TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Anton Kolisnetšenko 201751IVSB

Analysis of Common Security Vulnerabilities in the Bluetooth Low Energy Technology

Bachelor's thesis

Supervisor: Mohammad Tariq
Meeran
PhD

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Anton Kolisnetšenko 201751IVSB

Bluetooth Low Energy tehnoloogia enamlevinud turvanõrkuste analüüs

Bakalaureusetöö

Juhendaja: Mohammad Tariq
Meeran
PhD

Tallinn 2024

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Anton Kolisnetšenko

07.12.2023

Abstract

Bluetooth Low Energy has become a standard for connecting wearable wireless devices and interconnecting IoT appliances. However, the wireless nature of this standard makes the networks susceptible to exploitation. There are various different methods that can be used by attackers to identify vulnerabilities in BLE connection and exploit them.

In this thesis, the author analyzes the most prevalent vulnerabilities present in BLE, explores ways to exploit BLE vulnerabilities using low-cost devices and proposes mitigations.

The author presents a number of techniques for identifying and conducting low-level attacks based on existing work. Then experiments are conducted using low-cost devices. Results of the conducted experiments indicate that a few security holes are present. The discovered vulnerabilities are described and possible mitigations are discussed.

This thesis is written in English and is 43 pages long, including 6 chapters, 22 figures and 2 tables.

Annotatsioon
Bluetooth Low Energy tehnoloogia enamlevinud turvanõrkuste analüüs

Bluetooth Low Energy on muutunud kantavate traadita seadmete ühendamise ja IoT ühendamise standardiks. Selle standardi juhtmevaba olemus muudab võrgud aga ekspluateerimisele vastuvõtlikuks. Ründajad saavad BLE-ühenduse haavatavuste tuvastamiseks ja nende ärakasutamiseks kasutada erinevaid meetodeid.

Selles lõputöös analüüsib autor BLE-s esinevaid kõige levinumaid haavatavusi, uurib võimalusi BLE haavatavuste ärakasutamiseks odavate seadmete abil ja pakub välja leevendusi.

Autor esitab olemasoleva töö põhjal mitmeid tehnikaid madala tasemega rünnakute tuvastamiseks ja läbiviimiseks. Seejärel tehakse katseid odavate seadmetega. Läbiviidud katsete tulemused näitavad, et esinevad mõned turvaaugud. Kirjeldatakse avastatud turvaauke ja arutatakse võimalikke leevendusi.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 43 leheküljel, 6 peatükki, 22 joonist, 2 tabelit.

List of abbreviations and terms

ATT	Attribute Protocol
BLE	Bluetooth Low Energy
BR	Basic Rate
CRC	Cyclic Redundancy Check
CSRK	Connection Signature Resolving Key
EDR	Enhanced Data Rate
FHSS	Frequency-Hopping Spread Spectrum
GAP	Generic Access Profile
GATT	Generic Attribute Profile
HCI	Host-Controller Interface
HS	High Speed
IEEE	Institute of Electrical and Electronics Engineers
IoT	Internet of Things
IRK	Identity Resolving Key
ISM	Industrial, Scientific, and Medical
KNOB	Key Negotiation Of Bluetooth
LE-SC	LE Secure Connection
LL	Link Layer
L2CAP	Logic Link Control and Adaptation Protocol
MITM	Man-In-The-Middle
OOB	Out-Of-Band
PHY	Physical Layer
RF	Radio frequency
RSSI	Received Signal Strength Indicator
SKD	Session key diversifier
SMP	Security Manager Protocol
STK	Short Term Key
TK	Temporary Key
UUID	Universally unique identifiers

Table of Contents

List of Figures

List of Tables

1 Introduction

IEEE 802.15.1 standard, also known as Bluetooth, is one of the main components of wireless communications. In version 4.0 of the standard Bluetooth Low Energy (BLE) was introduced. BLE standard is targeted at low power and limited power supply devices with its ability to allow for low-energy device communication while maintaining standard communication range. [1, 2]

We are living in a time of seamless connection and intelligent interactions because to the widespread adoption of Bluetooth Low Energy (BLE) technology in many aspects of our everyday life. From fitness trackers to smart home devices, BLE has become a universal communication protocol, enabling the exchange of data between devices in an energy-efficient manner. But this widespread use of technology also raises questions about the security flaws in BLE implementations. [1]

The motivation for this research comes from the need to understand and address potential threats in a technology that has become integral to the Internet of Things (IoT) ecosystem. In this thesis, a general overview of the BLE protocol stack along with the description of the most common BLE vulnerabilities will be presented. After that, methods of identifiyng vulnerabilities using low-cost devices will be described and tested on practice. Finally, possible mitigations of found vulnerabilities will be discussed.

1.1 Problem statement

With the IoT technology becoming widespread and wearable wireless devices gaining in popularity, BLE has become a de facto standard for short-range wireless communication. As a result, securing this technology has become an essential task. However, despite the increasing popularity and usage of BLE enabled devices, there is a lack of comprehensive research and guidelines on addressing the security vulnerabilities and threats that come with this technology. Therefore, devices are still in danger due to many vulnerabilities being unpatched as well as misconfiguration by users and bad usage practices. [1]

1.2 Research questions

In order to address the problem of enhancing the security of Bluetooth Low Energy (BLE) technology, two questions have been formulated, focusing on identifying the most prevalent vulnerabilities within BLE and exploring effective mitigation strategies.

1. What are the most common vulnerabilities present in Bluetooth Low Energy (BLE) technology and what effective mitigation strategies can be employed to address these vulnerabilities?

2. How can an attacker exploit BLE vulnerabilities using low-cost tools?

1.3 Research goal

The aim of this thesis is to investigate the common security vulnerabilities associated with Bluetooth Low Energy protocol stack and to identify effective mitigation techniques and best practices to address these vulnerabilities.

1.4 Scope and limitations

The thesis will focus specifically on analyzing the security vulnerabilities of Bluetooth Low Energy technology, including weaknesses in authentication and authorization mechanisms, as well as issues related to spoofing or man-in-the-middle attacks. The thesis will not cover security of other wireless technologies, such as WiFi or cellular networks.

2 Literature review

To discuss potential threats and vulnerabilities in BLE devices it is important to first understand how BLE works. In this section, a general overview of Bluetooth will be given. Then, the protocol stack of BLE will be described and general function of each protocol will be explained.

2.1 Bluetooth overview

Bluetooth is a short range low power wireless communication technology. First version of Bluetooth came out in 1994 and became known as Bluetooth Basic Rate (BR). Initial versions of Bluetooth, 1.1 and 1.2 were ratified as the IEEE 802.15.1 standard. These versions introduced such important features as Host Control Interface and flow control and retransmission modes in Logic Link Control and Adaptation Protocol. Bluetooth BR supported data transmission speeds up to 1 Mbps. [2]

Version 2.0 of Bluetooth specification was released in 2004 and became known as Bluetooth Enhanced Data Rate (EDR), since it elevated transmission speeds from 1 Mbps to 3 Mbps. Version 2.1, released in 2007, introduced another important protocol – Secure Simple Pairing. [2]

In 2009, specification version 3.0 was introduced and it got adopted as the new IEEE 802.15.1 standard. It supported data transmission speeds up to 24 Mbps and became known as Bluetooth High Speed (HS). Bluetooth HS also included several improvements, such as enhanced power control and unicast connectionless data transmission. [2]

The IEEE 802.15.1 standard was updated again in 2010 with Bluetooth specification version 4.0. This version of Bluetooth introduced a new Bluetooth mode – Bluetooth Low Energy (BLE). Later versions of the standard, 4.1 and 4.2 introduced support for LTE and IPv6 and also included several additional security mechanisms. [2]

Bluetooth version 5.0 was released in 2016. The main update it brought was the strengthening of security mode 1, level 4. Specification made a key length of 128 bit mandatory, as opposed to the key length of 56 to 128 bits that devices could negotiate before. [2, 3]

Bluetooth version 5.2 was released in December of 2019, this version introduced security updates to Bluetooth Low Energy, introducing LE Power Control that enables devices to dynamically optimize the transmission power [4]. Bluetooth version 5.3 brought mainly enhancements to previous versions [5]. Current version of Bluetooth specification is 5.4, released in February 2023. It has brought two major improvements, such as Periodic Advertising with Responses and Encrypted Advertising Data. [6]

2.2 Overview of the Bluetooth Low Energy

Bluetooth protocol stack is divided into three layers: the Controller Layer, the Host Layer and the App Layer.

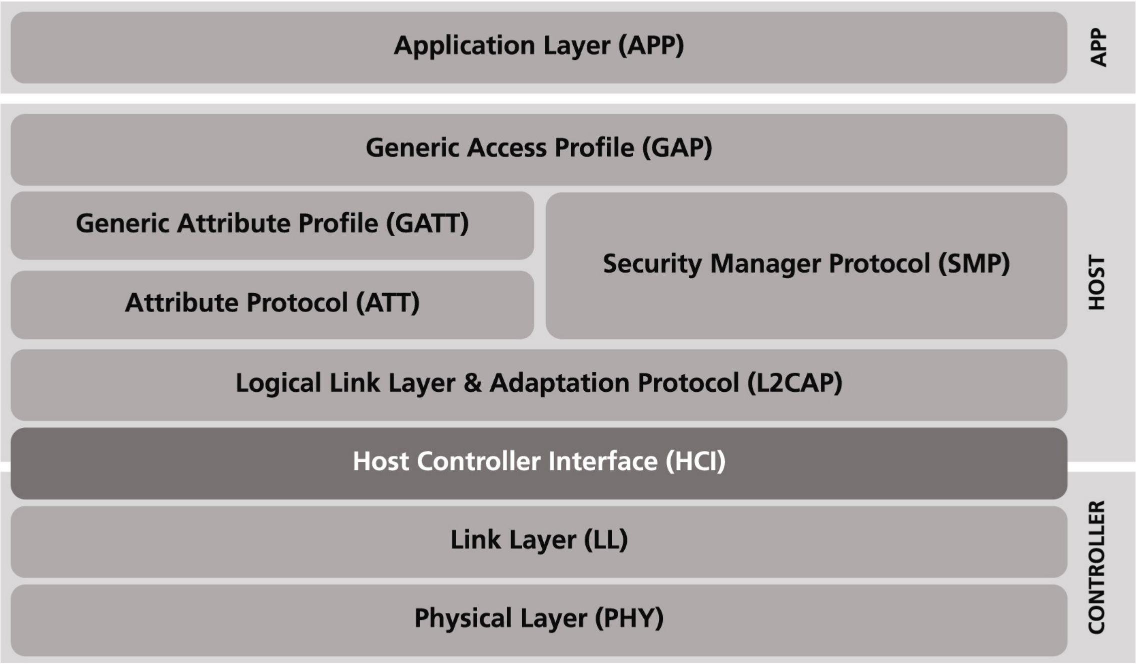

Application Layer (APP)
APP
Generic Access Profile (GAP)
Generic Attribute Profile (GATT)
Attribute Protocol (ATT)
Security Manager Protocol (SMP)
HOST
Logical Link Layer & Adaptation Protocol (L2CAP)
Host Controller Interface (HCI)
Link Layer (LL)
Physical Layer (PHY)
CONTROLLER

Figure 1: BLE protocol stack [3]

2.2.1 Controller Layer

Controller layer itself consists of two layers: Physical Layer (PHY) and Link Layer (LL). As the name suggests, Physical Layer mainly consists of the hardware components such as BLE radio and analog circuits. BLE utilizes unlicensed ISM band frequencies in range from 2.402 GHz to 2.480 GHz. The range is divided into 40 radio frequency (RF) channels. Devices use 37 of these channels, between which they can hop using Frequency-Hopping Spread Spectrum (FHSS), for transmitting data packets. Three other channels indexed at 37, 38, 39 are reserved for initial advertisement and connection establishment. [1, 3]

Link Layer interfaces with the PHY and handles low-level tasks such as device role management (advertising, scanning, creating and managing connections). LL is also responsible for time-critical operations, such as CRC generation and verification and PDU encryption and decryption. [1, 3, 7]

2.2.2 Host-Controller Interface

Host-Controller Interface (HCI) is a hidden layer that allows for the communication between a host and a controller. The implementation of the HCI may differ depending on the use case. In large devices controller part may be implemented in a standalone module and host part is running on the main CPU, whereas in smaller devices host and controller are often integrated into the main CPU. [1, 3]

2.2.3 Host Layer

The host layer consists of several protocols. The Logic Link Control and Adaptation Protocol (L2CAP) plays important role in BLE protocol stack. The main function of L2CAP is multiplexing – delivering packets from the LL to the correct upper layer protocol. L2CAP is also in charge of fragmentation and reassembly of transport frames – breaking down a large packet of information from upper layers into manageable BLE-sized chunks and combining smaller BLE packets into a single bigger packet to send to upper layers. Typically, the size of a chunk is 27 bytes. [1, 3, 7]

Security Manager Protocol (SMP) handles Bluetooth security. It is responsible for the pairing process, generating and exchanging security keys to establish an encrypted communication channel, as well as signing. [2, 3]

The Attribute Protocol (ATT) is a stateless client-server protocol. It is used by Generic Attribute Profile (GATT) to store device attributes, i.e. services, characteristics, in a lookup table. Each stored attribute has its own handle, type using 16-bit universally unique identifiers (UUID) that are defined either by Bluetooth SIG or the manufacturer of the device, a value and a set of permissions. GATT protocol organizes device attributes into services as shown in Table 1. [1, 3, 7]

Table 1: Examples of GATT attributes

Handle	Type (UUID)	Value	Permissions
0x0001	Device Name (0x2900)	Smart Lock	Read
0x0002	0x2A37	Heart Rate Measurement	Notify
0x0003	0x2A25	Serial Number String	Read

The Generic Access Profile (GAP) is the control layer responsible for describing profile roles such as central, peripheral, broadcaster and observer as well as managing device access methods and procedures based on these roles. [1, 3]

2.3 State of the art in BLE security

Bluetooth LE security is determined by devices' selected security mode, security level and pairing method. In this section these features will be described.

2.3.1 Security modes

Bluetooth Low Energy devices can potentially be used in a wide range of different scenarios, each with its own security requirements. In order to achieve this level of flexibility, each BLE device can decide its security requirements through the use of Security Modes. BLE devices support 3 Security Modes, Security Mode 1, Security Mode 2 and Security Mode 3 (introduced in Bluetooth version 5.2). Security Modes are further subdivided into Security Levels. Each Security Level is a combination of security attributes and requirements. [1, 2, 3]

Security Mode 1 is usually used for point-to-point communications. Two other security modes were developed to meet specific use case scenarios. BLE Security Mode 2 is used if the transmission of data should be fast and efficient. To achieve that, confidentiality is sacrificed. Security Mode 3 was introduced in the latest version of the Bluetooth standard, 5.2 for securing "LE Isochronous Channels". [1, 3]

2.3.2 Security Levels

Security Levels mainly define the type of authentication and encryption used in communication. Currently there are 4 Security Levels available, which are No Security, Authenticated pairing, Encryption and Data Signing. The only Security Mode + Security Level pairings that provide no security are Security Mode 1 + Security Level 1 and Security Mode 3 + Security Level 1. [1, 3]

2.3.3 Pairing

The pairing procedure is the base feature of BLE security that establishes and distributes keys that are later used for encrypting connections between the two devices. Bluetooth version 4.2 introduced new pairing method LE Secure Connection (LE-SC) in addition to already existing LE Legacy Pairing. Each pairing method contains different association models that define how pairing and authentication is done. Legacy Pairing models include Just Works, Passkey Entry and Out-Of-Band (OOB). LE-SC has all the same pairing models and Numeric Comparison in addition. [1, 8]

LE Legacy Pairing can be divided into three phases. Phase 1 begins with an exchange of Pairing Requests and Pairing Responses between devices that is done by SMP. Based on both devices' capabilities and flags set in Pairing Requests and Responses, association model that will be used for pairing is decided. Association models in LE Legacy Pairing include Just Works, Passkey Entry, and OOB. [1, 9]

Just Works association model requires no user interaction and is used when either one of the devices or both of them lack IO capabilities. In Passkey Entry model, a user is required to enter a 6 digit code displayed on one device into the other device. OOB model uses different means of exchanging data, e.g. through the use of NFC or QR-codes. [1, 9]

In Phase 2, the goal is to generate Short Term Key (STK) to encrypt the key exchange carried out in Phase 3. STK is generated based on Temporary Key (TK), obtained through the association model. In case of Just Works, TK is set to 0. If Passkey Entry association model is used, the TK is the 6-digit number exchanged between devices padded with 0 to reach 128-bit length. [1, 9]

The purpose of Phase 3 is to generate and exchange Long Term Key (LTK) and, if required by chosen Security Model and Security Level, other keys, such as Identity Resolving Key (IRK) used for generating and managing private addresses and Connection Signature Resolving Key (CSRK) needed for signing data packets. [1, 9]

LE Secure Connection follows similar pairing steps, but with a few important changes which make it more secure. Firstly, in Phase 1 of this pairing method, a new association model called Numeric Comparison becomes available. In this association model, a random 6-digit number is displayed on both devices and the user is required to confirm that the number is the same on both devices. Phase 2 of LE Secure Connection utilizes P-256 elliptic curve for LTK generation that does not rely on the 6-digit number entered earlier. Phase 3 follows similar process as used in LE Legacy Pairing. [3, 8, 9]

2.3.4 Session encryption

Paired devices have exchanged their encryption key (STK or LTK). This key is then used to encrypt communication sessions between devices. To do that, a random 128-bit session key diversifier (SKD) is generated on both devices. Then, SKD is encrypted with the encryption key. Lastly, a three-way handshake is performed and AES-CCM encrypted BLE connection is formed. [3]

2.4 Common attacks on BLE

Bluetooth LE security is susceptible to many different kinds of attacks. In this section, most common attacks against BLE devices will be described.

2.4.1 Jamming

Jamming attacks disrupt the communication between victim devices by emitting radio noise on frequencies used by Bluetooth devices. There are four types of jamming attacks that can be launched to disrupt communication channels between IoT devices. [1, 3]

The first type is the constant jamming attack which emits a continuous jamming signal. This type of attack is effective but highly energy inefficient and easy to detect. [1, 3]

The second type is the deceptive jamming attack which sends jamming signals at periodic intervals. This type of attack is energy efficient because it intermittently disrupts communication channels, making it harder to detect than the constant jamming attack. [1, 3]

The third type is the random jamming attack which combines both constant and deceptive jamming methods randomly, making it difficult to detect. This attack has an average energy efficiency. [1, 3]

The fourth type is the reactive jamming attack which is triggered when the jammer senses any network activity over the target channel. This attack is very specific, making it harder to detect, and generally more energy efficient than the other types of jamming attacks. [1, 3]

2.4.2 Man-In-The-Middle attacks

Man-In-The-Middle (MITM) attack if a form of eavesdropping that occurs when an attacker poses himself between two communicating parties and listens in to their conversation. Bluetooth devices, especially the ones using the Just Works association model, are susceptible to this kind of attack. [2, 3, 9]

Default MITM scenario, where an attacker inserts one device between two communicating parties is not possible in BLE, since one Bluetooth device is not able to be connected to two different groups of devices at the same time. To perform MITM attack on a Bluetooth connection, an attacker should insert two malicious devices, each facing one of the victim devices. Malicious devices should communicate with each other using different means, for example Websockets. [1, 9]

2.4.3 Battery exhaustion

IoT devices are usually designed to be energy efficient. Adversaries can use battery exhaustion as an exploit to disrupt services provided by systems that rely on the coordination of multiple devices, which may be located in inaccessible areas. BLE-enabled IoT devices are designed to conserve energy by operating in short bursts. However, if an attacker continuously bombards the target device with requests that prevent it from entering a sleep state, the device's lifespan will be significantly reduced due to its persistent awake state. [1]

2.4.4 KNOB attack

KNOB stands for Key Negotiation Of Bluetooth, and is also known as Key Negotiation Downgrade. During the attack, devices are forced to renegotiate their long-term key and session key in the link layer with the lowest entropy specified by the standard. The BLE specification does not require integrity protection during the feature exchange phase, which includes key negotiation. This allows the BLE encryption key entropy to be reduced to the minimum of 7 bytes, regardless of using authenticated secure connections. An attacker can then perform brute force attacks on the weak keys, resulting in the decryption of communication between two devices, channel jamming, and crafting valid decrypted Bluetooth packets. These threats can be disruptive in IoT applications that rely on BLE. Downgrading also enables MitM attacks in Android devices, and other major commercial operating systems seem to lack proper support for Secure Connection mode. [1, 10]

2.4.5 Btlejack

The Btlejacking attack targets BLE versions 4.0 and later and exploits the BLE supervision timeout in the CONNECT_REQ PDU to detect and intercept a connection. This timeout defines the time after which a connection is considered lost if no valid packets are received and is used by both the central and peripheral devices. The attacker intercepts the connection, jams the communication to interrupt it, and then hijacks the aborted connection to connect to the peripheral device and gain access to its data. This attack compromises the confidentiality and integrity of the device data. During the connection setup, the central device sets a supervision timeout value, and if a device does not receive any packet for that amount of time, the connection is considered lost. By jamming packets sent by the peripheral to the central device, the attacker can trigger the timeout in the central device and then take over the connection as the central device. [1, 2]

3 Methodology

This section is divided into three parts. In the first part, methods used by an attacker to gather information and identify vulnerabilities in target devices will be presented. The second part of this section concerns limitations that author has encountered during his work. The third part of this section will focus on ethical considerations regarding conducting experiments in wireless settings.

3.1 Methods for identifying vulnerabilities in BLE devices

First method for identifying vulnerabilities in BLE devices is scanning devices and networks. This method involves scanning for nearby BLE devices and analyzing their advertising packets to gather information about devices as well as indication of any possible vulnerabilities. This step can be performed using many different tools, ranging from mobile applications such as nRF Connect for Mobile, to different Linux tools like bettercap or hcitool, to capturing packets with BLE Sniffers running on SoC's like Ubertooth or nRF52 and analyzing captured packets in Wireshark. [11, 12]

Advertising packets are used by BLE devices to broadcast their presence to nearby devices, and can contain information such as the device name, Bluetooth address, supported services, and manufacturer data. Analysis of this data may lead to discovering vulnerabilities in the device. For example, after obtaining device name or model and looking for it in the Common Vulnerabilities and Exposures (CVE) database, an attacker can identify known vulnerabilities in specific BLE devices or software stacks. [11]

Sniffing BLE traffic involves intercepting and analyzing the data exchanged during BLE connections. One commonly used approach is to employ BLE sniffers, such as the nRF52 or Ubertooth, capable of capturing BLE packets at the Link Layer. Tools like Wireshark can then be utilized to dissect and interpret the captured data. [13]

During sniffing, researchers can analyze the exchanged packets to identify communication protocols, discover implemented services, and assess the security of data transmission.

Spoofing BLE advertising packets involves manipulating the information broadcasted by BLE devices to potentially deceive nearby devices. By crafting and transmitting custom advertising packets, it becomes possible to simulate the presence of specific BLE devices or inject misleading data into the environment. This method aids in assessing how devices respond to unexpected or malicious advertising data.

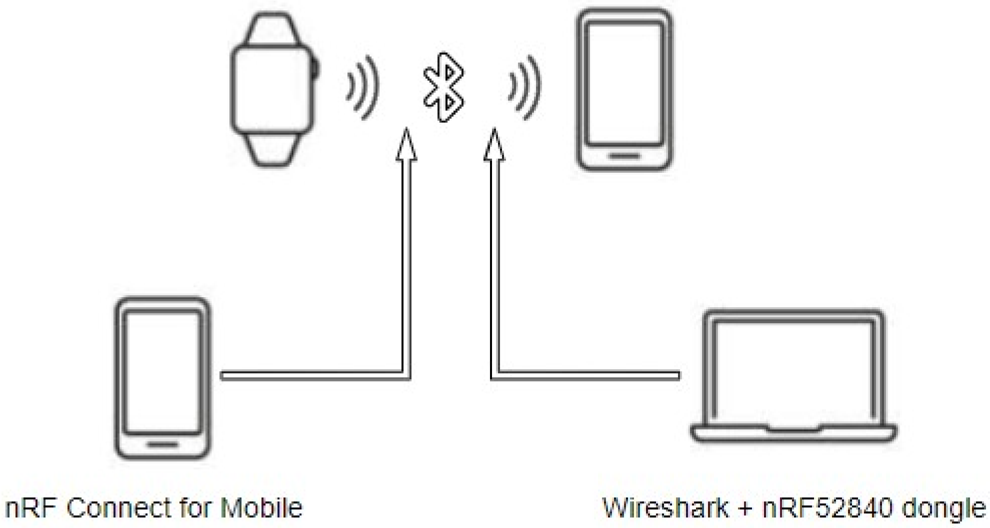


Figure 2: Experimental setup

3.2 Scope and limitations

Most of the attacks on BLE require use of expensive hardware. This work will focus on performing steps that do not rely on the use of expensive or specialized hardware, making the research findings more applicable to a wider range of users, researchers, and enthusiasts interested in BLE security.

3.3 Ethical considerations

The author made sure to only record data that is regarded as public because over-the-air data sniffing is involved, such as advertisement data. Only devices owned by the author were used.

4 Experimental setup

In this section, actual conducted experiments as well as results obtained will be described. The author will first go over the hardware used for experiments, their capabilities and reasoning behind the choice of hardware. After that, tools that were used to perform experiments will be described. Finally, the steps taken to perform vulnerability assessment will be described.

4.1 Bluetooth adapter setup

Bluetooth adapters installed from factory are designed with regular users in mind and, therefore, lack capabilities necessary to perform Bluetooth traffic sniffing and inspection. For that reason, additional hardware must be used to work with BLE traffic. There are several options available on the market, such as nRF52840, Ubertooth One, or Texas Instruments' microcontrollers.

Table 2: Available BLE sniffers

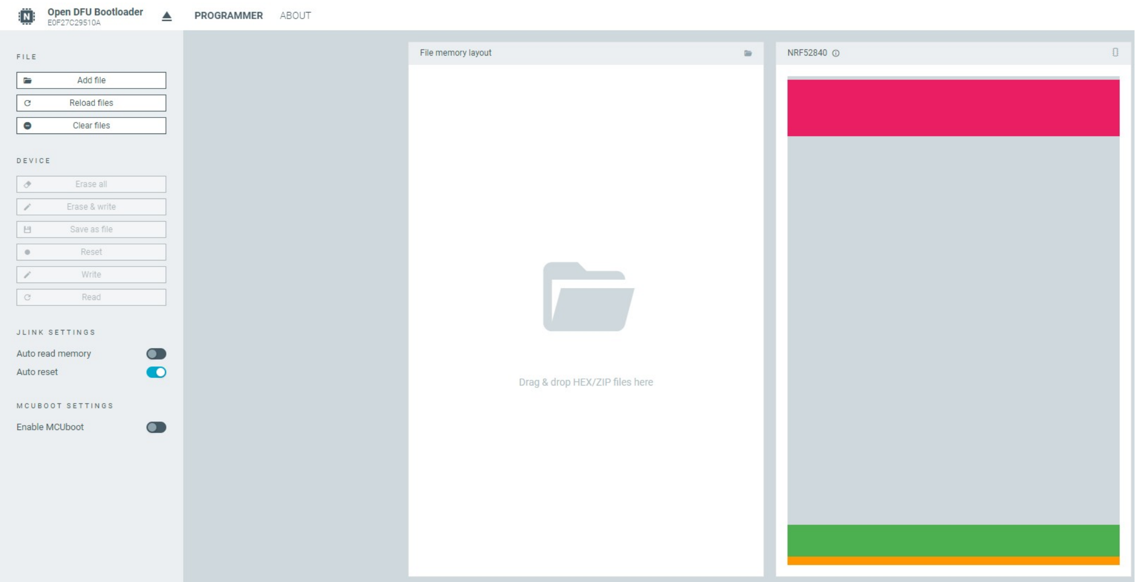
	NRF52840 DK	Ubertooth One	Texas Instruments' CC2540
Cost	10 €	120-200 €	40 €
Open Source	No	Yes	No
Ability to tap into existing connections	No	Yes	No

Available options differ in terms of price, capabilities, type of platform (open-source or closed source). After comparing available options, nRF52840 DK dongle was chosen.

Figure 3: nRF52840 DK

4.1.1 nRF52840 DK setup

nRF52840 DK is a single-board development kit built on the nRF52840 SoC. This dongle is capable of working with Bluetooth Low Energy among other protocols working on 2.4 GHz frequency. It works together with the companion application nRF Connect for Desktop which lets the user flash the dongle with different applications. In this example, we will be flashing the dongle with BLE Sniffer software. In order to do that, we first have to install the nRF Connect for Desktop application. After completing the installation process we insert our dongle into the USB port of the notebook and make sure that the application recognizes the connected dongle. [14, 15]



Auto read memory ○
Auto reset ●
Enable MCUboot ○

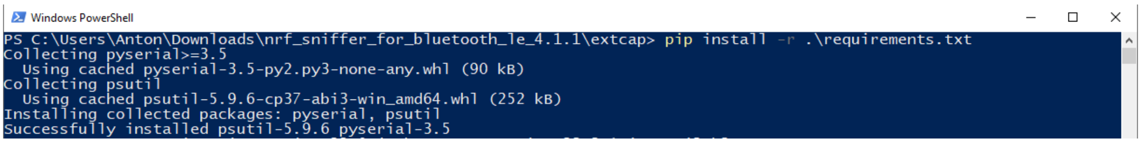
Figure 4: nRF52840 Programmer overview

Then we download BLE packet sniffer tool available at the official Nordic Semiconductor website [16], and flash the dongle with the right image.

4.1.2 Wireshark setup

While Wireshark does not support working with BLE packets out of the box, Nordic Semiconductor provides resources to enable Wireshark to work with BLE. Paired with nRF52840 dongle, it lets the user to capture BLE packets and inspect their contents. [12]

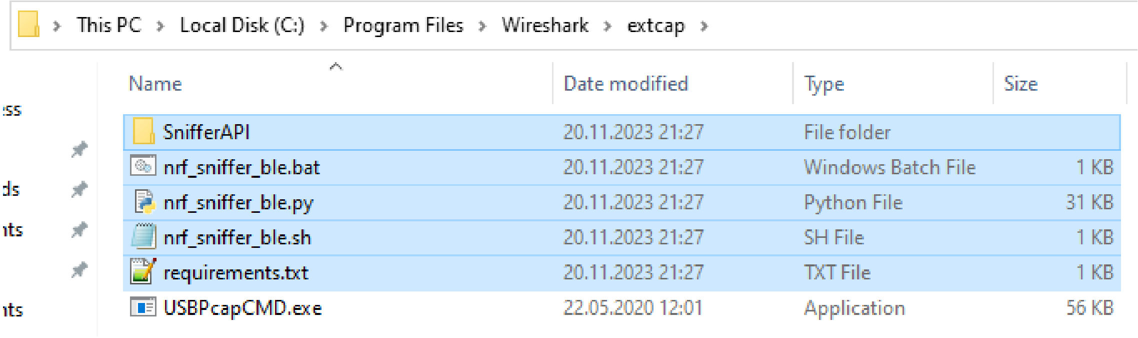
The nRF Sniffer for BLE is installed as an external capture plugin in Wireshark. Everything that we need is contained in the archive that we downloaded in the previous step. First, we have to install the Python dependencies required to run the software.



```
PS C:\Users\Anton\Downloads\nrf_sniffer_for_bluetooth_le_4.1.1\extcap> pip install -r .\requirements.txt
Collecting pyserial>=3.5
  Using cached pyserial-3.5-py2.py3-none-any.whl (90 kB)
Collecting psutil
  Using cached psutil-5.9.6-cp37-abi3-win_amd64.whl (252 kB)
Installing collected packages: pyserial, psutil
Successfully installed psutil-5.9.6 pyserial-3.5
```

Figure 5: Installing Python dependencies

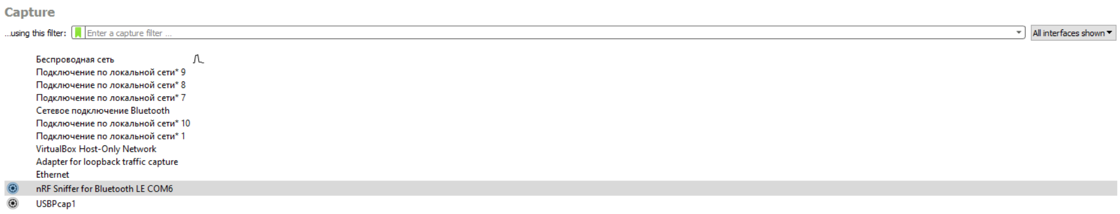
After installing dependencies, we have to copy the capture tool into the Wireshark. The tool is also contained in the archive, so we copy it and paste the contents into Wireshark's extcap folder.


This PC › Local Disk (C:) › Program Files › Wireshark › extcap ›
ss
ds
ts
ts

Name	Date modified	Type	Size
SnifferAPI	20.11.2023 21:27	File folder	
nrf_sniffer_ble.bat	20.11.2023 21:27	Windows Batch File	1 KB
nrf_sniffer_ble.py	20.11.2023 21:27	Python File	31 KB
nrf_sniffer_ble.sh	20.11.2023 21:27	SH File	1 KB
requirements.txt	20.11.2023 21:27	TXT File	1 KB
USBPcapCMD.exe	22.05.2020 12:01	Application	56 KB

Figure 6: Adding BLE capture tool to Wireshark

After performing these steps, the nRF Sniffer should be available in the Wireshark interface selection.



Беспроводная сеть
Подключение по локальной сети* 9
Подключение по локальной сети* 8
Подключение по локальной сети* 7
Сетевое подключение Bluetooth
Подключение по локальной сети* 10
Подключение по локальной сети* 1
VirtualBox Host-Only Network
Adapter for loopback traffic capture
Ethernet
nRF Sniffer for Bluetooth LE COM6
USBPcap1

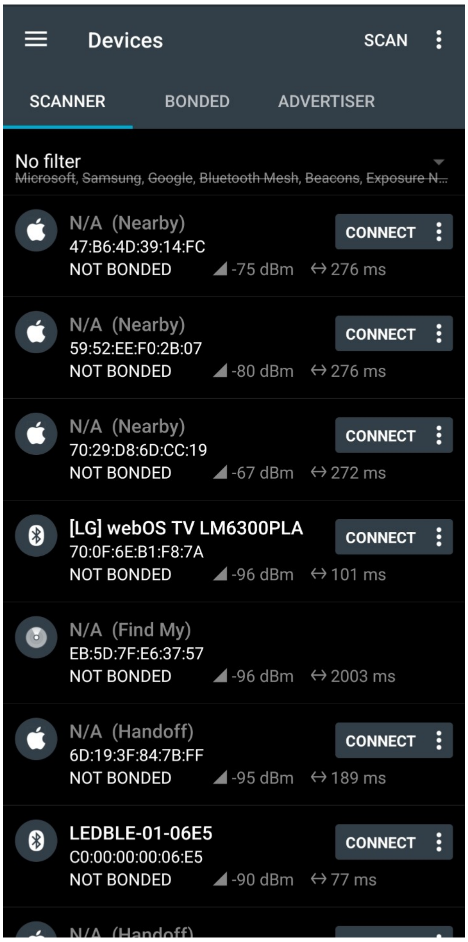
Figure 7: Wireshark interface after successful setup of the BLE tool

4.2 Vulnerability assessment

To perform vulnerability assessment, different tools can be used. In this sectin of the work, the detailed description of the process of vulnerability assessment using nRF Connect for Mobile and nRF52840 dongle with Wireshark will be described.

4.2.1 nRF Connect for Mobile

nRF Connect for Mobile can serve as a well-rounded BLE packet inspection tool. This mobile application lets the user inspect, analyze and interact with BLE devices. NRF Connect for Mobile provides means to establish connections, read and write characteristics and navigate through BLE services. Furthermore, it enables users to copy advertising packets or create their own, send them out and connect to other devices.



Devices SCAN

SCANNER BONDED ADVERTISER

No filter
~~Microsoft, Samsung, Google, Bluetooth Mesh, Beacons, Exposure N...~~

N/A (Nearby) CONNECT
47:B6:4D:39:14:FC
NOT BONDED -75 dBm ↔ 276 ms

N/A (Nearby) CONNECT
59:52:EE:F0:2B:07
NOT BONDED -80 dBm ↔ 276 ms

N/A (Nearby) CONNECT
70:29:D8:6D:CC:19
NOT BONDED -67 dBm ↔ 272 ms

[LG] webOS TV LM6300PLA CONNECT
70:0F:6E:B1:F8:7A
NOT BONDED -96 dBm ↔ 101 ms

N/A (Find My)
EB:5D:7F:E6:37:57
NOT BONDED -96 dBm ↔ 2003 ms

N/A (Handoff) CONNECT
6D:19:3F:84:7B:FF
NOT BONDED -95 dBm ↔ 189 ms

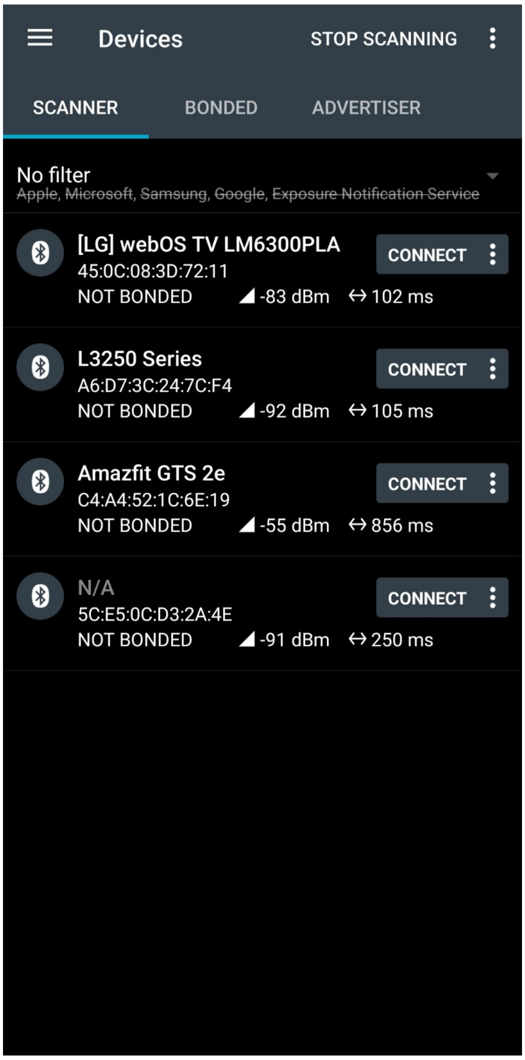
LEDBLE-01-06E5 CONNECT
C0:00:00:00:06:E5
NOT BONDED -90 dBm ↔ 77 ms

N/A (Handoff)

Figure 8: Nearby devices' BLE advertising packets

In the scanner tab shown in Figure 8 we can see different devices sending advertising packets around. As stated in the section 3.2, the author will only work with owned devices.

For this example, we will inspect advertising packets of a Amazfit GTS 2e smartwatch. To capture advertising packets of the watch, it was disconnected from the phone and a factory reset was performed. Then, the watch and the scanner in the nRF Connect application were turned on.



No filter
~~Apple, Microsoft, Samsung, Google, Exposure Notification Service~~

[LG] webOS TV LM6300PLA
45:0C:08:3D:72:11
NOT BONDED -83 dBm ↔ 102 ms

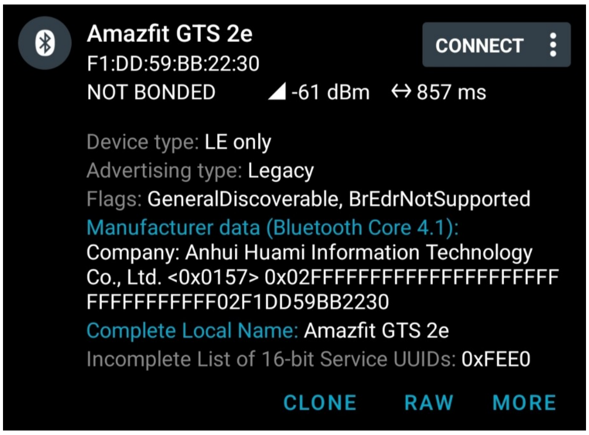
L3250 Series
A6:D7:3C:24:7C:F4
NOT BONDED -92 dBm ↔ 105 ms

Amazfit GTS 2e
C4:A4:52:1C:6E:19
NOT BONDED -55 dBm ↔ 856 ms

N/A
5C:E5:0C:D3:2A:4E
NOT BONDED -91 dBm ↔ 250 ms

Figure 9: Amazfit GTS 2e advertising packet captured

After filtering captured packets, it becomes evident that the target advertising packet was captured (Figure 9). When we tap the packet, we can analyze its contents.

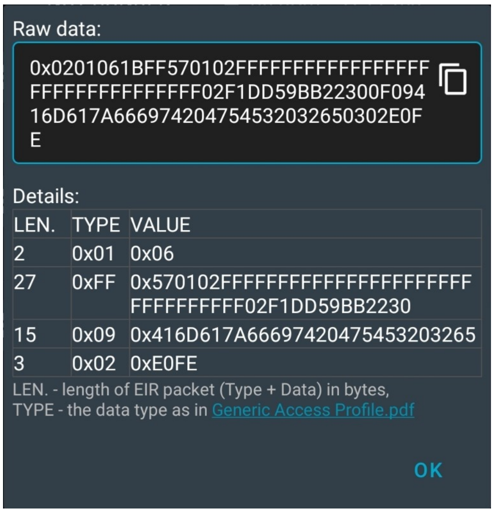


Amazfit GTS 2e
F1:DD:59:BB:22:30
NOT BONDED ◢ -61 dBm ↔ 857 ms

Device type: LE only
Advertising type: Legacy
Flags: GeneralDiscoverable, BrEdrNotSupported
Manufacturer data (Bluetooth Core 4.1):
Company: Anhui Huami Information Technology Co., Ltd. <0x0157> 0x02FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF02F1DD59BB2230
Complete Local Name: Amazfit GTS 2e
Incomplete List of 16-bit Service UUIDs: 0xFEE0

Figure 10: Advertising packet contents overview

Figure 10 indicates that this is an LE only device with Legacy advertising type. Advertising packet has GeneralDiscoverable and BrEdrNotSupported flags set that means that the device is actively broadcasting advertising packets to inform nearby devices of its presence and availability. The packet also contains manufacturer company name and device complete local name. By clicking on the Raw button, advertising packet raw data can be extracted for further analysis [17] or manipulation (Figure 11).



Raw data:

```
0x0201061BFF570102FFFFFFFFFFFFFFFFF
FFFFFFFFFFFFFFF02F1DD59BB22300F094
16D617A666974204754532032650302E0F
E
```

Details:

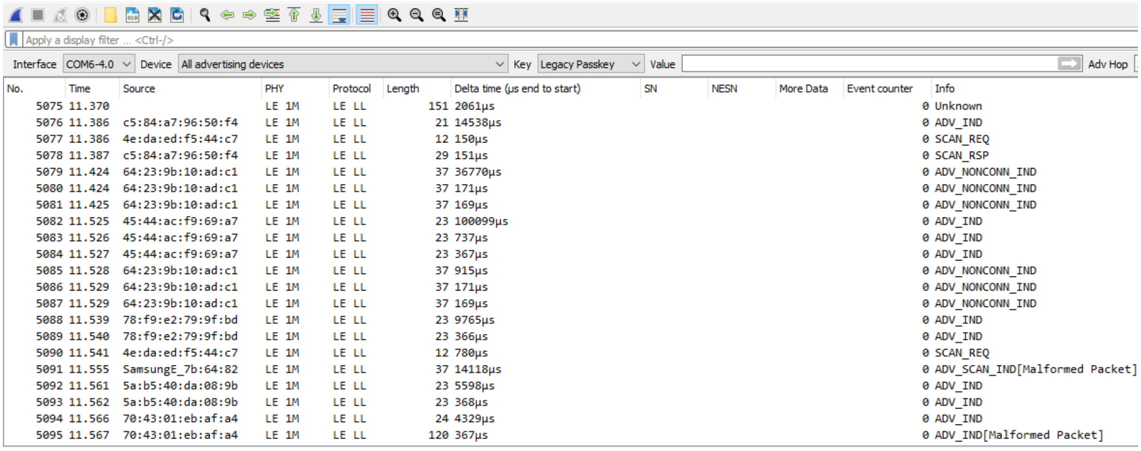
LEN.	TYPE	VALUE
2	0x01	0x06
27	0xFF	0x570102FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF02F1DD59BB2230
15	0x09	0x416D617A66697420475453203265
3	0x02	0xE0FE

LEN. - length of EIR packet (Type + Data) in bytes,
TYPE - the data type as in Generic Access Profile.pdf

Figure 11: Advertising packet's raw data

4.2.2 Wireshark

The second tool that was used to analyze BLE advertising packets is Wireshark. The setup process is described in section 4.1.2. An example of BLE traffic capture can be seen in Figure 12.



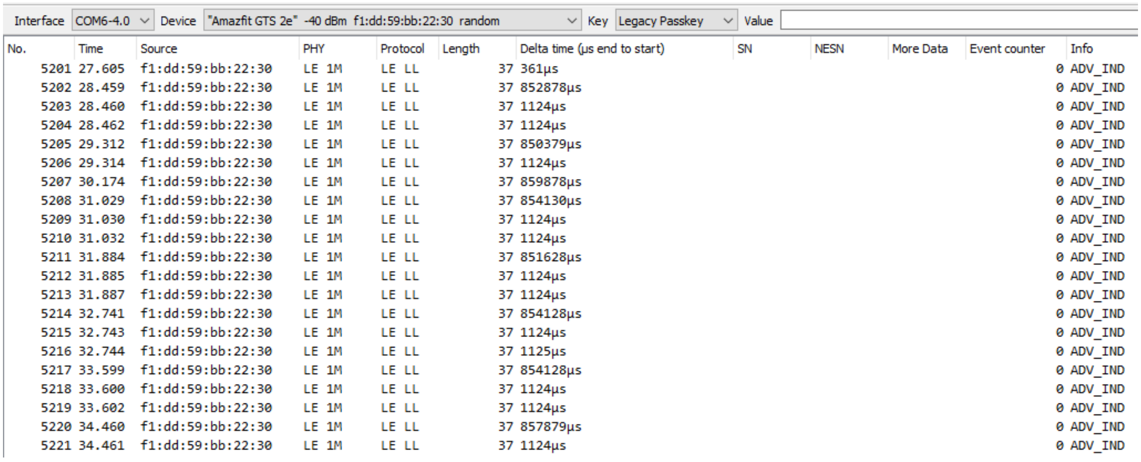
Apply a display filter ... <Ctrl-/>

Interface COM6-4.0 Device All advertising devices Key Legacy Passkey Value Adv Hop

No.	Time	Source	PHY	Protocol	Length	Delta time (µs end to start)	SN	NESN	More Data	Event counter	Info
5075	11.370		LE 1M	LE LL	151	2061µs				0	Unknown
5076	11.386	c5:84:a7:96:50:f4	LE 1M	LE LL	21	14538µs				0	ADV_IND
5077	11.386	4e:da:ed:f5:44:c7	LE 1M	LE LL	12	150µs				0	SCAN_REQ
5078	11.387	c5:84:a7:96:50:f4	LE 1M	LE LL	29	151µs				0	SCAN_RSP
5079	11.424	64:23:9b:10:ad:c1	LE 1M	LE LL	37	36770µs				0	ADV_NONCONN_IND
5080	11.424	64:23:9b:10:ad:c1	LE 1M	LE LL	37	171µs				0	ADV_NONCONN_IND
5081	11.425	64:23:9b:10:ad:c1	LE 1M	LE LL	37	169µs				0	ADV_NONCONN_IND
5082	11.525	45:44:ac:f9:69:a7	LE 1M	LE LL	23	100099µs				0	ADV_IND
5083	11.526	45:44:ac:f9:69:a7	LE 1M	LE LL	23	737µs				0	ADV_IND
5084	11.527	45:44:ac:f9:69:a7	LE 1M	LE LL	23	367µs				0	ADV_IND
5085	11.528	64:23:9b:10:ad:c1	LE 1M	LE LL	37	915µs				0	ADV_NONCONN_IND
5086	11.529	64:23:9b:10:ad:c1	LE 1M	LE LL	37	171µs				0	ADV_NONCONN_IND
5087	11.529	64:23:9b:10:ad:c1	LE 1M	LE LL	37	169µs				0	ADV_NONCONN_IND
5088	11.539	78:f9:e2:79:9f:bd	LE 1M	LE LL	23	9765µs				0	ADV_IND
5089	11.540	78:f9:e2:79:9f:bd	LE 1M	LE LL	23	366µs				0	ADV_IND
5090	11.541	4e:da:ed:f5:44:c7	LE 1M	LE LL	12	780µs				0	SCAN_REQ
5091	11.555	SamsungE_7b:64:82	LE 1M	LE LL	37	14118µs				0	ADV_SCAN_IND[Malformed Packet]
5092	11.561	5a:b5:40:da:08:9b	LE 1M	LE LL	23	5598µs				0	ADV_IND
5093	11.562	5a:b5:40:da:08:9b	LE 1M	LE LL	23	368µs				0	ADV_IND
5094	11.566	70:43:01:eb:af:a4	LE 1M	LE LL	24	4329µs				0	ADV_IND
5095	11.567	70:43:01:eb:af:a4	LE 1M	LE LL	120	367µs				0	ADV_IND[Malformed Packet]

Figure 12: BLE traffic captured with nRF52840

Wireshark nRF Sniffer Profile enables users to filter BLE traffic. In this example, the traffic is filtered so that only advertising packets from the Amazfit GTS 2e device is shown. As we can see in Figure 13, until the device is connected, it broadcasts advertising packets using Link Layer (LL) protocol.

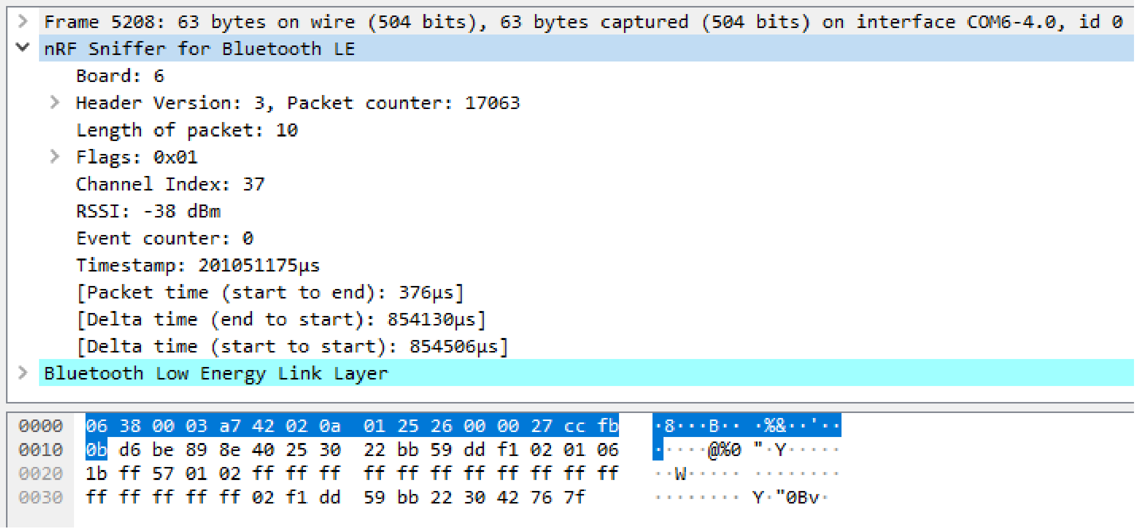


Interface COM6-4.0 Device "Amazfit GTS 2e" -40 dBm f1:dd:59:bb:22:30 random Key Legacy Passkey Value

No.	Time	Source	PHY	Protocol	Length	Delta time (µs end to start)	SN	NESN	More Data	Event counter	Info
5201	27.605	f1:dd:59:bb:22:30	LE 1M	LE LL	37	361µs				0	ADV_IND
5202	28.459	f1:dd:59:bb:22:30	LE 1M	LE LL	37	852878µs				0	ADV_IND
5203	28.460	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5204	28.462	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5205	29.312	f1:dd:59:bb:22:30	LE 1M	LE LL	37	850379µs				0	ADV_IND
5206	29.314	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5207	30.174	f1:dd:59:bb:22:30	LE 1M	LE LL	37	859878µs				0	ADV_IND
5208	31.029	f1:dd:59:bb:22:30	LE 1M	LE LL	37	854130µs				0	ADV_IND
5209	31.030	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5210	31.032	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5211	31.884	f1:dd:59:bb:22:30	LE 1M	LE LL	37	851628µs				0	ADV_IND
5212	31.885	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5213	31.887	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5214	32.741	f1:dd:59:bb:22:30	LE 1M	LE LL	37	854128µs				0	ADV_IND
5215	32.743	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5216	32.744	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1125µs				0	ADV_IND
5217	33.599	f1:dd:59:bb:22:30	LE 1M	LE LL	37	854128µs				0	ADV_IND
5218	33.600	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5219	33.602	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND
5220	34.460	f1:dd:59:bb:22:30	LE 1M	LE LL	37	857879µs				0	ADV_IND
5221	34.461	f1:dd:59:bb:22:30	LE 1M	LE LL	37	1124µs				0	ADV_IND

Figure 13: Amazfit GTS 2e advertising packets

Clicking on an advertising packet lets the user to inspect it closely. It can be seen, that the frame is split into two main parts – nRF Sniffer for Bluetooth LE and Bluetooth Low Energy Link Layer. The first part contains metadata – information about the packet itself, such as the Received Signal Strength Indicator (RSSI), length of the packet, timestamp and index of the channel the packet was captured on (Figure 14).

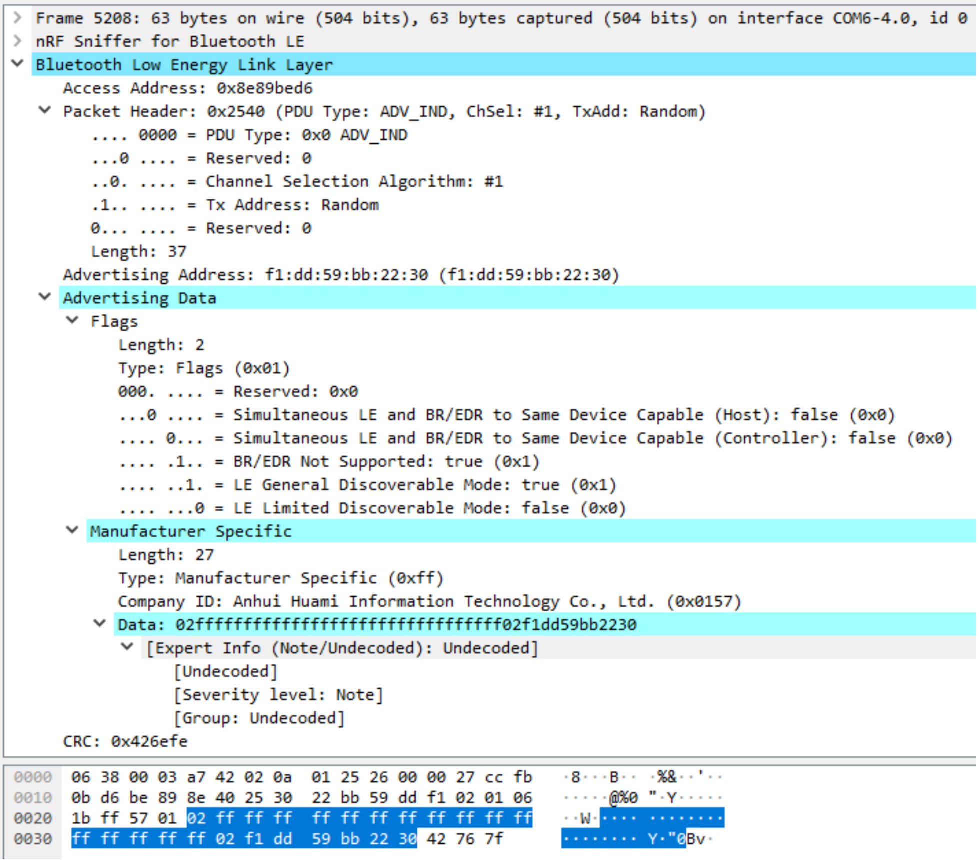


```
> Frame 5208: 63 bytes on wire (504 bits), 63 bytes captured (504 bits) on interface COM6-4.0, id 0
v nRF Sniffer for Bluetooth LE
     Board: 6
   > Header Version: 3, Packet counter: 17063
     Length of packet: 10
   > Flags: 0x01
     Channel Index: 37
     RSSI: -38 dBm
     Event counter: 0
     Timestamp: 201051175µs
     [Packet time (start to end): 376µs]
     [Delta time (end to start): 854130µs]
     [Delta time (start to start): 854506µs]
> Bluetooth Low Energy Link Layer

0000  06 38 00 03 a7 42 02 0a  01 25 26 00 00 27 cc fb   ·8···B·· ·%&··'··
0010  0b d6 be 89 8e 40 25 30  22 bb 59 dd f1 02 01 06   ·····@%0 "·Y·····
0020  1b ff 57 01 02 ff ff ff  ff ff ff ff ff ff ff ff   ··W····· ········
0030  ff ff ff ff ff 02 f1 dd  59 bb 22 30 42 76 7f      ········ Y·"0Bv·
```

Figure 14: First part of the captured advertising packet

The second part of the frame displayed in Figure 15 contains the actual advertising data. By analyzing the packet information we can confirm findings from nRF Connect for Mobile. The packet indeed has the BR/EDR Not Supported and LE General Discoverable Mode flags set and the manufacturer is Anhui Huami Information Technology Co., Ltd. With UUID 0x0157.



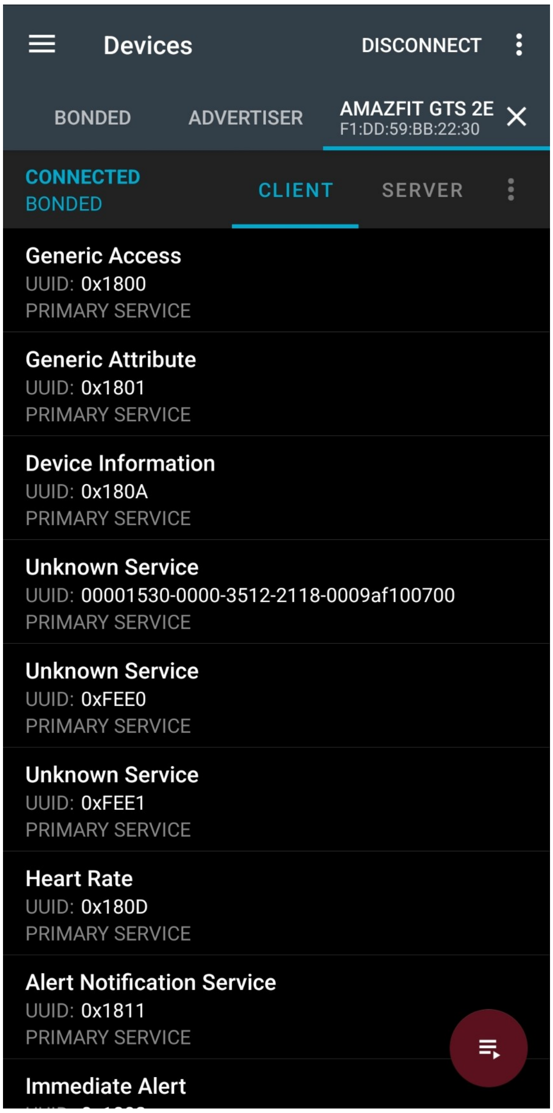
```
> Frame 5208: 63 bytes on wire (504 bits), 63 bytes captured (504 bits) on interface COM6-4.0, id 0
> nRF Sniffer for Bluetooth LE
v Bluetooth Low Energy Link Layer
     Access Address: 0x8e89bed6
   v Packet Header: 0x2540 (PDU Type: ADV_IND, ChSel: #1, TxAdd: Random)
        .... 0000 = PDU Type: 0x0 ADV_IND
        ...0 .... = Reserved: 0
        ..0. .... = Channel Selection Algorithm: #1
        .1.. .... = Tx Address: Random
        0... .... = Reserved: 0
        Length: 37
     Advertising Address: f1:dd:59:bb:22:30 (f1:dd:59:bb:22:30)
   v Advertising Data
      v Flags
           Length: 2
           Type: Flags (0x01)
           000. .... = Reserved: 0x0
           ...0 .... = Simultaneous LE and BR/EDR to Same Device Capable (Host): false (0x0)
           .... 0... = Simultaneous LE and BR/EDR to Same Device Capable (Controller): false (0x0)
           .... .1.. = BR/EDR Not Supported: true (0x1)
           .... ..1. = LE General Discoverable Mode: true (0x1)
           .... ...0 = LE Limited Discoverable Mode: false (0x0)
      v Manufacturer Specific
           Length: 27
           Type: Manufacturer Specific (0xff)
           Company ID: Anhui Huami Information Technology Co., Ltd. (0x0157)
         v Data: 02ffffffffffffffffffffffffffffffff02f1dd59bb2230
            v [Expert Info (Note/Undecoded): Undecoded]
                 [Undecoded]
                 [Severity level: Note]
                 [Group: Undecoded]
     CRC: 0x426efe

0000  06 38 00 03 a7 42 02 0a  01 25 26 00 00 27 cc fb   ·8···B·· ·%&··'··
0010  0b d6 be 89 8e 40 25 30  22 bb 59 dd f1 02 01 06   ·····@%0 "·Y·····
0020  1b ff 57 01 02 ff ff ff  ff ff ff ff ff ff ff ff   ··W····· ········
0030  ff ff ff ff ff 02 f1 dd  59 bb 22 30 42 76 7f      ········ Y·"0Bv·
```

Figure 15: Second part of the captured advertising packet

4.3 Sniffing BLE conversation

By clicking Connect button in the nRF Connect for Mobile application, the author was able to successfully connect to the smartwatch. Connecting to the device allows us to inspect GATT services, their characteristics and descriptors (Figure 16).



Generic Access
UUID: 0x1800
PRIMARY SERVICE

Generic Attribute
UUID: 0x1801
PRIMARY SERVICE

Device Information
UUID: 0x180A
PRIMARY SERVICE

Unknown Service
UUID: 00001530-0000-3512-2118-0009af100700
PRIMARY SERVICE

Unknown Service
UUID: 0xFEE0
PRIMARY SERVICE

Unknown Service
UUID: 0xFEE1
PRIMARY SERVICE

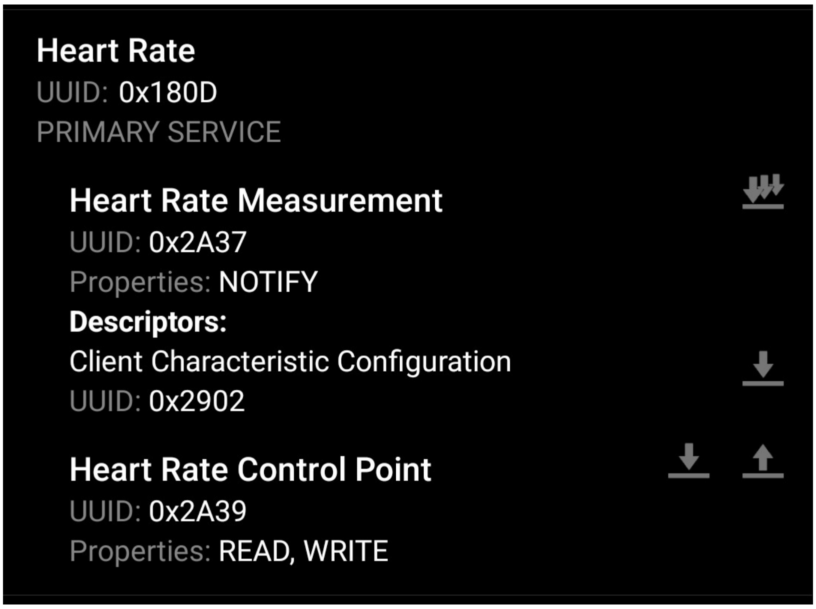
Heart Rate
UUID: 0x180D
PRIMARY SERVICE

Alert Notification Service
UUID: 0x1811
PRIMARY SERVICE

Immediate Alert

Figure 16: Amazfit GTS 2e GATT services

For the purposes of this study, the author decided to concentrate on the Heart Rate service (Figure 17).



Heart Rate
UUID: 0x180D
PRIMARY SERVICE

Heart Rate Measurement
UUID: 0x2A37
Properties: NOTIFY
Descriptors:
Client Characteristic Configuration
UUID: 0x2902

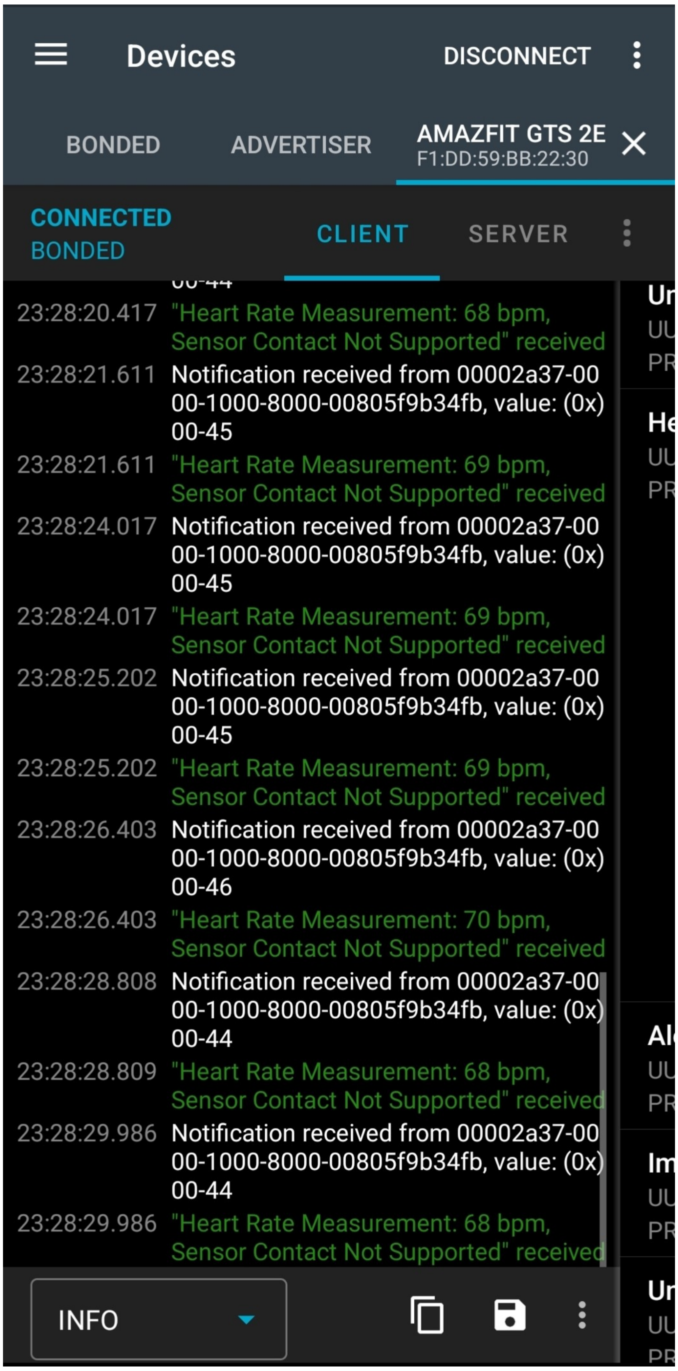
Heart Rate Control Point
UUID: 0x2A39
Properties: READ, WRITE

Figure 17: Amazfit GTS 2e Heart Rate service

Heart Rate service with UUID 0x180D contains two characteristics – Heart Rate Measurement and Heart Rate Control Point. As per Blueooth Heart Rate Profile documentation [18], Heart Rate Measurement is a characteristic designed to transmit real-time heart rate data from peripheral device to the central. The Heart Rate Measurement has "NOTIFY" property, indicating that the heart rate sensor can send notifications to the connected device whenever there is a change in heart rate data.

Heart Rate Control Point characteristic facilitates communication between a central device (such as a smartphone) and a heart rate sensor to control and configure certain aspects of heart rate monitoring. For example, if the Heart Rate Sensor has the ability to report the Energy Expanded field in the Heart Rate Measurement characteristic, the controller may use the Heart Control Point characteristic to reset the Energy Expanded Field to zero. [18]

The buttons on the right-side of the screen allow the application user to subscribe to notifications of the characteristic, write or read data. Here, author subscribed to Heart Rate Measurement characteristic's notifications. After that, the watch was connected to an official app and heart rate monitoring was launched. When the heart rate was measured, packets containing Heart Rate Measurement data were captured (Figure 18).



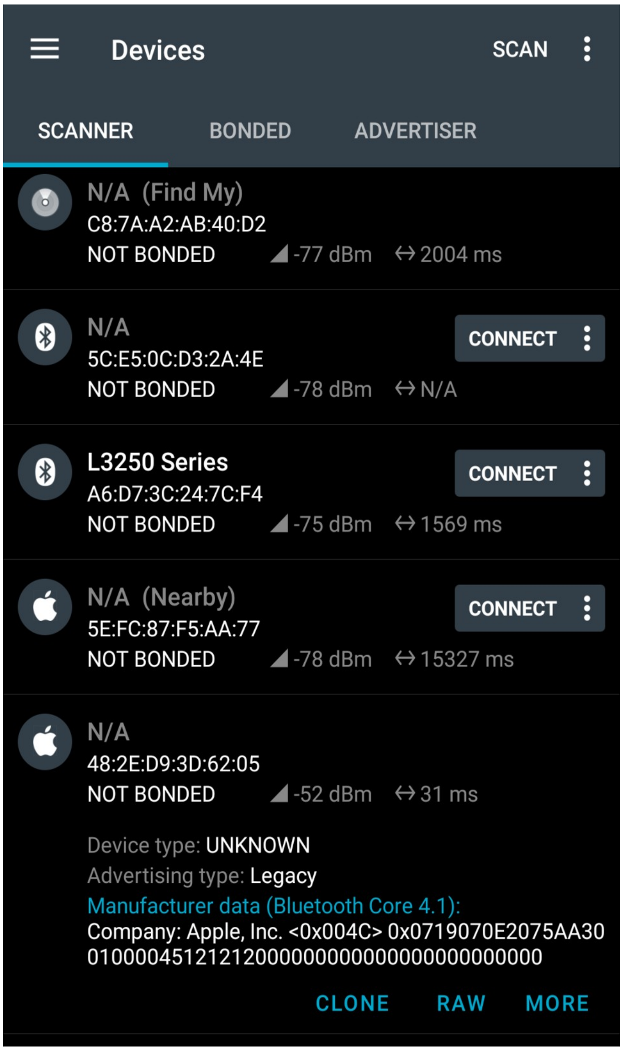
```
             00-44
23:28:20.417 "Heart Rate Measurement: 68 bpm,
             Sensor Contact Not Supported" received
23:28:21.611 Notification received from 00002a37-00
             00-1000-8000-00805f9b34fb, value: (0x)
             00-45
23:28:21.611 "Heart Rate Measurement: 69 bpm,
             Sensor Contact Not Supported" received
23:28:24.017 Notification received from 00002a37-00
             00-1000-8000-00805f9b34fb, value: (0x)
             00-45
23:28:24.017 "Heart Rate Measurement: 69 bpm,
             Sensor Contact Not Supported" received
23:28:25.202 Notification received from 00002a37-00
             00-1000-8000-00805f9b34fb, value: (0x)
             00-45
23:28:25.202 "Heart Rate Measurement: 69 bpm,
             Sensor Contact Not Supported" received
23:28:26.403 Notification received from 00002a37-00
             00-1000-8000-00805f9b34fb, value: (0x)
             00-46
23:28:26.403 "Heart Rate Measurement: 70 bpm,
             Sensor Contact Not Supported" received
23:28:28.808 Notification received from 00002a37-00
             00-1000-8000-00805f9b34fb, value: (0x)
             00-44
23:28:28.809 "Heart Rate Measurement: 68 bpm,
             Sensor Contact Not Supported" received
23:28:29.986 Notification received from 00002a37-00
             00-1000-8000-00805f9b34fb, value: (0x)
             00-44
23:28:29.986 "Heart Rate Measurement: 68 bpm,
             Sensor Contact Not Supported" received
```

Figure 18: Capturing Heart Rate measurements

4.4 Spoofing BLE advertising packets

Capabilities of nRF Connect for Mobile are not restricted with only analyzing captured packets. The application also enables users to clone captured advertising packets, modify them or create their own and then broadcast them. In this section the process of spoofing BLE advertising packets will be shown. Specifically, we will be spoofing Apple Airpods Pro advertising packets.

Before heading to the application, Apple Airpods Pro were disconnected and unpaired from the phone. Then, the right advertising packet was found in the nRF app (Figure 19).



N/A (Find My)
C8:7A:A2:AB:40:D2
NOT BONDED -77 dBm ↔ 2004 ms

N/A
5C:E5:0C:D3:2A:4E
NOT BONDED -78 dBm ↔ N/A

L3250 Series
A6:D7:3C:24:7C:F4
NOT BONDED -75 dBm ↔ 1569 ms

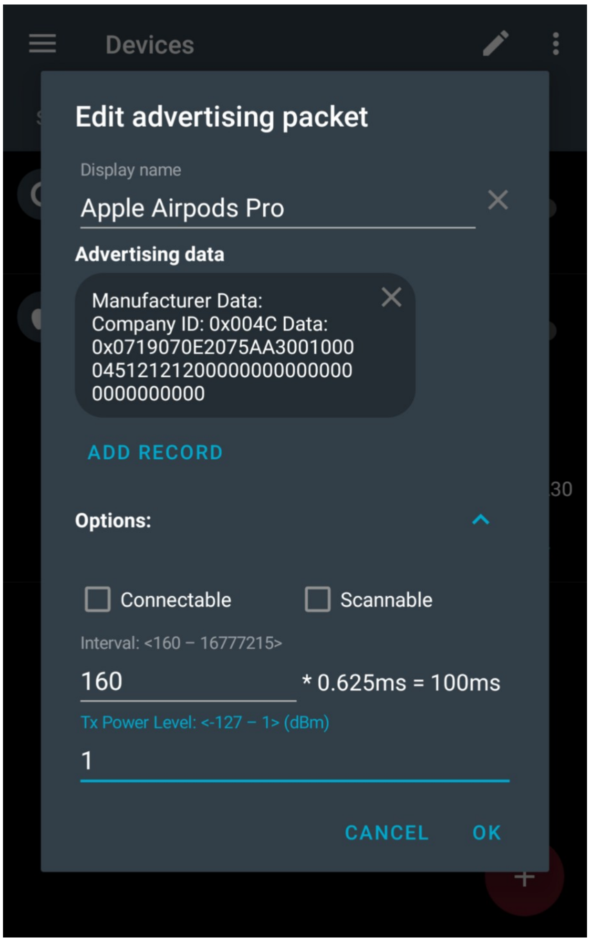
N/A (Nearby)
5E:FC:87:F5:AA:77
NOT BONDED -78 dBm ↔ 15327 ms

N/A
48:2E:D9:3D:62:05
NOT BONDED -52 dBm ↔ 31 ms

Device type: UNKNOWN
Advertising type: Legacy
Manufacturer data (Bluetooth Core 4.1):
Company: Apple, Inc. <0x004C> 0x0719070E2075AA30
01000045121212000000000000000000000000

Figure 19: Airpods Pro advertising packet

After finding the correct advertising packet, we can clone it. The author also decided to rename the packet to make it more easily identifiable (Figure 20).



Edit advertising packet

Display name

Apple Airpods Pro

Advertising data

Manufacturer Data:
Company ID: 0x004C Data:
0x0719070E2075AA3001000
045121212000000000000000
0000000000

Options:

☐ Connectable ☐ Scannable

Interval: <160 – 16777215>

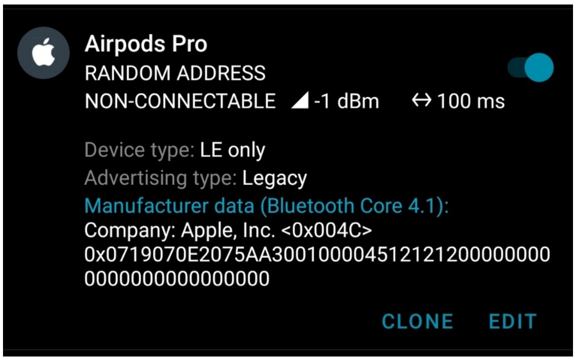
160 * 0.625ms = 100ms

Tx Power Level: <-127 – 1> (dBm)

1

Figure 20: Renaming captured packet

The next step is starting the broadcast which is done by clicking a single button (Figure 21).

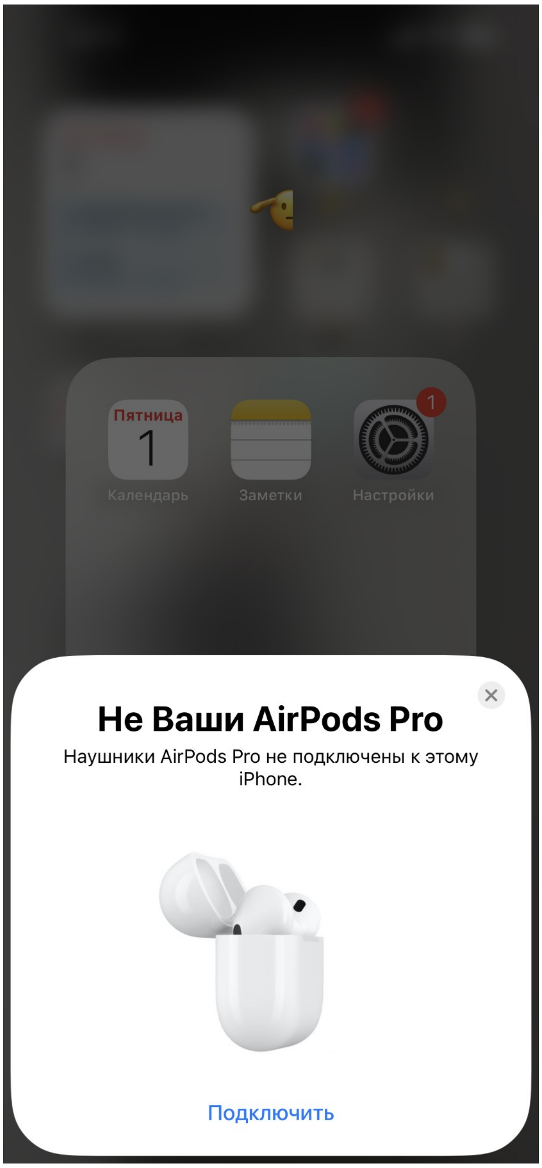


Airpods Pro
RANDOM ADDRESS
NON-CONNECTABLE ◢ -1 dBm ↔ 100 ms

Device type: LE only
Advertising type: Legacy
Manufacturer data (Bluetooth Core 4.1):
Company: Apple, Inc. <0x004C>
0x0719070E2075AA30010000451212121200000000
0000000000000000

Figure 21: Broadcasting spoofed packet

A few seconds after the broadcast was started, a connect card appeared on a nearby iPhone (Figure 22).



Не Ваши AirPods Pro

Наушники AirPods Pro не подключены к этому iPhone.

Figure 22: iPhone connect screen reacting to the spoofed packet

4.5 Mitigating identified vulnerabilities

Mitigating the scanning of advertising packets in Bluetooth Low Energy (BLE) communications presents a unique challenge due to the wireless nature of the technology. Advertising packets are broadcasted openly to facilitate device discovery, so attempting to entirely prevent scanning is impractical. However, security measures can be implemented to enhance privacy and minimize the potential risks associated with scanning. One effective approach is the use of random addresses for each advertising packet. By employing this strategy, BLE devices periodically change their Bluetooth addresses during advertising, making it more challenging for unauthorized entities to track or identify devices consistently.

Mitigating BLE sniffing and spoofing can be done by using secure pairing methods, such as LE Secure Connect, to establish a secure connection between BLE devices. Also, using the Just Works association model should be avoided wherever possible, since it allows an attacker to connect to the target device effortlessly. This ensures that only authenticated devices can communicate, mitigating the risk of unauthorized access and man-in-the-middle attacks.

5 Results and recommendations

The purpose of this study was to explore and analyze potential vulnerabilities in Bluetooth Low Energy (BLE) communication. The experiments were conducted using the nRF52840 DK dongle, Wireshark, and nRF Connect for Mobile tools to intercept, analyze, and manipulate BLE advertisement packets and communication sessions. This section outlines the key findings and implications derived from these experiments.

Using Wireshark and nRF Connect for Mobile, advertisement packets from BLE devices were intercepted and analyzed. The investigation focused on understanding the structure and content of these packets. The results revealed important details such as device identifiers, manufacturer data and offered services.

Sniffing a BLE conversation between an Amazfit GTS 2e smartwatch and a phone gave an opportunity to look into the data exchange and communication patterns. The intercepted data highlighted the importance of securing BLE communications by using a secure pairing mode and association model to prevent eavesdropping and unauthorized access. This means using LE Secure Pairing mode if devices support it and avoiding the Just Works association model if possible.

In the second experiment, the study successfully spoofed an AirPods Pro advertisement packet to deceive an iPhone into displaying the connection screen. This demonstrated a potential security loophole wherein malicious actors could manipulate BLE advertising data to trick users into connecting to unauthorized devices. The results show the importance of robust security mechanisms in BLE devices to prevent spoofing attacks. To secure devices and mitigate spoofing attacks, Security Mode 3 introduced in Bluetooth Core Specification 5.2 can be used [5].

The findings from the experiments emphasize the significance of addressing security concerns in BLE technology. Vulnerabilities observed during advertisement packet analysis and conversation sniffing highlight potential risks, including unauthorized access to sensitive data and privacy breaches. The successful spoofing of an AirPods

Pro advertisement packet raises awareness about the need for enhanced authentication and validation mechanisms in BLE connections.

6 Conclusions

The goal of this thesis was to analyze common BLE vulnerabilities and show the process of exploitation of found vulnerabilities using low-cost devices, as well as discuss possible solutions and mitigations of found vulnerabilities. In order to achieve set goal, two research questions were formulated.

A thorough review and analysis of existing literature on the topic of BLE vulnerabilities was conducted in order to answer the first research question. This involved a thorough analysis of scientific articles and technical publications in order to identify important information concerning the most common vulnerabilities of BLE technology.

In order to answer the second research question experiments were designed. During the experimental part of the thesis author was able to gather data about the peripheral device from its advertising packet, intercept data from BLE traffic by conducting a sniffing attack and spoof BLE advertising packet. After conducting analysis of BLE advertising packets and performing sniffing and spoofing attacks, possible mitigations were described. The author recommends enabling advertising address randomization and using Secure Connect pairing method and avoiding Just Works association model.

Future research involving the use of more expensive hardware, such as Ubertooth One could be conducted. Also, it would be beneficial to investigate the effectiveness of proposed mitigation techniques and security measures

References

[1] A. Lacava, V. Zottola, A. Bonaldo, F. Cuomo, and S. Basagni, “Securing Bluetooth Low Energy networking: An overview of security procedures and threats,” Computer Networks, vol. 211, p. 108953, Jul. 2022, doi: 10.1016/j.comnet.2022.108953.

[2] A. Lonzetta, P. Cope, J. Campbell, B. Mohd, and T. Hayajneh, “Security Vulnerabilities in Bluetooth Technology as Used in IoT,” J. Sens. Actuator Netw., vol. 7, no. 3, p. 28, Jul. 2018, doi: 10.3390/jsan7030028.

[3] M. Cäsar, T. Pawelke, J. Steffan, and G. Terhorst, “A survey on Bluetooth Low Energy security and privacy,” Computer Networks, vol. 205, p. 108712, Mar. 2022, doi: 10.1016/j.comnet.2021.108712.

[4] D. Hollander, "New Core Specification v5.3 Feature Enhancements," Bluetooth, Jul. 15, 2021. [Online]. Available: https://www.bluetooth.com/blog/new-core-specification-v5-3-feature-enhancements/. [Accessed Dec. 7, 2023]

[5] M. Woolley, "Bluetooth Core Specification Version 5.2 Feature Overview," Bluetooth, Dec. 9, 2020. [Online]. Available: https://www.bluetooth.com/wp-content/uploads/2020/01/Bluetooth_5.2_Feature_Overview.pdf. [Accessed Dec. 7, 2023]

[6] P. Kanafek, "What's new in Bluetooth v5.4: An overview," Nordic Semiconductor, Feb. 15, 2023. [Online]. Available: https://devzone.nordicsemi.com/nordic/nordic-blog/b/blog/posts/whats-new-in-bluetooth-v5-4-an-overview. [Accessed Dec. 7, 2023]

[7] S. M. Darroudi and C. Gomez, “Bluetooth Low Energy Mesh Networks: A Survey,” Sensors, vol. 17, no. 7, p. 1467, Jun. 2017, doi: 10.3390/s17071467.

[8] H. Pieterse and M. S. Olivier, “Bluetooth Command and Control channel,” Computers & Security, vol. 45, pp. 75-83, Sep. 2014, doi: 10.1016/j.cose.2014.05.007.

[9] Zhang, Yue & Weng, Jian & Dey, Rajib & Fu, Xinwen. (2019). B Bluetooth Low Energy (BLE) Security and Privacy. 10.1007/978-3-319-32903-1_298-1.

[10] D. Antonioli, N.O. Tippenhauer, and K. Rasmussen, "Key negotiation downgrade attacks on Bluetooth and Bluetooth Low Energy," ACM Transactions on Privacy and Security (TOPS), vol. 23, no. 3, pp. 1-28, 2020.

[11] Attify, "Exploiting Bluetooth Low Energy using Gattacker for IoT - Step-by-Step Guide," Mar. 01, 2018. [Online]. Available: https://blog.attify.com/hacking-bluetooth-low-energy/. [Accessed Dec. 7, 2023]

[12] Nordic Semiconductor, "nRF Sniffer for Bluetooth LE v4.1.x User Guide," [Online]. Available: https://infocenter.nordicsemi.com/pdf/nRF_Sniffer_BLE_UG_v4.1.x.pdf. [Accessed Dec. 7, 2023]

[13] Nordic Semiconductor, "Bluetooth Low Energy Fundamentals," [Online]. Available: https://academy.nordicsemi.com/courses/bluetooth-low-energy-fundamentals/. [Accessed Dec. 7, 2023]

[14] Nordic Semiconductor, "nRF52840 DK," [Online]. Available: https://www.nordicsemi.com/Products/Development-hardware/nrf52840-dk. [Accessed Dec. 7, 2023]

[15] Nordic Semiconductor, "nRF52840 Dongle," [Online]. Available: https://www.nordicsemi.com/Products/Development-hardware/nRF52840-Dongle. [Accessed Dec. 7, 2023]

[16] Nordic Semiconductor, "nRF Sniffer for Bluetooth LE," [Online]. Available: https://www.nordicsemi.com/Products/Development-tools/nRF-Sniffer-for-Bluetooth-LE/Download?lang=en#infotabs. [Accessed Dec. 7, 2023]

[17] Silicon Labs, "Bluetooth Advertising Data Basics," [Online]. Available: https://docs.silabs.com/bluetooth/4.0/general/adv-and-scanning/bluetooth-adv-data-basics. [Accessed Dec. 7, 2023]

[18] Bluetooth, "Heart Rate Profile 1.0," [Online]. Available: https://www.bluetooth.com/specifications/specs/heart-rate-profile-1-0/. [Accessed Dec. 7, 2023]

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I Anton Kolisnetšenko

1 Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Analysis of Common Security Vulnerabilities in the Bluetooth Low Energy Technology", supervised by Mohammad Tariq Meeran

1.1 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2 I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

3 I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

07.12.2023

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.