Final Year Project Report

Full Unit – Interim Report

MIMT attack in Bluetooth Low Energy device

Hanju Wang

A report submitted in partial fulfilment of the degree of

BSc (Hons) in Information Security

Supervisor: Hurley-Smith, Darren

Department of Computer Science

Royal Holloway, University of London

December 06, 2019

Declaration

This report has been prepared on the basis of my work. Where other published and unpublished source materials have been used, these have been acknowledged.

Word Count:4657

Student Name: Hanju Wang

Date of Submission: 6th Dec 2019

Signature: Hanju Wang

Table of Contents

Abstract

Bluetooth technology to become standard inner world in a short time, the main reason is that it not only can make many kinds of intelligent wireless device, you can transfer files, support for voice communication, can establish a data link, such as application of Bluetooth wireless office environment, the typical environment of automotive industry, information home appliances, medical equipment, and the school education and factory automation control, etc. Especially in the individual and family applications, has brought a lot of convenience to our lives.

In the coming Internet of things (IoT) world, biomedical devices must have intelligent, low-power and security features, making Bluetooth Low Energy wireless connectivity an industry design focus. The complementarity and ubiquity of low-power Bluetooth make it an important and growing technology in the healthcare industry.

Meanwhile, the security features of Bluetooth are different from the traditional wired network. The topology of the Bluetooth network may change at any time, and the communication between devices is not necessarily directly performed and may be conveyed by a device in the network. In the network, security issues are quite complex. Although Bluetooth defines the service for data transmission at the bottom and upper layers to achieve security functions similar to wired communication.

The service layer mandatory security mode and the link-layer mandatory security mode still cannot ensure that the network is not easy to leak and be protected from attacks. This paper discusses in detail a man-in-the-middle attack in the Bluetooth wireless local area network interaction environment, in order to raise people's security issues on wireless networks s concern.

Project Specification

The aim of this project is to investigate the Man in Middle attack in the context of Bluetooth-enabled medical or fitness IoT devices.

However, since there are different versions of Bluetooth and each version is dependent upon the integrated hardware, this instigates many security concerns such as Man in Middle attack. It is important to test all Bluetooth version available on medical and fitness devices to check if they are prone to this type of attack.

Data security and authorization management mechanisms are critical in the medical field. The low power Bluetooth standard supports the AES128 encryption protocol, as well as more complex secure hashing algorithms, long keys, and electronic curve cryptography through the MCU. Finally, it is proved to be co-existence. Low-power Bluetooth is also based on ISM and 2.4ghz standard frequency band with ZigBee and wi-fi. In order to ensure co-existence with devices using other wireless standard protocols in hospitals and avoid affecting data collection and transmission due to interference, low-power Bluetooth introduces adaptive frequency hopping and low-duty cycle physical layer control technology.[2]

The LE device uploads the data to the database, which consolidates the same data. Data consolidation can also lead to privacy violations and national security concerns. A large amount of user information collected by LE equipment may be uploaded to the cloud in real-time, and the data can be obtained through illegal attacks and integrated analysis, so as to obtain information of commercial value, track the behaviour path of specific people and understand the disease spectrum of the whole country or region, resulting in serious consequences.

Avoid connecting important data with wearable devices, prevent others from using wearable devices to record personal or work area information data, and avoid the problem that personal privacy and core data of work area are stolen due to disrespect for others' privacy.

There is no liability law for the privacy protection of medical big data. Also, the blurring of data boundaries in the era of big data increases the difficulty of protecting the network and information security. Therefore, it is particularly important to establish a strict regulatory mechanism for medical wearable devices.

First of all, we need to learn about Bluetooth technology and its connection matching method in detail. Second, we need to learn about the attack principle of MITM attack and the existing MITM attack mode in Bluetooth technology. Then we will do the frame design, we will do the simple frame design first, we will connect the LE device to the victim, and establish an attacker, and give him message monitoring and tampering attacks.

Chapter 1: **Introduction**

With the development of computer science technology and mobile communication device technology, portable mobile devices have become one of the necessities of people's daily office and life. These devices include laptop computers, personal digital assistants, smart mobile phones, tablets and wearable equipment, etc. If the information between these devices cannot be shared, the usefulness of mobile devices will be greatly limited. Most of the interconnections between devices are connected by physical cables, such as twisted-pair cables, USB (Universal Serial Bus, universal serial bus) data lines, and so on. With the increasing number of communication protocols, the number of types of connection cables is also increasing. Users are more inclined to wireless connections than physical cables.

Bluetooth technology came into being in this environment. The word "Bluetooth" is derived from the historical allusion that the Danish king Harald Blatand loved the bluebird in the 10th century AD. Bluetooth technology uses the free universal band 2.4GHz ISM (industrial, scientific, medical) [1], originally proposed by an engineer of communications giant Ericsson Mobile Communications Company in 1994. Subsequently, in 1998, the Bluetooth Special Interest Group (BT SIG) was established by five companies: Ericsson, Nokia, IBM, Toshiba, and Intel. Bluetooth technology. With the continuous development of Bluetooth technology, the members of the Bluetooth Technology Alliance have reached more than 20,000, providing strong technical support for the development of Bluetooth [3-8]. It uses frequency hopping technology to make mobile devices form a wireless personal area. Other devices join this wireless personal area to achieve interconnection between devices, information sharing, and wireless Communication "last 10 meters" problem. Bluetooth has become one of the world's short-range wireless connection standards, and it is one of the most important wireless communication methods recognized by the industry to connect portable mobile devices with other human-computer interaction devices (such as headsets, keyboards, and mice).

In cryptography and computer security, a man-in-the-middle attack (MITM) is an attack in which an attacker secretly passes and may alter the communication between two parties who believe they are communicating directly. An example of a MITM attack is active eavesdropping, where the attacker makes the relationship between the victim and the victim and the messaging between them so that they believe they are talking about a direct private connection to each other, when in fact the whole conversation is controlled by the attacker.

Man-in-the-middle attacks steal information by adding third parties. When Bluetooth device A and device B establish a connection, the attacker adds device C between A and B. Device A and device B will not communicate directly but communicate through device C. Assume that device A is the master and device B is the slave. When device A sends a connection request, attacker device C sends a connection response before device B, and device C sends a connection request to device B at the same time. In this way, both device A and device B think that a connection has been established with each other, but the fact is that device C has established a connection with device A and device B, respectively, and device C conducts attacks such as message monitoring and tampering between device A and device B.

MITM attacks can be prevented or detected in two ways: authentication and tamper detection. Authentication provides some assurance that a given message is from a legitimate source. Tamper detection shows only evidence that the message may have been tampered with.[10]

Chapter 2: The architectural components of Bluetooth Low energy

A communication protocol refers to the rules that both entities must follow if they want to successfully communicate with each other to realize a certain service. Generally, the contents of the agreement will define the unit format of the communication data, the meaning of the information represented by the information unit, and the communication. Connection and the timing of information when receiving and sending, to ensure the smooth transmission of data during communication. The Bluetooth low energy protocol is the core of the Bluetooth low energy network system. The architecture of the Bluetooth low energy protocol is shown in Figure1:

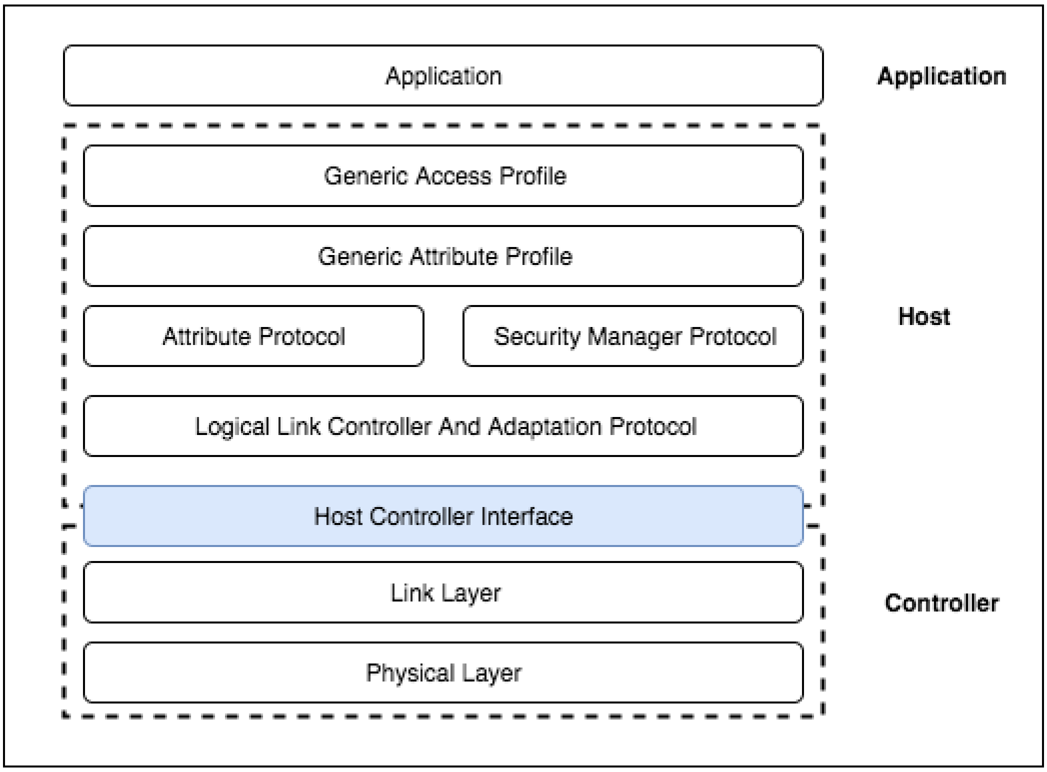

Application
Application
Generic Access Profile
Generic Attribute Profile
Attribute Protocol
Security Manager Protocol
Host
Logical Link Controller And Adaptation Protocol
Host Controller Interface
Link Layer
Controller
Physical Layer

Figure 1: The architecture of Bluetooth Low Energy [8]

In the Bluetooth Core specification, there are three major architectural layers: Controller, Host and Application. At the host layer, there is a module called Security Manager (SM) that defines the matching and key distribution methods and protocols, the corresponding Security toolkit, and the Security Manager Protocol (SMP) that defines the matching command frame format, frame structure, and timeout limits. A controller is usually a physical device that can send and receive radio signals and understands how to translate these electrical signals into data packets that carry information. The controller has both the physical layer and the link layer, as well as the lower half of the direct test mode and the Host-Controller Interface (HCI) layer. The controller communicates with the outside world

The antenna is connected to the host through the host controller interface.

Physical layer: The physical layer is the part that uses 2.4GHz radio to complete data transmission and reception. In low-power Bluetooth, a modulation method called Gaussian

Frequency Shift Keying (GFSK) is used to change the frequency of radio waves and transmit 0 or 1 information.

Direct test mode: The direct test mode allows the tester to let the physical layer of the controller send a series of test data packets or accept a series of test data packets, and then determine whether the physical layer complies with the radio frequency specifications according to the number of data packets.

Link layer: The link layer is the most complex part of the Bluetooth low energy protocol system. The link layer is responsible for broadcasting, scanning, establishing and maintaining connections, and at the same time ensuring that the contents of the data packets are organized in the correct way, the checksums are calculated correctly, and the encryption sequence.

Main controller interface: The main controller interface provides a standard interface for the host to communicate with the controller. It consists of two parts: a logical interface and a physical interface. Because the host controller interface is located between the controller and the host, the part located in the controller is often referred to as the lower part of the host controller interface, and the part located in the host is often referred to as the upper half of the host controller interface.

The host is usually a software stack that manages how two or more devices communicate and how to use the radio to provide several different services at the same time. There are three protocols in the host: logical link control and adaptation protocol (L2CAP), attribute protocol (ATT), security manager protocol (SM), and two specifications: Generic Attribute Specification Profile (GATT) and Generic Access Profile (GAP)

Logical link control and adaptation protocol: L2CAP is a multiplexing layer of Bluetooth low energy, which enables Bluetooth low energy to multiplex three different channels. It also supports data segmentation and reassembly functions, enabling larger messages to be transmitted on the underlying radio.

Security manager protocol: The security manager defines a simple pairing and key distribution protocol. At the same time, he also provides a security toolbox, which is responsible for generating the hash value of the data, the confirmation value, and the short-term key used in the pairing process. The security manager ensures that data can be exchanged securely in the Bluetooth low energy network.

Attribute protocol: The attribute protocol defines a set of rules for accessing data on the peer device. It allows the client to discover and obtain properties on the property server.

Common attribute specification: The common attribute specification is above the attribute protocol, and defines the type of the attribute and the method of using the attribute. It specifies a standard way of discovering and using services, features, and descriptors.

Chapter 3: **Design of MITM attack in the device**

As I know the MITM attack in Bluetooth is the High-risk password vulnerability named CVE-2018-5383. The Bluetooth allows an unauthorized NFC device to intercept a user's Bluetooth connection and steal data. The Bluetooth vulnerability affects devices made by major manufacturers such as Apple, Broadcom and Intel. It is not known if it will affect devices running Android and Linux.

The vulnerability stems from the operating system's secure connection pairing mode for Bluetooth connection in Bluetooth low power mode and the Bluetooth simple secure connection mode BR/EDR in firmware. Researchers at the Technion institute of technology in Israel have found that Bluetooth devices use both connection modes without verifying the device's public key. Because key verification in these two connection modes is not required, some manufacturers' Bluetooth products do not use elliptic curve parameters to generate verification keys during Bluetooth secure connection.

Therefore, as long as the attacker USES an unauthorized Bluetooth device at close range, he can launch a man-in-the-middle attack on the target and intercept the Bluetooth connection key used by the device. This allows these Bluetooth devices, once connected, to steal communications data from the device or install malware.

Moreover, the Bluetooth SIG has also added testing for this vulnerability within its Bluetooth Qualification Process.

The CERT/CC published a security advisory on the flaw that includes technical details.

The U.S. computer emergency response team coordination centre (CERT/CC) also issued a security notice. Bluetooth connections use the ECDH key negotiation algorithm to establish connections between devices. The ECDH key negotiation algorithm consists of a private key and a public key, which is used to generate a connection pair code for a Bluetooth device. Because some Bluetooth devices do not use elliptic curve parameters for security verification during each connection, they are vulnerable to attack by some malicious Bluetooth devices. To fix this, the Bluetooth technology alliance has enhanced Bluetooth connection security, forced public key verification by default, and added this test to Bluetooth authentication. The bug's update package needs to be installed at both device and operating system levels to maximize the security of the Bluetooth connection.

Android and Linux operating systems have yet to determine whether the vulnerability, which affects Bluetooth chips made by Apple, Broadcom, Intel and Qualcomm, is affected.

Apple and Intel have released updates to their systems that address the vulnerability. Intel on Monday released firmware and software updates that fully fix the Bluetooth bug in its dual-channel wireless hotspot products. Broadcom has released a solution to the vulnerability to oems, while Qualcomm has not released any statements about the vulnerability. No large-scale security risk incidents caused by the vulnerability have been identified.[11]

Now we further hypothesize this attack:

1) The thermometer is the master device, and the portable computer is a Bluetooth-connected slave device. Since the Bluetooth protocol allows the role to be switched between master and slave, this assumption is not fixed. An attacker computer as a man in the middle can force a master-slave conversion.

2) Initially, the thermometer and laptop have been paired with Bluetooth, and a semi-permanent Bluetooth link key K has been derived, and the attacker does not know K.

3) The EAP-AKA authentication method is implemented between the thermometer and the authentication server. Both the thermometer and the authentication server derive two keys: one is the master key of the encryption and integrity key from UMTS, and one is from Master session key MSK (Master Session Key).

In this experiment, a laptop computer requests the use of an IP-based service over a wireless network, with the thermometer assisting in authentication and authorization by the authentication server. The attacker uses a wireless node and a device that can intercept the victim's thermometer and the laptop Communicating Bluetooth computer launches an attack.

MSK is used as the encryption key of the wireless connection. Moreover, the authentication server transmits the MSK to the wireless node through an AAA protocol (such as RAD 1US) (assuming that the communication between the authentication server and the node has been properly protected by TLS or IPSec); and the thermometer uses the Bluetooth protocol to transfer MSK is transmitted to the laptop.

The attack is divided into two phases. The first phase: the attacker's computer passively records the Bluetooth session, during which the victim's thermometer sends the MSK to the laptop, and the attacker can also obtain the MSK by hacking into a legitimate node used by the laptop. Second stage: The attacker's computer replays the Bluetooth session recorded in the first stage, forcing the laptop to reuse the compromised MSK. As a result, the laptop is connected to the attacker's node without any knowledge.

Chapter 4: **Specific attack process**

Your report will be structured as a collection of number sections at different levels of detail. For example, the heading to this section is a first-level heading (it's called *Heading 1*) and has been defined with a particular set of font and spacing characteristics. At the start of a new section, you need to select the appropriate heading style, *Heading 1* in this case, by clicking *Heading 1* on the style toolbar.

The attacker's computer records the Bluetooth packet transmitted between the thermometer and the laptop. The captured packet must include: authentication, encrypted commands, and the encrypted session key MSK.

Initially, the thermometer and the laptop used the Bluetooth link management protocol LMP to authenticate each other.

The thermometer (master device) sends a challenge RAND1 to the portable computer using the au_rand message of the LMP; the laptop calculates the RES1 response and uses the sres message of the LMP to reply the RES1 to the thermometer. Similarly, the portable computer authenticates by sending the challenge RAND2 to the thermometer Thermometer and verify the RES2 response sent from the thermometer.

Then, the thermometer (master device) uses the start_encryption_req message of LMP to send an encrypted random number EN_RAND to the portable computer to start Bluetooth encryption. This means that the subsequent information transmitted between the thermometer and the portable computer is encrypted.

Figure 2 depicts how an attacker replays a recorded session to force a laptop to reuse a compromised MSK.

Initially, the attacker's computer (hereafter referred to as the attacker) and the laptop authenticated each other:

The attacker first sends the challenge RAND1 recorded in the first phase to the laptop and receives the RES1 response from the laptop; then, the laptop sends a new challenge RAND3 to the attacker, and the attacker transfers it to the thermometer. Thermometer calculation RES3 responds and sends it to the attacker, who then forwards it to the laptop.

Next, the attacker sends the encrypted random number EN_RAND recorded in the first stage to the laptop to start Bluetooth encryption. Then, the attacker replays the EAP authentication sequence, and the laptop directly transfers it to the attacker's wireless node The EAP message is not forwarded by the attacker's wireless node. Then, the attacker sends a message to the laptop, including the compromised M SK recorded in the first stage. Finally, the laptop sends the encryption to the attacker's wireless node Data, and it is believed to be on a legitimate network. In this way, the attacker replays the Bluetooth communication to the laptop, and the laptop connects to the attacker's node without even knowing it.

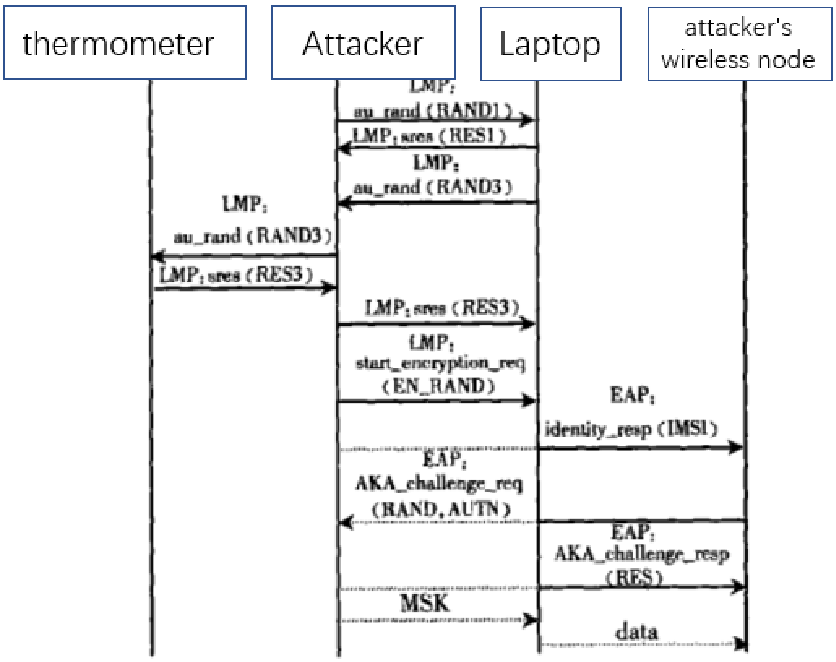


Figure 2 Attack process

Chapter 5: Experiment outcomes report

During the Bluetooth authentication process, the thermometer sends a Bluetooth challenge RAND1 to the laptop. The laptop responds with RES1 via Bluetooth. When the thermometer receives the RES1 response, it calculates the authentication encryption offset: ACO = E1 (K, RAND1, ADD), where E is the Bluetooth authentication algorithm; K is a semi-permanent Bluetooth link key derived from the thermometer and the portable computer; ADD is the Bluetooth adapter address in the portable computer. If mutual authentication is used, the portable computer also Send a challenge RAND2 to the thermometer and the thermometer returns a response RES2 to the laptop

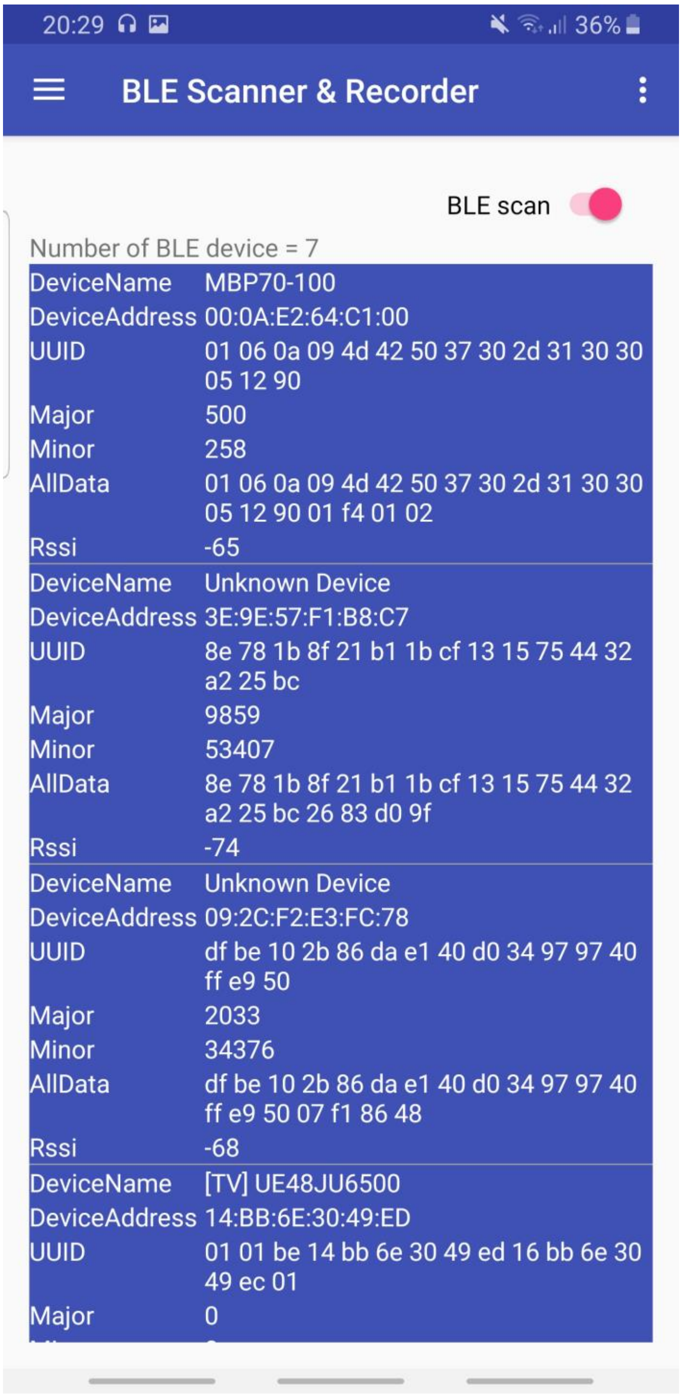
Then, the thermometer sends an encryption command to the portable computer and provides an encrypted random number EN_RAND. This random number is used to calculate the value of the Bluetooth encryption key (KC) KC = E3 (K, EN_RAND, ACO). Here, E3 is an Algorithm for calculating the encryption key; K is the current link key.

Finally, the calculation of the keystream K_Cipher is obtained through the E0 algorithm. When using E0 to calculate the keystream K_Cipher, parameters such as the encryption key KC, the clock and address of the thermometer Bluetooth adapter are required as inputs.

Since K_Cipher's calculation does not rely on any random number generated by the laptop, the attacker's computer can replay the sequence of Bluetooth encrypted information to the laptop.

Chapter 6: Presentation issues

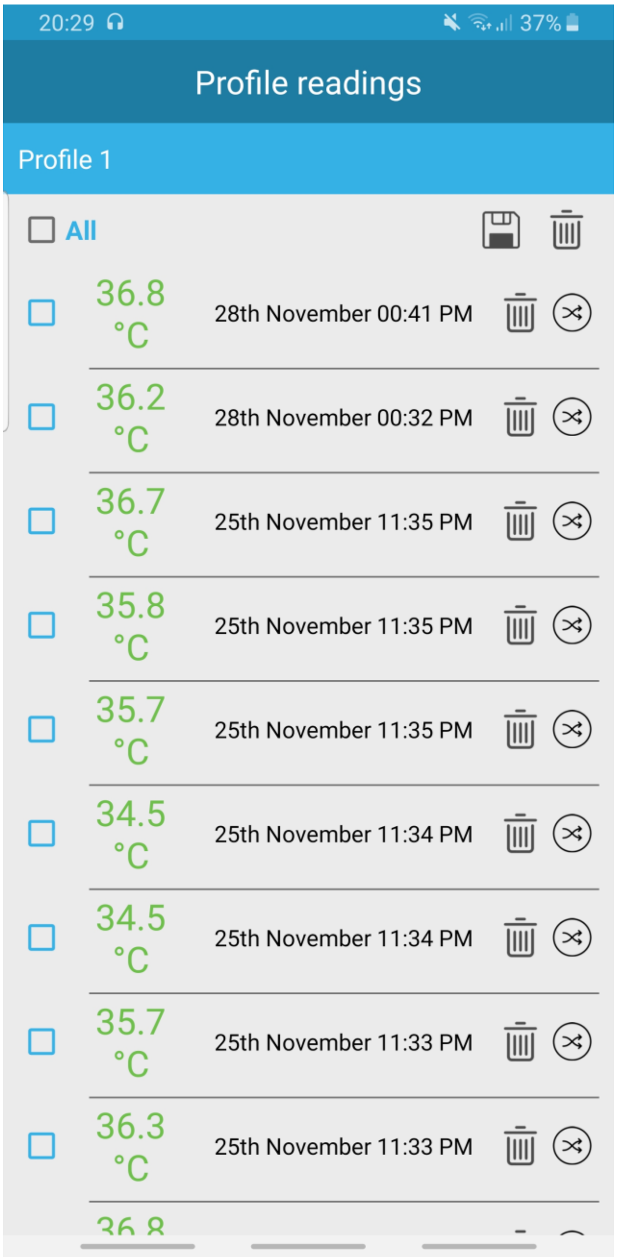
6.1 Figures, Charts and Tables



Number of BLE device = 7

DeviceName	MBP70-100
DeviceAddress	00:0A:E2:64:C1:00
UUID	01 06 0a 09 4d 42 50 37 30 2d 31 30 30 05 12 90
Major	500
Minor	258
AllData	01 06 0a 09 4d 42 50 37 30 2d 31 30 30 05 12 90 01 f4 01 02
Rssi	-65
DeviceName	Unknown Device
DeviceAddress	3E:9E:57:F1:B8:C7
UUID	8e 78 1b 8f 21 b1 1b cf 13 15 75 44 32 a2 25 bc
Major	9859
Minor	53407
AllData	8e 78 1b 8f 21 b1 1b cf 13 15 75 44 32 a2 25 bc 26 83 d0 9f
Rssi	-74
DeviceName	Unknown Device
DeviceAddress	09:2C:F2:E3:FC:78
UUID	df be 10 2b 86 da e1 40 d0 34 97 97 40 ff e9 50
Major	2033
Minor	34376
AllData	df be 10 2b 86 da e1 40 d0 34 97 97 40 ff e9 50 07 f1 86 48
Rssi	-68
DeviceName	[TV] UE48JU6500
DeviceAddress	14:BB:6E:30:49:ED
UUID	01 01 be 14 bb 6e 30 49 ed 16 bb 6e 30 49 ec 01
Major	0

Figure 3 BLE Scanner & Recorder



20:29 37%

Profile readings

Profile 1

□ All

□	36.8 °C	28th November 00:41 PM
□	36.2 °C	28th November 00:32 PM
□	36.7 °C	25th November 11:35 PM
□	35.8 °C	25th November 11:35 PM
□	35.7 °C	25th November 11:35 PM
□	34.5 °C	25th November 11:34 PM
□	34.5 °C	25th November 11:34 PM
□	35.7 °C	25th November 11:33 PM
□	36.3 °C	25th November 11:33 PM
	36.8	

Figure 4 Data collected from the thermometer

Actions	Details	Done? Yes/No
Learn about Bluetooth technology	Learn about Bluetooth pairing, connecting, communication protocol,	Yes
Know about Bluetooth application in Medical and fitness devices	Low energy Bluetooth is a simple, safe, low power and high compatibility medical wireless technology	Yes
Know about historic attack Bluetooth device by M ITM attack	The update to the Bluetooth system will fix the high-risk password vulnerability. For example, CVE-2018-5383, Bluetooth exploit can track and identify	Yes

	iOS, Microsoft mobile device users	
Design of testing framework for running MITM attack	Chapter 3 and Chapter 4 in this report	Yes
Device_connect	The device connects to the mobile phone by Bluetooth	Yes
Get data from the device	The APP display the data from the device	Yes
Attacker simulates the mobile phone	The attacker gets data from the device and gives the wrong data to the mobile phone.	Yes
Device and mobile phone set High-level key to establishing connect	Both the device and the authentication server derive two keys: one is the master key of the encryption and integrity key from UMTS, and one is from Master session key MSK (Master Session Key)	Yes
After setting the key the attacker can get the data from the device or not	attacker replays a recorded session to force a laptop to reuse a compromised MSK.	Yes
The experiment comes outs	Chapter 5	Yes

4.2 Source Code

The code style is Python. There are 7 files in my zip source code.: 6 code files and 1 txt file for reading me.

1. helpers.py: Setup logging configuration

2. Bluetooth_handler.py: Representing a Bluetooth LE interface, and offering to scan, connecting, transmitting, etc. Build upon the socket of the underlying socket_handler and Scapy HCI commands.

3. Socket_handler.py: Handles everything low-Level ranging from Socket to Host-Controller-Interface, though usually by providing a convenient interface to scapy.

4. Mitm_handler.py: Abstracting the use of two Bluetooth Stacks by passing commands to the appropriate Handler in accordance to the idea behind BTLE MITM, especially who has to communicate with whom after which action and vice versa.

5. Interactive_session.py: Starting an Interactive Console and constantly checking

The program scans for advertising peripherals, offers the option to connect to one, then connects to this advertising peripheral, shutting down the advertising of this peripheral, then imitates the exact same advertising packages from the peripheral itself with an arbitrarily spoofed BD_Addr (can be the same as peripheral, as it allows to change the own Hardware BD_Addr as well). A victim will then connect to the spoofed peripheral, thinking it would be the real desired peripheral due to its identical advertising data. Any Attribute Protocol (ATT) data exchanged between the real peripheral and the real central will then be tunnelled through the MITM relay station and displayed while upholding both separate connections independently.

As the MITM station requires two Bluetooth stacks (one each the for the fake central and fake peripheral) does the main-routine initialize two separate pairs of underlying sockets and Bluetooth stacks. The socket_handler takes care of everything low level from the Host Controller Interface (HCI) to the handling of the socket, while the bluetooth_handler implements the Bluetooth stack logic and provides the functional interfaces. The mitm_handler unites both Bluetooth stacks and directs the attack commands to the appropriate stack, while also implementing the Man-in-the-Middle logic. The Interactive_Session does not add functionality but does instead allow the interactive operation of the mitm_handler, while constantly requesting the handling of new incoming data.

Object Interfaces are kept to a minimum and only directed upwards (socket_handler <- bluetooth_handler <- mitm_handler <- interactive_session) to maximize cohesion and minimize coupling (though this is broken once in regards to the bluetooth_handlers as they are made known to each other to enable direct forwarding of ATT Data to the respectively other Bluetooth stacks.

Main:

```
import logging

import datetime

from .helpers import setup_logging

from .socket_handler import SocketHandler

from .bluetooth_handler import BluetoothHandler

from .mitm_handler import MITMHandler

from .interactive_session import InteractiveSession

# Set up logging

setup_logging()

logger = logging.getLogger(__name__)

def main():

    print("Starting Bluetooth Low-Energy MITM. Timestamp: {}".format(datetime.datetime.now()))
    logger.info("\n\n\n\nStarting Bluetooth Low-Energy MITM. Timestamp: {}".format(datetime.datetime.now()))

    # Initialize two seperate (Peripheral and Central) Socket Handler and acquire the Bluetooth sockets

    socket_handler_peripheral = SocketHandler("__socket_handler_peripheral__", 0)

    socket_handler_central = SocketHandler("__socket_handler_central__", 1)

    # Initialize two seperate (Peripheral and Central) Bluetooth Handlers (representing the actual Bluetooth Interface)

    # and connect them with their according Socket Handler

    bluetooth_handler_peripheral = BluetoothHandler("__bluetooth_handler_peripheral__", socket_handler_peripheral)

    bluetooth_handler_central = BluetoothHandler("__bluetooth_handler_central__", socket_handler_central)

    # Unite both Bluetooth Handlers in a Man-in-the-Middle Handler, abstracting scanning, connecting, mimicking etc

    # to one interface and directing the commands to the appropriate Bluetooth Handler

    mitm_handler = MITMHandler(bluetooth_handler_peripheral, bluetooth_handler_central)

    # Start Interactive Session that allows for input commands such as scanning, connecting, etc and controls

    # controls both Bluetooth stacks via the Man-in-the-Middle Handler

    InteractiveSession(mitm_handler)

    # Close Sockets before exiting

    mitm_handler.close_sockets()

if __name__ == '__main__':

    main()
```

Chapter 7: Table of Contents and References

[1] Heydon R, Hunn N. Bluetooth low energy[J]. CSR Presentation, Bluetooth SIG https://www. bluetooth. org/DocMan/handlers/DownloadDoc. ashx, 2012.

[2] Silabs.com. (2016). 生物医学设备为何偏好"低功耗蓝牙". [online] Available at: https://www.silabs.com/community/chinese-blog.entry.html/2016/11/28/_-nhkw [Accessed 9 Nov. 2019].

[3] Bluetooth SIG. Bluetooth specification version 2.0+EDR[J]. Bluetooth SIG, 2004.11. [Accessed 17 Nov. 2019].

[4] Bluetooth SIG. Bluetooth specification version 2.1+EDR [J]. Bluetooth SIG, 2007.7.26. [Accessed 17 Nov. 2019].

[5] Bluetooth SIG. Bluetooth specification version 3.0+HS[J]. Bluetooth SIG, 2009.4.21. [Accessed 17 Nov. 2019].

[6] Bluetooth SIG. Bluetooth specification version 4.0[J]. Bluetooth SIG, 2010.6.30. [Accessed 17 Nov. 2019].

[7] Bluetooth SIG. Bluetooth specification version 4.1 [J]. Bluetooth SIG, 2013.12.3. [Accessed 17 Nov. 2019].

[8] Bluetooth SIG. Bluetooth specification version 4.2[J]. Bluetooth SIG, 2014.12.2. [Accessed 17 Nov. 2019].

[9] Bhargava, M. (2017). IoT Projects with Bluetooth Low Energy. [online] Subscription.packtpub.com. Available at: https://subscription.packtpub.com/book/hardware_and_creative/9781788399449 [Accessed 28 Nov. 2019].

[10] En.wikipedia.org. (n.d.). Man-in-the-middle attack. [online] Available at: https://en.wikipedia.org/wiki/Man-in-the-middle_attack [Accessed 28 Nov. 2019].

[11] Paganini, P. (2018). CVE-2018-5383 Bluetooth flaw allows attackers to monitor and manipulate traffic. [online] Security Affairs. Available at: https://securityaffairs.co/wordpress/74719/hacking/cve-2018-5383-bluetooth-flaw.html [Accessed 1 Dec. 2019].

Chapter 8: Project Information and Rules

The details about how your project will be assessed, as well as the rules you must follow for this final project report, are detailed in the project booklet.

You must read that document and strictly follow it.