOIT)*, 16(4):1–25, 2016.

[85] Sandra Siby, Rajib Ranjan Maiti, and Nils Tippenhauer. Iotscanner: Detecting and classifying privacy threats in iot neighborhoods. *arXiv preprint arXiv:1701.05007*, 2017.

[86] Ravi Inder Singh, Manasa Sumeeth, and James Miller. Evaluating the readability of privacy policies in mobile environments. *International Journal of Mobile Human Computer Interaction (IJMHCI)*, 3(1):55–78, 2011.

[87] Arunan Sivanathan. Iot behavioral monitoring via network traffic analysis. *arXiv preprint arXiv:2001.10632*, 2020.

[88] Arunan Sivanathan, Hassan Habibi Gharakheili, and Vijay Sivaraman. Detecting behavioral change of iot devices using clustering-based network traffic modeling. *IEEE Internet of Things Journal*, 2020.

[89] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Adam Radford, Chamith Wijenayake, Arun Vishwanath, and Vijay Sivaraman. Characterizing and classifying iot traffic in smart cities and campuses. In *2017 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 559–564. IEEE, 2017.

[90] Arunan Sivanathan, Daniel Sherratt, Hassan Habibi Gharakheili, Vijay Sivaraman, and Arun Vishwanath. Low-cost flow-based security solutions for smart-home iot devices. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE, 2016.

[91] Rocky Slavin, Xiaoyin Wang, Mitra Bokaei Hosseini, James Hester, Ram Krishnan, Jaspreet Bhatia, Travis D Breaux, and Jianwei Niu. Toward a framework for detecting privacy policy violations in android application code. In *Proceedings of the 38th International Conference on Software Engineering*, pages 25–36, 2016.

[92] S.Perera. Taxonomy of iot usecases: Seeing iot forest from the trees. `https://iwringer.wordpress.com/2015/10/08/taxonomy-of-iot-usecases-seeing-iot-forest-from-the-trees/`, 2020.

[93] Statista Research Department. Internet of things (iot) connected devices installed base worldwide from 2015 to 2025. `https://www.statista.com/`

statistics/471264/iot-number-of-connected-devices-worldwide/, 2020.

[94] Alanoud Subahi and George Theodorakopoulos. Ensuring compliance of iot devices with their privacy policy agreement. In *2018 IEEE 6th International Conference on Future Internet of Things and Cloud (FiCloud)*, pages 100–107. IEEE, 2018.

[95] Alanoud Subahi and George Theodorakopoulos. Detecting iot user behavior and sensitive information in encrypted iot-app traffic. *Sensors*, 19(21):4777, 2019.

[96] Harald Sundmaeker, Patrick Guillemin, Peter Friess, and Sylvie Woelfflé. Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision*, 3(3):34–36, 2010.

[97] Ali Sunyaev, Tobias Dehling, Patrick L Taylor, and Kenneth D Mandl. Availability and quality of mobile health app privacy policies. *Journal of the American Medical Informatics Association*, 22(e1):e28–e33, 2015.

[98] M Sweeney. What is pii, non-pii, and personal data? https://piwik.pro/blog/what-is-pii-personal-data/, 2020.

[99] Jasper Tan and Simon GM Koo. A survey of technologies in internet of things. In *2014 IEEE International Conference on Distributed Computing in Sensor Systems*, pages 269–274. IEEE, 2014.

[100] Ali Tekeoglu and Ali Saman Tosun. Investigating security and privacy of a cloud-based wireless ip camera: Netcam. In *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–6. IEEE, 2015.

[101] Ali Tekeoglu and Ali Şaman Tosun. A testbed for security and privacy analysis of iot devices. In *2016 IEEE 13th International Conference on Mobile Ad Hoc and Sensor Systems (MASS)*, pages 343–348. IEEE, 2016.

[102] Ilaria Torre, Giovanni Adorni, Frosina Koceva, and Odnan Sanchez. Preventing disclosure of personal data in iot networks. In *2016 12th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*, pages 389–396. IEEE, 2016.

[103] Ding Wang, Debiao He, Haibo Cheng, and Ping Wang. fuzzypsm: A new password strength meter using fuzzy probabilistic context-free grammars. In *2016 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 595–606. IEEE, 2016.

[104] Ding Wang, Zijian Zhang, Ping Wang, Jeff Yan, and Xinyi Huang. Targeted online password guessing: An underestimated threat. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 1242–1254. ACM, 2016.

[105] He Wang, Ted Tsung-Te Lai, and Romit Roy Choudhury. Mole: Motion leaks through smartwatch sensors. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 155–166, 2015.

[106] Rolf H Weber. Internet of things–new security and privacy challenges. *Computer law & security review*, 26(1):23–30, 2010.

[107] Jacob West, Tadayoshi Kohno, David Lindsay, and Joe Sechman. Wearfit: Security design analysis of a wearable fitness tracker. *IEEE Center for Secure Design*, 2016.

[108] Shomir Wilson, Florian Schaub, Aswarth Abhilash Dara, Frederick Liu, Sushain Cherivirala, Pedro Giovanni Leon, Mads Schaarup Andersen, Sebastian Zimmeck, Kanthashree Mysore Sathyendra, N Cameron Russell, et al. The creation and analysis of a website privacy policy corpus. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1330–1340, 2016.

[109] Stephanie Winkler and Sherali Zeadally. Privacy policy analysis of popular web platforms. *IEEE Technology and Society Magazine*, 35(2):75–85, 2016.

[110] Jessica D Young. Commitment analysis to operationalize software requirements from privacy policies. *Requirements Engineering*, 16(1):33–46, 2011.

[111] Jessica D Young and Annie I Anton. A method for identifying software requirements based on policy commitments. In *2010 18th IEEE International Requirements Engineering Conference*, pages 47–56. IEEE, 2010.

[112] Tianlong Yu, Vyas Sekar, Srinivasan Seshan, Yuvraj Agarwal, and Chenren Xu. Handling a trillion (unfixable) flaws on a billion devices: Rethinking network

security for the internet-of-things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, pages 1–7, 2015.

[113] Kai Zhao and Lina Ge. A survey on the internet of things security. In *2013 Ninth international conference on computational intelligence and security*, pages 663–667. IEEE, 2013.

[114] Sebastian Zimmeck, Ziqi Wang, Lieyong Zou, Roger Iyengar, Bin Liu, Florian Schaub, Shomir Wilson, Norman Sadeh, Steven Bellovin, and Joel Reidenberg. Automated analysis of privacy requirements for mobile apps. In *2016 AAAI Fall Symposium Series*, 2016.