

BLUETOOTH ATTACKS- a Survey on smart watches
data leakage through open Bluetooth connections

A project report submitted to

**Rajiv Gandhi University of Knowledge Technologies
SRIKAKULAM**

**In partial fulfillment of the requirements for the
Award of the degree of**

**BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE AND ENGINEERING**

Submitted by

3rd year B. Tech 2nd semester

Paidi Revathi - S170286

Uppala Chaitanya-S170020

Devudala Hema Kiran-S170914

Under the Esteemed Guidance of

Asst.Prof. Sri CH. Satish Kumar Sir



Rajiv Gandhi University of Knowledge Technologies SRIKAKULAM

CERTIFICATE

This is to certify that the thesis work titled “**Bluetooth Attacks-A survey on smart watches data leakage through open Bluetooth connection**” was successfully completed by P Revathi(**S170286**), U Chaitanya(**S170020**), D Hemakiran(**S170914**) In partial fulfillment of the requirements for the Mini Project in Computer Science and Engineering of Rajiv Gandhi University of Knowledge Technologies under my guidance and output of the work carried out is satisfactory.

Prof.Sri. CH. Satish Kumar Sir

Project Guide

Prof.Sri. K. Dileep Kumar Sir

Project Coordinator

DECLARATION

I declared that this thesis work titled “**Bluetooth Attacks-A survey on smart watches data leakage through open Bluetooth connection**” is carried out by me during the year 2021-22 in partial fulfillment of the requirements for the Mini Project in **Computer Science and Engineering**.

I further declare that this dissertation has not been submitted elsewhere for any Degree. The matter embodied in this dissertation report has not been submitted elsewhere for any other degree. Furthermore, the technical details furnished in various chapters of this thesis are purely relevant to the above project and there is

no deviation from the theoretical point of design, development and implementation.

Paidi Revathi (S170286)

Uppala Chaitanya (S170020)

Devudala Hema Kiran(S170914)

ACKNOWLEDGEMENT

I would like to articulate my profound gratitude and indebtedness to my project guide **Prof. Sri Ch.Satish Kumar**, Assistant Professor who has always been a constant motivation and guiding factor throughout the project time. It has been a great pleasure for me to get an opportunity to work under his guidance and complete the thesis work successfully.

I wish to extend my sincere thanks to **Prof.Smt.S. Lakshmi Sri**, Head of the Computer Science and Engineering Department, for her constant encouragement throughout the project.

I am also grateful to other members of the department without their support. My work would have been carried out so successfully.

I thank one and all who have rendered help to me directly or indirectly in the completion of my thesis work.

Project Team Members:

Paidi Revathi (S170286)

Uppala Chaitanya (S170020)

Devudala Hema Kiran(S170914)

ABSTRACT

Bluetooth is a short-range wireless communication technology that are widely used in today's world .It allows devices such as mobile phones, computers, and peripherals(IOT devices) to transmit data or voice wirelessly over a short distance. The purpose of Bluetooth is to replace the cables that normally connect devices, while still keeping the communications between them secure.

Nowadays the usage of smart watches is widely spread in every age group because of various inbuilt features such as calling, messaging, health tracking, and GPS tracks etc. Some of these features collect sensitive data and store them for analytics purposes. The security aspects provided by these devices depend on the price range.

Our study is to identify the data leakages through open Bluetooth connection smart watches. We conducted a survey on it and observed that there might be a chance to connect to Bluetooth without any Permission from the master device. Our study aim's at to find flaws regarding open Bluetooth connections and prevent attacks such as data leakage, identifying and protecting from hackers.

INDEX

CH.NO	CONTENTS	PG.
1	INTRODUCTION	
1.1	Introduction.....	1
1.2	Statement of the problem.....	1
1.3	Objective.....	2
1.4	Goals.....	2
1.5	Scope.....	2
1.6	Limitations.....	2
2.	LITERATURE SURVEY	
2.1	Collect Information.....	3
2.2	Study.....	3
2.3	Benefits.....	3
2.4	Summary.....	3
3.	SYSTEM ANALYSIS	
3.1	Existing System.....	4
3.2	Disadvantages.....	4
3.3	Proposed System.....	4
3.4	Advantages.....	4
3.5	System Requirements.....	4
4.	SYSTEM DESIGN	
4.1	Design of the System.....	5
4.1.1	Class Diagram.....	5
4.1.2	Use Case Diagram.....	6
4.1.3	Sequence Diagram.....	7
4.1.4	Data Flow Diagram.....	8

5. SYSTEM IMPLEMENTATION

5.1 Real time observation.....	9
--------------------------------	---

6. SOURCE CODE

6.1 Test Code for bluetooth enabling(python).....	14
6.2 Code for near by scanning(python).....	15
6.3 Kali Linux commands.....	16

7. SYSTEM TESTING

7.1 TESTING INTRODUCTION.....	19
7.2 TYPES OF TESTING.....	19
7.3 LEVELS OF TESTING.....	20

8. PRECAUTIONS 22

9. CONCLUSION 23

10. FUTURE ENHANCEMENT 24

11. REFERENCES 25

CHAPTER 1

INTRODUCTION

1.1 Introduction

Bluetooth technology is used to connect various devices wirelessly. This makes life much simpler when it comes to sharing files, photos and music with other gadgets within range. There are a few things you need to know about connecting your Bluetooth device before you get started. Because not everyone has the same experience pairing a device or finding the right instructions.

Bluetooth is a short-range wireless communications protocol operating in the 2.4GHz that allows devices such as cell phones, laptops, and personal music players to connect to each other. It's useful for staying connected while on the go. In this article, we will discuss some of the basics of Bluetooth technology including its working principle, different types of Bluetooth devices and how they can all be used conveniently together.

The Bluetooth specification is split into two major parts. One part is called Bluetooth BR/EDR, also known as Bluetooth classic, the other one is called Bluetooth Low Energy (BLE) which was added in version 4.0. Both are nearly completely independent protocols. Our focus here is on core BLE. Applications can be found in all areas, including ones with high requirements on safety and security such as electronic locks, alarm systems, process monitoring, or medical devices. These devices are often controlled and monitored via BLE by smartphones or laptops.

Being wireless, a BLE interface is particularly exposed to potential attacks. An attacker does not need physical access to the device and has a low risk of being caught in action. This makes security of BLE interfaces a major concern. Potential attack goals include sniffing, denial-of-service (DoS), spoofing, injection of messages, partial or full takeover of a connection, tracking, and localization. The Bluetooth specification offers security measures against most of these threats, introducing multiple device pairing schemes, optional encryption and authentication of connections, or address randomization.

However, those security measures are only effective if properly specified, used, and implemented correctly. Several security measures defined in early versions are flawed and serious weaknesses have been found that implementers need to be aware of. Weaknesses within earlier versions of the specification are still relevant. BLE modules, SoCs and devices commonly remain in the market and in service for several years without providing an upgrade option. So even for newly designed devices it is often desirable to remain backwards compatible with older versions of the protocol.

1.2 Statement of the problem

Nowadays the digital world is expanding. For example, the usage of smart watches is widely spread in every age group because of various inbuilt features such as calling, messaging, health tracking, and GPS tracks etc. Some of these features collect sensitive data and store them for analytics purposes. So, this kind of data can be prone to several attacks from hackers. Our study is to identify the data leakages through open Bluetooth connections on smart watches.

1.3 Objective

1. To create awareness of Bluetooth attacks while connecting with BLE devices.
2. To communicate with BLE devices in a secure way.
3. To acknowledge BLE device users about prevention methods.

1.4 Goals

1. Privacy loopholes
2. Integrity
3. Confidentiality
4. Authentication
5. Communication

1.5 Scope

Nowadays the digital world is widely Spreading.

1. Smart watches also use Digital Technology. It uses Bluetooth technology to provide most of its features.
2. In the digital world, data plays a major role which may lead to cyber attacks.
3. This project scope remains until smartwatch is used in this digital world.

1.6 Limitations

1. BLE devices do not work with large amounts of data.
2. Weak support
3. No authentication and Less security

Chapter 2

LITERATURE SURVEY

2.1 Collect Information

We have taken the information from the other sources like IEEE papers and other online websites to check how they are categorized and organized, and we proposed this ourselves.

2.2 Study

1. Smart Watch features
2. Bluetooth Attacks
3. BLE devices and connection management

2.3 Benefits

1. Prevention methods to avoid Security issues in smart watches
2. To acknowledge the data leakage of BLE devices such as smart watches.

Summary

Bluetooth Low Energy (BLE) is a wireless standard, widely used to communicate with Android and iOS mobile applications with devices of many kinds. These include home security, medical and other which may exchange sensitive data or perform sensitive operations. It's critical to establish a secure communication using a proper pairing mode. So, in this case study we studied about the data leakage from BLE devices i.e Smart Watch because of open Bluetooth connection establishment.

Chapter -3

SYSTEM ANALYSIS

3.1 Existing system

1. MAC Spoofing: It is a method of attack which is done at a time of pairing and connects to a Bluetooth device using the trusted device's mac address to gain access to the device.
2. MITM Attack: The attacker intercepts a packet sent by one of the devices, modifies it and then send it to The other devices

3.2 Disadvantages

1. It can not be used for higher data rates as offered by wifi and cellular technologies. It supports 1 Mbps & 2 mbps data rates.
2. It can not be used for long distance wireless communications unlike cellular and wifi devices. It supports upto 200 meters in LOS (Line of Sight).
3. It is open to interception and attack due to wireless transmission/reception.

3.3 Proposed System

Surveillance attack:

We conduct experiments with some of our friend's smart watches who wear watches on their non-dominant wrist. We also attach a smartphone close to the smart watch on the user's wrist to collect the sensitive data. The BLE(Bluetooth Low Energy) traffic between the smartwatch and the user's smartphone is sniffed using their own personal application. The data is collected while the smart watch app is open in the foreground on an android mobile.

3.4 Advantages

1. Low power consumption
2. Automatic

3.5 System Requirements

Software Requirements:

1. Windows 10
2. Linux
3. Python idle
- 4.pybluez (python module for Bluetooth functions)

Hardware Requirements:

1. RAM: 4GB above
2. Hard disk: 500 GB above

CHAPTER- 4

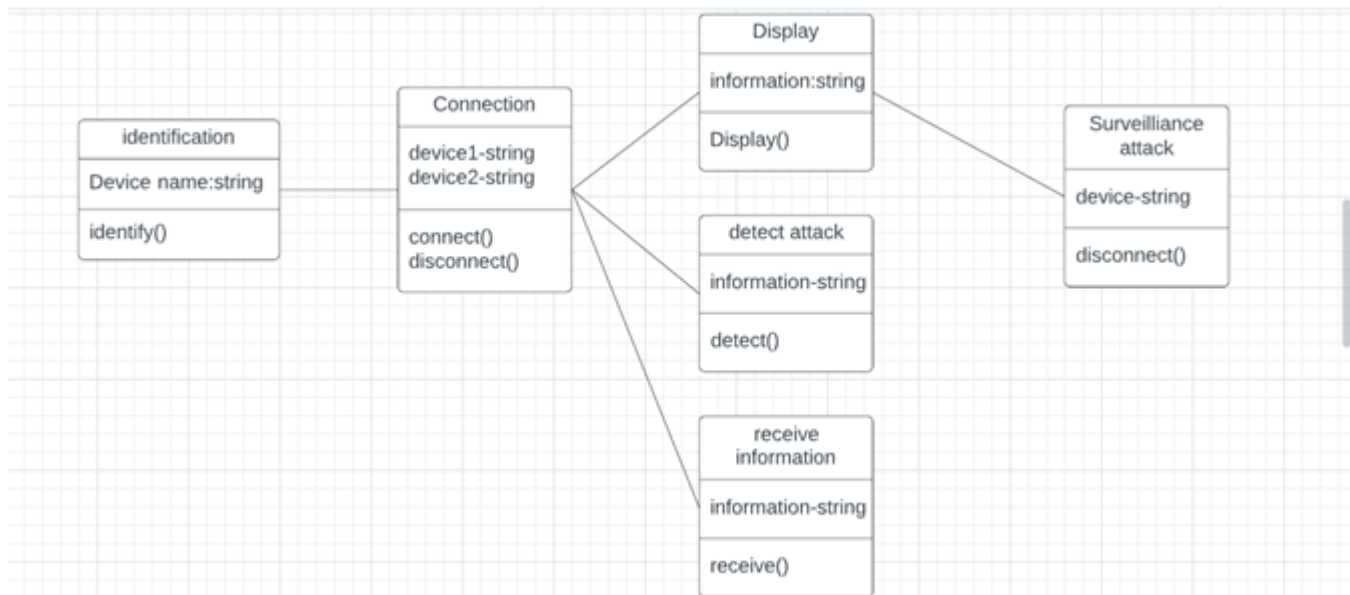
SYSTEM DESIGN

4.1 Design of the System:

Unified Modeling Language (UML) was created in 1995 by merging diagramming conventions used by three application development methodologies: OMT by James Rumbaugh, Objectory by Ivar Jacobson and the Brooch procedure by using Grady Brooch. Previous to this time, these three amigos, together with a few dozen other practitioners had promoted competing methodologies for systematic program development, each and every with its own system of diagramming conventions. The methodologies adopted a sort of cookbook sort of pushing an application task via a succession of life cycle stages, culminating with a delivered and documented software. One purpose of UML was once to slash the proliferation of diagramming techniques by way of standardizing on an original modeling language, as result facilitating verbal exchange between builders. It achieved that goal in 1997 when the (international) Object administration team (OMG) adopted it as a commonplace. Some critics don't forget that UML is a bloated diagramming language written by means of a committee. That said, I do not forget it to be the nice manner to be had today for documenting object-oriented program progress. It has been and is fitting more and more utilized in industry and academia. Rational Rose is a pc Aided program Engineering (CASE) software developed by way of the Rational organization underneath the course of Brooch, Jacobson and Rumbaugh to support application progress using UML. Rational Rose is always complex due to its mission of wholly supporting UML. Furthermore, Rational Rose has countless language extensions to Ada, C++, VB, Java, J2EE, and many others. Rational Rose supports ahead and reverse engineering to and from these languages. However, Rational Rose does not aid some usual design tactics such as knowledge drift diagrams and CRC cards, due to the fact that these will not be a part of UML. Considering that Rational Rose has so many capabilities it is a daunting task to master it. Happily, loads can be executed making use of only a small subset of these capabilities. These notes are designed to introduce beginner builders into making productive use of this sort of subset.

4.1.1 Class Diagram

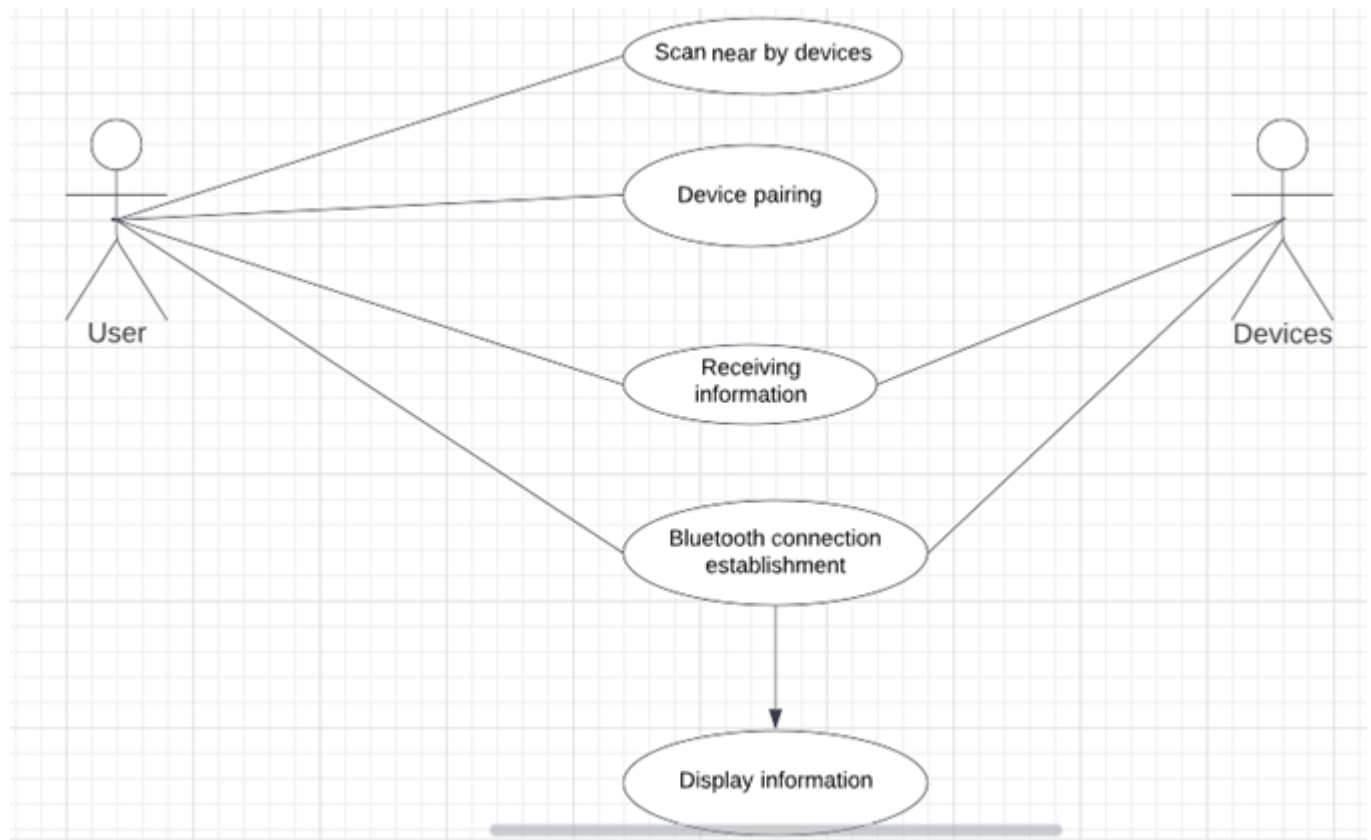
Class diagram in the Unified Modelling Language (UML), is a kind of static structure diagram that describes the constitution of a process through showing the system's classes, their attributes, and the relationships between the class. The motive of a class diagram is to depict the classes within a model. In an object-oriented software, classes have attributes (member variables), operations (member capabilities) and relation



4.1.2 Use Case Diagram

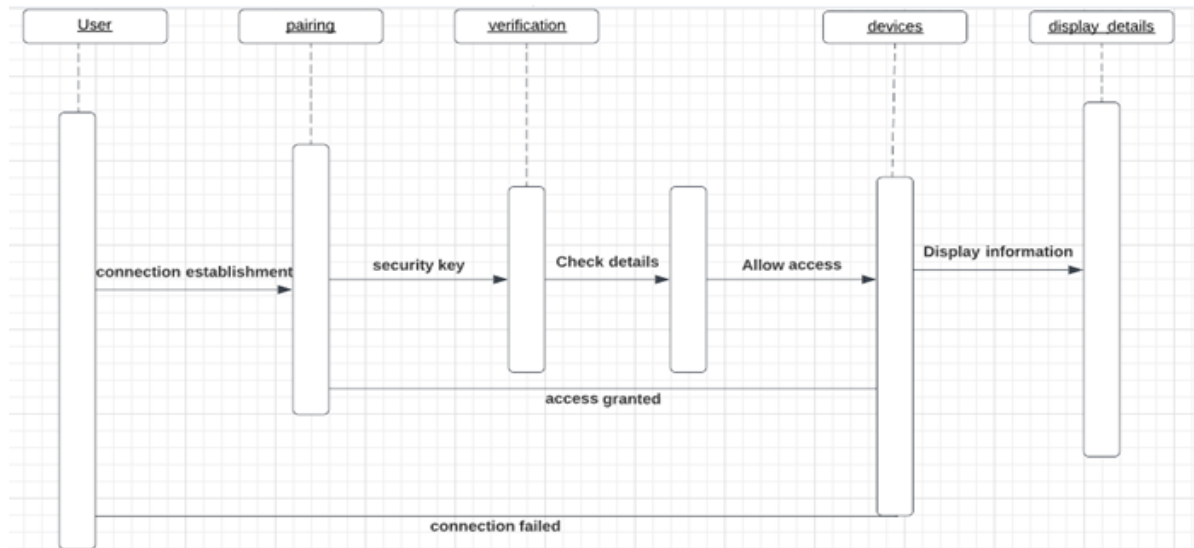
It is a visual representation of what happens when an actor interacts with a system. A use case diagram captures the functional aspects of a system.

The system is shown as a rectangle with the name of the system inside the actor are shown as stick figures, the use cases are shown as solid bordered ovals labeled with the name of the use case and relationships are lines or arrows between actor and use cases. Symbols used in Use cases are as follows-Relationship.



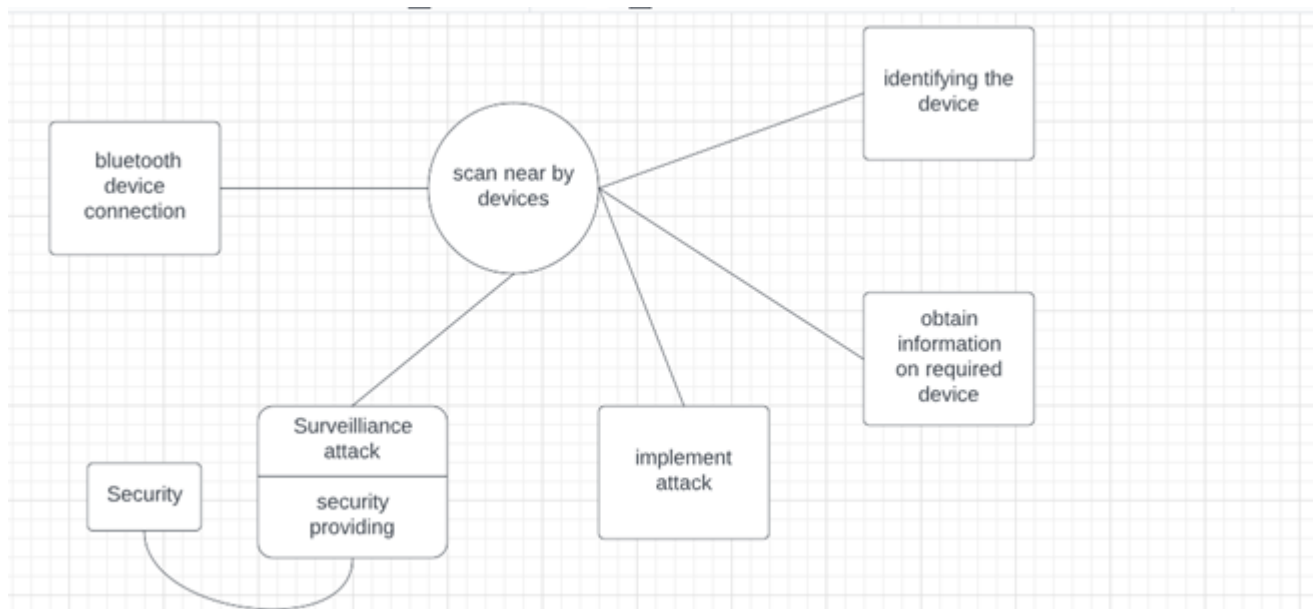
4.1.3 Sequence Diagram

A sequence diagram in Unified Modeling Language (UML) is one variety of interaction diagram that suggests how methods operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are quite often referred to as event-hint diagrams, event situations, and timing diagrams. A sequence diagram suggests, as parallel vertical traces (lifelines), special systems or objects that are residing at the same time, and, as horizontal arrows, the messages exchanged between them, within the order of the place they occur.



4.1.4 Data Flow Diagram

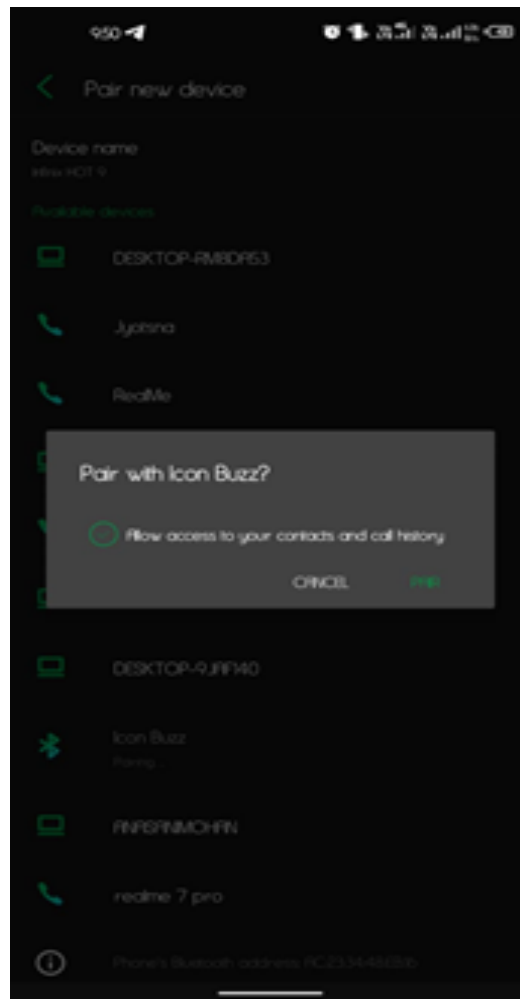
A data flow diagram or bubble chart (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. Often they are a preliminary step used to create an overview of the system which can later be elaborated. DFDs can also be used for the visualization of data processing (structured design). A DFD shows what kinds of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of processes, or information about whether processes will operate in sequence or in parallel (which is shown on a flowchart).



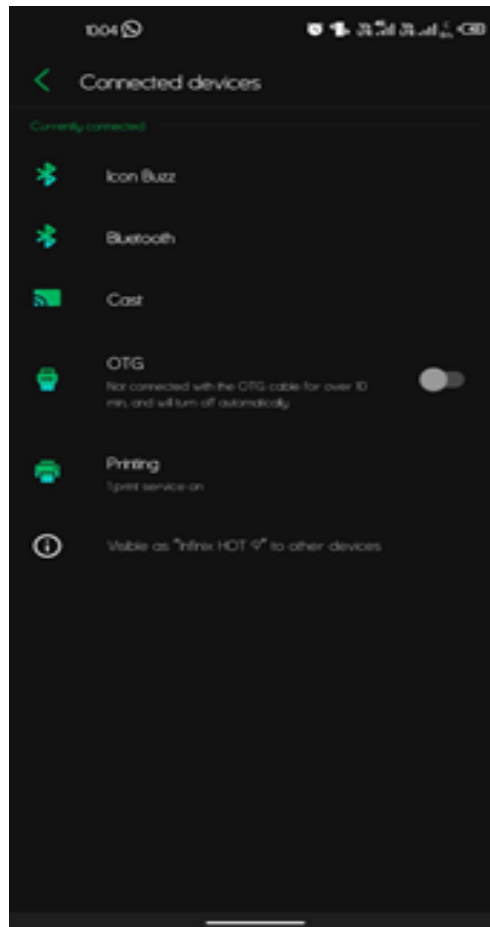
CHAPTER-5

SYSTEM IMPLEMENTATION

5.1 Real time observation

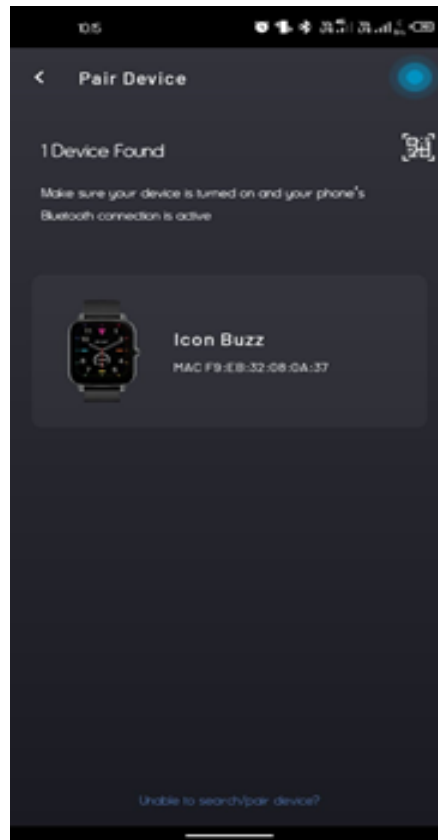


Pop up message while connecting(which is only pops up in smart phones but not in BLE devices)



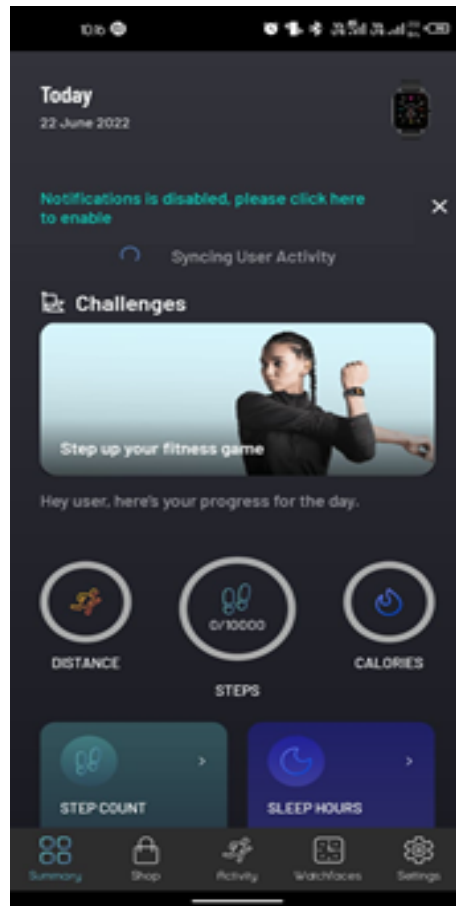
Bluetooth connection establishment

(connection is established without BLE device acknowledgement)



Binding with smart watch application

(this step was done to access the smart watches sensitive data)



Data stored in app for further analysis

(Where data received from smart watch is stored for further analysis when connected to smartphone via Bluetooth service)



(Data obtained using smartwatches preferred applications)

6.1 Test code

For Bluetooth enabling automatically in windows:

```
from winrt.windows.devices import radios

async def bluetooth_power(turn_on):

    all_radios = await radios.Radio.get_radios_async()

    for this_radios in all_radios:

        if this_radios.kind == radios.RadioKind.BLUETOOTH:

            if turn_on:

                result = await this_radios.set_state_async(radios.RadioState.ON)

            else:

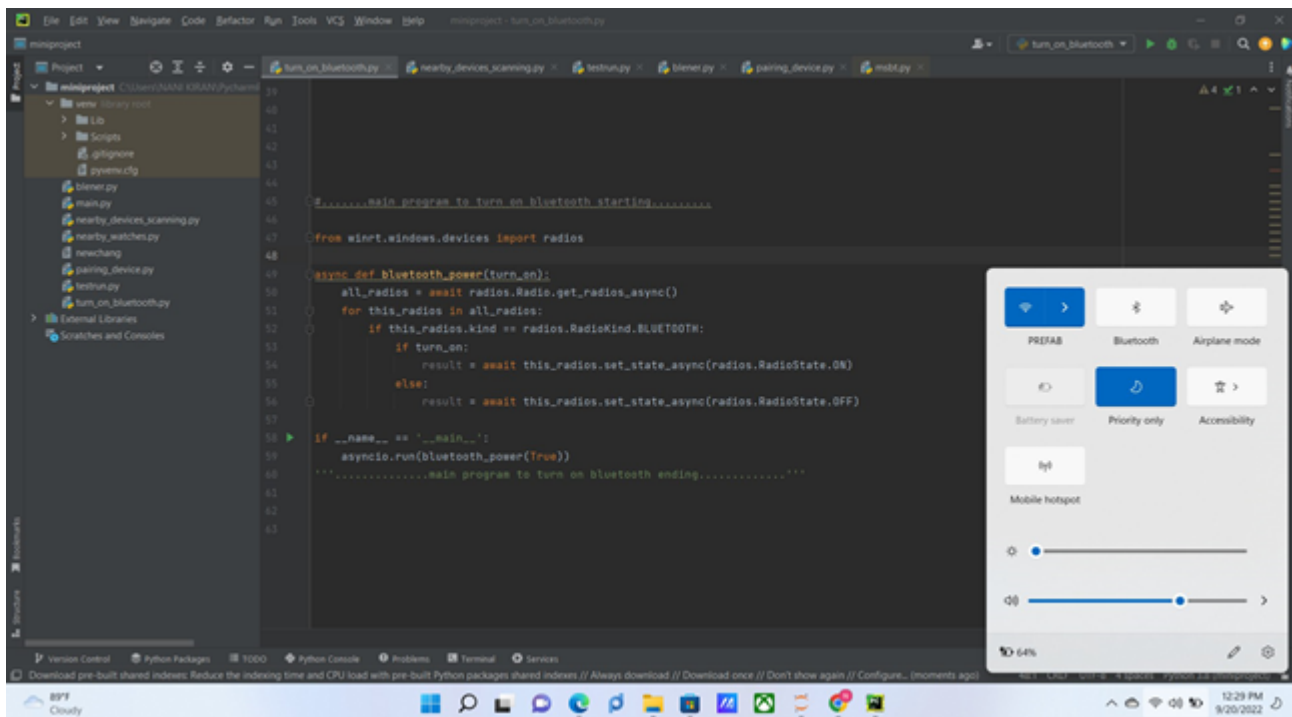
                result = await this_radios.set_state_async(radios.RadioState.OFF)

    if name == '__main__':

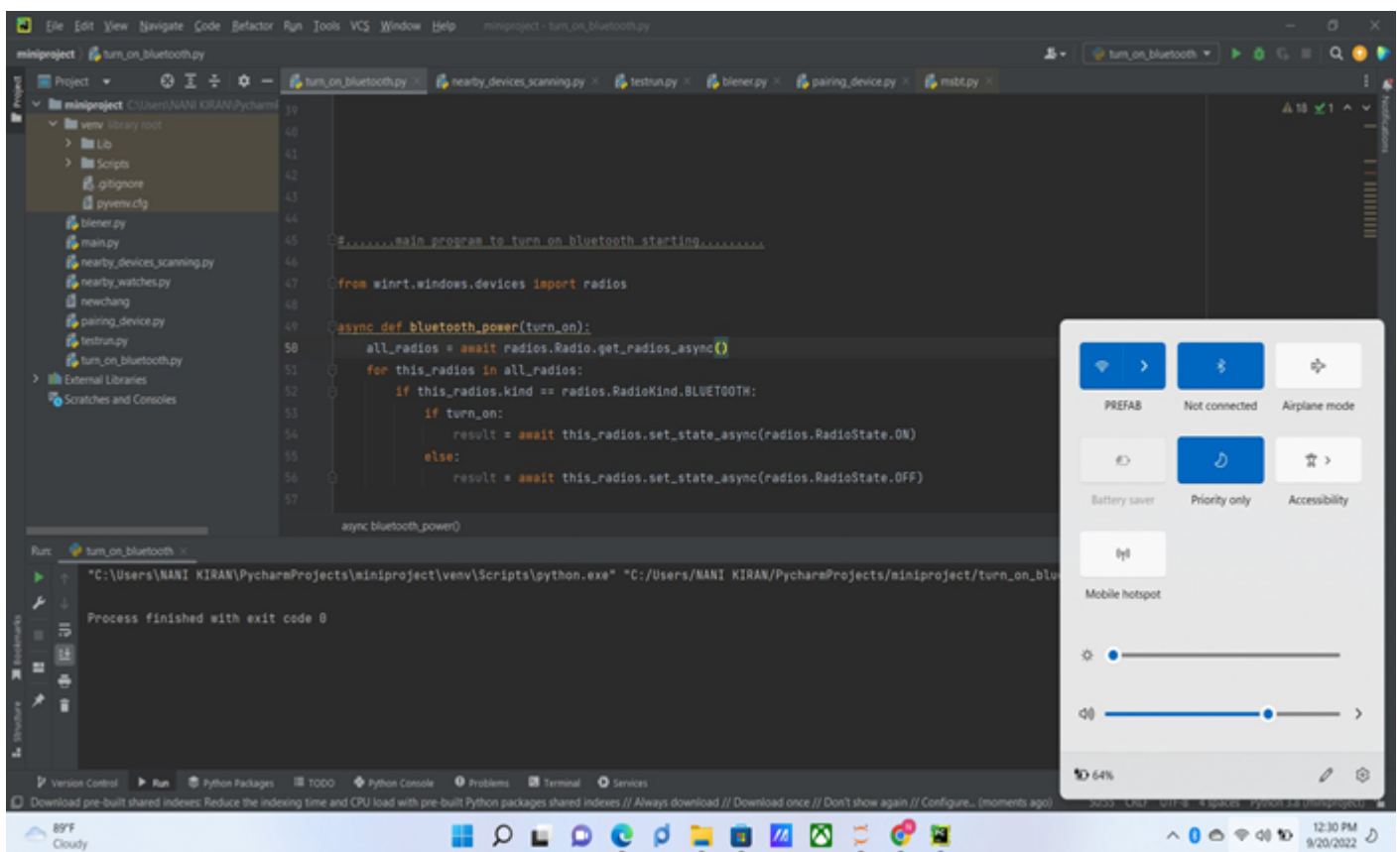
        asyncio.run(bluetooth_power(True))
```

Output1:Before executing the code (Bluetooth does not enabled)

Bluetooth Attacks



Output2: Bluetooth automatically enabled



6.2 Test code for scanning nearby devices:

```
import bluetooth #importing bluetooth module

nearby_devices = bluetooth.discover_devices(lookup_names='true',lookup_class='true')# scanning for bluetooth
devices in range

print('\n')

print('....scanning for nearby devices.....')

print('\n')

no_of_devices = len(nearby_devices)# finding no of devices found

print('number of devices found :',no_of_devices)

print('\n')

for addr,name,device_class in nearby_devices: #finding address,name,devices class for every device found

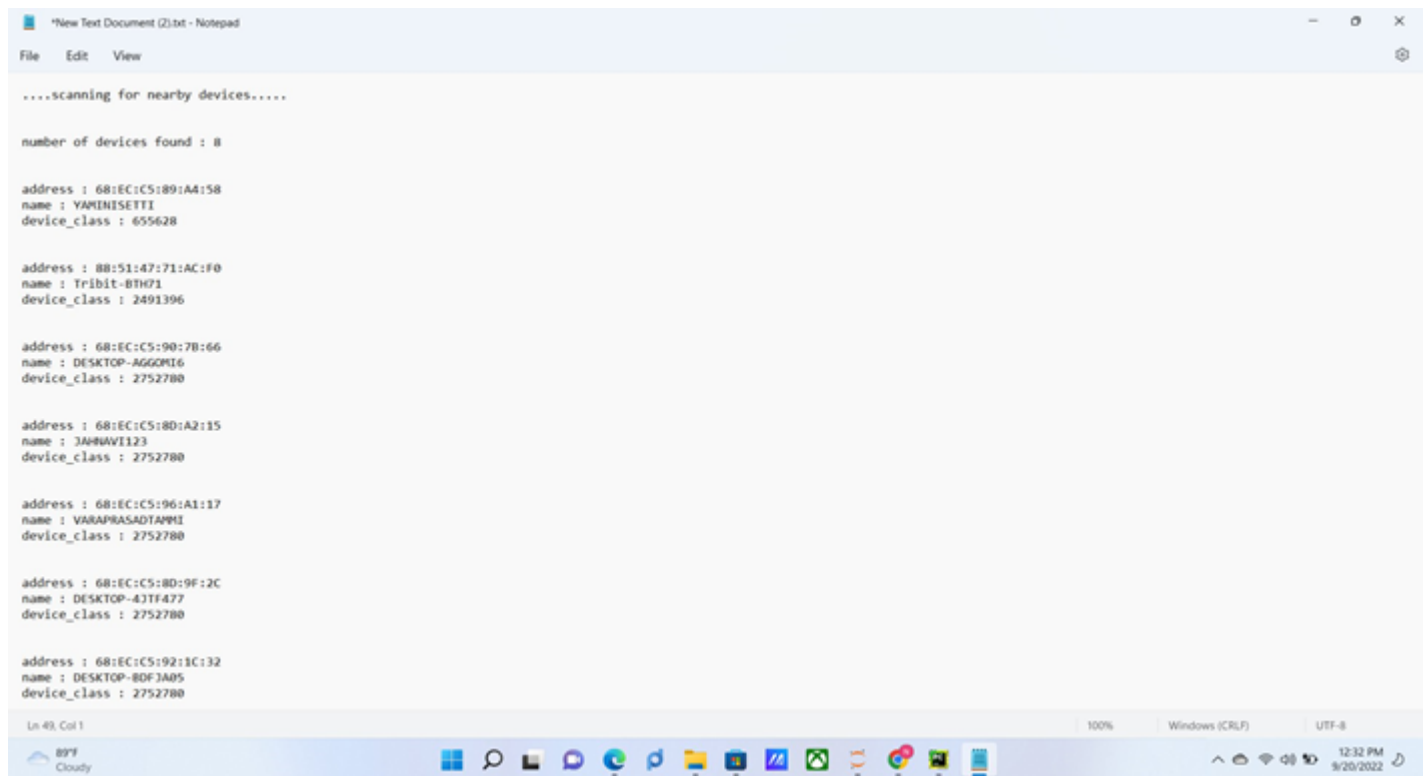
print('address :',addr)

print('name :',name)

print('device_class :',device_class)

print('\n')
```

Output: Scanning for nearby devices



```

....scanning for nearby devices....

number of devices found : 8

address : 68:EC:C5:89:A4:58
name : YAMINISSETTI
device_class : 655628

address : 88:51:47:71:AC:F0
name : Tribit-BTH01
device_class : 2491396

address : 68:EC:C5:90:7B:66
name : DESKTOP-AGGOMI6
device_class : 2752780

address : 68:EC:C5:8D:A2:15
name : JAHNAVI123
device_class : 2752780

address : 68:EC:C5:96:A1:17
name : VARAPRASADTAMMI
device_class : 2752780

address : 68:EC:C5:8D:9F:2C
name : DESKTOP-4JTF477
device_class : 2752780

address : 68:EC:C5:92:1C:32
name : DESKTOP-8DFJA05
device_class : 2752780

```

In this process of getting result we scanned every device around within range but unable to scan the smart watches

This occurred due to the no proper module maintenance which supports Bluetooth functions.

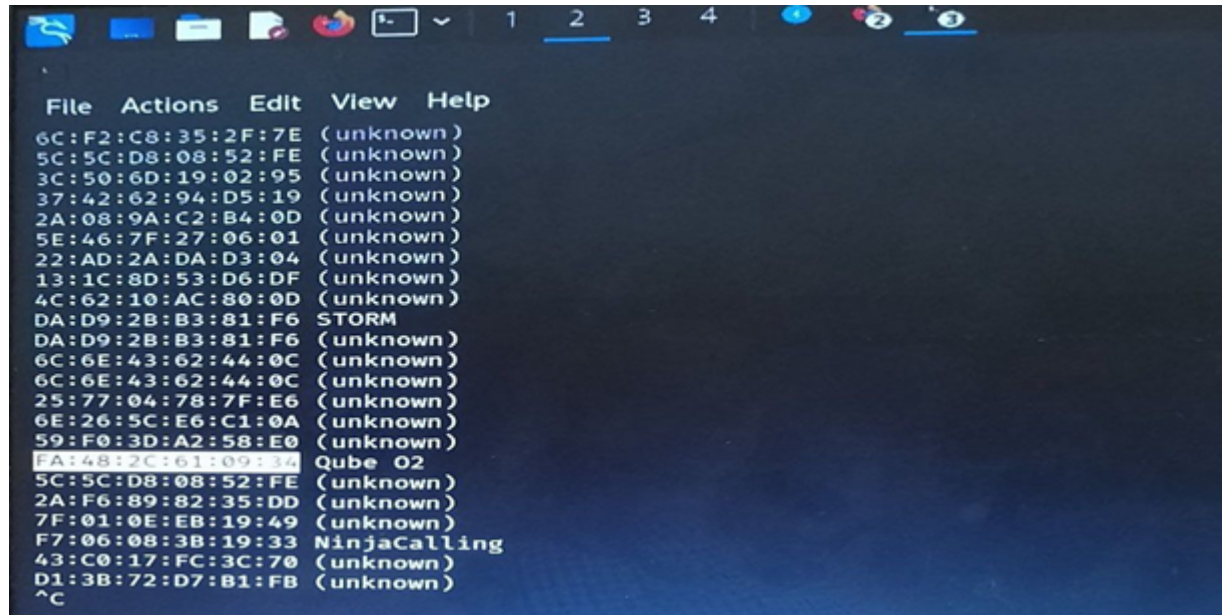
Code for scanning the Bluetooth devices and connection establishment in Kali linux:

>>Service Bluetooth start (to enabling the bluetooth service in kali linux operating system)

>>hcitool lescan (to start scanning for Bluetooth low energy devices)

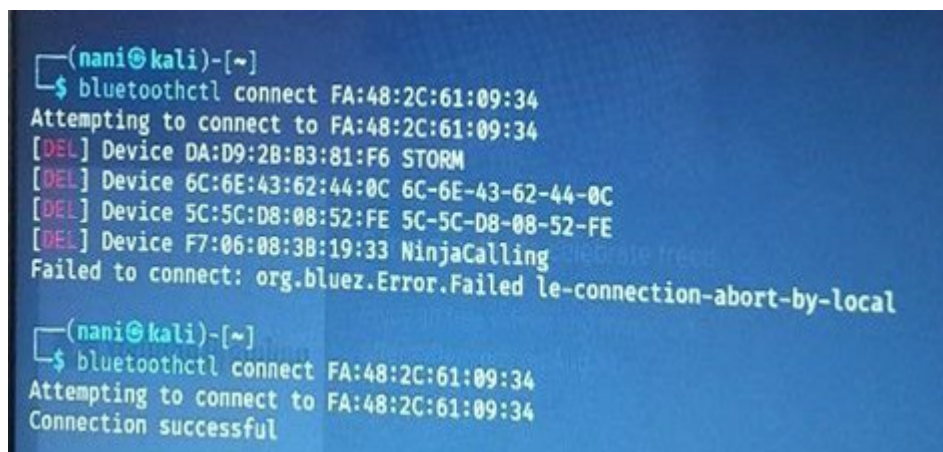
Output:

Available Bluetooth Low Energy Devices



>>>bluetoothctl connect **FA:48:2C:61:09:34** (to connect Bluetooth of Smart watch using Bluetooth address)

Output: Connection establishment with “Qube 02” smartwatch



We have succeeded to connect to the target device once but afterwards we can not able to connect again.

Using bettercap tool:

```
(nani@kali)-[~]
$ sudo bettercap
bettercap v2.32.0 (built for linux amd64 with go1.18.1) [type 'help' for a list of commands]

192.168.16.0/24 > 192.168.16.180 » [22:04:20] [sys.log] [war] Could not find mac for 192.168.16.179
192.168.16.0/24 > 192.168.16.180 » ble.recon on
[22:04:32] [sys.log] [err] unknown or invalid syntax "ble.recon on", type help for the help menu.
192.168.16.0/24 > 192.168.16.180 » █
```

In this module there should be BLE function to support bluetooth function ,but no more available to do work on it.

>>hcitool lescan (find out the storm watch bluetooth address)

>>gatttool -t random -b DD:C3:C6:05:65:7C -I

[DD:C3:C6:05:65:7C][LE]> connect

Output:

```
(nani@kali)-[~]
$ gatttool -t random -b DD:C3:C6:05:65:7C -I
[DD:C3:C6:05:65:7C][LE]> connect
Attempting to connect to DD:C3:C6:05:65:7C
Connection successful
[DD:C3:C6:05:65:7C][LE]> primary
attr handle: 0x0001, end grp handle: 0x0009 uuid: 00001800-0000-1000-8000-00005f9b34fb
attr handle: 0x000a, end grp handle: 0x000d uuid: 00001801-0000-1000-8000-00005f9b34fb
attr handle: 0x000e, end grp handle: 0xffff uuid: 00000af4-0000-1000-8000-00005f9b34fb
[DD:C3:C6:05:65:7C][LE]> Characteristics
handle: 0x0002, char properties: 0x0a, char value handle: 0x0003, uuid: 00002a00-0000-1000-8000-00005f9b34fb
handle: 0x0004, char properties: 0x02, char value handle: 0x0005, uuid: 00002a01-0000-1000-8000-00005f9b34fb
handle: 0x0006, char properties: 0x02, char value handle: 0x0007, uuid: 00002a04-0000-1000-8000-00005f9b34fb
handle: 0x0008, char properties: 0x02, char value handle: 0x0009, uuid: 00002a06-0000-1000-8000-00005f9b34fb
handle: 0x000b, char properties: 0x20, char value handle: 0x000c, uuid: 00002a05-0000-1000-8000-00005f9b34fb
handle: 0x000f, char properties: 0x0e, char value handle: 0x0010, uuid: 00000afa-0000-1000-8000-00005f9b34fb
handle: 0x0011, char properties: 0x12, char value handle: 0x0012, uuid: 00000afb-0000-1000-8000-00005f9b34fb
handle: 0x0014, char properties: 0x12, char value handle: 0x0015, uuid: 00000af6-0000-1000-8000-00005f9b34fb
handle: 0x0017, char properties: 0x0e, char value handle: 0x0018, uuid: 00000af5-0000-1000-8000-00005f9b34fb
[DD:C3:C6:05:65:7C][LE]> char-desc
handle: 0x0001, uuid: 00001800-0000-1000-8000-00005f9b34fb
handle: 0x0002, uuid: 00001803-0000-1000-8000-00005f9b34fb
handle: 0x0003, uuid: 00002a00-0000-1000-8000-00005f9b34fb
handle: 0x0004, uuid: 00001803-0000-1000-8000-00005f9b34fb
handle: 0x0005, uuid: 00002a01-0000-1000-8000-00005f9b34fb
handle: 0x0006, uuid: 00001803-0000-1000-8000-00005f9b34fb
handle: 0x0007, uuid: 00002a04-0000-1000-8000-00005f9b34fb
handle: 0x0008, uuid: 00001803-0000-1000-8000-00005f9b34fb
handle: 0x0009, uuid: 00002a06-0000-1000-8000-00005f9b34fb
handle: 0x000a, uuid: 00001800-0000-1000-8000-00005f9b34fb
handle: 0x000b, uuid: 00001803-0000-1000-8000-00005f9b34fb
handle: 0x000c, uuid: 00002a05-0000-1000-8000-00005f9b34fb
handle: 0x000d, uuid: 00002902-0000-1000-8000-00005f9b34fb
handle: 0x000e, uuid: 00001800-0000-1000-8000-00005f9b34fb
handle: 0x000f, uuid: 00001803-0000-1000-8000-00005f9b34fb
handle: 0x0010, uuid: 00000afa-0000-1000-8000-00005f9b34fb
handle: 0x0011, uuid: 00001803-0000-1000-8000-00005f9b34fb
handle: 0x0012, uuid: 00000afb-0000-1000-8000-00005f9b34fb
handle: 0x0013, uuid: 00002902-0000-1000-8000-00005f9b34fb
```

CHAPTER 7

SYSTEM TESTING

INTRODUCTION

The cause of testing is to detect mistakes. Making an attempt out is the technique of looking to realize each viable fault or weakness in a piece product. It presents a method to determine the performance of add-ons, sub-assemblies, assemblies and/or a completed product. It is the method of exercising a program with the intent of constructing certain that the application procedure meets its necessities and client expectations and does no longer fail in an unacceptable process. There are rather plenty of forms of scan. Each experiment sort addresses a special trying out requirement.

TYPES OF TESTS:

Unit testing:

Unit checking out involves the design of scan circumstances that validate that the Internal application good judgment is functioning safely, and that program inputs produce legitimate outputs. All decision branches and interior code floats must be validated. It's the checking out of character application items of the application. It is achieved after the completion of a person unit earlier than integration. It is a structural checking out that relies on competencies of its construction and is invasive. Unit exams participate in common exams at component level and scan a distinct business approach, utility, and/or process configuration. Unit assessments are certain that every specified course of an industry method performs appropriately to the documented requisites and involves clearly outlined inputs and anticipated results.

Integration testing:

Integration Testing is designed to scan built-in program accessories to determine within the occasion that they evidently run as one software. Trying out is occasion driven and is more concerned with the fundamental final result of screens or fields. Integration assessments reveal that despite the fact that the accessories had been for my part pleasure, as proven through effectively unit checking out, the combo of accessories is correct and regular. Integration checking out is chiefly aimed at exposing the issues that come up from the performance of different components.

Functional testing:

Functional Testing checks provide systematic demonstrations that capabilities established are to be had as particular by means of the business and technical specifications, method documentation, and consumer manuals. Functional testing is working on below mentioned data.

Legitimate input: identified lessons of legitimate input ought to be accredited.

Invalid enter: recognized lessons of unacceptable effort must be rejected.

Capabilities: recognized features ought to be exercised.

Output: recognized courses of software outputs have got to be exercised.

Systems/Procedures: performance of the system here was invoked.

Individual and team work of useful checks is fascinated by specifications, key capabilities, or special scan instances. Moreover, systematic insurance plans concerning establishing business method flows; data fields, predefined processes, and successive strategies have to be regarded for trying out. Before useful trying out is whole, extra checks are recognized and the strong price of present checks be strong minded.

System testing:

scheme difficulty ensures that the whole included agenda process meets principles. It examines a pattern to make sure an identified and predictable outcome. An illustration of procedure testing is the configuration oriented approach integration scan. System testing is based on approach descriptions and flows, emphasizing pre-driven system links and integration aspects.

White Box Testing:

This testing is a trying out wherein the application tester has competencies of the interior workings, constitution and software language, or at least its cause. It's rational. It's used to test areas that can't be reached from a black box stage.

Black Box Testing:

This is testing the software with none advantage of the inside workings, establishment or words of the unit life form veteran. Black field checks, as most other sorts.

LEVELS OF TESTING :

Unit testing strategy

Unit checking out is most commonly performed as a part of a mixed code and unit experiment part of the software lifecycle, though it is not exceptional for coding and unit checking to be performed as two targeted phases.

Test strategy and approach:

Field testing can be carried out manually and sensible assessments shall be written in element.

Test objectives

Each field must be worked correctly.

Each page must be activated through the specified link.

Features to be tested Verify that the entries are of the correct format No duplicate entries should be allowed

Integration testing strategy

Software integration testing is the incremental integration checking out of two otherwise further included software gears on top of a solo stage to fabricate failure induced with the aid of interface defects. The project of the mixing scan is to check that components or program applications, e.g., Components in a program approach or œ one step up œ software purposes at the company degree œ interact without error.

Test Results:

All of the scan circumstances recounted above passed efficiently. No defects encountered.

Acceptance Testing

User Acceptance testing trying out is a crucial section of any mission and requires enormous participation by the tip user. It additionally ensures that the procedure meets the functional specifications.

Test Results:

The entire test cases recounted above passed effectively. No defects Encountered .

PRECAUTIONS

The Following points are some precautions to protect from attacks through bluetooth connections

1. Make a connection in a private place when connecting to BLE devices.

When a connection is established in a private place there is no scope of establishing a new connection to our smart watch.

2. Enable Bluetooth only when you need it.

Turning off Bluetooth when not required will avoid unnecessary troubles of losing sensitive data.

(can not be done in all BLE devices, but if some devices have this feature)

3. Make every connection slot occupied to our BLE devices.

Every BLE smart watch device has multiple connections to pair with different devices simultaneously, making all connections occupied to BLE devices to avoid open connections issues and establish unauthorized accesses.

4.Synchronize the smart watch application with mail to store the sensitive data.

When we synchronize the smart watch application in mobile with mail ids then when another unknown device or even a trusted second device connected to smart watch cannot access sensitive data until the mail was synchronized.

Extra tips for all other Bluetooth enabled devices

- 1.Reject all unexpected pairing requests
- 2.Update your mobile phone firmware to the latest version.
3. Keep the device in non-discoverable (hidden) mode

Conclusion

BLE is one of the most utilized technologies in the IoT sector throughout the world due to its power efficiency and reliable data transfer. In the present world comfort comes along with their own risks which leads to vulnerable threats in pairing and communication with BLE devices.

In our case study we observed a data leakage in smart watch because of open Bluetooth connection in a simple way of experiment

We are not able to provide any security measures with our knowledge on networking but tried to solve as much as we can, we worked in some areas and failed to establish a connection with BLE smart watches.

So, we are providing precautions to avoid the open Bluetooth connection threats and acknowledge users to make a secure way of using BLE devices.

Future Enhancement

With a goal of providing security to BLE devices specially on smart watches we worked on some of the modules which support Bluetooth function to work with, But cannot proceed further as we don't have active maintenance of Bluetooth supported modules.

We provided a way of using Bluetooth with precautions to secure our devices.

But every user who wants to use the available technology of Bluetooth should follow the safety precautions.

In order to make the users of BLE devices show no concern about worrying the threats with BLE devices, our work may be helpful to do further work to provide security.

REFERENCES

1. A survey on security threats and vulnerability attacks on bluetooth communication” Trishna Panse, Prashant Panse International Journal of Computer Science and Information Technologies 4 (5), 741-746, 2013
2. “A study on vulnerabilities and threats to wearable devices” Felton Blow, Yen-Hung Hu, Mary Hoppa Journal of The Colloquium for Information Systems Security Education 7 (1), 7-7, 2020
3. “Bluetooth security” Juha T Vainio Proceedings of Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, Seminar on Internetworking: Ad Hoc Networking, Spring 5, 2000
4. A. Barua, M. A. Al Alamin, M. S. Hossain and E. Hossain, "Security and Privacy Threats for Bluetooth Low Energy in IoT and Wearable Devices: A Comprehensive Survey," in *IEEE Open Journal of the Communications Society*, vol. 3, pp. 251-281, 2022, doi: 10.1109/OJCOMS.2022.3149732
5. WEARABLE TECHNOLOGY DEVICES SECURITY AND PRIVACY VULNERABILITY ANALYSIS Ke Wan Ching and Manmeet Mahinderjit Singh School of Computer Sciences,Universiti Sains MalaysiaPenang, Malaysia
6. An Information Security Awareness Model of Bluetooth Attacks on Smartwatches: A Case of a Kenyan University
7. How secure your smart watch