



**University of  
Zurich**<sup>UZH</sup>

# **Portable Internet of Things (IoT) Security Scoring Application**

*Qianhui Wang  
Zurich, Switzerland  
Student ID: 21-740-410*

Supervisor: Katharina O. E. Muller, Dr. Bruno Rodrigues  
Date of Submission: October 26, 2023



# Abstract

Since the release of AirTag in 2021, consumers have been concerned about the security of personal tracking devices. Thus, HomeScout was developed primarily focusing on tracker detection and filtering. In light of the pervasive prevalence of Bluetooth devices nowadays, it has become imperative to expand the scope of HomeScout from trackers to encompass a broader range of everyday Internet of Things (IoT) devices. Unfortunately, there is limited research on this domain. Some concentrate on IoT devices connected via the Internet, while some primarily offer security guidelines without articulating specific scoring criteria. To bridge this gap, this research designs a three-level security scoring criteria tailored for IoT devices connected via Bluetooth Low Energy (BLE). This guideline draws its foundations from the fundamental structure of BLE advertising packets and the information carried within it. Furthermore, the research extends the HomeScout prototype to incorporate a general security assessment functionality for the neighboring devices, grounded in alignment with the proposed scoring criteria. The practical application of these criteria is validated through a series of experimental trials. After analyzing these experimental results, discernible insights emerge regarding the security levels of BLE devices existing within real-world environments. The findings reveal that more than half of the assessed devices exhibit a moderate level of security. Among the remaining devices, the number of highly secure devices surpasses the count of those classified as low-security ones.

Seit der Veröffentlichung des AirTags im Jahr 2021 sind die Verbraucher zunehmend besorgt über die Sicherheit von persönlichen Ortungsgeräten. Aus diesem Grund wurde das HomeScout entwickelt, das sich insbesondere auf die Erkennung und Filterung solcher Tracker konzentriert. Angesichts der weitreichenden Verbreitung von Bluetooth-Geräten in der heutigen Zeit und ihrer Schwachstellen ist es notwendig geworden, den Anwendungsbereich von HomeScout von Trackern auf eine breitere Palette an alltäglichen Internet der Dinge (IoT) Geräten auszuweiten. Leider ist die Forschung in diesem Bereich noch begrenzt. Einige Forschungsarbeiten befassen sich mit IoT-Geräten, die mit dem Internet verbunden sind, während andere hauptsächlich Sicherheitsrichtlinien ohne konkrete Bewertungskriterien vorlegen. Um diese Wissenslücke auszufüllen, wurde in dieser Forschungsarbeit ein dreistufiges Sicherheitsbewertungssystem speziell für IoT-Geräte entwickelt, die über die Bluetooth Niedrigenergie (BLE) kommunizieren. Diese Richtlinie basiert auf der grundlegenden Struktur der BLE-Werbepakete und den darin übertragenen Informationen. Darüber hinaus erweitert die Forschungsarbeit den HomeScout-Prototyp um eine allgemeine Sicherheitsbewertungsfunktion für die Nachbargeräte, die sich an den vorgeschlagenen Bewertungskriterien orientiert. Die praktische Anwendung dieser Kriterien wird durch eine Reihe von experimentellen Versuchen validiert. Nach der Auswertung dieser experimentellen Ergebnisse ergeben sich deutliche Erkenntnisse über

das Sicherheitsniveau von BLE-Geräten in realen Umgebungen. Die Ergebnisse deuten darauf hin, dass mehr als die Hälfte der untersuchten Geräte ein moderates Sicherheitsniveau aufweisen. Unter den verbleibenden Geräten sind mehr hochsichere Geräte als solche, die als gering gesichert eingestuft wurden.

# Acknowledgments

First and foremost, I would like to express my sincere gratitude to K for her continuous support throughout this work. As my supervisor, she consistently dedicated her time to engaging in insightful discussions and offering invaluable, constructive feedback. I am also deeply appreciative of the assistance provided by B during the final stages of this project. It has been a truly enriching experience to do this thesis within the Communication Systems Research Group (CSG) at the Department of Informatics, University of Zurich. The topic is exceptionally captivating. Lastly, I would like to express my sincere appreciation to my parents for their unwavering support throughout my academic pursuits. Their encouragement has been a constant source of motivation and power.



# Contents

<b>Abstract</b>	<b>i</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivation . . . . .	2
1.2 Description of Work . . . . .	2
1.3 Thesis Outline . . . . .	3
<b>2 Related Work</b>	<b>5</b>
<b>3 Background</b>	<b>9</b>
3.1 BLE Technology . . . . .	9
3.1.1 Overview Architecture . . . . .	9
3.1.2 Link Layer . . . . .	11
3.2 IoT . . . . .	16
3.3 Security Scoring . . . . .	17
3.3.1 Existing Security Guidelines and Scoring Systems . . . . .	17
<b>4 Security Criteria Design</b>	<b>21</b>
4.1 Nmap Scanning Results . . . . .	21
4.2 BLE Broadcast Packets . . . . .	23
4.3 Sniffing Results via An Android Smartphone . . . . .	29
4.4 Exploration of Open Source Intelligence (OSINT) . . . . .	31

4.5	Exploration of MUD . . . . .	33
4.6	Scoring Criteria . . . . .	33
4.6.1	Criteria Features . . . . .	33
4.6.2	Device Scoring . . . . .	36
4.6.3	Security Levels . . . . .	38
<b>5</b>	<b>Implementation</b>	<b>41</b>
5.1	Build Configurations . . . . .	41
5.2	Workflow of the Extended Function . . . . .	41
5.3	Architecture . . . . .	42
5.3.1	User Interface (UI) . . . . .	42
5.3.2	Data Exchange . . . . .	43
5.3.3	Security Scoring Function . . . . .	44
<b>6</b>	<b>Evaluation and Results</b>	<b>49</b>
6.1	Evaluation of the Single Known Device . . . . .	49
6.2	Evaluation in Real Environments . . . . .	53
6.2.1	The Experiment in A Residential District . . . . .	53
6.2.2	The Experiment in A University Building . . . . .	55
6.2.3	The Experiment in A Railway Station . . . . .	56
<b>7</b>	<b>Conclusion and Future Work</b>	<b>59</b>
7.1	Conclusion . . . . .	59
7.2	Limitation and Future Work . . . . .	60
	<b>Abbreviations</b>	<b>69</b>
	<b>Glossary</b>	<b>71</b>
	<b>List of Figures</b>	<b>71</b>
	<b>List of Tables</b>	<b>75</b>
<b>A</b>	<b>Experiments</b>	<b>79</b>



# Chapter 1

## Introduction

Since the emergence of the AirTag in 2021, consumers have been concerned about potential vulnerabilities associated with Internet of Things (IoT) devices, particularly in scenarios involving personal tracking [34]. Consequently, several mobile applications, including AirGuard [33] and HomeScout, were developed to identify and assess potentially risky devices. With the previous development of the HomeScout application [50], it has evolved to detect and filter out various personal trackers, such as the AirTag, Tile, Samsung Galaxy SmartTag+, Chipolo One Spot, and the OpenHaystack tag. This application allows users to identify personal tracking devices, regardless of time or location.

Moreover, an in-depth analysis of the scanned Bluetooth Low Energy (BLE) devices and the data encapsulated within transmitted packets has revealed that BLE-based IoT devices can also be utilized for object or people tracking. In contemporary society, the widespread utilization of Bluetooth devices, such as Bluetooth headphones and smart-watches, has become a pervasive facet of daily life. These devices predominantly employ BLE technology for communication. However, the BLE protocol suffers from some security and privacy susceptibilities. Consequently, BLE devices are susceptible to potential threats from malicious attackers, such as eavesdropping and replay attacks. Furthermore, it is vital to recognize that BLE devices possess the capacity to compromise user privacy. Hostile attackers can leverage these devices to track users or formulate conjectures about users' activities by closely analyzing the communication patterns of BLE devices. Therefore, it is important to expand the horizons of HomeScout beyond personal trackers towards a diverse range of IoT devices that permeate various aspects of daily life.

One of the biggest challenges in IoT is ensuring the security of a highly diverse IoT network, which results from a heterogeneous landscape of devices, implementations, manufacturers, and communication protocols. Given the sheer magnitude of potential combinations, devising a one-size-fits-all solution becomes a nearly impossible task. Consequently, researchers have advocated for adopting open standards, exemplified by initiatives like IoTAG [25], which support the inclusion of comprehensive security details within transmitted packets. However, adopting such standards hinges on manufacturers' willingness to implement the changes, resulting in a sluggish pace.

Furthermore, there have been endeavors to define general ontologies [52], [30], with a particular emphasis on security [6] or privacy [26], to elucidate the fundamental aspects

demanding consideration. Nevertheless, this approach often struggles with the transition from theoretical to practical [57].

Another methodology involves analyzing security aspects through a passive sniffing approach, exemplified by IoTHunter [38]. This approach relies on network traffic analysis and device-specific keyword identification to categorize IoT devices. By integrating security assessment with the detection or classification of devices within mobile applications, the security evaluation process may attain newfound levels of accessibility and efficiency.

Existing research on the security scoring of IoT devices is limited. Some studies only propose a reference framework for security assessment without giving specific scoring criteria[54], [9]. The proposed detailed scoring guidelines in current studies are for specific IoT devices[15], [61] or generally applicable to IoT devices relying on an Internet connection[45], [25], ignoring BLE devices. To fill the gap, this thesis designs security scoring criteria suitable for unknown BLE devices and implements an extended functionality on the HomeScout prototype based on the proposed criteria.

## 1.1 Motivation

This thesis aims to enhance the capabilities of HomeScout, enabling it to serve as a tool for conducting security evaluations of BLE devices in general. The motivation behind leveraging the Android framework to develop a security rating application lies in exploring an approach that can be universally applied to assess the security of BLE devices. The primary questions encountered in achieving a credible security assessment are:

- What kind of information is accessible via a smartphone?
- What is the significance of the acquired data in the context of security?
- Is it feasible to use the obtained information for security scoring?

## 1.2 Description of Work

The main contributions of this work are:

1. Investigation on the knowledge of IoT, BLE, Kotlin, and the current state-of-the-art security scoring criteria solutions, regarding security, IoT and BLE.
2. Design the security scoring criteria for BLE devices.
3. Implement security scoring as an extended function of HomeScout.
4. Conduct experiments to verify the feasibility and rationality of the prototype and analysis of the experiment results.

This Master Thesis consists of three distinct phases. The initial phase is dedicated to research, during which related literature is explored, and essential background knowledge is acquired. Subsequently, the second phase centers on the design process, formulating security scoring criteria, conceptualizing a prototype extension for the HomeScout application, and planning experiments to evaluate both the security scoring methodology and the prototype. Finally, the third phase involves the practical execution of the project, entailing the implementation of the prototype extension and the execution of experiments for subsequent evaluation.

The first step is to become familiar with the BLE protocol, IoT security, and different existing security scoring criteria for IoT or specifically for BLE. To maintain all pertinent data and security scoring, it is also necessary to look into practical data forms, such as Manufacturer Usage Description (MUD) files [10], and prospective expansions or extensions. In this stage, it is critical to understand the theoretical components of this topic and the data available through the HomeScout BLE scanner to allow for the conceptualization of feasible security scoring criteria, which can later be prototyped and added to the HomeScout application. It's also necessary to obtain knowledge of the Kotlin programming language and the existing HomeScout application.

In the second stage, utilizing the knowledge gained from stage one is vital to determine which BLE data and metadata are most significant and appropriate for a security evaluation. Herein, it is essential to distinguish between data that is useful and accessible from the BLE data package frame and data that is genuinely accessible depending on the results of the BLE scanning via an Android smartphone. These details will then be used to develop the security scoring criteria and prototypical security evaluation, considering the constraints based on theoretical and practical data availability. Therefore, this stage includes establishing the security scoring criteria and designing the HomeScout extension prototype.

The third stage implements a security scoring functionality according to the proposed criteria for surrounding BLE devices based on the HomeScout prototype. After the implementation, several meticulously designed experiments are conducted to assess the feasibility and the rationality of the applied security scoring criteria in this stage. The paperwork and a final thesis report will be developed along the three phases.

## 1.3 Thesis Outline

This thesis is structured as follows: chapter 2 presents related work in IoT, BLE, and security. Chapter 3 illustrates the basic knowledge about BLE, IoT, and existing security scoring criteria. Chapter 4 describes the security rating indicators and final criteria proposed in this research. Chapter 5 presents the details of the implemented extended HomeScout prototype. Chapter 6 describes the experiments, the corresponding results, and their analysis. Chapter 7 discusses this thesis's findings, limitations, and future work.



# Chapter 2

## Related Work

BLE is widely used in IoT and smart devices for communication due to its benefits of ultra-low power consumption, ease of development, and compatibility with ubiquitous smartphones. Many studies have investigated various aspects of BLE technology, including its performance, features, and applications. [36] measures the power consumption of a peripheral device in BLE connection running version 1.2 of BLE stack and calculates the lifetime of the battery of this device. In addition, [63] concludes that when it comes to the amount of data sent per joule used, BLE is really incredibly energy-efficient by comparing the energy consumption of real BLE devices and ZigBee/802.15.4 devices. The overhead is acceptable even for IPv6-based BLE communication. Through simulation studies, [28] outlines BLE's features and specific performance, and promising applications. [37] conducts a comparative analysis between BLE4 and BLE5, and evaluates the communication performance of BLE in the presence of ZigBee interference. [22] explores the performance of BLE mesh networks, estimating aspects like encapsulation overhead and energy consumption. [44] provides a comprehensive overview of BLE, highlighting key features such as reachability range, transmission delay, and power consumption.

In addition to research on BLE performance and features, studies have also delved into its applications, for example, indoor localization [12], [14], [27] and contact tracking [20], [32], [60]. [27] investigates the utilization of BLE devices, modern smartphones carried by visitors, to estimate the location of museum visitors. [12] employs a combination of indoor sensors and BLE beacons attached to users to achieve indoor localization. Furthermore, [31] proposes an angle detection method based on BLE for indoor localization, and [14] introduces a BLE-based indoor localization system that utilizes natural-inspired optimization algorithms in logistics warehouse environments.

Additionally, BLE has been utilized in contact tracing applications during epidemic outbreaks [20], [32], [60]. [20] describes the implementation of contact tracing through BLE communication. Moreover, [60] and [32] also explore the use of BLE for contact tracing in the context of epidemics.

Many researchers have attempted to apply BLE in different IoT applications such as healthcare, transportation, and smart homes due to its low-energy and wireless nature. [67] presents a thorough overview of BLE applications in healthcare systems, along with

a list of commercial medical devices that leverage BLE technology. [53] offers a solution wherein data is transmitted from an electric toothbrush to a mobile health application using BLE to support oral healthcare decision-making processes. [62] proposes a method based on BLE to continuously monitor a user's gait speed, aiming to detect early signs of severe cognitive diseases. [11] explores the potential of BLE in optimizing patient turnaround time, which refers to the duration between a patient's arrival at the hospital and their departure. Additionally, [21] develops a tailored BLE system designed to quantify healthcare worker proximity networks and patient close contact and conducts a pilot study within a hospital setting, highlighting the feasibility of the proposed system for monitoring healthcare worker-patient interactions. Furthermore, research endeavors have explored the application of BLE in the domain of vehicles and transportation. [66] devises a vehicle control system that empowers users to access vehicle data acquired through sensors and manipulate the car using a BLE connection facilitated by their smartphones. [23] develops a wearable solution focused on detecting and warning about blind spots in vehicles. [48] designs a system to assist individuals with visual impairments, specifically those who are blind, in utilizing public transportation. [55] proposes an innovative system aimed at optimizing transportation operations. By employing BLE beacons and tablets, data pertaining to transportation routes is collected and subsequently analyzed to identify areas for improvement. [41] employs BLE technology to enhance IoT connectivity in smart homes. [64] focuses on developing a smart home system using BLE mesh networking, which reduces restoration costs while maintaining performance. Furthermore, [18], [19] introduce two energy management approaches based on BLE technology for smart homes to reduce electricity consumption costs and enhance user comfort. To validate these methods, they perform simulation experiments, ensuring their effectiveness in practice. Similarly, [42] presents a BLE-based energy efficiency management system that effectively addresses energy consumption concerns within the context of smart homes. Additionally, [24] introduces a smart home power management system that utilizes BLE and WiFi sensor devices to optimize the management of the electricity consumption of plug-in electric vehicles and home appliances.

However, as BLE is used more frequently in a variety of contexts, including medical systems [67], [53], [62], [11], [21], and smart homes [41], [64], [18], [19], [42], [24] concerns regarding IoT security and data privacy have been raised. [56] executes four attacks on each of the three Bluetooth devices, including bluejacking, blue-smacking, bluesnarfing, and social engineering, and finds that hackers can easily perform Distributed Denial-of-Service (DDoS) attacks, resource misappropriation, and message modification on Bluetooth devices. [58] introduces an extended calculation formula for authentication to assess vulnerabilities in IoT systems utilizing BLE wireless networks, revealing the high vulnerability of such networks. Furthermore, [40] reviews the latest specifications related to BLE authentication, describing the vulnerabilities and possible attacks on authentication. [17] highlights the weaknesses in BLE device communication security, particularly regarding the algorithm employed by BLE for key distribution. [59] conducts an extensive survey of BLE applications and outlines BLE protocol security issues like man-in-the-middle attacks and battery depletion attacks. Additionally, [16] provides a systematic overview of BLE security, including the security properties of different BLE versions and features of known weaknesses and attacks. [13] presents a comprehensive survey focusing on insecure implementations of encryption, authentication, and user privacy in BLE, and proposes a

comprehensive classification approach for BLE security and privacy issues. [68] exposes flaws in Media Access Control (MAC) address randomization schemes that can expose devices to replay attacks due to side channels introduced by the whitelist. [40] focuses on the threats faced by BLE mesh.

To help people, regardless of their level of expertise, to understand whether the IoT devices they use are secure, developing an IoT security scoring system is a valuable approach. The scoring system assigns a score or a level to reflect the severity of device security vulnerabilities, allowing people to determine the proactive measures and their priorities to mitigate risks. However, to my knowledge, there is limited research on IoT security scoring systems. Table 2.1 summarizes related references. [45] proposes a 3-level scale for security assessment, encompassing four dimensions: data confidentiality, data integrity, access control, and reflection attacks originating from IoT devices. Nevertheless, this scale lacks component weights and an integrated formula for calculating the overall device score. [15] proposes a metric to measure the security level of a device based on the Common Vulnerability Scoring System (CVSS) score and the information about known vulnerabilities. [25] conducts network scans using Nmap to obtain a list of device host names and the services provided by the open ports. This data is utilized to evaluate the security of functions like encryption and update behavior, and produce overall numerical ratings. [61] uses CVSS to calculate vulnerability scores for IoT devices. [54] proposes a framework for risk assessment of IoT devices that uses subjective logic to infer relevant vulnerabilities under uncertainty combining with the information such as type of device, manufacturer, and firmware version and publicly available resources and determined severity levels based on the vulnerability's CVSS score. [8] proposes a framework based on the Markov chain and CVSS to compute vulnerability scores of attacks for medical IoT devices. [9] designs a vulnerability quantification framework for IoT systems based on CVSS with the combination of threat intelligence and machine learning models.

Ref.	Quantification Framework Used	Pros	Cons
[45]	Own criteria from 4 aspects	Security and privacy threats were both considered	Lack the weights for 4 components and the overall scores for devices
[15]	Based on CVSS score and known vulnerabilities	Provides overall scores of IoT devices	Evaluate the security level of known devices that cannot be easily applied to unknown devices
[25]	Own criteria based on Nmap scanning results	Non-technician friendly security scoring criteria for IoT devices	Network scans may not provide enough information
[61]	Use CVSS scores directly		
[54]	Based on publicly available vulnerabilities information and CVSS scores	Non-technician friendly security scoring system for IoT devices	Uncertainty about the reliability of firmware information inferences
[8]	Based on Markov chain and CVSS scores	12 attacks in IoMT environment were considered	Only provides scores for specific attacks instead of vulnerabilities
[9]	Based on CVSS with the combination of threat intelligence and machine learning models	A security scoring framework better suited to IoT devices	Need for analysts specialized in different fields to set up the proposed indicators

Table 2.1: A summary of references related to security scoring system.



# Chapter 3

## Background

This chapter introduces BLE technology, IoT, and security scoring systems that are currently applied.

### 3.1 BLE Technology

The official Bluetooth Special Interest Group (SIG) BLE Primer, Bluetooth Core Specification Version 5.4, and Heydon's The Developer's Handbook serve as the foundation for the technical introduction of BLE in this section.

Since the 1990s, SIG has managed the Bluetooth specification. The original plan was to create a system that could take the role of cables when it came to short-range data transmission between devices. It was soon used in devices such as wireless mice. As the technology developed, it was applied in a growing variety of applications, such as file transfers and wireless headsets. Both Bluetooth 2.0 and Bluetooth 3.0 are accelerating data rates to accommodate the faster data transmission rates required by these novel application scenarios. BLE, on the other hand, goes in another direction. Instead of focusing on quick speeds or massive data throughput, it focuses on ultra-low energy consumption. Because of this, BLE helps small, wireless devices that send status data, such as sensors and embedded medical device equipment, keep available for extended periods of time with no need for frequent battery replacement.

#### 3.1.1 Overview Architecture

The BLE architecture is composed of three components, as illustrated in Figure 3.1: application, host, and controller. The controller is often a physical device that sends and receives radio waves and demodulates the waves to retrieve information packets. The host, a software stack, manages numerous services and enables device connectivity. Applications implement a use case by using the software stack and, consequently, the controller. This

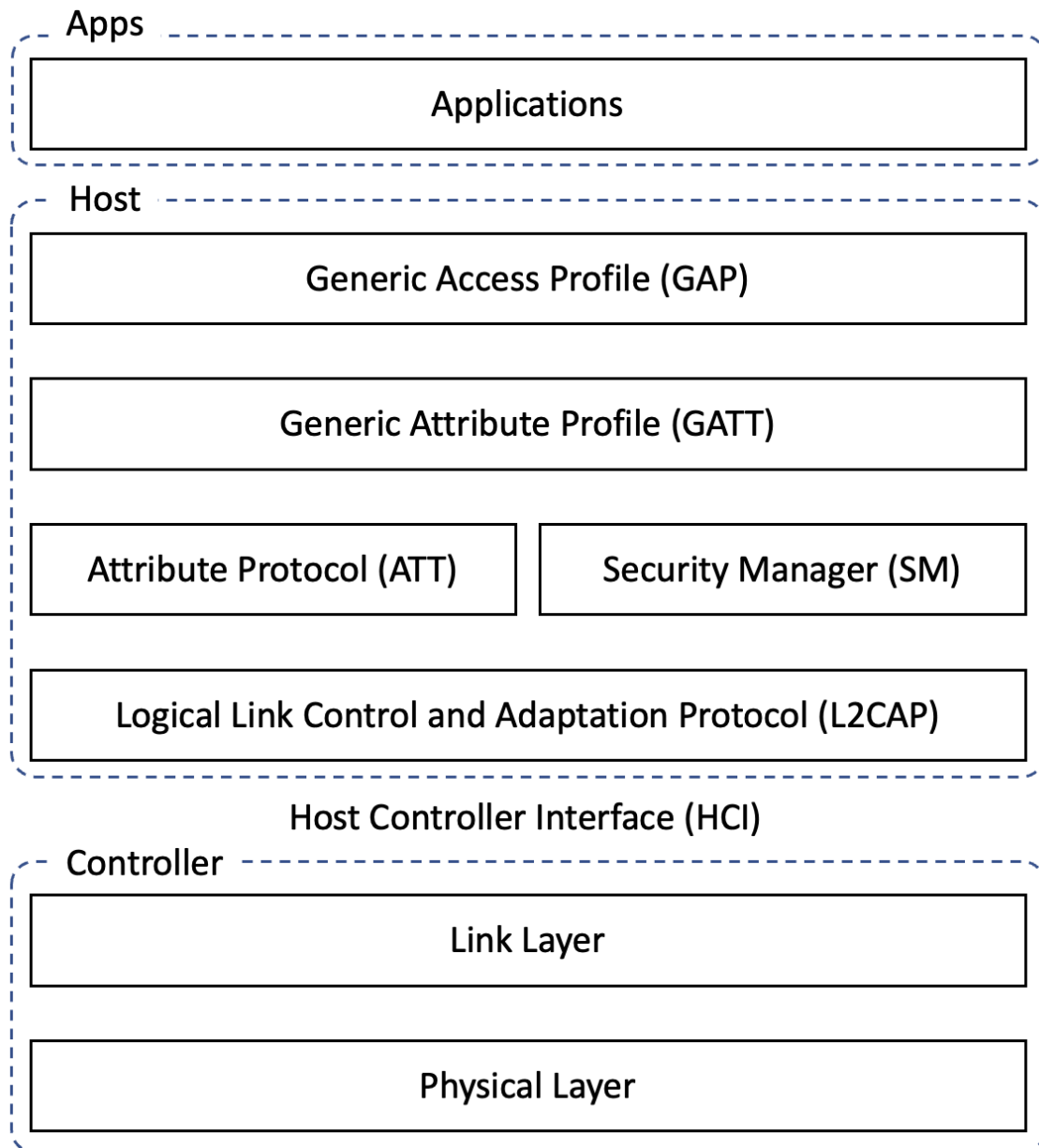


Figure 3.1: The architecture for BLE. Combined with the figure 3-1 from [35] and the figure 2 from [65].

three-tiered BLE architecture provides an effective and scalable foundation for IoT devices to run flawlessly in a variety of settings.

At the very bottom of the architecture is the controller part, which encompasses the Physical Layer (PHY) and the Link Layer. Above it is the host part containing three vital protocols: the Logical Link Control and Adaptation Protocol (L2CAP), the Attribute Protocol (ATT), and the Security Manager (SM). Moreover, the host layer accommodates the Generic Attribute Profile (GATT) and Generic Access Profile (GAP), which collectively enhance the functionality and interoperability of BLE devices. Facilitating compatible communication between the controller and the host, the Host Controller Interface (HCI) serves as a logical interface, offering available bi-directional communication between the host and the controller, even if they are from different manufacturers.

The PHY defines all parts of BLE technology related to the utilization of radio, for example, radio frequency. The three PHYs, LE 1 Megabit PHY (LE 1M), LE 2 Megabit PHY (LE 2M), and LE Coded, which are predetermined configurations of PHY parameters, are available demodulation schemes. In addition to defining packet formats, bit stream processing procedures, and protocols for over-the-air communication and link control, the Link Layer also specifies the various methods for communication using radio. L2CAP is in charge of segmenting and reassembling data units between its lower layer (the Link Layer) and its upper layer (the Attribute Protocol and the Security Manager). The Security Manager specifies a straightforward protocol for distribution and pairing. ATT clients can access and use information in the server's attribute table thanks to GATT. Furthermore, based on the fundamental attributes in the attribute table, GATT defines high-level data types. GAP defines operational modes and procedures, security levels and modes, and some user interface standards for applications.

### 3.1.2 Link Layer

The Link Layer plays a pivotal role in facilitating data communication and ensuring the integrity of transmitted packets. The detailed information on the Link Layer is helpful in augmenting our understanding of the communication methods and data formats transmitted by BLE devices, thus holding crucial significance for this research.

#### State Machine

The Link Layer operates according to a well-defined state machine with seven states, as shown in Figure 3.2. The descriptions of each state are summarised in Table 3.1. During the establishment of a link, two distinct roles come into play: the central role, which assumes to initiate the connection, and the peripheral role, which passively awaits the connection. In typical scenarios, smartphones often assume the central role, while smart home devices, smartwatches, Bluetooth headsets, and other similar devices take on the peripheral role. In the absence of a connection, the peripheral one operates in a broadcast state, transmitting advertising packets. When the central one seeks to establish a connection with a peripheral role, it responds to the advertising packet, switching from

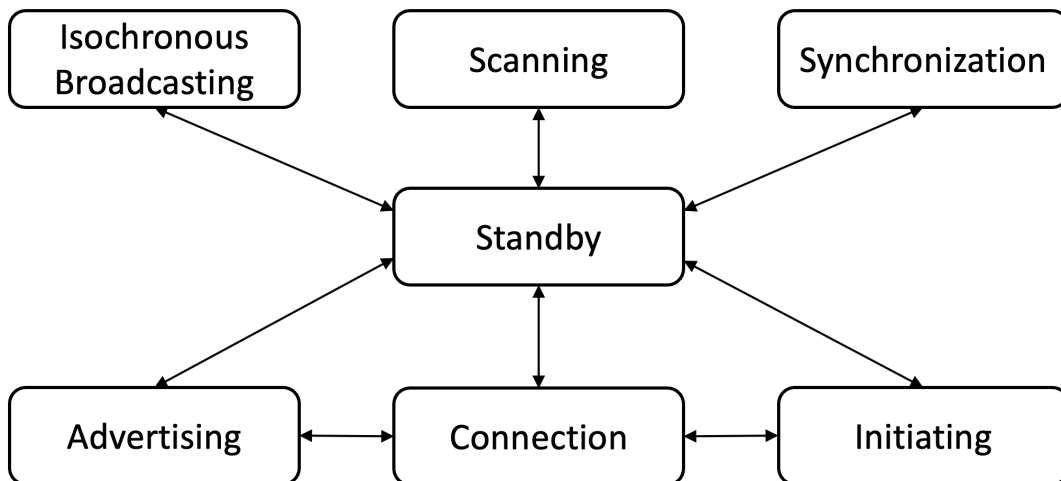


Figure 3.2: The Link Layer state machine. Source: Figure 9 from [65].

Standby	Device neither transmits nor receives packets.
Initiating	Responds to advertising packets from a particular device to request a connection.
Advertising	Transmits advertising packets and potentially processes packets sent in response to advertising packets by other devices.
Connection	In a connection with another device.
Scanning	Listening for advertising packets from other devices.
Isochronous Broadcasting	Broadcasts isochronous data packets.
Synchronization	Listens for periodic advertising belonging to a specific advertising train transmitted by a particular device.

Table 3.1: The descriptions of seven states. Source: Table 2 from [65].

the initiating state to the connection state. Concurrently, the peripheral role receives and processes the response packets dispatched by the central role, transitioning its state to a connection state.

## Packets

The Link Layer exists two distinct packet types: one employed by the uncoded PHYs, LE 1M and LE 2M, and the other utilized by the LE coded PHYs. The packet structure adopted by the LE coded PHYs is too intricate. The focus of this work is the packet structure utilized by the uncoded PHYs.

The packet structure, as illustrated in Figure 3.3, comprises several key components: the preamble, access address, Protocol Data Unit (PDU), and Cyclic Redundancy Check (CRC). The preamble, a very simple alternating sequence, serves the purpose of precise

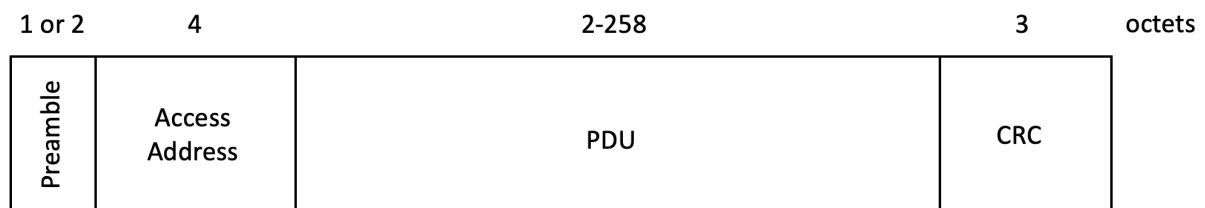


Figure 3.3: The packet structure of uncoded PHYs. Source: Figure 7 from [65].

signal frequency synchronization, automatic gain control, and symbol timing estimation. By analyzing the access address field, the receiver determines whether the packet is pertinent to itself or not, thereby determining if the packet should be received. Three distinct types of PDUs exist: the advertising physical channel PDU, the data physical channel PDU, and the isochronous PDU. While the advertising physical channel PDU solely encompasses the protocol header and payload, the latter two types consist of three fundamental elements: the protocol header, payload, and message integrity check. Finally, the CRC, situated at the end of the packet, functions as an error-detection mechanism, identifying error bits that may have been received due to transmission-related issues.

## Channels

BLE operates within the 2.4GHz band, which is subdivided into a total of 40 channels. Among these channels, three are defined as advertising channels, while the remaining 37 channels serve as general-purpose or data channels. The advertising channels are exclusively employed for broadcasting packet transmission. In contrast, the 37 general-purpose channels are primarily utilized for a majority of communication purposes, including a wide range of data exchange scenarios.

## Address

The address of a BLE device can be used as an identifier. There are two types of device addresses: public device addresses and random device addresses. Public device addresses are assigned by the Institute of Electrical and Electronics Engineers (IEEE). On the other hand, random device addresses are generated randomly in compliance with Bluetooth Core Protocol specifications.

Random device addresses encompass three variations, namely static, non-resolvable private, and resolvable private addresses. The two Most Significant Bits (MSB) of the address, located at the rightmost part of the address, indicate the address type.

As shown in Figure 3.4, the static address, except for the last two MSBs which are both 1, the remaining 46 bits are randomly generated. They are not disguised and do not provide device privacy. They remain unchanged unless regenerated upon power failure or reboot. Figure 3.5 illustrates the format of a non-resolvable address.

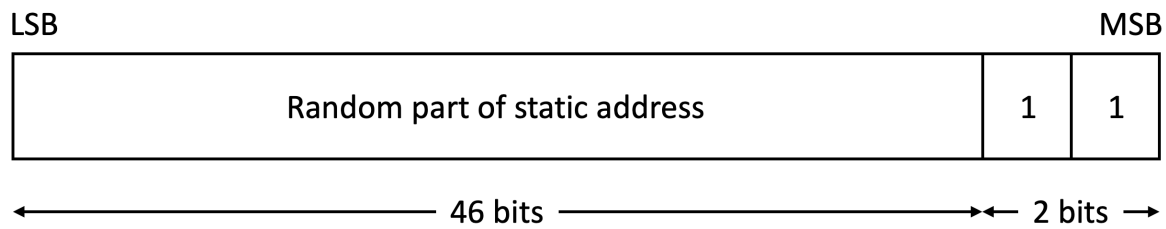


Figure 3.4: Format of a static address, where LSB is the abbreviation of the least significant bit, and MSB represents the most significant bit. Source: Figure 1.2 from [29].

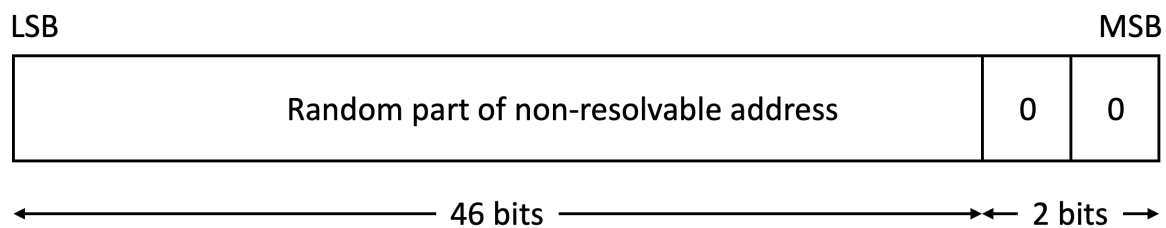


Figure 3.5: Format of a non-resolvable private address, where LSB is the abbreviation of the least significant bit, and MSB represents the most significant bit. Source: Figure 1.3 from [29].

Similar to static addresses, a non-resolvable private address contains 46 randomly generated bits, but with different MSBs "00". They can be used to protect the privacy of Bluetooth devices and cannot be resolved by any other device.

Unlike non-resolvable addresses, resolvable addresses protect the privacy of the Bluetooth device while allowing one or more trusted parties to identify and resolve the device. As indicated in Figure 3.6, a resolvable private address consists of two components: a hash value and a prand. The prand consists of 22 bits randomly generated and the two MSBs. The hash value is computed by applying the random address function, as defined in the Bluetooth Core Specification, to the prand and the device's identity resolution key (IRK). Consequently, trusted entities possessing the IRK for a device with a resolvable private address have the capability to compare the hash value within the address against the one calculated through the random address function using the corresponding IRK and prand. A successful match signifies the resolution of the device's identity.

## LE Advertising Broadcast

LE Advertising Broadcast is a connectionless communication mode. In general, advertising packets can be received by all scanning devices in range, facilitating one-to-many communication scenarios. This mechanism can be used by a peripheral role to broadcast

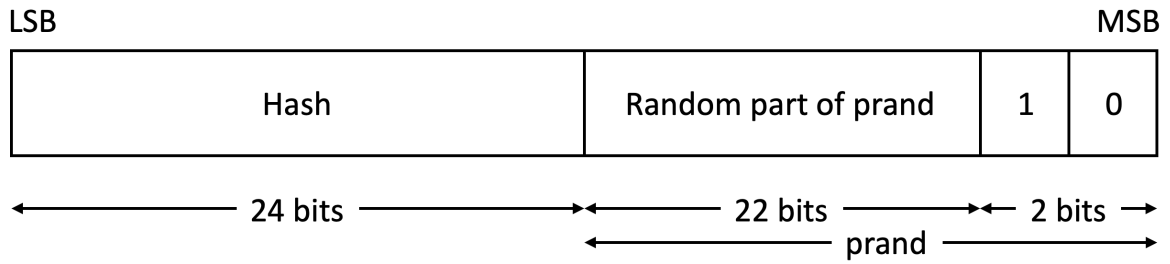


Figure 3.6: Format of a resolvable private address, where LSB is the abbreviation of the least significant bit, and MSB represents the most significant bit. Source: Figure 1.4 from [29].

its own state to an unknown central device, providing information for the central role to initiate a connection.

Furthermore, a special type of advertising allows one device to advertise specifically to another device. A device can send an advertising packet containing its own Bluetooth address and the Bluetooth address of the target device. When the target device scans the advertising packet containing its own address, it responds by sending a connection request to the sender of the advertisement, paving the way for further communication between the devices.

Advertisements are generally considered unreliable transmissions because the receiver does not send a confirmation of receipt to the broadcaster after accepting the advertising packet. As a result, the sender lacks a mechanism to ensure the successful and accurate reception of the message.

In the Bluetooth Core Specification, there are two types of advertising: legacy and extended. Legacy advertising uses three dedicated channels (numbered 37, 38, and 39) and applies the LE 1M PHY for transmission. These channels, referred to as the primary advertising channels, cannot be used simultaneously in a single advertising event. However, they can be utilized in any sequential order as needed in most advertising scenarios without special requirements. The Bluetooth core specification defines seven types of PDUs for legacy advertising, which can be used for connectable or non-connectable, scannable, or non-scannable advertising events, respectively. Detailed information about the PDUs is summarized in Table 3.3. The PDU type indicates whether active scanning is allowed, the requests to which the receiver will further reply, and whether a connection can be established.

Extended advertising introduces secondary channels numbered 0-36, expanding the capacity for broadcasting larger data volumes. In contrast to legacy advertising, which exclusively utilizes the primary advertising channel for transmitting all advertising packets, extended advertising employs a different approach. Specifically, it transmits only the header data, including a significant component called AuxPtr, over the primary advertising channel. The payload, which constitutes the majority of the data, is subsequently transmitted over the general-purpose channels 0-36, also known as the secondary chan-

<b>PDU Name</b>	<b>Description</b>	<b>Transmitted By</b>	<b>Scannable</b>	<b>Connectable</b>
ADV_IND	Undirected advertising	P	Yes	Yes
ADV_DIRECT_IND	Directed advertising	P	No	Yes
ADV_NONCONN_IND	Undirected, non-connectable, non-scannable advertising	P	No	No
ADV_SCAN_IND	Undirected, scannable advertising	P	Yes	No
SCAN_REQ	Scan request	C	N/A	N/A
SCAN_RSP	Scan response	P	N/A	N/A
CONNECT_IND	Connect response	C	N/A	N/A

Table 3.2: The detailed information of seven PDUs used for legacy advertising, where P represents Peripheral and C means Central. Source: Table 4 from [65].

nels. The AuxPtr field serves as a reference to an associated auxiliary packet, facilitating the receiver’s navigation to the secondary channel for payload retrieval.

## 3.2 IoT

IoT is considered a global network that enables the seamless connectivity of various devices, ranging from everyday objects like wearables and household appliances to industrial machinery and infrastructure systems. These interconnected devices, operating independently and having unique addresses, engage in intelligent data exchange over wired or

<b>PDU Name</b>	<b>Description</b>	<b>Channels</b>	<b>PHY(s)</b>	<b>Transmitted By</b>
ADV_EXT_IND	Extended advertising	Primary	1M,Coded	P
AUX_ADV_IND	Subordinate extended advertising	Secondary	All	P
AUX_SCAN_REQ	Auxiliary scan request	Secondary	All	C
AUX_SCAN_RSP	Auxiliary scan response	Secondary	All	C

Table 3.3: The detailed information of four PDUs used for extended advertising, where P represents Peripheral and C means Central; 1M represents LE 1M, Coded is LE Coded, and All indicates LE 1M, LE 2M and LE Coded. Source: A part of Table 5 from [65].



wireless networks [46].

The inherent strength of IoT technologies resides in their capacity for pervasive connectivity, presenting immense potential across various domains [51]. In industrial settings, their implementation enhances productivity by facilitating efficient communication between operators and machines. Similarly, their integration in the development of smart cities fosters an improved quality of life, enabling enhanced convenience and efficiency in daily living. In the field of healthcare, IoT applications contribute to elevated service standards for patients, encompassing aspects such as disease prevention and rehabilitation through wearable devices or embedded sensors.

Nevertheless, the IoT area is not devoid of security concerns. Given the frequent exchange of data among IoT devices, the risk of unauthorized interception and the potential inference or synthesis of sensitive information from seemingly innocuous data [49] exists. For instance, data collected by smart bracelets, such as heart rate, step count, and exercise logs, could inadvertently disclose an individual's health status and lifestyle patterns. Additionally, the compromise of a single device within an IoT network can serve as a launching pad for fraudulent activities aimed at exploiting other network nodes [49].

The increasing prevalence of IoT has permeated various domains of people's lives, facilitating the integration of sensors and enabling the collection of private data. However, the direct exchange of private data among IoT devices introduces a potential vulnerability, as it opens the door for malicious attackers to exploit sensitive information [47]. Consequently, cyber security emerges as a pressing concern within the IoT area. In this context, the implementation of an effective security scoring system becomes essential to promptly identify and assess the severity of potential threats.

### 3.3 Security Scoring

IoT security scoring involves evaluating the security level of an IoT device or system based on predefined criteria and assigning a score or rating that represents its level of security. The purpose of a security scoring system is to offer a standardized and quantitative measure of security, enabling individuals to assess the security of a device, even if they lack the technological expertise to do so.

#### 3.3.1 Existing Security Guidelines and Scoring Systems

Several voluntary guides, best practices, and frameworks have emerged, such as the Internet of Things Security Foundation (IoTSF) and Nessus, with the aim of aiding the establishment of robust security measures that facilitate secure connectivity and foster an environment conducive to heightened security and superior quality within the realm of IoT. These resources serve as invaluable references, offering comprehensive guidance and recommendations to help practitioners focus on and bolster the security of IoT devices, systems, and networks. By adhering to these guidelines, industry and government can

fortify their defenses against potential threats, thereby promoting the integrity, confidentiality, and availability of IoT ecosystems.

### **Cybersecurity Framework (CSF)**

The CSF, crafted by the National Institute of Standards and Technology (NIST), constitutes a compendium of guidelines that advocates a supply and risk-centric methodology for governing cybersecurity risks across various domains, encompassing the realm of the IoT. The framework provides a collection of cybersecurity activities and the corresponding outcomes in easy-to-understand terms, enabling discussions about risk tolerance, task prioritization, and related topics. However, it is noteworthy that the CSF does not provide a formal scoring system for evaluating cybersecurity posture.

### **The Internet of Things Security Foundation (IoTSF)**

The IoTSF is an industry-driven organization. It suggests the IoT Security Compliance Framework, which aims to advance best practices and increase public knowledge of security risks affecting various IoT device and system domains. Notably, the IoTSF framework does not encompass the inclusion of a formal scoring system for the purpose of evaluating security posture.

### **CVSS**

CVSS, a published standard used by organizations worldwide, is not specific to the IoT area but is widely used to assess and rate the severity of computer systems and software vulnerabilities. CVSS was developed by the National Infrastructure Advisory Committee (NIAC) and is managed by the Forum of Incident Response and Security Teams (FIRST). Its primary goal is to convey the characteristics of a vulnerability and provide a quantitative and qualitative representation of its severity.

CVSS incorporates three metric groups, namely basic, temporal, and environmental metrics. The basic metric group evaluates of the inherent characteristics intrinsic to a vulnerability, including factors like access complexity, authentication requirements, and others. These basic features may exhibit fluctuations over time and under diverse environmental conditions. The basic metrics generate an initial numerical score within the range of 0 to 10 (where higher scores signify more severe vulnerabilities). Subsequently, the score is adjusted in light of temporal and environmental metrics. Finally, the numeric score is transformed into a qualitative representation, encompassing classifications such as "critical," "high," "medium," "low," and "none," making it easier to comprehend how serious the vulnerability is.

In essence, CVSS plays a pivotal role in providing a standardized means for organizations to evaluate and comprehend the severity of vulnerabilities present in computer systems and software. By employing a consistent framework, CVSS aids in prioritizing remedial efforts,

ensuring that critical vulnerabilities are promptly addressed and mitigated. In alignment with this idea, VARIOt, an EU project funded by HaDEA's CEF Telecom program, continuously collects information on IoT vulnerabilities and evaluates them through CVSS to create a database dedicated to IoT vulnerabilities. This publicly accessible repository provides brief descriptions of each vulnerability in its database alongside the corresponding CVSS scores and levels. People can search for information about vulnerabilities by vendor, model and version of IoT devices.

### **Nessus**

Nessus is a general-purpose vulnerability assessment software developed by Tenable. Using a wide range of detection technologies and supporting multiple protocols, it can assess both traditional IT assets and dynamic modern attack surfaces, including cloud environments, mobile devices, and unknown external attack surfaces, across different operating systems. Nessus has a large, continuously updated database of known vulnerabilities, including those listed in the Common Vulnerabilities and Exposures (CVE) database. This allows Nessus to compare scanned system characteristics with known vulnerabilities, enabling precise and timely risk assessments. Nessus also offers a scoring system that rates vulnerabilities based on their severity, often using the CVSS as a reference.



# Chapter 4

## Security Criteria Design

This section analyzes the actual captured BLE broadcast packets and the scanning results via Nmap and describes the security scoring criteria and rationale for the design.

### 4.1 Nmap Scanning Results

Inspired by [25], network scans are performed on an Apple MacBookPro laptop using Nmap to collect information about devices and services in a WiFi network. All open Transmission Control Protocol (TCP) and User Datagram Protocol (UDP) ports and their corresponding services are shown in Table 4.1. Hypertext Transfer Protocol (HTTP), Session Initial Protocol (SIP), and File Transfer Protocol (FTP) do not encrypt data during communication. In contrast, all Secure Shell Protocol (SSH) or Hypertext Transfer Protocol Secure (HTTPS) traffic is encrypted and, therefore, more secure than the others.

Services	Port	Comments
FTP	21	Unencrypted
SSH	22	Encrypted
HTTP	80	Unencrypted
HTTPS	443	Encrypted
UPnP	5000	Insecure
SIP	5060	Unencrypted
Domain	53	Domain Name System
MySQL	3306	MySQL Database System
Netconf-tls	6513	Network Configuration Portocol over TLS
afs3-fileserver	7000	File Server
vmware-fdm	8182	VMware Fault Domain Manager
ProRemote	8183	For Remote Control

Table 4.1: Overview of open ports and services.

Furthermore, the analysis of the Nmap scan results unveiled some privacy vulnerabilities. Notably, a distinct privacy concern arises from the names of Apple devices, which consistently include the device name in broadcast packets of the Multicast Domain Name Resolution System (MDNS) protocol, regardless of whether it's a query or response, as shown in Figure 4.1 and Figure 4.2. Most Apple device names contain details like the device's type, as well as the user's first and last names [39], thereby engendering a potential risk to user privacy. While iPads and iPhones employ encryption to safeguard their respective device model numbers, it is noteworthy that the model numbers of MacBook Pros and MacBook Airs remain unencrypted and are presented in plaintext form, as depicted in Figure 4.3. Remarkably, based on the readily available plaintext model identifier, an individual can access comprehensive details concerning the device by referencing the official Apple websites [2], [3]. This information encompasses pertinent specifications such as display size, chip information, the newest compatible operating systems, and so forth. However, this accessible data introduces dual jeopardy, potentially compromising both the security of the device itself and the privacy of the user. For instance, malevolent actors equipped with this knowledge could exploit known vulnerabilities associated with the specific device model, thereby facilitating the execution of targeted attacks.

```

  Answers
  _companion-link._tcp.local: type PTR, class IN, Qianhui的MacBook Pro._companion-link._tcp.local
    Name: _companion-link._tcp.local
    Type: PTR (domain name PoinTeR) (12)
    .000 0000 0000 0001 = Class: IN (0x0001)
    0... .... .... .... = Cache flush: False
    Time to live: 4500 (1 hour, 15 minutes)
    Data length: 24
    Domain Name: Qianhui的MacBook Pro._companion-link._tcp.local

```

Figure 4.1: A part of MDNS packet for answer.

```

  Queries
  QianhuideMacBook-Pro: type ANY, class IN, "QM" question
    Name: QianhuideMacBook-Pro
    [Name Length: 20]
    [Label Count: 1]
    Type: * (A request for all records the server/cache has available) (255)
    .000 0000 0000 0001 = Class: IN (0x0001)
    0... .... .... .... = "QU" question: False

```

Figure 4.2: A part of MDNS packet for query.

```

  ▾ Qianhui的MacBook Pro._device-info._tcp.local: type TXT, class IN
    Name: Qianhui的MacBook Pro._device-info._tcp.local
    Type: TXT (Text strings) (16)
    .000 0000 0000 0001 = Class: IN (0x0001)
    0... .... .... .... = Cache flush: False
    Time to live: 4500 (1 hour, 15 minutes)
    Data length: 41
    TXT Length: 20
    TXT: model=MacBookPro17,1
    TXT Length: 10
    TXT: osxvers=22
    TXT Length: 8
    TXT: icolor=2

```

Figure 4.3: Model information in one of MDNS packets.

However, Nmap is designed for computer operating systems, including Linux, Windows, and Mac OS X [4]. Thus, it is hard to implement this scanning on a mobile phone. In an effort to explore alternative solutions to Nmap on an Android smartphone, the "Nmap Wrapper for Android" application within the Play Store, the official application store for Android (Realme) phones, is found. In Figure 4.4, the outcomes of this application's scanning capabilities using the Nmap command are illustrated. Notably, these results exclusively comprise information concerning open ports and their respective protocols, devoid of specific packet-level data.

Regrettably, the absence of packet-level data renders this application inadequate for inferring comprehensive information about the target devices. Consequently, its utility within the context of this work, particularly concerning the security scoring of Bluetooth devices, is severely limited. Additionally, the implementation of Nmap command scanning results on Android, capable of returning more detailed information and an adaption to Bluetooth is beyond the scope of this work and was not further explored.

## 4.2 BLE Broadcast Packets

Initially, Ubertooth One is connected to a Mac laptop to sniff BLE broadcast packets. Figure 4.5 and Figure 4.6 show one of the broadcast packets captured with Ubertooth One and Wireshark. It contains arrival time, source address, PDU type, signal strength, and manufacturer information.

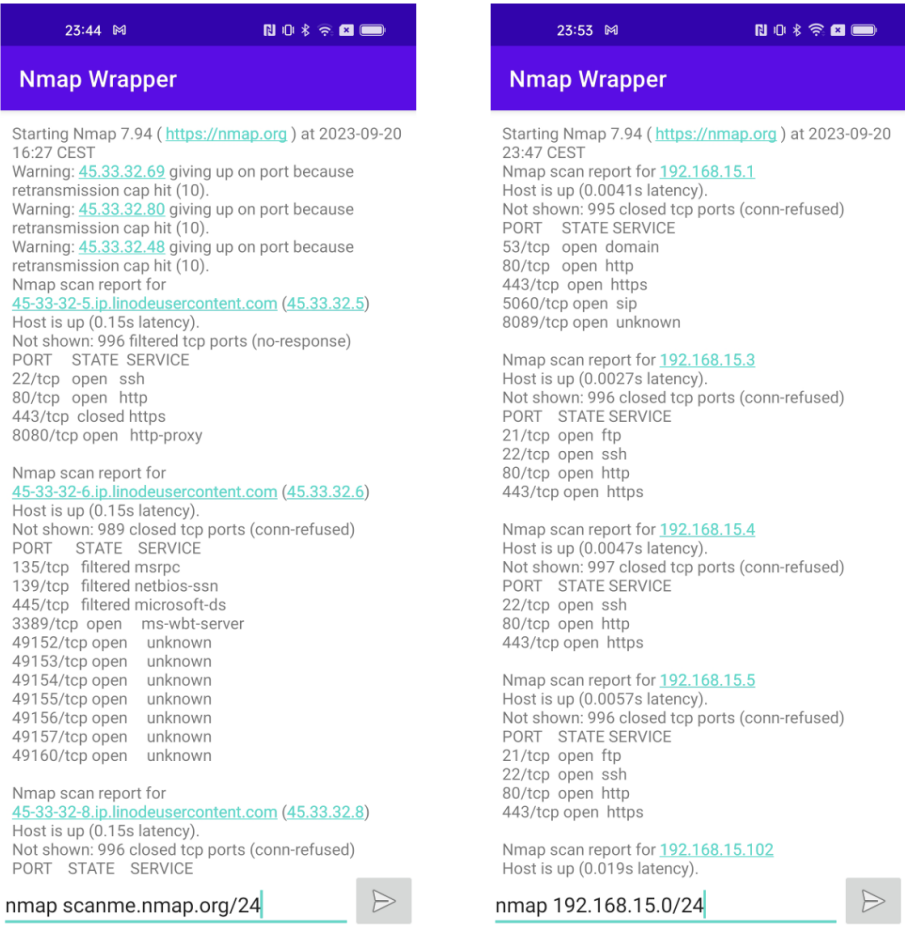


Figure 4.4: The running outcomes of the Nmap Wrapper for Android on a Realme phone.



```

  ▾ Frame 12: 44 bytes on wire (352 bits), 44 bytes captured (352 bits) on interface /tmp/pipe, id 0
    Section number: 1
    ▾ Interface id: 0 (/tmp/pipe)
      Interface name: /tmp/pipe
      Encapsulation type: Bluetooth Low Energy Link Layer RF (161)
      Arrival Time: Jun 26, 2023 21:17:15.447936233 CEST
      [Time shift for this packet: 0.000000000 seconds]
      Epoch Time: 1687807035.447936233 seconds
      [Time delta from previous captured frame: 0.001692400 seconds]
      [Time delta from previous displayed frame: 0.001692400 seconds]
      [Time since reference or first frame: 0.127878500 seconds]
      Frame Number: 12
      Frame Length: 44 bytes (352 bits)
      Capture Length: 44 bytes (352 bits)
      [Frame is marked: False]
      [Frame is ignored: False]
      [Protocols in frame: bluetooth:btle_rf:btle:btcommon]
    ▾ Bluetooth
      [Source: 53:68:f4:2b:a6:77 (53:68:f4:2b:a6:77)]
      [Destination: Broadcast (ff:ff:ff:ff:ff:ff)]
    ▾ Bluetooth Low Energy RF Info
      RF Channel: 0, 2402 MHz, Advertising channel 37
      Signal dBm: -53
      Noise dBm: -128
      Access Address Offenses: 0
      Reference Access Address: 0x8e89bed6
    ▾ Flags: 0x0037
      .... 1 = Dewhitened: True
      .... 1. = Signal Power Valid: True
      .... 1.. = Noise Power Valid: True
      .... 0... = Decrypted: False
      .... 1 .... = Reference Access Address Valid: True
      .... 1. .... = Access Address Offenses Valid: True
      .... 0.. .... = Channel Aliased: False
      .... 00 0... = PDU Type: Advertising or Data (Unspecified Direction) (0)
      .... 0.. .... = CRC Checked: False
      .... 0... .... = CRC Valid: False
      .... 0 .... = MIC Checked: False
      .... 0. .... = MIC Valid: False
      .... 00.. .... = PHY: LE 1M (0)

```

Figure 4.5: A frame of a captured broadcast packet with UberTooth One in Wireshark.

```

  ▾ Bluetooth Low Energy Link Layer
    ▾ Access Address: 0x8e89bed6
      ▾ [Expert Info (Note/Protocol): AccessAddress matched at capture]
        [AccessAddress matched at capture]
        [Severity level: Note]
        [Group: Protocol]
      ▾ Packet Header: 0x1940 (PDU Type: ADV_IND, ChSel: #1, TxAdd: Random)
        .... 0000 = PDU Type: 0x0 ADV_IND
        ...0 .... = Reserved: 0
        ..0. .... = Channel Selection Algorithm: #1
        .1.. .... = Tx Address: Random
        0... .... = Reserved: 0
        Length: 25
        Advertising Address: 53:68:f4:2b:a6:77 (53:68:f4:2b:a6:77)
      ▾ Advertising Data
        ▾ Flags
          Length: 2
          Type: Flags (0x01)
          000. .... = Reserved: 0x0
          ...1 .... = Simultaneous LE and BR/EDR to Same Device Capable (Host): true (0x1)
          .... 1... = Simultaneous LE and BR/EDR to Same Device Capable (Controller): true (0x1)
          .... 0... = BR/EDR Not Supported: false (0x0)
          .... ..1. = LE General Discoverable Mode: true (0x1)
          .... ...0 = LE Limited Discoverable Mode: false (0x0)
        ▾ Tx Power Level
          Length: 2
          Type: Tx Power Level (0x0a)
          Power Level (dBm): 7
        ▾ Manufacturer Specific
          Length: 12
          Type: Manufacturer Specific (0xff)
          Company ID: Apple, Inc. (0x004c)
        ▾ Data: 10076f1f7ca0974518
          ▾ [Expert Info (Note/Undecoded): Undecoded]
            [Undecoded]
            [Severity level: Note]
            [Group: Undecoded]
        CRC: 0xbfab77

```

Figure 4.6: A frame of captured broadcast packets with UberTooth One in Wireshark.

The presented Figure 4.7 illustrates the unprocessed payload of this packet, consistently comprising units that include attributes such as length, common data type, and associated values. Notably, the initial byte serves as the length indicator, facilitating the precise delineation of unit boundaries. The meaning of the common data type number is attainable through consultation with the official SIG documentation. This specific packet contains three units.

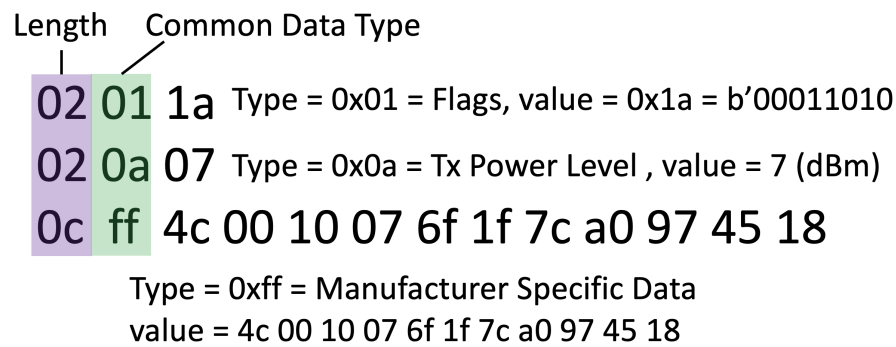


Figure 4.7: Original payload of the same packet showed in Wireshark before and its explanation.

The first unit pertains to "flags," with a maximum length of 1 byte, meaning the potential configuration of 8 distinct flags. Within this broadcast data, the binary values of bits 1, 3, and 4 are set to 1. Specifically, bit 1 denotes the LE General Discoverable Mode, while bits 3 and 4 collectively denote the concurrent operation of LE and BR/EDR controllers and hosts. In addition, bit 0 indicates LE Limited Discoverable Mode, bit 2 indicates whether BR/EDR is supported, and bits 5-7 are reserved for future use. The subsequent unit involves the "Tx Power Level," indicating the strength of the signal. The last unit furnishes details regarding the manufacturer of the device. The official SIG documentation provides a compilation of company names accompanied by their respective identifiers.

The packet shown in the Figure 4.5 and Figure 4.6 is analyzed via Wireshark already, which is consistent with the information analyzed manually but is easier to understand. This thesis primarily centers upon an in-depth exploration of the packet header and the inherent Advertising data.

The packet header section encompasses both the PDU type and address type. To investigate the potential utility of PDU types for security scoring, a review of the literature was conducted. [43] proposes that PDU types serve to delineate the intended function of the device. Furthermore, [65] states that various broadcast modes, such as Legacy Advertising and Extended Advertising, use different PDU types. Notably, the available studies did not go into great detail about PDU type security. Figure 4.8 lists packets with different types of Advertising PDU. They exhibit a consistent structural format. Notably, the header section of these packets uniformly contains essential information, including a descriptor denoting the PDU type. However, distinctions arise in the Flags, TxPower, and response data fields. It's worth noting that the Flags and TxPower fields do not contribute substantively to security scoring, and the response data remains undecipherable. To succinctly summarize, the assessment of PDU security exclusively based on its type remains a challenging task, lacking comprehensive insight and references. Consequently, the utilization of PDU types as a metric for security scoring is not pursued in this work. For instance, in Figure 4.6, the MAC address of the device, denoted as 53:68:f4:2b:a6:77, is represented in binary as 0b1010011011010001111010000101010111010011001110111. Notably, the two leading bits of this address are '11', signifying that it falls under the category of a random static address, concordant with the random address type specified in the packet

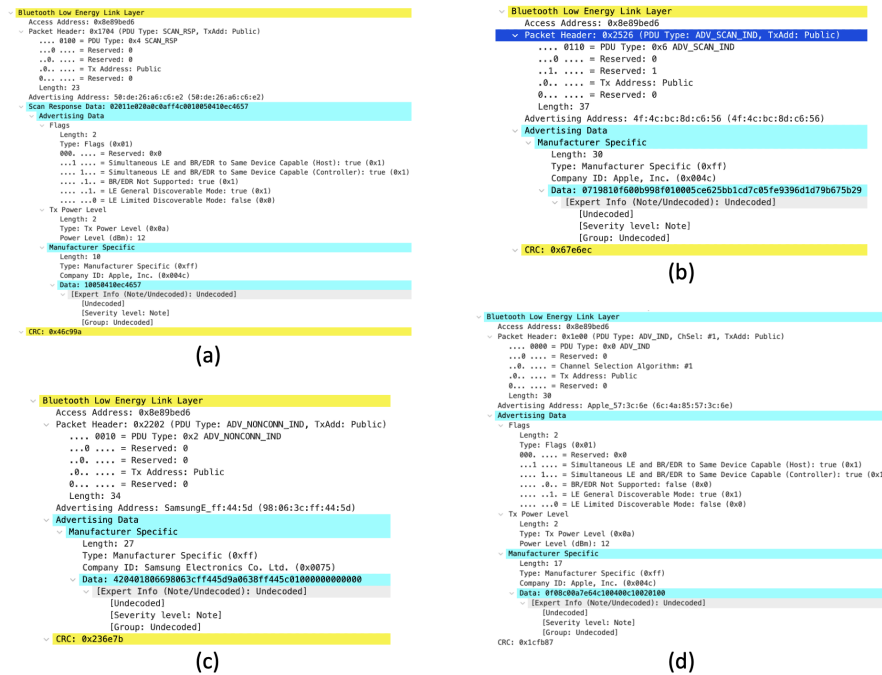


Figure 4.8: Packets with different types of PDU, where sub-figure (a) is from a SCAN\_RSP PDU, sub-figure (b) is from an ADV\_SCAN\_IND PDU, sub-figure (c) is from an ADV\_NONCONN\_IND PDU, and sub-figure (d) is from an ADV\_IND PDU.

header. This analytical procedure was repeated for 30 captured MAC addresses, consistently yielding congruent results with the address types stipulated in the packet header. Additionally, two distinct categories of random addresses were identified in this process. The method is used to infer address types within the scope of this study in light of this validation process.

The Advertising data contains a section named "Manufacturer Specific," which imparts insights pertaining to the device's manufacturer. For instance, the packet shown as an example in Figure 4.6 comes from an Apple device. It is crucial to stress that the information contained in the "Data" sub-component of the Manufacturer Specific section has been encoded, and cannot be decoded using common data types defined in the Bluetooth Core Specification. Notably, even ASCII encoding proves inadequate for the decipherment of this particular segment of information. Therefore, this part of the information cannot be parsed, so the significance of this part of the information for security scoring is not discussed in this work.

Subsequent to sniffing with Ubertooth One, it becomes evident that a considerable proportion of the acquired packets lack the Manufacturer Specific field. There is a special example that is exemplified in Fig.4.9, where certain device addresses appear to potentially contain manufacturer-related information, notably "Samsung," while concurrently lacking the Manufacturer Specific field. This circumstance prompts contemplation of the possibility that the Ubertooth One device may be either collecting incomplete information or misinterpreting the parsing process. Further investigation is conducted in response to this uncertainty, using Wireshark and the nRF52840 Development Kit (nRF52840 DK) to

```

> Frame 40: 53 bytes on wire (424 bits), 53 bytes captured (424 bits) on interface /tmp/pipe, id 0
> Bluetooth
> Bluetooth Low Energy RF Info
< Bluetooth Low Energy Link Layer
  < Access Address: 0x8e89bed6
    < [Expert Info (Note/Protocol): AccessAddress matched at capture]
      [AccessAddress matched at capture]
      [Severity level: Note]
      [Group: Protocol]
  > Packet Header: 0x2201 (PDU Type: ADV_DIRECT_IND, ChSel: #1, TxAdd: Public, RxAdd: Public)
    Advertising Address: SamsungE_ff:44:dd (98:06:3c:ff:44:dd)
    Target Address: 04:42:00:75:ff:1b (04:42:00:75:ff:1b)
  < CRC: 0x800166
    < [Expert Info (Warning/Checksum): Incorrect CRC]
      [Incorrect CRC]
      [Severity level: Warning]
      [Group: Checksum]

```

Figure 4.9: Broadcast packet details with MAC addresses containing the word Samsung captured by Ubertooth One.

improve the process of data gathering and processing. After scrutinizing the outcomes derived from the nRF52840 DK sniffing procedure, a broad semblance to the results acquired through using the Ubertooth One device is observed. Nevertheless, certain disparities were discerned. The MAC address discovered earlier, which contained manufacturer-related data, exists in the data procured via nRF52840 DK sniffing. The key distinction, however, lies in the presence of the Manufacturer Specific information in the latter collection. It is noteworthy that the manufacturer information embedded within the Manufacturer Specific field (specifically, Samsung Electronics Co. Ltd.) harmoniously corroborates the manufacturer-related details indicated in the address itself (i.e., SamsungE\_ff:44:5d), as shown in the accompanying Figure 4.10.

### 4.3 Sniffing Results via An Android Smartphone

An investigation into the range of information that can feasibly be ascertained through BLE sniffing on an Android smartphone is necessary given the purpose of implementing the security scoring system within an Android mobile application.

An exhaustive examination of the unprocessed scan data originating from the Android device is conducted and the entirety of attributes encapsulated within the scanning result data structure are printed out. As elucidated within Figure 4.11, the scan results encompass an array of essential properties, including the MAC address, signal strength, ServiceUuids, manufacturer information, device name, and timestamp data associated with the device. Moreover, there are two fields that incorporate insightful details concerning the broadcast methodology, discerning whether the packet originated from legacy advertising and determining the device's state of connectability.

The results presented within Figure 4.11 pertain to two distinct devices, both are connectable and employ the legacy advertising protocol. Notably, subfigure (a) and (b)

```

> Frame 101: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface /dev/cu.usbmodem0006837215421-4.0, id 0
> nRF Sniffer for Bluetooth LE
  > Bluetooth Low Energy Link Layer
    Access Address: 0x8e89bed6
    > Packet Header: 0x2202 (PDU Type: ADV_NONCONN_IND, TxAdd: Public)
    Advertising Address: SamsungE_ff:44:5d (98:06:3c:ff:44:5d)
  > Advertising Data
    > Manufacturer Specific
      Length: 27
      Type: Manufacturer Specific (0xff)
      Company ID: Samsung Electronics Co. Ltd. (0x0075)
      > Data: 420401806698063cff445d9a0638ff445c01000000000000
        > [Expert Info (Note/Undecoded): Undecoded]
          [Undecoded]
          [Severity level: Note]
          [Group: Undecoded]
    > CRC: 0x236e7b
      > [Expert Info (Warning/Checksum): Incorrect CRC]
        [Incorrect CRC]
        [Severity level: Warning]
        [Group: Checksum]

```

Figure 4.10: Broadcast packet details with MAC addresses containing the word Samsung captured by nRF52840 DK.

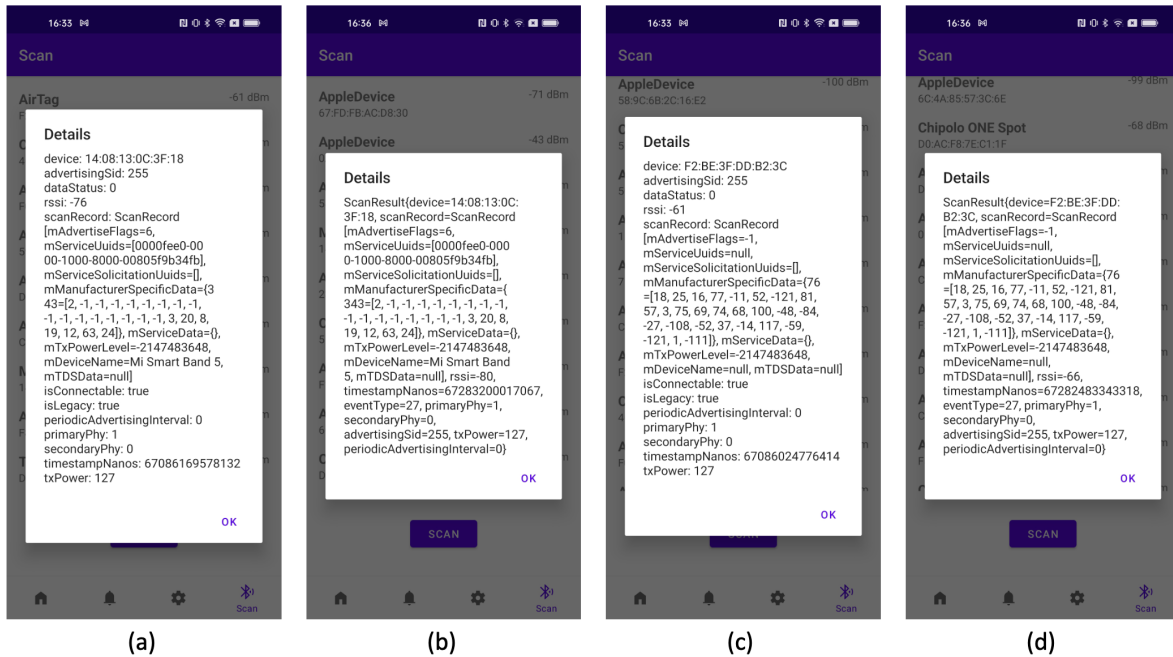


Figure 4.11: Two raw scanning results and corresponding printed fields from an Android smartphone. Subfigure (a) and (b) are from a Mi Smart Band, while (c) and (d) are from an Airtag. Additionally, (a) and (c) show all fields in the scanning results of two different devices; (b) and (d) demonstrate the raw scanning results of the two different devices.

represent data from the Mi Smart Band, while subfigure (c) and (d) depict broadcast packets originating from the Airtag device.

A discernible discrepancy emerges when contrasting the broadcast packets of Airtag with those of the Mi Smart Band, specifically with respect to the null value of ServiceUids and device name in the former. Prompted by this observation, an in-depth scrutiny of the scanned data ensued, revealing a prevailing pattern wherein the device name remains null within most broadcast packets. Even in instances where it would be reasonable to expect a device name, such as AirPods, the captured broadcast packets consistently yield null values in the device name field. A parallel trend was also identified in relation to ServiceUids, which predominantly manifested as null values across the sampled packets.

While both device names and ServiceUids carry intrinsic significance for the purposes of deducing device types and evaluating security implications, their suitability for security scoring is rendered infeasible due to the preponderance of null values. Consequently, these attributes have been excluded from consideration within the context of security scoring within the scope of this study.

The format of the device MAC address within the scan results obtained from the Android smartphone remains consistent with that collected using the laptop. However, a notable disparity arises in the representation of the ManufacturerSpecificData, which is evidently manifested in a distinct format compared to the presentation observed in Wireshark.

After browsing the Android Application Programming Interface (API) Reference [5], it was ascertained that a sub-component of ScanResult class, ScanRecord, represents a scan record from Bluetooth LE scan. The ScanRecord class offers a public method: getManufacturerSpecificData() which returns manufacturer identifiers and their corresponding manufacturer-specific data.

In light of this insight, the first number in mManufacturerSpecificData, such as "343" in sub-figures (a) and (b), and "76" in sub-figures (c) and (d), signifies the company's unique identifier. The array after it is the manufacturer-specific data. To correlate this identifier with the respective company, refer to the official Bluetooth document "Assigned Numbers" [1] is necessary. However, a prerequisite for this lookup procedure is the conversion of these decimal numerical identifiers into four-digit hexadecimal representations, prefixed with "0x", as the same as the format shown in the Wireshark. In cases where the resultant hexadecimal number comprises fewer than four digits, it is imperative to pad it with leading zeros.

## 4.4 Exploration of Open Source Intelligence (OSINT)

OSINT methodologies are also considered to broaden the scope of information. An exploration on the Shodan website is conducted, a specialized search engine designed for connected devices on the internet. Shodan meticulously gathers data pertaining to all devices directly linked to the Internet, a fact that has led to certain individuals employing it for unauthorized access, such as connecting to webcams to view their captured footage.

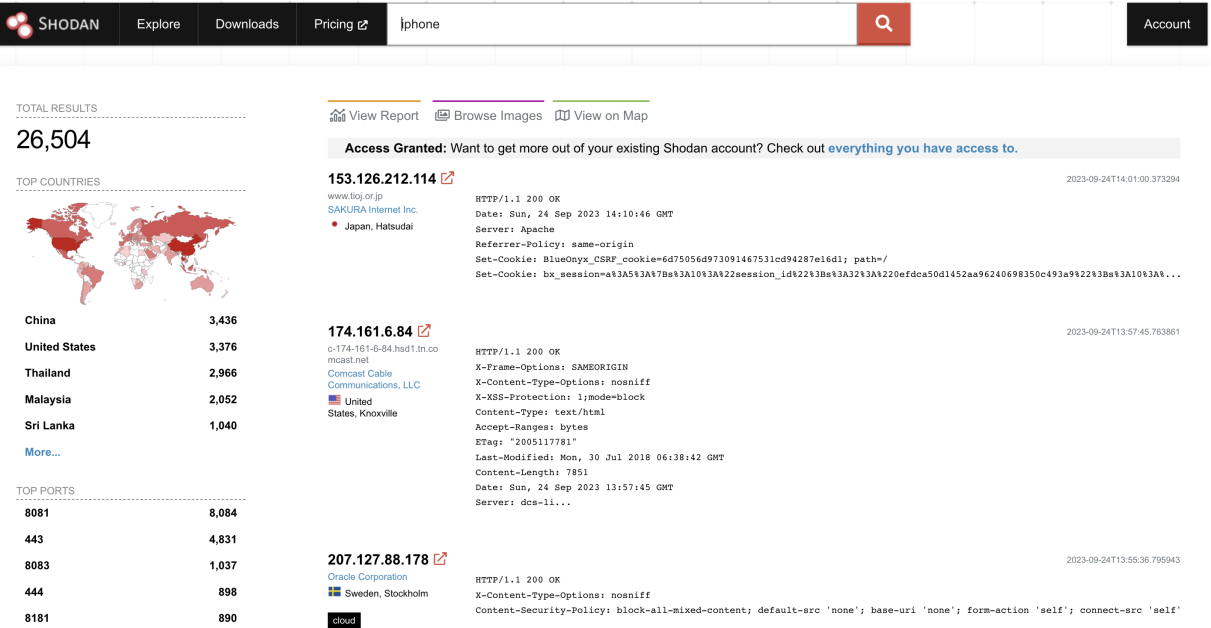


Figure 4.12: Search result on Shodan with the keyword "iPhone".

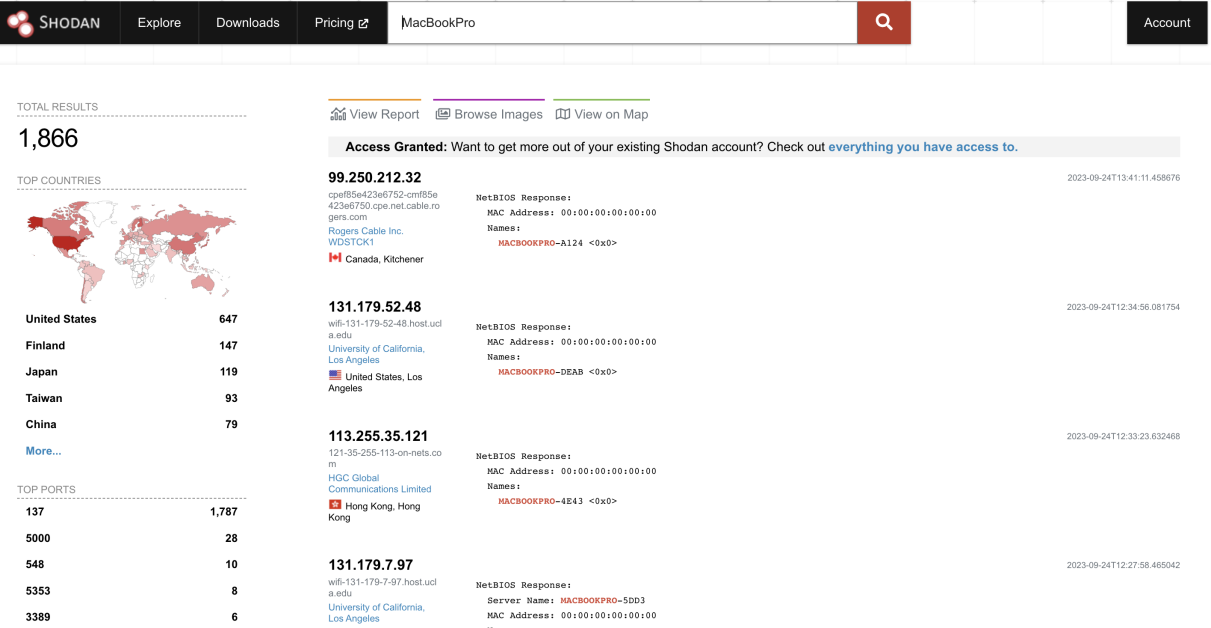


Figure 4.13: Search result on Shodan with the keyword "MsczbookPro".

My inquiry on Shodan entails searches using the keywords "iPhone" and "MacBookPro," the outcomes of which are presented in Figure 4.12 and Figure 4.13 for reference. However, it is noteworthy that Shodan lacks the capability to filter results based on the current geographical location, thereby limiting its capacity to analyze nearby devices. Moreover, the search yielded an unexpectedly meager dataset. To be specific, there are only 55



results for the keyword "iPhone" with a location in Switzerland, and only 17 results for the keyword "MacBookPro" with a location in Switzerland. The constraints of available information, in terms of both quantity and depth, make it inadequate for data collection in support of security scoring via OSINT methodologies.

## 4.5 Exploration of MUD

To gather additional device-related insights, MUDs are considered. MUDs serve the purpose of enabling end devices to communicate their required access permissions and network functionalities to the network infrastructure for proper operation. However, it's essential to note that MUD solutions necessitate IoT devices to have internet connectivity and employ the HTTPS protocol for interactions with MUD File Servers. As this approach primarily relies on internet-based networks rather than Bluetooth connections, it falls outside the scope of the present study.

## 4.6 Scoring Criteria

The proposed security scoring criteria are based on BLE broadcast packets. After a meticulous examination of the amassed data, a notable observation emerged: the availability of informative data suitable for safety scoring is constrained. This circumstance presents a formidable challenge in the conversion of abstract security levels into precise numerical scores. Moreover, the complexity arises from the intricate task of precisely assigning weights to various facets, given that an overarching security assessment may encompass multiple dimensions and considerations. It is evident that the allocation of weights among these facets significantly influenced the final numerical scores. As a means to enhance the rationality and practicality of the security scoring process, 3-level scale security scoring is a judicious approach for assessing the security of Bluetooth devices, allowing for a more intuitive and manageable evaluation of their security.

### 4.6.1 Criteria Features

Inspired by IoTAG [25], a scholarly work that leverages packets captured via the Nmap command for the identification and security assessment of IoT devices, the current study conducts a broad analysis to determine the feasibility of using particular security-related attributes for the purpose of security scoring with regard to Bluetooth devices.

### Wi-Fi Technology, Protocols and Communication

It is evident that distinct wireless network encryption technology and Internet protocols inherently possess varying levels of security. For instance, the Wi-Fi Protected Access 3 (WPA3) standards are recognized for their superior security compared to WPA, and

HTTPS exhibits greater security in comparison to HTTP. However, the acquisition of packet-level information pertaining to these aspects from smartphones is a challenge.

Additionally, considering that many IoT devices are connected via Bluetooth, targeting Bluetooth devices that communicate using Bluetooth low-power technology is meaningful. Therefore, devices that communicate using Wireless Local Area Network (WLAN) technologies like WiFi or Local Area Network (LAN) are omitted. As a result, the communication technologies and running protocols are not taken into account while evaluating security.

### **Easy Passwords**

The "IoTAG" study highlights a concerning problem of IoT devices, where users often employ default usernames and passwords allocated by the manufacturers, which are commonly shared across multiple devices. Furthermore, it is noted that some users set simplistic and easily guessable passwords, such as "123456" or "111111." In response to these security concerns, "IoTAG" employs a default password test to score device security.

However, it is paramount to clarify that the IoT Bluetooth devices investigated within the scope of this study exclusively utilize BLE technology for communication. These devices do not necessitate password-based authentication for device connection. Consequently, there exists no imperative for security evaluation with regard to passwords in this particular context.

### **Firmware Version**

The inclusion of device-specific version information within communication packets can potentially enable the retrieval of known security vulnerabilities by accessing public databases. Consequently, the inadvertent leakage of version information may introduce security risks, with newer firmware versions often offering enhanced security features compared to their older counterparts.

However, it is pertinent to emphasize that the Bluetooth broadcast packets examined within this study do not contain firmware version information. Moreover, when analyzing the packets captured via the Nmap command, accessible version information is solely discernible for MacBookPro and MacBookAir devices. As a result, the practice of inferring security scores based on version information is rendered infeasible within the context.

### **ServiceUUIDs**

Certain BLE broadcast packets may include information regarding the serviceUUID, prompting an attempt to deduce the overarching service category based on the serviceUUID itself. However, it is noticeable that the official "Assigned Numbers" documentation lacks thorough service-specific details. Instead, it primarily records the company's name that has requested the UUID and the associated 4-digit hexadecimal identifier.

In contrast, the serviceUUIDs encountered in the collected broadcast packets are longer character strings.

It is discovered after an investigation that it is possible to match the first four serviceUUID digits in the broadcast packet to those listed in the official repository. This process aids in identifying the originating company linked to the service; however, it falls short of providing precise information about the specific nature of the service itself.

Furthermore, serviceUUID is excluded as a metric for security scoring due to the prevalence of broadcast packets where the serviceUUID field is null, rendering it infeasible for comprehensive analysis.

### **Manufacturer**

As previously elucidated, the format of the Manufacturer Specific Data contained within BLE broadcast packets collected by Android smartphones, though not mirroring precisely the format found within the Assigned Numbers repository, can indeed be aligned with company identifiers documented therein with some necessary transformations. Consequently, the manufacturer can be inferred through the gathered packets.

A noteworthy observation pertains to the official documentation, wherein a specific company identifier is assigned to Chipolo (0x08C3). However, a broadcast packet originating from a Chipolo One Spot device intriguingly attributes the device's manufacturer as Apple (76 converted to hexadecimal is 4C, corresponding to Apple's ID, 0x004C). This phenomenon can be attributed to the utilization of Apple-made hardware within the Chipolo One Spot device. This observation confers a distinctive aspect to the security scoring process. The identification of security issues associated with a manufacturer's products may subsequently extend to devices that employ the hardware of that manufacturer, even if they are from different companies. Furthermore, devices of unknown origin introduce a considerable degree of uncertainty and associated risks.

It is imperative to acknowledge that manufacturers of Bluetooth devices have implemented varying degrees of security measures, theoretically enabling a more nuanced security assessment of devices stemming from diverse manufacturers. However, the practicality of this approach is constrained by the sheer volume of assigned IDs, which presently number 3,318 companies (as of September 21, 2023) in the official documentation. Evaluating the security ratings of each of these companies would entail a substantial workload. Consequently, within the scope of this research, manufacturer information is employed as a basis for security scoring, but no further granular assessment of individual manufacturer companies is pursued.

### **Advertising Address Type**

The device address extracted from the packet data obtained through Android smartphone sniffing aligns consistently with the representation of the device address derived from packets captured by nRF52840 DK and Wireshark. As previously mentioned, obtaining

the MSB through the conversion of the device's MAC address into its binary equivalent can serve as a reliable means to ascertain the precise random address type. In addition, public addresses are assigned by the IEEE and necessitate registration with the IEEE for acquisition. These addresses comprise an Organizationally Unique Identifier (OUI) as assigned by the IEEE and a company-assigned device identifier. Therefore, OUIs can help identify the public address.

Notably, each broadcast packet encompasses information related to the device address, which emerges as a viable metric for the purpose of security scoring. This approach affords valuable insights into the security posture of the respective Bluetooth devices.

In addition, the Bluetooth core specification (Vol 3, Part C) recommends updating the random non-resolvable and random resolvable addresses every 15 minutes to improve security. It is also feasible to monitor address updates through timestamp information.

## Other Features

The collected BLE broadcast packets also contain some additional information. Among these details, the device name stands out as a potential asset for deducing device type and version, which may provide significant insights into security scoring considerations. Regrettably, the device name field remains predominantly null in the majority of broadcast packets. In the presence of this pervasive absence of substantive data, this particular attribute proves unsuitable as a basis for security scoring of the devices.

Moreover, the broadcast packets involve various other features, including Received Signal Strength Indicator (RSSI), Transmit Power Level (TxPowerLevel), primary and secondary attributes, PDU Types, and Advertising Flags. Both RSSI and TxPowerLevel are related to signal strength-related information, with minimal direct implications for device security. Furthermore, a literature review reveals a dearth of research concerning the divergent security implications of different PDU Types, various Advertising Flags, and distinct PHY configurations. Therefore, these attributes are not integrated into the security scoring framework due to their lack of established relevance within existing research.

### 4.6.2 Device Scoring

In this section, the scoring process for different aspects of safety metrics will be described. The proposed security criteria are listed in Table 4.2.

audit criteria	score
manufacturer	
known company	<b>2</b>
unknown company	<b>0</b>
advertising address type	
public address	<b>0</b>
random static address	<b>0</b>
random private resolvable address	<b>0.5</b>
random private non-resolvable address	<b>1.5</b>
address update	
no update more than 15 minutes	<b>0</b>
update within 15 minutes	<b>0.5</b>

Table 4.2: Security criteria.

### Manufacturer

Devices originating from unknown manufacturers inherently bear undefined security risks, lacking any information about security measures. Consequently, devices emanating from manufacturers of unverified identity are assigned a rating of 0 for this particular evaluation criterion.

In light of the impracticality of fine-tuning security scores for the extensive list of 3,318 companies documented within the official records, a simplified approach is adopted. All devices traceable to known manufacturers are uniformly assigned a rating of 2.

### Advertising Address Type

A public address, characterized by its globally unique and unchanging nature, is exceptionally straightforward to trace, warranting a security score of 0. Static random addresses are randomly generated during device reboots or resets. However, they remain susceptible to tracking if a device is not subject to such reinitialization. Hence, a device employing a static random address featuring an MSB of "11" is also assigned a security score of 0.

Conversely, non-resolvable private addresses offer a high degree of security, as they defy resolution by any external entities. Devices employing non-resolvable private addresses with an MSB of "00" merit a rating of 1.5 points.

In the case of resolvable private addresses, their resolution is contingent upon possessing the requisite encryption key. While resolvable private addresses benefit from the added security afforded by asymmetric encryption, studies have revealed potential vulnerabilities, particularly concerning replay attacks when the whitelist is set [68]. In consequence, devices utilizing resolvable private addresses featuring an MSB of "01" are rated at 0.5 points, representing an intermediate level of security in this context.

The maximum score of this indicator has been changed from 2 to 1.5 in order to offer a balanced assessment framework for both the manufacturer's information and the device's

address security features. This modification takes into account an additional offset value of evaluation related to the lifecycle of addresses, which serves as a complementary component in calculating the overall security of MAC addresses. For clarity and context, a more thorough description of the offset is offered in the next subsection.

### **Address Update Monitoring**

In adherence to the Bluetooth core specification, it is recommended to update the private address at intervals, with a suggested interval of 15 minutes. Inspired by this recommendation, the address update test in security scoring is applied. Typically, random private addresses, including both resolvable and non-resolvable addresses, are often updated regularly. Similarly, random static addresses may be updated through regular device reboots, facilitating the periodic update of the address. The rationale behind this is rooted in the notion that periodically updated addresses are less likely to be tracked and thus have higher security. Thus, the determination of address update status emerges as an essential component for further elaborating on the security evaluation of address types, enabling a refined assessment of security scores. Considering that whether an address is updated or not is complementary to address type security, the score is rated with the security score of the address update as an offset to the address type security. To be specific, if this prescribed address update interval exceeds 15 minutes, the proposed security scoring criteria will apply a corrective measure to the score. Specifically, a deduction of 0.5 is subtracted from the base score corresponding to the relevant device address type.

### **4.6.3 Security Levels**

Due to the constraints posed by the limited data collected, equipment security has been determined to be divided into three distinct levels. The full range of possible score combinations is laid forth in Table 4.3, along with descriptions of each corresponding situation. Devices from known manufacturers employing private non-resolvable addresses attain a classification denoting a high degree of security. Devices originating from known manufacturers utilizing either private resolvable addresses or periodically updated random static addresses, specifically, those with a lifespan of less than 15 minutes, are ascribed a status of moderate security. In addition, devices utilizing public addresses or static addresses with a lifespan exceeding 15 minutes, alongside all devices of unverified manufacturers, are relegated to the low-security classification.

Device Descriptions	Scores	Security
From a known manufacturer with non-resolvable address that is updated in 15 minutes	$2+1.5+0.5=4$	High
From a known manufacturer with non-resolvable address that is not updated in 15 minutes	$2+1.5+0=3.5$	High
From a known manufacturer with resolvable address that is updated in 15 minutes	$2+0.5+0.5=3$	Medium
From a known manufacturer with resolvable address that is not updated in 15 minutes	$2+0.5+0=2.5$	Medium
From a known manufacturer with static address that is updated in 15 minutes	$2+0+0.5=2.5$	Medium
From a known manufacturer with static address that is not updated in 15 minutes	$2+0+0=2$	Low
From a known manufacturer with unique public address	$2+0+0=2$	Low
From an unknown manufacturer with non-resolvable address that is updated in 15 minutes	$0+1.5+0.5=2$	Low
From an unknown manufacturer with non-resolvable address that is not updated in 15 minutes	$0+1.5+0=1.5$	Low
From an unknown manufacturer with resolvable address that is updated in 15 minutes	$0+0.5+0.5=1$	Low
From an unknown manufacturer with resolvable address that is not updated in 15 minutes	$0+0.5+0=0.5$	Low
From an unknown manufacturer with static address that is updated in 15 minutes	$0+0+0.5=0.5$	Low
From an unknown manufacturer with static address that is not updated in 15 minutes	$0+0+0=0$	Low
From an unknown manufacturer with unique public address	$0+0+0=0$	Low
Public addresses are unique, unchanging addresses and never be updated.		

Table 4.3: All possible scores and corresponding descriptions.





# Chapter 5

## Implementation

Based on the previous design, an extension to the HomeScout Android application was developed. This section describes its configuration, workflow schematic, and application interface.

### 5.1 Build Configurations

Android Studio employs the Gradle build system for the compilation of projects. This comprehensive process encompasses the transformation of resources, source code, packages, and all essential dependencies into an Android Application Package (APK), which is managed by Gradle. Upon downloading the resultant APK file, it becomes possible to install and execute the application on an Android smartphone. Notably, this project leverages Android Gradle Plugin version 7.4.1 in conjunction with Gradle version 7.5 for its development and build management.

### 5.2 Workflow of the Extended Function

The functionality introduced entails the scanning of all proximate BLE devices and the subsequent assessment of their security. Figure 5.1 illustrates the workflow of various services within the HomeScout application, with the distinct pink segment representing the functionality expansion introduced by this work. The user initiates the application, navigates to the feature page, and commences the scanning process. As BLE advertisements are captured, the security scoring algorithm conducts an evaluation of the security posture of the associated BLE device, according to the insights from the accumulated advertisement packets. The result will be saved as a .csv file and displayed on the interface after the evaluation is completed. A comprehensive exposition of this feature will be provided in subsequent sections for a detailed understanding.

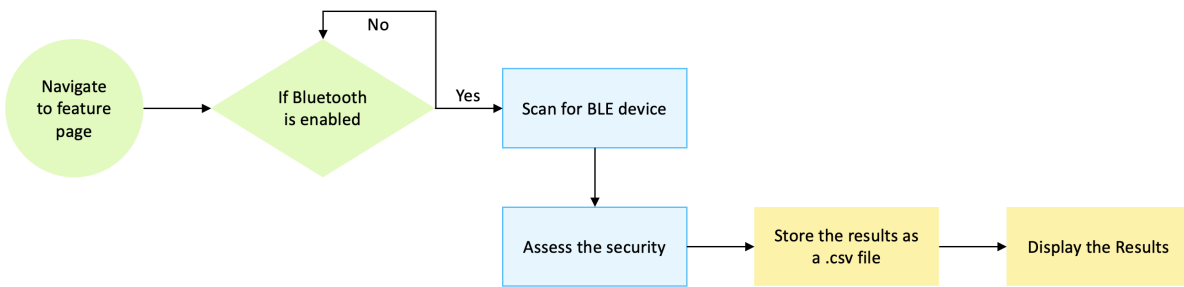


Figure 5.1: The workflow of the extended function.

## 5.3 Architecture

This work introduces a new user interface tailored for security scoring and the integration of a local data file into the preexisting architectural framework of the HomeScout [50] application.

### 5.3.1 User Interface (UI)

The HomeScout application adheres to the principles outlined in the Android Developer Guide and is architecturally designed based on the Model-View-ViewModel (MVVM) pattern. The user interface has a Main Activity with a bottom navigation bar, providing users with the switch between distinct segments.

Initially, HomeScout features four primary fragments: Welcome, Notifications, Settings, and Scan, each serving specific functionalities. This extension introduces a new Security fragment into the application's framework. This new fragment allows users to manually initiate scans for nearby BLE devices and offers insights into the security status and related security features associated with the scanned devices.

The MVVM pattern segregates and specifies the channels of communication between the Model, the View, and the ViewModel components. The Model is responsible for data sources, while the ViewModel serves as the manager of data pertinent to the user interface. Meanwhile, the View takes the role of facilitating the visual presentation of content.

In the context of user interactions with the interface, the View captures and forwards the UI events to the ViewModel, which handles the interaction's logic. Throughout the handling process, the ViewModel may interact with the Model for data updates. Subsequently, as the ViewModel updates its state, the View observes and updates automatically via data binding and presents the updated data to the user interface.

Figure 5.2 illustrates the interface of the security scoring function. An integral aspect of this function is the need for scanning information over a long time span, over 15 minutes, to effectively ascertain whether a given address has been updated. To guide users in this

regard, a succinct advisory message is thoughtfully placed at the bottom of the page, suggesting, 'It is recommended that two scans be performed at intervals of 15 minutes'.

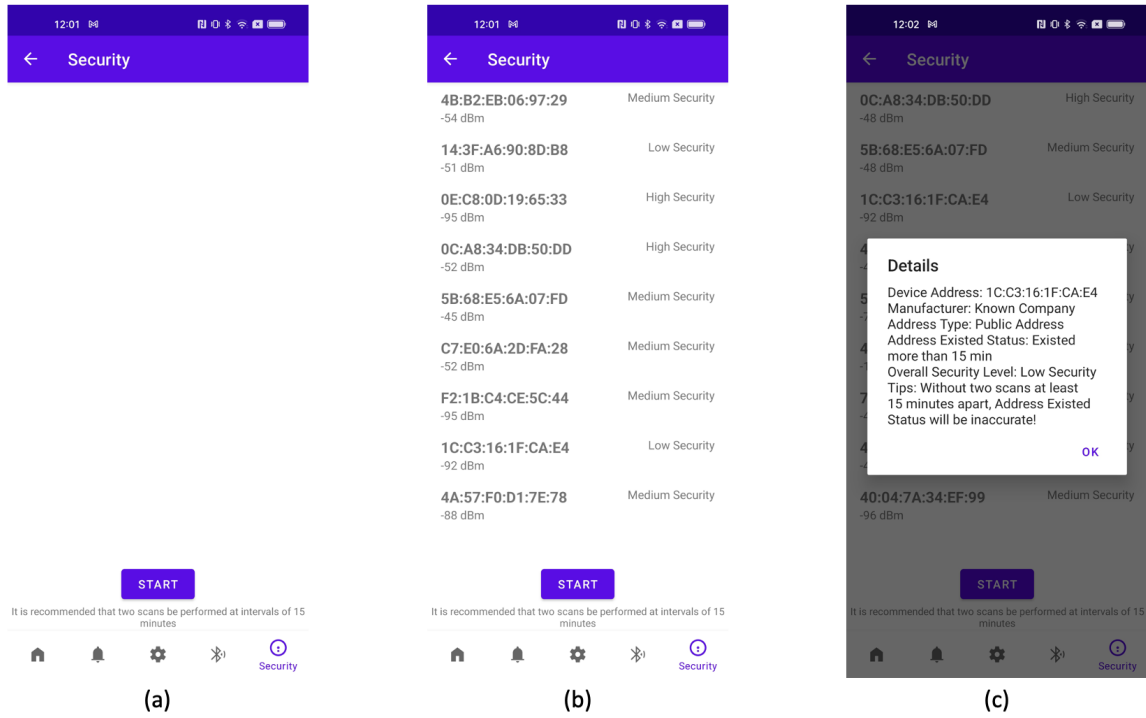


Figure 5.2: Screenshots of the security scoring feature. (a) is the initial page of this feature, (b) is the result list after clicking the start button, and (c) is the pop-up window after clicking one item.

After the user clicks the 'start' button, the application initiates the scanning of Bluetooth devices while concurrently conducting security evaluations. The results of these assessments are meticulously organized and presented in a comprehensive list format. Users can scroll down to access more results.

Furthermore, for users seeking more in-depth insights into the security attributes of specific devices, a convenient mechanism is provided. By clicking individual items within the list, a pop-up window emerges, affording users the opportunity to delve into the detailed security information, thereby enabling a more comprehensive consideration of the device's security, according to their specific requirements. There is a trade-off between providing information and preserving privacy. The security details provided serve to discern whether a device originates from a known manufacturer, all while safeguarding the specific company's identity from disclosure.

### 5.3.2 Data Exchange

Figure 5.3 depicts the data exchange procedure essential to the security scoring function. The Security Fragment captures and handles the subsequent interaction events after a

user initiates interacting with the UI. In this context, the Fragment may seek to access or write data to a local file via the Adapter.

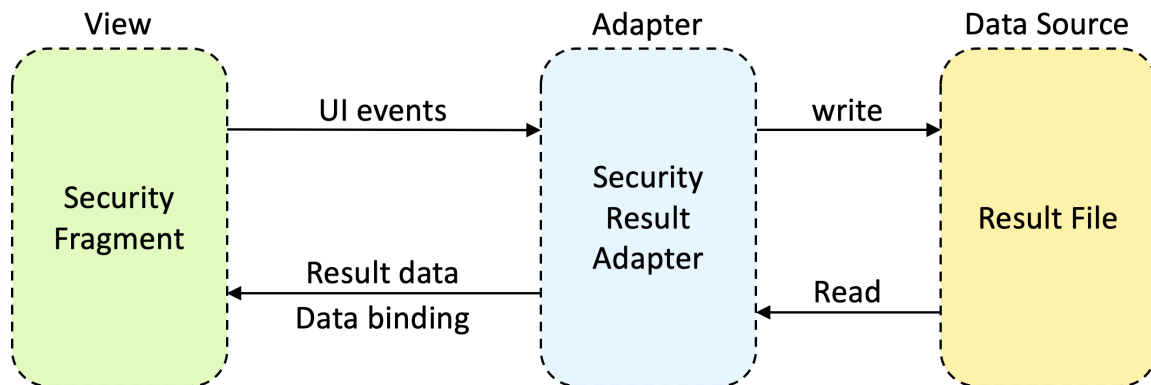


Figure 5.3: The process of data exchange integral to the security scoring function.

The Adapter, after receiving these requests, acting as an intermediary, proceeds to retrieve data from the local file. Subsequently, it processes this data, transforming it into individual items for presentation within the RecyclerView.

As the Adapter proceeds to populate the RecyclerView with these prepared items, the UI is updated to reflect the data. These updates serve to provide users with confirmation that their interactions have been not only acknowledged but also effectively processed and acted upon.

### 5.3.3 Security Scoring Function

Several crucial facets define the essence of this function, including the following key components: BLE scanning, the acquisition of manufacturer information, the inference of address types, and address update detection.

#### BLE Scanning

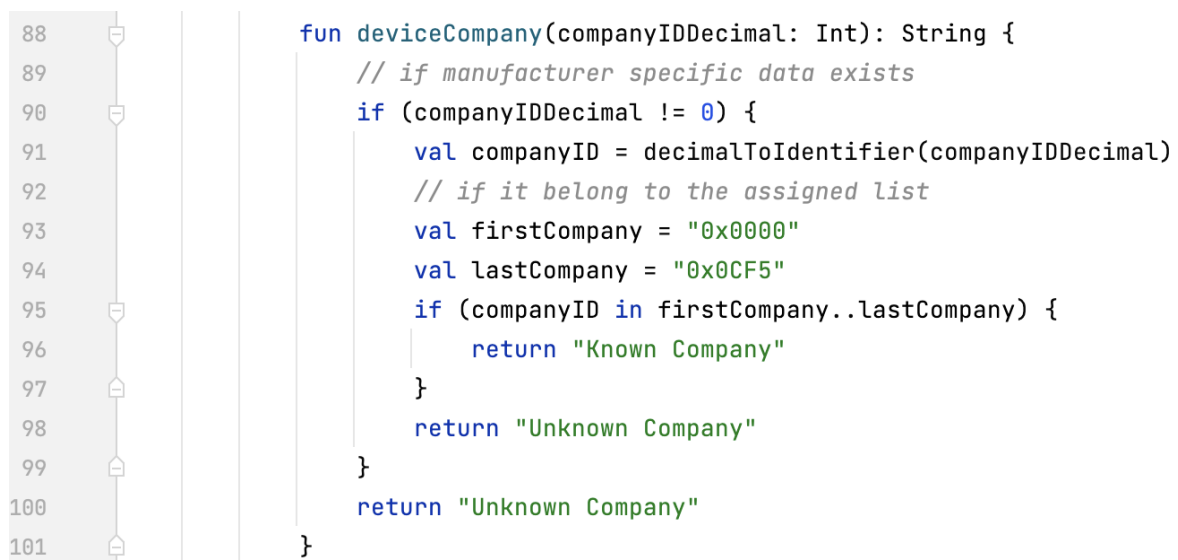
This part leverages the pre-existing BluetoothScanningService from the original HomeScout application to conduct BLE device scans. Within this service, a constant named SCAN\_PERIOD is defined to signify the scanning duration. Following the scanning process, one can invoke the `bleScanner.stopScan()` function to manually terminate scanning, or alternatively, the system will autonomously cease scanning at the culmination of the predefined scanning period.

It is important to note that in the original HomeScout application, the scanning period was set to 12 seconds. However, practical testing revealed that in situations where a multitude of BLE devices coexist in close proximity, such as in densely populated environments

like a university lecture hall, the application exhibited instability during runtime. This instability may be attributed to memory constraints resulting from the reception and processing of a large volume of BLE device advertising packets. As a result, in this implementation, the scanning period has been carefully adjusted to 3 seconds.

### Manufacturer Information

As previously discussed, the `manufacturerSpecificData` in the `ScanResult.ScanRecord` class of BLE scans contains information in the format of `343=[2, -1, ..., -1, 3, 20, 8, 19, 12, 63, 24]`. In this structure, '343' represents the decimal manufacturer's identifier, while the subsequent array signifies the manufacturer-specific data, which often remains undecipherable. Figure 5.4 illustrates the code snippet designed to ascertain whether the device originates from a known company based on the decimal manufacturer identifier.



```
88 fun deviceCompany(companyIDDecimal: Int): String {
89     // if manufacturer specific data exists
90     if (companyIDDecimal != 0) {
91         val companyID = decimalToIdentifier(companyIDDecimal)
92         // if it belong to the assigned list
93         val firstCompany = "0x0000"
94         val lastCompany = "0x0CF5"
95         if (companyID in firstCompany..lastCompany) {
96             return "Known Company"
97         }
98         return "Unknown Company"
99     }
100     return "Unknown Company"
101 }
```

Figure 5.4: Code snippet of the `deviceCompany` function, aiming to check if the device comes from a known manufacturer with its decimal company identifier.

To ensure the possibility for precise identification of device manufacturers in the future, the implementation has opted to convert decimal identifiers into hexadecimal counterparts, as prescribed in the official documentation. It's important to note that, in order to protect device owners' privacy, this work refrains from disclosing specific manufacturer company information. Furthermore, in accordance with the consecutiveness of company identifiers, as officially documented, an advertising package is categorized as originating from a known manufacturer under two conditions: firstly, when the `manufacturerSpecificData` is not null, and secondly, when the manufacturer identifier falls within the range of officially assigned identifiers.

In terms of implementation, the application harnesses Android's `BluetoothLeScanner` to initiate and terminate scanning operations. The `'startScan()'` method is invoked with a



Following the conversion of the scanned address into binary format, as previously elucidated, an address commencing with '11' signifies a random static address, '00' indicates a random non-resolvable private address, and '01' implies a random resolvable private address. Addresses that fail to satisfy these criteria are deemed invalid. It is noteworthy that, although there is a theoretical possibility of a random address being the same as an OUI, the likelihood of generating a random 22-bit binary sequence that is identical to an OUI (discounting the two MSBs from the first 24-bit) is exceedingly low and, therefore, not a consideration.

```

122 fun getAdvertisingAddressType(macAddr: String): String {
123     val publicOUIs = readOUIs(fileName)
124     val binaryMacAddress = macAddressToBinary(macAddr)
125     val first6 = firstSixDigits(macAddr)
126     val first6hex = firstSixDigits(macAddr).toLong( radix: 16)
127     val notOUI1 : List<String> = listOf("0009FA", "00242D", "00243E", "002457", "002534", "00253F", "0025F8", "002649", "00264B")
128
129     val AddrType: String = when {
130         // filter out several constant sequence in the OUI list to avoid large txt file
131         publicOUIs.contains(first6) -> "Public Address"
132         first6hex <= "000832".toLong( radix: 16) -> "Public Address"
133         first6hex >= "00084E".toLong( radix: 16) && first6hex <= "002722".toLong( radix: 16) && first6 !in notOUI1 -> "Public Address"
134         first6hex >= "003000".toLong( radix: 16) && first6hex <= "0030FF".toLong( radix: 16) -> "Public Address"
135         first6hex >= "004000".toLong( radix: 16) && first6hex <= "0040FF".toLong( radix: 16) -> "Public Address"
136         first6hex >= "006000".toLong( radix: 16) && first6hex <= "0060FF".toLong( radix: 16) -> "Public Address"
137         first6hex >= "008000".toLong( radix: 16) && first6hex <= "0080FF".toLong( radix: 16) -> "Public Address"
138         first6hex >= "009000".toLong( radix: 16) && first6hex <= "0090FF".toLong( radix: 16) -> "Public Address"
139         first6hex >= "00A000".toLong( radix: 16) && first6hex <= "00A0FF".toLong( radix: 16) -> "Public Address"
140         first6hex >= "00C000".toLong( radix: 16) && first6hex <= "00C0FF".toLong( radix: 16) -> "Public Address"
141         first6hex >= "00D000".toLong( radix: 16) && first6hex <= "00D0FF".toLong( radix: 16) -> "Public Address"
142         first6hex >= "00E000".toLong( radix: 16) && first6hex <= "00E0FF".toLong( radix: 16) -> "Public Address"
143         binaryMacAddress.startsWith( prefix: "11") -> "Random Static Address"
144         binaryMacAddress.startsWith( prefix: "00") -> "Random Non-resolvable Private Address"
145         binaryMacAddress.startsWith( prefix: "01") -> "Random Resolvable Private Address"
146         else -> "Invalid"
147     }
148     return AddrType
149 }

```

Figure 5.6: Code snippet of the function for device address type inference.

## Detection of Address Update

During the process of device scanning and security evaluation, security-related data, including the device's address, its corresponding manufacturer, the address type, address update status, security level, and the timestamp denoting the capture of the advertisement packet, is meticulously inscribed into a local .csv file for archival purposes. In Figure 5.7, a code excerpt is presented, illuminating the function of address update detection.

Public addresses are unique and unchanged, so they never update. Following the execution of a scan, an update detection ensues for all random addresses. It entails an examination of the local .csv file to ascertain the existence of a record corresponding to the currently scanned MAC address. In instances where the address remains unlisted within the local .csv file, it signifies that the address is undergoing its first scan, and it will be regarded as a regular update by default.

```

151 fun addressUpdate(priorAddrs: List<String>, addr: String, timeStamp: Long): String {
152     val resultFile = "result_test.csv"
153     val file = File(view.context.filesDir, resultFile)
154     var dif = 0L
155     if (addr in priorAddrs){
156         try {
157             BufferedReader(file.reader()).use { reader ->
158                 var line: String?
159                 while (reader.readLine().also { line = it } != null) {
160                     val lastmac = line?.split(Regex( pattern: ","))?.firstOrNull()
161                     if (lastmac == addr){
162                         val earliestTime = line?.split(Regex( pattern: ","))?.lastOrNull()
163                         if (earliestTime != null) {
164                             dif = abs( n: (timeStamp - earliestTime.toLong()) / NANSECONDS_TO_MINUTES)
165                             println("$addr : $dif")
166                             break
167                         } else {
168                             dif = 0L
169                         }
170                     }
171                 }
172             }
173         } catch (e: Exception) {
174             e.printStackTrace()
175         }
176     } else {
177         dif = 0
178     }
179     return if (dif >= 15L){
180         "No Update"

```

Figure 5.7: Code snippet of the function for update detection of random addresses.

On the other hand, when a match is made with an already existing MAC address item, a comparison of temporal data is conducted. This involves examining the timestamp associated with the currently scanned advertising packet in relation to the timestamp when the address was first recorded in the local file. If the temporal difference exceeds the officially recommended update interval, which is set at 15 minutes, it indicates that the address has remained unchanged for an extended period.



# Chapter 6

## Evaluation and Results

Throughout this study, a series of carefully planned experiments were conducted, which fall into two primary categories. The experiments were performed using a Realme 9 Pro 5G smartphone of Android Version 12, with the HomeScout app.

The first stage comprises an experiment aimed at evaluating the results of known devices' security levels with the combination of HomeScout's identification and categorization features. Section 6.1 provides a detailed explanation of this experiment and an in-depth analysis of its results.

The second part of the experimentation involves an evaluation of the application's availability. This evaluation was conducted across several diverse environments with a substantial prevalence of Bluetooth devices. Section 6.2 will provide a comprehensive delineation of the experimental process, an analysis of the results, and a discussion of potential contributing factors.

### 6.1 Evaluation of the Single Known Device

The experiment aims to validate the security assessment results for several known devices. It will be conducted in conjunction with HomeScout's original recognition classification feature for a set of target devices. It is essential to note that the experimental setting is deliberately simplified, featuring a controlled environment characterized by a defined list of devices, as illustrated in the Table 6.1. This choice is based on the understanding that straightforward environments, with a limited and stable number of active Bluetooth devices, provide a more conducive setting for precise device identification and analysis compared to complex, public environments.

The challenge of creating an environment exclusively containing known devices arises due to the comprehensive nature of Bluetooth scanning. Consequently, in this particular experiment, the target devices are placed in a separate room where no other Bluetooth-enabled devices are present. Drawing on previous test results obtained by HomeScout

Central Devices	Peripheral Devices
iPhone	AirTag
MacBookPro	Chipolo ONE Spot
	HUAWEI Smart Band
	Sony Headphone WH-1000XM3

Table 6.1: The list of target known devices.

[50], it is observed that the RSSI for the Tracker devices out the room consistently registers below -85. In contrast, the RSSI for the device placed in close proximity to the smartphone running the HomeScout application typically falls within the range of -30 to -50. Therefore, the target devices' RSSI is expected to be in the range of -30 to -85.

The experimental methodology relies on combining the device category information provided by HomeScout with the RSSI information. This combination enhances the reliability of device identification by establishing a correspondence between the scanned Bluetooth address and a known device.

The experiment starts with a comprehensive security assessment of the Tracker device. During the experiment, the scanning operation was executed once, followed by a corresponding scoring, and this sequence was reiterated multiple times. In Figure 6.1, the scanning and rating results for the AirTag device are presented. It's important to note that the specific AirTag employed in this experiment was categorized as a low-security device.

Upon closer examination of the AirTag's characteristics, it was revealed that this particular device utilizes random static addresses, which have not been refreshed for a duration exceeding 15 minutes. However, the scan results involved the presence of an unknown AirTag device that employed a random resolvable private address with an update cycle of less than 15 minutes. The observed discrepancy in address types and updating behaviors observed in the AirTags during the experiment may be attributed to their usage history. Specifically, the tracker was not used for a long period, thereby resulting in the Bluetooth MAC address remaining static.

However, as demonstrated in Figure 6.2, both the known and unknown Chipolo devices, whether the targets or others, update their address with an interval shorter than 15 minutes. Even though both tracker devices exhibit different security characteristics in the security rating. Obviously, Chipolo ONE Spot performs better in terms of security.

Subsequently, experiments were conducted on portable devices that are commonly owned by individuals, such as smart brands and Bluetooth-enabled headsets. Specifically, the Huawei Smart Band and Sony WH-1000XM3 were selected as the test devices for experimentation. As delineated in Figure 6.3, sub-figures (a) and (f), these two devices are discernible through the classification results obtained from the scanning process. Furthermore, the confirmation of the addresses of the target devices is achieved through the information derived from the two central devices, the laptop and the mobile phone, as depicted in Figure 6.3, sub-figures (d) and (e). Sub-figures (b) and (c) present the security-related insights concerning the two target devices.

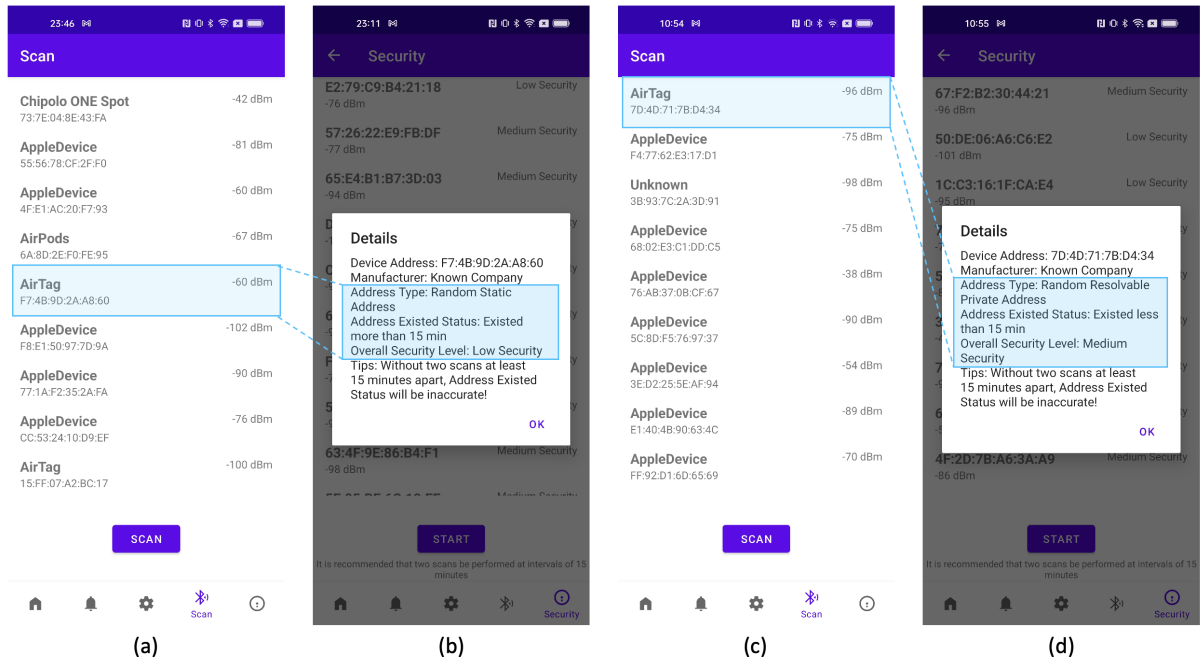


Figure 6.1: The results of AirTag devices, where sub-picture (a), (b) are from a target known AirTag in the experiment room, while (c), (d) are from an unknown AirTag out of the test room.

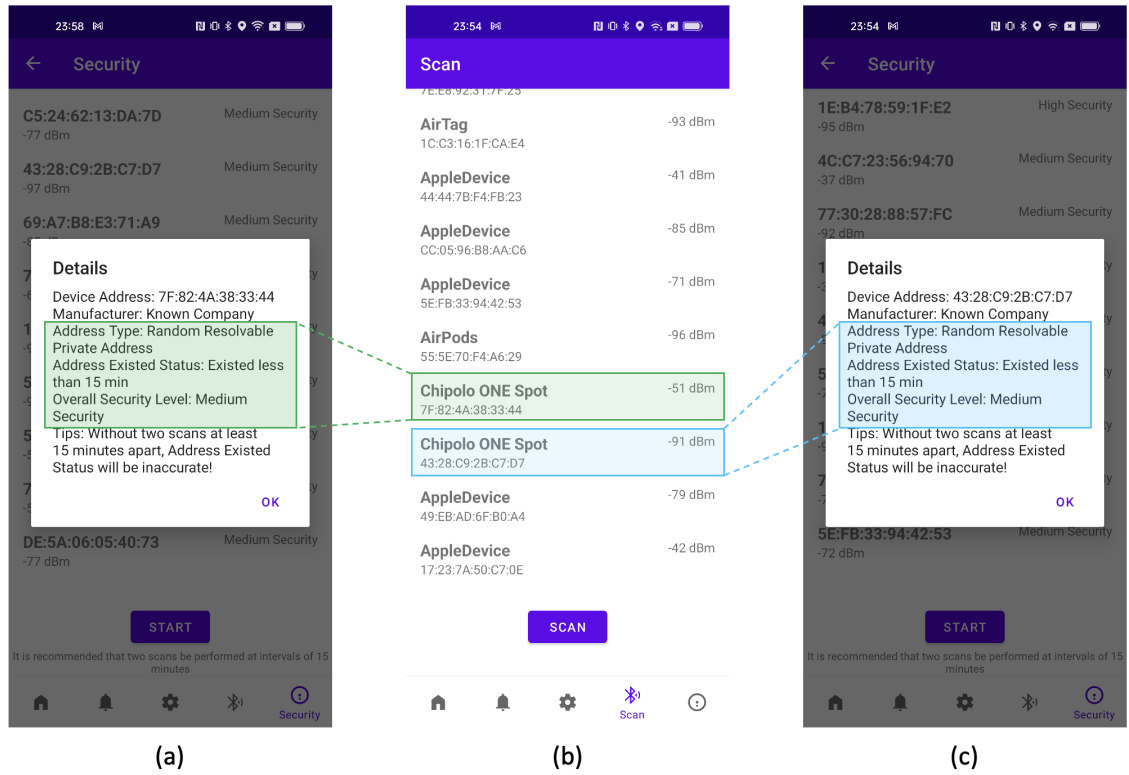


Figure 6.2: The results of AirTag devices, where sub-picture (b) is the results from Scan Page, and (a), (c) are security results from known and unknown devices.

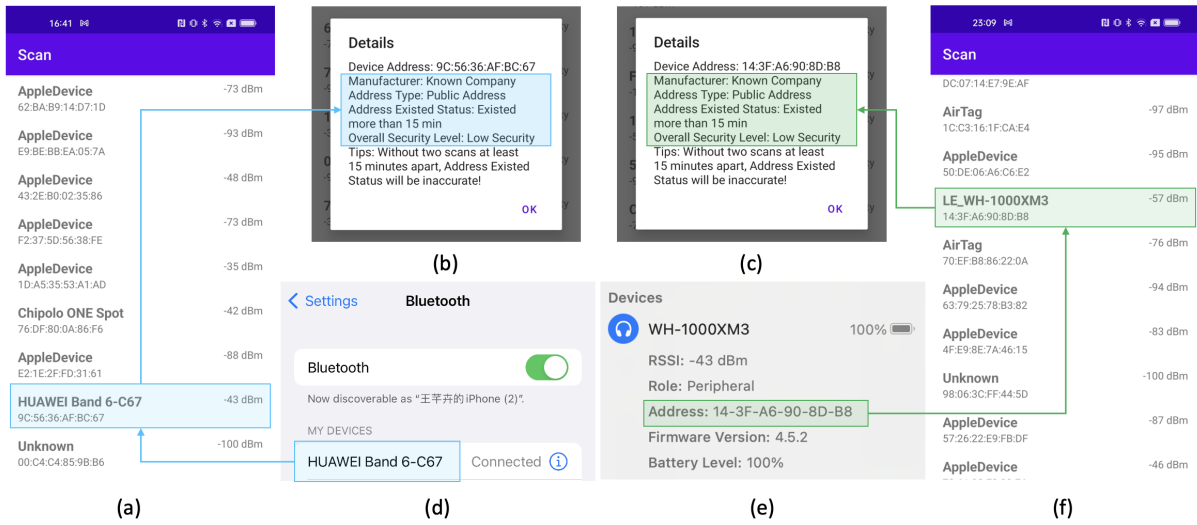


Figure 6.3: The results of two commonly carried peripheral devices, where (a) and (f) present part of the results of scanning, (b) and (c) illustrate the security details of the two devices, and sub-figure (d), (e) show the Bluetooth information obtained by central devices.

Both of the target devices have been categorized as low-security devices due to their unique and unchanging public addresses. The official documentation reveals that the OUI 9C-56-36 is allocated to Huawei Device Co., Ltd, while 14-3F-A6 is attributed to Sony Home Entertainment & Sound Products Inc. The public address OUIs are consistent with the manufacturer of the known devices. In Sony’s case, the OUI further discloses details about the device type, suggesting it may belong to the category of entertainment sound devices, such as headphones or stereo products.

Notably, the Huawei Smart Band is exclusively scanned by HomeScout when it is not engaged in an active connection, signifying that it is in a state of readiness for connection establishment. On the other hand, the Sony headphones are scanned by the application regardless of their Bluetooth connection status, including periods of Bluetooth pairing and even during usage after a successful connection. However, it’s important to acknowledge that BLE broadcast packets lack sufficient information to definitively determine the state of a BLE device. Therefore, classifying both of these test devices as low-security is a reasonable and consistent choice, given the limited data available for fine-grained security differentiation.

Regarding both central devices, their Bluetooth addresses are easily identifiable. As shown in Figure 6.4, sub-figure (a) displays the Bluetooth address of the MacBookPro, and sub-figure (b) shows the Bluetooth address of the iPhone. It’s important to note that neither the undiscoverable MacBookPro nor the discoverable iPhone appears in the scan results. It is due to the central device’s role as the primary connection initiator. Central devices always receive advertising packets instead of broadcasts, resulting in the absence of captured advertising packets during passive monitoring.



Figure 6.4: The Bluetooth addresses of the two central devices.

## 6.2 Evaluation in Real Environments

The experiments in this phase were conducted in a real and complex environmental setting. The presence of numerous BLE devices in public environments allowed for the collection of extensive and diverse datasets.

### 6.2.1 The Experiment in A Residential District

An initial experiment took place in a residential district, during which the scoring process was repeated twice after each interval spanning more than 15 minutes. The total temporal window of this process is three hours. Consequently, a total of 20 scanning sessions were conducted. Subsequently, this protocol yielded a dataset consisting of 1179 security rating data items. After removing duplicates, 226 distinct Bluetooth addresses are extracted. The rationale for this relatively high degree of repetition can be attributed to the temporal context of the experiment, which was executed during the night. During this period, BLE devices within the residential area displayed characteristic stability, with relatively few new devices entering or departing from the experimental area during the experiment.

Sub-figure (a) in Figure 6.5 provides an illustrative summary of the statistical insights obtained from the experimental dataset. The preeminent segment of the pie chart corresponds to 'random resolvable private addresses,' constituting a substantial majority at 67.44% in terms of prevalence among the observed devices. Following this, 'random static addresses' represent 19.02% of the dataset. In contrast, 'random unresolvable addresses' and 'public addresses' exhibit notably lower. This discrepancy may be attributed to the inherent characteristics of these addresses. Random unresolvable addresses are not possible to be resolved by any other devices. Additionally, the IEEE's allocation of a limited number of OUIs. Thus, impose constraints on the use of these two address types.

Sub-figure (b) in Figure 6.5 provides an overview of the address update frequencies among the devices. A noteworthy 94.25% of the addresses are observed to undergo updates within

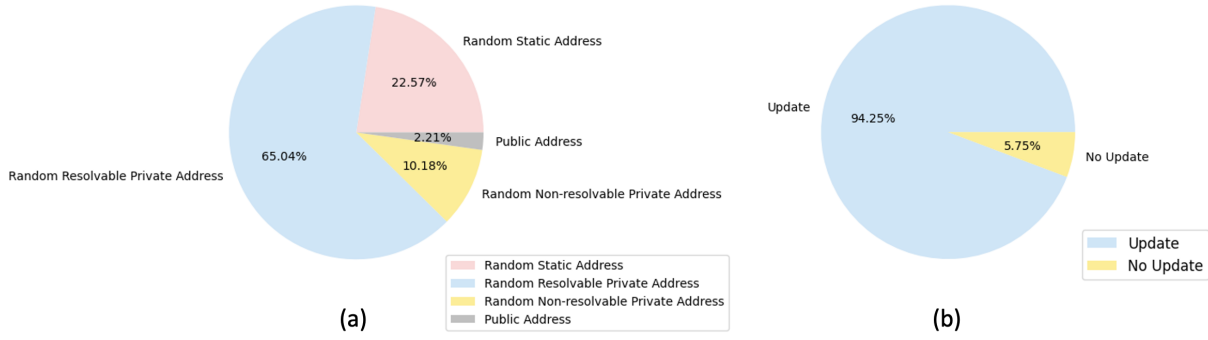


Figure 6.5: The distribution of address types of scanned BLE devices.

Static Address	Resolvable Address	Non-resolvable Address
5.88%	2.72%	4.35%

Table 6.2: The proportions of slow update devices of each address type.

a 15-minute interval. Only 3.54% of the addresses exhibit update periods exceeding 15 minutes after excluding unchanged public addresses. For all types of random addresses, there are instances of protracted address update intervals. Given the substantial variability in the device population employing distinct address types, Table 6.2 outlines the percentage of devices whose addresses are updated at a slow rate for each address type. Interestingly, devices utilizing random static addresses display the highest percentage of infrequent updates. This phenomenon may be attributed to the necessity of a device reboot for updating the addresses of this type, rendering frequent high-frequency updates a challenging endeavor.

Figure 6.6 portrays a visual representation of the statistical findings pertaining to the security levels of the 226 collected Bluetooth addresses. Among these, a significant 86.28% were categorized as medium-security devices, while 10.18% were designated as high-security devices. Conversely, a mere 3.54% were classified as low-security devices.

It is important to emphasize that, as elucidated in previous discussions, in the case where all devices are from known companies, the security levels are mainly based on certain address characteristics. Specifically, low-security devices are using public addresses or random static addresses that remain unaltered for durations exceeding 15 minutes. Considering that distinction rests on the understanding that both resolvable and non-resolvable private addresses undergo periodic updates, with the sole distinction residing in the update interval. Static addresses, on the other hand, exhibit a resemblance to public addresses, as they tend to remain unaltered throughout the device's lifecycle. Thus, if no address update is detected within a 15-minute timeframe, the static address is categorized as low-security, while devices that update their address within this interval are classified as medium-security. High-security devices use non-resolvable private addresses, as they exhibit a heightened degree of security. To be specific, resolvable private addresses, despite employing asymmetric encryption, may still be susceptible to replay attacks when whitelisting is set [68], thereby warranting their classification as moderately secure in contrast to highly secure non-resolvable private addresses.

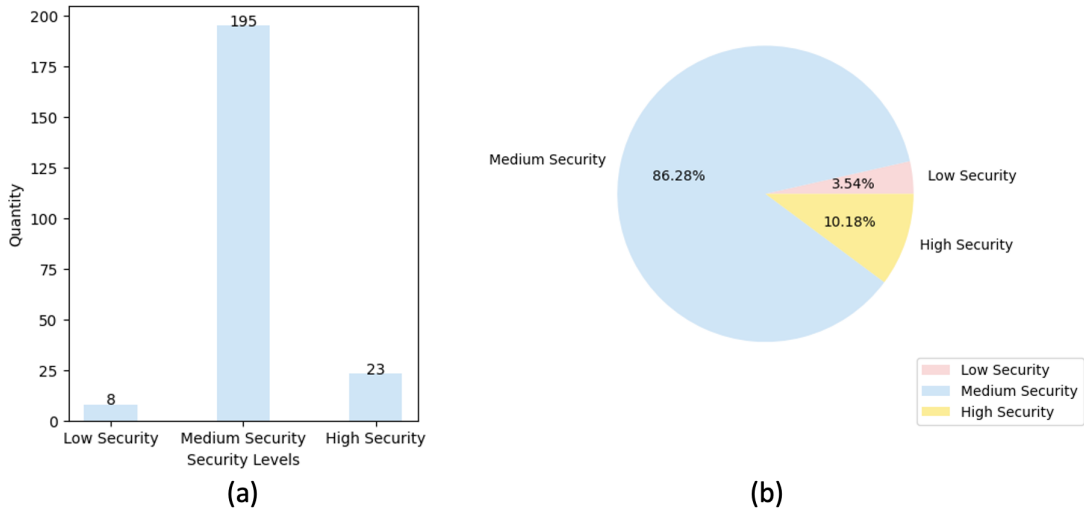


Figure 6.6: The distribution of security levels of scanned BLE devices in a residential environment.

### 6.2.2 The Experiment in A University Building

Comparable experiments were undertaken at the teaching building within the Oerlikon campus of the University of Zurich. The experiment was scheduled on a weekday, a deliberate choice attributed to the heightened prevalence of BLE devices within the building during the active period. Therefore, the experiment was conducted with a reduced frequency of running the application, factoring in the relatively long time span during which individuals typically remain within the academic buildings on weekdays due to classes or self-learning.

In this particular experiment, a series of 10 security scoring operations were executed over 3 hours. Subsequent to the data collection process, a total of 680 device security assessments were obtained, with 275 distinct data entries after eliminating duplicate records.

In a manner akin to the previous experiment, it is noteworthy that all devices incorporated in the dataset are traceable to known manufacturer companies. Fig.6.7 serves to elucidate the distribution of address types and update status of acquired devices. Notably, a similar pattern was observed in the academic building context, where resolvable private addresses were the majority, accounting for more than 50% of the recorded addresses. However, within the school building, a substantial presence of non-resolvable private addresses is evident, distinguishing it from the residential environment. Additionally, there is a marginally heightened proportion of devices undergoing updates within the 15-minute window, reflecting a heightened emphasis on user privacy and security within the public academic environment.

This phenomenon indicates that academic buildings are host to a greater number of devices that hold stringent requirements pertaining to user data protection, as manifested by the adoption of non-resolvable private addresses. Fig.6.8, on the other hand, provides insights into the quantitative and proportional representation of devices categorized within

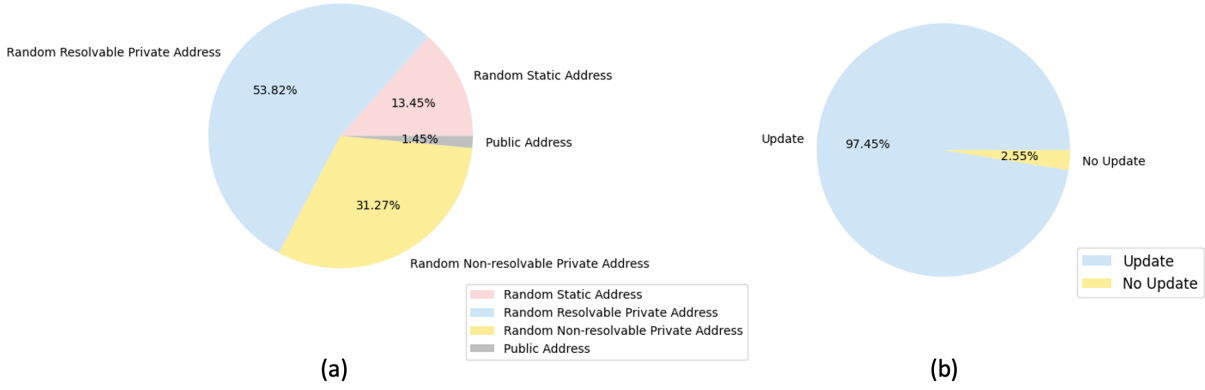


Figure 6.7: The distribution of address types and update status of observed devices in a university building.

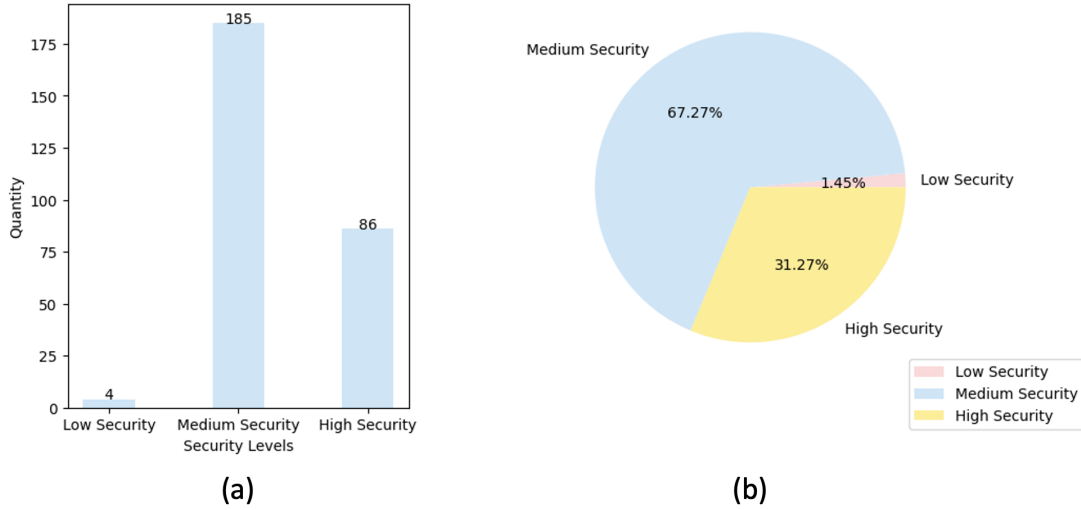


Figure 6.8: The distribution of security levels of scanned BLE devices in a university building.

each level of security. In comparison to the residential area, the academic institutions demonstrate a relatively larger percentage of high-security devices, while the presence of low-security devices is noticeably reduced. This difference can be attributed to the inherent tendency of academic-related devices in public teaching environments to prioritize data privacy and security.

### 6.2.3 The Experiment in A Railway Station

Lastly, the experiment was carried out within the Oerlikon train station, a place characterized by a constant influx and egress of individuals. This continuous stream of commuters and travelers inherently led to relatively brief intervals during which BLE devices remained within the station premises. The heightened mobility of the devices afforded a favorable environment for the accumulation of substantial datasets.



The experimental process involved conducting two security scorings every 15 minutes over a period of one and a half hours, 12 scans in total. This resulted in a dataset of 1035 data points, which, after eliminating duplicates, was refined into 415 distinct records. This approach to data collection leverages the station’s dynamic nature, facilitating the procurement of an intricate dataset within the confines of this high-mobility environment.

Unlike the two previous experiments, this particular study observed devices coming from unknown manufacturers, constituting a modest 2.41% of the overall dataset. This deviation from the previous experiments is likely due to the considerably more complex and dynamic environment encountered within train stations, as opposed to the relatively more stable residential and school settings.

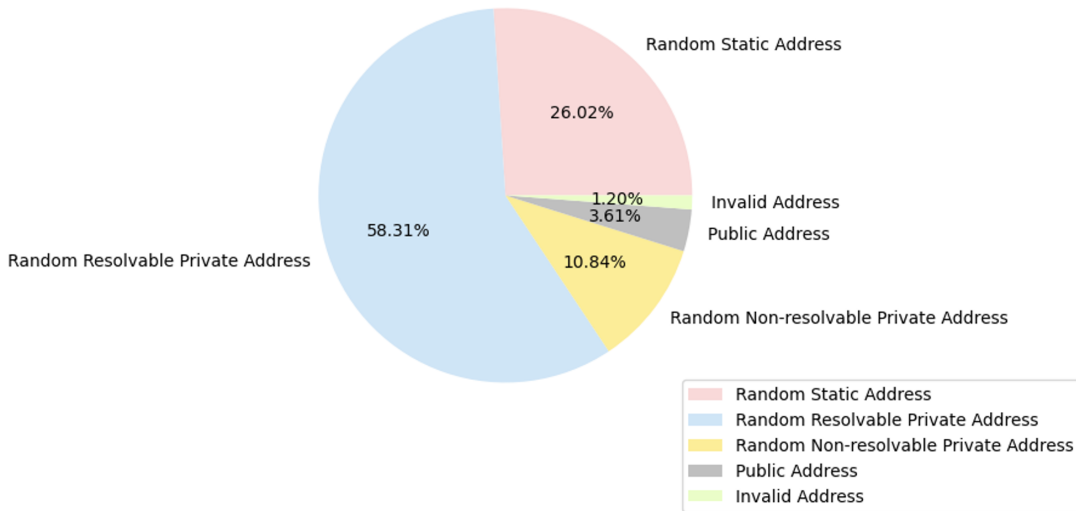


Figure 6.9: The distribution of address types of scanned BLE devices in a train station.

Figure 6.9 is illustrative of the distribution of the collected device address types. It is worth noting that there exists a minor fraction of 1.2% designated as “invalid addresses,” which do not adhere to the conventional criteria of public addresses, lacking an Organizationally Unique Identifier (OUI) as an address prefix, nor do they align with the established patterns of the three primary random address categories. However, the distribution of address types is similar to that observed in residential areas. Random resolvable private addresses predominate, surpassing the 50% of the total, followed by random static addresses, random non-resolvable private addresses, and public addresses. A noteworthy observation is that in living environments like train stations and residential areas, devices utilizing random static addresses outnumber those using random unresolvable private addresses. This phenomenon may be attributed to the popularity of random static addresses as a cost-effective and easily maintainable alternative to public addresses, as they do not incur additional expense, and the lack of data protection in entertaining devices like Sony headphones.

Figure 6.10 delineates the quantity and percentage representation of devices within each security tier. Evidently, there are more low-security devices within train stations in comparison to residential and school settings, revealing that there are more insecure devices

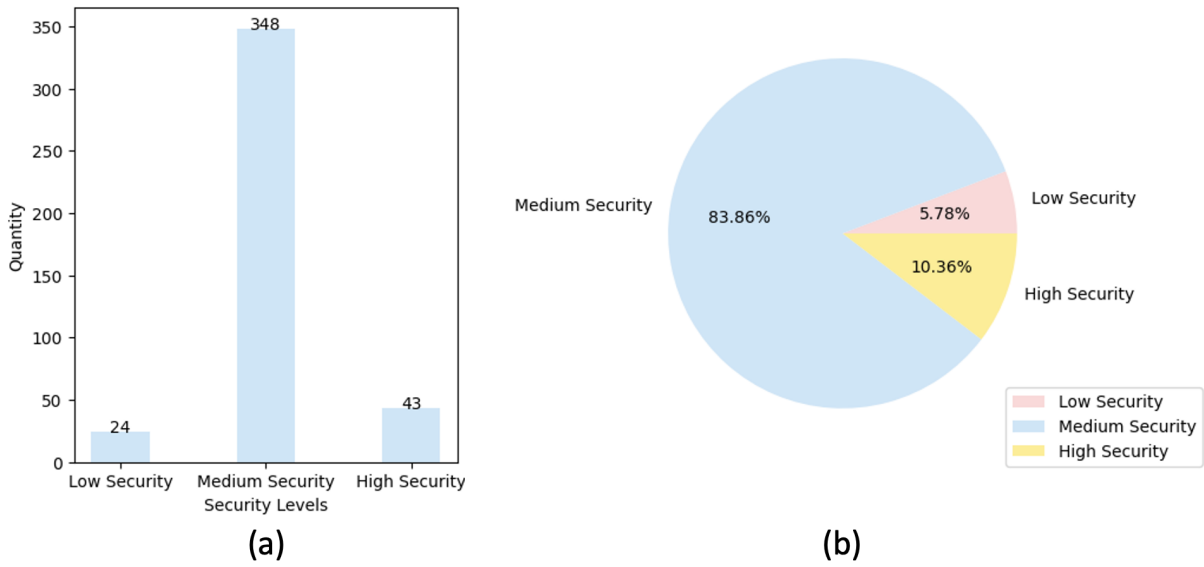


Figure 6.10: The distribution of security levels of scanned BLE devices in a train station.

in a complex environment with high personnel mobility. Nevertheless, it is important to acknowledge the presence of potential biases due to the intrinsic uncertainties and the high mobility characteristic of the train station. Despite this, a general trend in the distribution remains consistent, with medium-security devices surpassing the 50%, and an excess of high-security devices in relation to low-security ones.

# Chapter 7

## Conclusion and Future Work

The concluding chapter highlights the contributions and limitations of this study. Additionally, it provides recommendations for advancing the development of HomeScout and outlines potential avenues for future research in security scoring for BLE devices.

### 7.1 Conclusion

The primary objective of this research is to provide Android users, regardless of their expertise, with an understanding of the security situations of surrounding devices. This thesis explores existing literature on security scoring, particularly for IoT and BLE devices, and identifies deficiencies and gaps in current studies.

This thesis's first significant contribution lies in its comprehensive examination of the knowledge domain involving IoT and BLE technologies. Notably, it reveals a dearth of research on establishing robust security scoring frameworks for BLE devices. In current literature, some approaches involve inspecting data packets exchanged over the Internet to evaluate the security posture of IoT devices. In contrast, some other approaches offer security recommendations or guidelines but lack specific, structured scoring criteria. Consequently, users are often left to rely on their expertise to conduct security assessments based on these guidelines. This study seeks to address these limitations and provide an easier and more structured framework for security assessment in the context of BLE devices.

Taking inspiration from IoTAG [25], this thesis delves into the comprehensive analysis of Bluetooth broadcast packet data, acquired from the laptop and the smartphone. Subsequently, it formulates precise security scoring criteria tailored for BLE devices, particularly based on passive monitoring of BLE broadcast packets. This contribution constitutes the second pillar of this study.

Recognizing that evaluating devices against these guidelines necessitates a certain level of expertise, this effort goes even further. It aims to cater to most Android users, who may lack the required technical proficiency but seek a tangible means to assess the security of

the BLE devices in their nearby environment. To address this need, the analysis expands the HomeScout prototype and implements the security scoring functionality within it, marking the third significant contribution of this research.

Furthermore, this thesis includes the design and execution of a series of meticulously designed experiments. These experiments validate the practicality and reasonableness of the newly introduced security scoring feature. Ultimately, the analysis of the results of these security scoring experiments, conducted in complex real-world settings, sheds light on prevailing security attributes exhibited by BLE devices. Notably, there is a preponderance of devices categorized as medium-secure, and high-security devices outnumber their low-security counterparts in the remaining devices..

In summary, the Android application that has been extended demonstrates the capability to furnish security rating of BLE devices, catering to a diverse user spectrum with varying levels of expertise. By employing the proposed security levels or delving into the security details, users are afforded the means to gain insights into the security attributes of the nearby devices. Furthermore, a series of experiments has been conducted to scrutinize the availability and reasonableness of the application's functionality.

This thesis is a fundamental cornerstone for forthcoming studies in conjunction with the extended Android application. The future work is poised to refine the granularity of the security scoring system, thereby providing users with a more comprehensive and nuanced security evaluation. The overarching objective remains to enhance the user's capacity to intuitively understand the security implications of the BLE devices encountered in their daily life.

## 7.2 Limitation and Future Work

Several constraints need to be considered when elucidating the findings of this work. First limitations is the exclusive reliance on the passive monitoring of BLE advertising packets as the foundation for security ratings. This approach inevitably yielded relatively limited security-relevant indicators for security scoring criteria, resulting in less comprehensive security ratings. In future studies, other approaches could be considered to broaden the scope of data acquisition. For instance, retrieving data packets transferring through the Internet enables an analysis of the security posture of central devices. Simultaneously, pursuing a proactive strategy, involving the attempts to send connection requests to scanned peripheral devices, can supplement additional security-related insights.

Simultaneously, it is essential to acknowledge the inherent complexities in designing a precise and meaningful numerical scoring system, primarily stemming from the limitation of available dimensions for security assessment. Consequently, this research elects to adopt a pragmatic approach by employing a three-level security rating system to facilitate users' approximate comprehension of a device's overall security. Furthermore, it is recognized that users with diverse requirements may employ distinct criteria when evaluating device security, a relatively abstract concept.

In the context of this thesis, the scoring criteria are encapsulated within the application's functionality, allowing users to access security details and the accompanying security level proposed by the application. Future research could enhance the flexibility by permitting users to independently adjust the weight assigned to each security metric, tailoring the assessment process to their specific preferences.

Acknowledging the monumental volume of manufacturer data, encompassing a vast registry of 3,318 manufacturers as documented officially, is crucial. This wealth of information, coupled with the absence of a comprehensive examination pertaining to the security of the devices from different manufacturers, necessitates that this application refrains from offering a highly granular scoring system that delves into the nuances of security measures undertaken by distinct manufacturing entities. Prospective research could involve the development of a more refined and nuanced security scoring system, one that takes into account variations in security measures among different manufacturers, thereby fostering a more comprehensive and differentiated assessment.

The application in this thesis is a prototype and not ready for the market. Moreover, certain limitations are inherent in the implementation of the extended functionality. Due to the limitation of time and the absence of an urgent requirement for the prototype to accommodate extensive data retrieval, the extension incorporates a CSV file for the archival of historical scoring data. While this methodology offers direct data storage and sharing capabilities, it concurrently constrains the application's capacity to effectively manage substantial datasets. In prospective iterations, the prototype feature may be subject to refinement and consideration for development utilizing the MVVM architectural pattern, which can potentially enhance scalability and data management capabilities.



# Bibliography

- [1] Assigned Numbers | Bluetooth® Technology Website — bluetooth.com. <https://www.bluetooth.com/specifications/assigned-numbers/>. [Accessed 14-08-2023].
- [2] Identify your MacBook Air model — support.apple.com. <https://support.apple.com/en-us/HT201862>. [Accessed 07-08-2023].
- [3] Identify your MacBook Pro model — support.apple.com. <https://support.apple.com/en-us/HT201300>. [Accessed 07-08-2023].
- [4] Nmap: the Network Mapper - Free Security Scanner — nmap.org. <https://nmap.org/>. [Accessed 07-08-2023].
- [5] ScanRecord | Android Developers — developer.android.com. <https://developer.android.com/reference/kotlin/android/bluetooth/le/ScanRecord>. [Accessed 20-09-2023].
- [6] VARIOt database entry ontology — variotdbs.pl. <https://www.variotdbs.pl/ref/variotentry/>. [Accessed 09-10-2023].
- [7] Welcome to The Public Listing For IEEE Standards Registration Authority — regauth.standards.ieee.org. <https://regauth.standards.ieee.org/standards-ra-web/pub/view.html#registries>. [Accessed 19-10-2023].
- [8] Maha Ali Allouzi and Javed I Khan. Identifying and modeling security threats for iomt edge network using markov chain and common vulnerability scoring system (cvss). *arXiv preprint arXiv:2104.11580*, 2021.
- [9] Pooja Anand, Yashwant Singh, Arvind Selwal, Pradeep Kumar Singh, and Kayhan Zrar Ghafoor. Ivqfiot: An intelligent vulnerability quantification framework for scoring internet of things vulnerabilities. *Expert Systems*, 39(5):e12829, 2022.
- [10] Vafa Andalibi, Eliot Lear, DongInn Kim, and L Jean Camp. On the analysis of mud-filesâ interactions, conflicts, and configuration requirements before deployment. In *The Fifth International Conference on Safety and Security with IoT: SaSeIoT 2021*, pages 137–157. Springer, 2022.
- [11] Ganes Raj Muthu Arumugam, Saravanan Muthaiyah, and Thein Oak Kyaw Zaw. Towards optimization of patients’ turnaround time using bluetooth low energy based solutions. *Journal of Telecommunications & the Digital Economy*, 10(4), 2022.

- [12] Lu Bai, Fabio Ciravegna, Raymond Bond, and Maurice Mulvenna. A low cost indoor positioning system using bluetooth low energy. *Ieee Access*, 8:136858–136871, 2020.
- [13] Arup Barua, Md Abdullah Al Alamin, Md Shohrab Hossain, and Ekram Hossain. Security and privacy threats for bluetooth low energy in iot and wearable devices: A comprehensive survey. *IEEE Open Journal of the Communications Society*, 2022.
- [14] Primož Bencak, Darko Hercog, and Tone Lerher. Indoor positioning system based on bluetooth low energy technology and a nature-inspired optimization algorithm. *Electronics*, 11(3):308, 2022.
- [15] Rafael I Bonilla, Juan J Crow, Luigi S Basantes, and Luis G Cruz. A metric for measuring iot devices security levels. In *2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech)*, pages 704–709. IEEE, 2017.
- [16] Matthias Cäsar, Tobias Pawelke, Jan Steffan, and Gabriel Terhorst. A survey on bluetooth low energy security and privacy. *Computer Networks*, 205:108712, 2022.
- [17] Thomas Chiu, David Calero Luis, and Vinesh Jethva. Internet of things ble security. In *Proceedings of the 6th Annual Conference on Research in Information Technology*, pages 37–37, 2017.
- [18] Mario Collotta and Giovanni Pau. A novel energy management approach for smart homes using bluetooth low energy. *IEEE Journal on selected areas in communications*, 33(12):2988–2996, 2015.
- [19] Mario Collotta and Giovanni Pau. A solution based on bluetooth low energy for smart home energy management. *Energies*, 8(10):11916–11938, 2015.
- [20] Mathieu Cunche, Antoine Boutet, Claude Castelluccia, Cédric Lauradoux, and Vincent Roca. *On using Bluetooth-Low-Energy for contact tracing*. PhD thesis, Inria Grenoble Rhône-Alpes; INSA de Lyon, 2020.
- [21] Stephanie J Curtis, Asanka Rathnayaka, Fan Wu, Abdulla Al Mamun, Craig Spiers, Gordon Bingham, Colleen L Lau, Anton Y Peleg, Mehmet Rasit Yuce, and Andrew J Stewardson. Feasibility of bluetooth low energy wearable tags to quantify healthcare worker proximity networks and patient close contact: A pilot study. *Infection, disease & health*, 27(2):66–70, 2022.
- [22] Seyed Mahdi Darroudi, Carles Gomez, and Jon Crowcroft. Bluetooth low energy mesh networks: A standards perspective. *IEEE Communications Magazine*, 58(4):95–101, 2020.
- [23] Nick De Raeve, Quinten Van den Brande, Matthias De Schepper, Jo Verhaevert, Patrick Van Torre, and Hendrik Rogier. Wearable bluetooth low energy based miniaturized detection node for blind spot detection and warning system on vehicles. In *2021 6th International Conference on Smart and Sustainable Technologies (SpliTech)*, pages 1–5. IEEE, 2021.



- [24] Ruben E Figueiredo, Vitor Monteiro, Joao C Ferreira, Joao L Afonso, and Jose A Afonso. Smart home power management system for electric vehicle battery charger and electrical appliance control. *International Transactions on Electrical Energy Systems*, 31(4):e12812, 2021.
- [25] Sebastian Fischer, Katrin Neubauer, Lukas Hinterberger, Bernhard Weber, and Rudolf Hackenberg. Iotag: An open standard for iot device identification and recognition. In *The Thirteenth International Conference on Emerging Security Information, Systems and Technologies*, pages 107–113, 2019.
- [26] Mohamad Gharib, Paolo Giorgini, and John Mylopoulos. Copri v. 2âa core ontology for privacy requirements. *Data & Knowledge Engineering*, 133:101888, 2021.
- [27] Romeo Giuliano, Gian Carlo Cardarilli, Carlo Cesarini, Luca Di Nunzio, Francesca Fallucchi, Rocco Fazzolari, Franco Mazzenga, Marco Re, and Alessandro Vizzarri. Indoor localization system based on bluetooth low energy for museum applications. *Electronics*, 9(6):1055, 2020.
- [28] Carles Gomez, Joaquim Oller, and Josep Paradells. Overview and evaluation of bluetooth low energy: An emerging low-power wireless technology. *sensors*, 12(9):11734–11753, 2012.
- [29] Core Specification Working Group et al. Bluetooth core specification, v5.3. *BlueTooth SIG: Kirkland, WA, USA*, 7:1819–1845, 2021.
- [30] Michael Gruninger. Methodology for the design and evaluation of ontologies. In *Proc. IJCAI’95, Workshop on Basic Ontological Issues in Knowledge Sharing*, 1995.
- [31] Zohreh Hajiakhondi-Meybodi, Mohammad Salimibeni, Konstantinos N Plataniotis, and Arash Mohammadi. Bluetooth low energy-based angle of arrival estimation via switch antenna array for indoor localization. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–6. IEEE, 2020.
- [32] Gary F Hatke, Monica Montanari, Swaroop Appadwedula, Michael Wentz, John Meklenburg, Louise Ivers, Jennifer Watson, and Paul Fiore. Using bluetooth low energy (ble) signal strength estimation to facilitate contact tracing for covid-19. *arXiv preprint arXiv:2006.15711*, 2020.
- [33] Alexander Heinrich, Niklas Bittner, and Matthias Hollick. Airguard-protecting android users from stalking attacks by apple find my devices. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pages 26–38, 2022.
- [34] Alexander Heinrich, Milan Stute, Tim Kornhuber, and Matthias Hollick. Who can find my devices? security and privacy of apple’s crowd-sourced bluetooth location tracking system. *arXiv preprint arXiv:2103.02282*, 2021.
- [35] Robin Heydon and Nick Hunn. Bluetooth low energy. *CSR Presentation, Bluetooth SIG* <https://www.bluetooth.org/DocMan/handlers/DownloadDoc.aspx>, 2012.

- [36] Sandeep Kamath and Joakim Lindh. Measuring bluetooth low energy power consumption. *Texas instruments application note AN092, Dallas*, 2010.
- [37] Heikki Karvonen, Konstantin Mikhaylov, Dinesh Acharya, and Md Moklesur Rahman. Performance evaluation of bluetooth low energy technology under interference. In *13th EAI International Conference on Body Area Networks 13*, pages 147–156. Springer, 2020.
- [38] Pratibha Khandait, Neminath Hubballi, and Bodhisatwa Mazumdar. Iothunter: Iot network traffic classification using device specific keywords. *IET Networks*, 10(2):59–75, 2021.
- [39] Bastian Könings, Christoph Bachmaier, Florian Schaub, and Michael Weber. Device names in the wild: Investigating privacy risks of zero configuration networking. In *2013 IEEE 14th International Conference on Mobile Data Management*, volume 2, pages 51–56. IEEE, 2013.
- [40] Andrea Lacava, Valerio Zottola, Alessio Bonaldo, Francesca Cuomo, and Stefano Basagni. Securing bluetooth low energy networking: An overview of security procedures and threats. *Computer Networks*, 211:108953, 2022.
- [41] Cristian LazaroIU and Mariacristina Roscia. Ble to improve iot connection in the smart home. In *2021 10th International Conference on Renewable Energy Research and Application (ICRERA)*, pages 282–287. IEEE, 2021.
- [42] Jiazheng Lina. Design of smart home energy efficiency management system based on ble technology. In *2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*, pages 201–206. IEEE, 2022.
- [43] Joakim Lindh. Bluetooth low energy beacons. *Texas Instruments*, 2, 2015.
- [44] Chendong Liu, Yilin Zhang, and Huanyu Zhou. A comprehensive study of bluetooth low energy. In *Journal of Physics: Conference Series*, volume 2093, page 012021. IOP Publishing, 2021.
- [45] Franco Loi, Arunan Sivanathan, Hassan Habibi Gharakheili, Adam Radford, and Vijay Sivaraman. Systematically evaluating security and privacy for consumer iot devices. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, pages 1–6, 2017.
- [46] Somayya Madakam, Vihar Lake, Vihar Lake, Vihar Lake, et al. Internet of things (iot): A literature review. *Journal of Computer and Communications*, 3(05):164, 2015.
- [47] Parushi Malhotra, Yashwant Singh, Pooja Anand, Deep Kumar Bangotra, Pradeep Kumar Singh, and Wei-Chiang Hong. Internet of things: Evolution, concerns and security challenges. *Sensors*, 21(5):1809, 2021.
- [48] Salvador Martínez-Cruz, Luis A Morales-Hernández, Gerardo I Pérez-Soto, Juan P Benitez-Rangel, and Karla A Camarillo-Gómez. An outdoor navigation assistance system for visually impaired people in public transportation. *IEEE Access*, 9:130767–130777, 2021.

- [49] Francesca Meneghello, Matteo Calore, Daniel Zucchetto, Michele Polese, and Andrea Zanella. Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices. *IEEE Internet of Things Journal*, 6(5):8182–8201, 2019.
- [50] Katharina OE Müller, Louis Bienz, Bruno Rodrigues, Chao Feng, and Burkhard Stiller. Homescout: Anti-stalking mobile app for bluetooth low energy devices. In *2023 IEEE 48th Conference on Local Computer Networks (LCN)*, pages 1–9. IEEE, 2023.
- [51] Sandro Nizetić, Petar Šolić, Diego López-de-Ipiña González-De, Luigi Patrono, et al. Internet of things (iot): Opportunities, issues and challenges towards a smart and sustainable future. *Journal of Cleaner Production*, 274:122877, 2020.
- [52] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: a guide to creating your first ontology. 2001. See <http://protege.stanford.edu/publications>, 2004.
- [53] Aeddula Omsri Kumar, Flyborg Johan, Larsson Tobias, Anderberg Peter, Johan Sanmartin Berglund, and Renvert Stefan. A solution with bluetooth low energy technology to support oral healthcare decisions for improving oral hygiene. In *Proceedings of the 5th International Conference on Medical and Health Informatics*, pages 134–139, 2021.
- [54] Pascal Oser, Sebastian Feger, Paweł W Woźniak, Jakob Karolus, Dayana Spagnuolo, Akash Gupta, Stefan Lüders, Albrecht Schmidt, and Frank Kargl. Safer: Development and evaluation of an iot device risk assessment framework in a multinational organization. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–22, 2020.
- [55] Sebeom Park and Yosoon Choi. Analysis and diagnosis of truck transport routes in underground mines using transport time data collected through bluetooth beacons and tablet computers. *Applied Sciences*, 11(10):4525, 2021.
- [56] Nishitkumar Patel, Hayden Wimmer, and Carl M Rebman. Investigating bluetooth vulnerabilities to defend from attacks. In *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, pages 549–554. IEEE, 2021.
- [57] Fahad Qaswar, M Rahmah, Muhammad Ahsan Raza, A Noraziah, Basem Alkazemi, Z Fauziah, Mohd Khairul Azmi Hassan, and Ahmed Sharaf. Applications of ontology in the internet of things: A systematic analysis. *Electronics*, 12(1):111, 2022.
- [58] Yanzhen Qu and Philip Chan. Assessing vulnerabilities in bluetooth low energy (ble) wireless network based iot systems. In *2016 IEEE 2nd International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS)*, pages 42–48. IEEE, 2016.
- [59] Khatod Varsha Ritesh, Agata Manolova, and Maria Nenova. Abridgment of bluetooth low energy (ble) standard and its numerous susceptibilities for internet of things and

- its applications. In *2017 IEEE International Conference on Microwaves, Antennas, Communications and Electronic Systems (COMCAS)*, pages 1–5. IEEE, 2017.
- [60] Ronald L Rivest, Daniel Weitzner, Louise Ivers, Israel Soibelman, and Marc Zissman. Pact: Private automated contact tracing. *Retrieved December, 2:2020*, 2020.
- [61] Syed Rizvi, Ryan Pipetti, Nicholas McIntyre, Jonathan Todd, and Iyonna Williams. Threat model for securing internet of things (iot) network at device-level. *Internet of Things*, 11:100240, 2020.
- [62] Emilio Sansano-Sansano, Raúl Montoliu, Óscar Belmonte-Fernández, Fernando J Aranda, and Fernando J Álvarez. Continuous non-invasive assessment of gait speed through bluetooth low energy. *IEEE Sensors Journal*, 22(8):8183–8195, 2022.
- [63] Matti Siekkinen, Markus Hienkari, Jukka K Nurminen, and Johanna Nieminen. How low energy is bluetooth low energy? comparative measurements with zigbee/802.15.4. In *2012 IEEE wireless communications and networking conference workshops (WCNCW)*, pages 232–237. IEEE, 2012.
- [64] Seher İnci Taştan and Gökhan Dalkılıç. Smart home system using internet of things devices and mesh topology. In *2021 6th International Conference on Computer Science and Engineering (UBMK)*, pages 407–412. IEEE, 2021.
- [65] Martin Woolley. The bluetooth low energy primer. *Retrieved September, 15:2022*, 2022.
- [66] Kun Xia, Haibo Wang, Nan Wang, Wei Yu, and Tong Zhou. Design of automobile intelligence control platform based on bluetooth low energy. In *2016 IEEE Region 10 Conference (TENCON)*, pages 2801–2805. IEEE, 2016.
- [67] Ting Zhang, Jiang Lu, Fei Hu, and Qi Hao. Bluetooth low energy for wearable sensor-based healthcare systems. In *2014 IEEE healthcare innovation conference (HIC)*, pages 251–254. IEEE, 2014.
- [68] Yue Zhang and Zhiqiang Lin. When good becomes evil: Tracking bluetooth low energy devices via allowlist-based side channel and its countermeasure. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 3181–3194, 2022.

# Abbreviations

API	Application Programming Interface
APK	Android Application Package
ATT	Attribute Protocol
BLE	Bluetooth Low Energy
CRC	Cyclic Redundancy Check
CSF	Cybersecurity Framework
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
DDoS	Distributed Denial of Service
EU	European Union
FIRST	Forum of Incident Response and Security Teams
FTP	File Transfer Protocol
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HCI	Host Controller Interface
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IoTSF	Internet of Things Security Foundation
IRK	Identity Resolution Key
L2CAP	Logical Link Control and Adaption Protocol
LAN	Local Area Network
LSB	Least Significant Bit
MAC	Media Access Control
MDNS	Multicast Domain Name Resolution System
MSB	Most Significant Bit
MUD	Manufacturer Usage Description
MVVM	Model-View-ViewModel
NIAC	National Infrastructure Advisory Committee
NIST	National Institute of standards and Technology
OSINT	Open Source Intelligence
OUI	Organizationally Unique Identifier
PDU	Protocol Data Unit
PHY	Physical Layer
RSSI	Received Signal Strength Indicator

SIG	Special Interest Group
SIP	Session Initial Protocol
SM	Security Manager
SSH	Secure Shell Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UI	User Interface
UUID	Universally Unique Identifier
VARIoT	Vulnerability and Attack Repository for IoT
WLAN	Wireless Local Area Network
WPA	Wi-Fi Protocol Access

# Glossary

**BLE Device** Any wireless communication device that uses BLE technology for short-range connectivity.

**Central Device** Any BLE device that can initiate a connection.

**MAC Address** Media Access Control Address of a device.

**Peripheral Device** Any BLE device that is advertising, discovered by central devices, and can accept a connection.

**Public Address** A kind of Bluetooth address that is globally fixed and must be registered with the IEEE.

**Random Non-resolvable Private Address** A kind of Bluetooth address that cannot be resolved by any other device.

**Random Resolvable Private Address** A kind of Bluetooth address that be resolved through a pre-shared hash key.

**Random Static Address** A kind of Bluetooth address that is randomly generated.

**Scan Period** The time duration of the app scanning for BLE devices.





# List of Figures

3.1	The architecture for BLE. Combined with the figure 3-1 from [35] and the figure 2 from [65]. . . . .	10
3.2	The Link Layer state machine. Source: Figure 9 from [65]. . . . .	12
3.3	The packet structure of uncoded PHYs. Source: Figure 7 from [65]. . . . .	13
3.4	Format of a static address, where LSB is the abbreviation of the least significant bit, and MSB represents the most significant bit. Source: Figure 1.2 from [29]. . . . .	14
3.5	Format of a non-resolvable private address, where LSB is the abbreviation of the least significant bit, and MSB represents the most significant bit. Source: Figure 1.3 from [29]. . . . .	14
3.6	Format of a resolvable private address, where LSB is the abbreviation of the least significant bit, and MSB represents the most significant bit. Source: Figure 1.4 from [29]. . . . .	15
4.1	A part of MDNS packet for answer. . . . .	22
4.2	A part of MDNS packet for query. . . . .	22
4.3	Model information in one of MDNS packets. . . . .	23
4.4	The running outcomes of the Nmap Wrapper for Android on a Realme phone. . . . .	24
4.5	A frame of a captured broadcast packet with UberTooth One in Wireshark. . . . .	25
4.6	A frame of captured broadcast packets with UberTooth One in Wireshark. . . . .	26
4.7	Original payload of the same packet showed in Wireshark before and its explanation. . . . .	27
4.8	Packets with different types of PDU, where sub-figure (a) is from a SCAN_RSP PDU, sub-figure (b) is from an ADV_SCAN_IND PDU, sub-figure (c) is from an ADV_NONCONN_IND PDU, and sub-figure (d) is from an ADV_IND PDU. . . . .	28

4.9	Broadcast packet details with MAC addresses containing the word Samsung captured by Ubertooth One. . . . .	29
4.10	Broadcast packet details with MAC addresses containing the word Samsung captured by nRF52840 DK. . . . .	30
4.11	Two raw scanning results and corresponding printed fields from an Android smartphone. Subfigure (a) and (b) are from a Mi Smart Band, while (c) and (d) are from an Airtag. Additionally, (a) and (c) show all fields in the scanning results of two different devices; (b) and (d) demonstrate the raw scanning results of the two different devices. . . . .	30
4.12	Search result on Shodan with the keyword "iPhone". . . . .	32
4.13	Search result on Shodan with the keyword "MsczbookPro". . . . .	32
5.1	The workflow of the extended function. . . . .	42
5.2	Screenshots of the security scoring feature. (a) is the initial page of this feature, (b) is the result list after clicking the start button, and (c) is the pop-up window after clicking one item. . . . .	43
5.3	The process of data exchange integral to the security scoring function. . . .	44
5.4	Code snippet of the deviceCompany function, aiming to check if the device comes from a known manufacturer with its decimal company identifier. . .	45
5.5	Two examples of device address type inference. . . . .	46
5.6	Code snippet of the function for device address type inference. . . . .	47
5.7	Code snippet of the function for update detection of random addresses. . .	48
6.1	The results of AirTag devices, where sub-picture (a), (b) are from a target known AirTag in the experiment room, while (c), (d) are from an unknown AirTag out of the test room. . . . .	51
6.2	The results of AirTag devices, where sub-picture (b) is the results from Scan Page, and (a), (c) are security results from known and unknown devices. .	51
6.3	The results of two commonly carried peripheral devices, where (a) and (f) present part of the results of scanning, (b) and (c) illustrate the security details of the two devices, and sub-figure (d), (e) show the Bluetooth information obtained by central devices. . . . .	52
6.4	The Bluetooth addresses of the two central devices. . . . .	53
6.5	The distribution of address types of scanned BLE devices. . . . .	54
6.6	The distribution of security levels of scanned BLE devices in a residential environment. . . . .	55

6.7	The distribution of address types and update status of observed devices in a university building. . . . .	56
6.8	The distribution of security levels of scanned BLE devices in a university building. . . . .	56
6.9	The distribution of address types of scanned BLE devices in a train station.	57
6.10	The distribution of security levels of scanned BLE devices in a train station.	58



# List of Tables

2.1	A summary of references related to security scoring system. . . . .	8
3.1	The descriptions of seven states. Source: Table 2 from [65]. . . . .	12
3.2	The detailed information of seven PDUs used for legacy advertising, where P represents Peripheral and C means Central. Source: Table 4 from [65]. .	16
3.3	The detailed information of four PDUs used for extended advertising, where P represents Peripheral and C means Central; 1M represents LE 1M, Coded is LE Coded, and All indicates LE 1M, LE 2M and LE Coded. Source: A part of Table 5 from [65]. . . . .	16
4.1	Overview of open ports and services. . . . .	21
4.2	Security criteria. . . . .	37
4.3	All possible scores and corresponding descriptions. . . . .	39
6.1	The list of target known devices. . . . .	50
6.2	The proportions of slow update devices of each address type. . . . .	54



# Appendix A

## Experiments

All raw, and processed experimental data can be found here: <https://github.com/qianhui36/Extended-HomeScout/tree/master/experiments%20results>