MINISTRY OF ELECTRONIC AND INFORMATION TECHNOLOGY (MEITY)

BLUETOOTH WIRELESS SECURITY

UNDER GUIDANCE OF: MR. SUSHIL NEHRA
MR. RANJAN KUMAR

SUBMITTED BY : AKSHAT UPADHYAY

Index

Abstract

Bluetooth technology has become an integral part of this modern society. The availability of mobile phones, game controllers, Personal Digital Assistant (PDA) and personal computers has made Bluetooth a popular technology for short range wireless communication. It is an open standard for short-range radio frequency communication. Bluetooth technology is used primarily to establish wireless personal area networks (WPANs), and it has been integrated into many types of business and consumer devices. This publication provides information on the security capabilities of Bluetooth technologies and gives recommendations to organisations employing Bluetooth technologies on securing them effectively. However, as the Bluetooth technology becomes widespread, vulnerabilities in its security protocols are increasing which can be potentially dangerous to the privacy of a user's personal information. The security issues of bluetooth have been an active area of research for the last few years. This report presents the vulnerabilities in the security protocols of this technology along with some past security threats and possible countermeasures as reported in the literatures which have been surveyed and summarised in this report. It will also presents some tips that end-users can implement immediately to become more cautious about their private information. Finally, the report concludes with some recommendations for future security enhancements that can be implemented in the Bluetooth standard.

Introduction

Bluetooth technology has been considered as a cheap, reliable, and power efficient replacement of cables for connecting electronic devices. This technology was officially approved in the summer of 1999 .Bluetooth is a combination of hardware and software technology. The hardware is riding on a radio chip. On the other hand, the main control and security protocols have been implemented in the software. By using both hardware and software Bluetooth has become a smart technology for efficient and flexible wireless communication system. Bluetooth radio chip supports communication among a group of electronic devices. Once the hardware radio chips are installed into the International Journal of Distributed and Parallel Systems (IJDPS) Vol.3, No.128 electronic devices, wireless communication can be established among these devices. The operating distance between two Bluetooth devices ranges from 10 and 100 meters. By using a directional antenna and an amplifier the range of Bluetooth can be extended over a mile away. One of the major advantages of Bluetooth technology is that it operates in a license-free Industrial, Scientific and Medical (ISM) band ranging from 2.4 to 2.4835 MHz. This band is divided into 79 channels each being 1MHz wide. Each Bluetooth chip has a unique identity code. The 'master-slave' concept is the core of a Bluetooth based network. The 'master' works as the moderator during the communication between itself and the slave as well as among the slaves themselves. In Bluetooth a trusted relationship between two devices called 'pairing' are formed by exchanging shared secret codes referred to as PINs. A 'master' device has the option of pairing with up to seven 'slave' devices establishing a network called a piconet. Two or more piconets together form a scatternet, which can be used to eliminate Bluetooth range restrictions. A scatternet is formed when the devices act as 'master' or 'slave' devices in multiple piconets at the same time. A more detail description of Bluetooth technology can be found in .

1. Need of Project:

The need of the project is to provide more secure Bluetooth connectivity since Bluetooth connections are not considered as a secure connection but still used in most of the mobile applications.

2. Intent of Project:

The intention of Project is to provide a secure and much transparent environment and Bluetooth Connections, so that users could understand and visualise in a better way the working of Bluetooth .

3. Work Plan

Work plan involves the following steps:

- Information gathering (get as much as information on relevant and relatable topics)
- Analysis of published worked
- Understanding of background and internal working
- Code writing
- Code implementations
- Conclusion

How Bluetooth Works

The Bluetooth standard, like WiFi, uses the FHSS technique (Frequency-Hopping Spread Spectrum), which involves splitting the frequency band of 2.402-2.480 GHz into 79 channels (called hops), each 1MHz wide. Then it transmits the signal using a sequence of channels known to both the sending and receiving stations. Thus, by switching channels as often as 1600 times a second, the Bluetooth standard can avoid interference with other radio signals.

Bluetooth sends and receives radio waves in a band of 79 different frequencies (channels) centred on 2.45 GHz, set apart from radio, television, and cellphones, and reserved for use by industrial, scientific, and medical gadgets. Don't worry: you're not going to interfere with someone's life-support machine by using Bluetooth in your home, because the low power of your transmitters won't carry your signals that far! Bluetooth's short-range transmitters are one of its biggest plus points. They use virtually no power and, because they don't travel far, are theoretically more secure than wireless networks that operate over longer ranges, such as Wi-Fi. (In practice, there are some security concerns.)

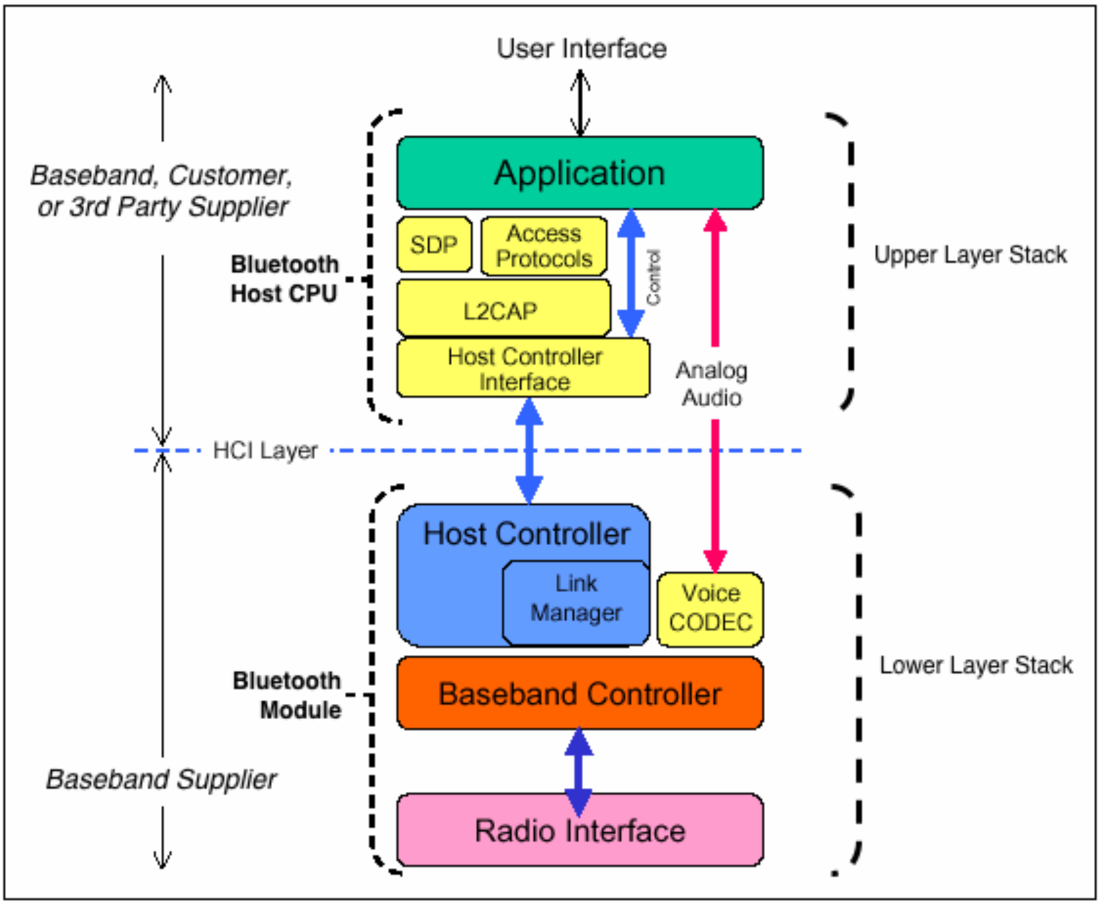
Bluetooth devices automatically detect and connect to one another and up to eight of them can communicate at any one time. They don't interfere with one another because each pair of devices uses a different one of the 79 available channels. If two devices want to talk, they pick a channel randomly and, if that's already taken, randomly switch to one of the others (a technique known as spread-spectrum frequency hopping). To minimise the risks of interference from other electrical appliances (and also to improve security), pairs of devices constantly shift the frequency they are using—thousands of times a second.

When a group of two or more Bluetooth devices are sharing information together, they form a kind of ad-hoc, mini computer network called a piconet. Other devices can join or leave an existing piconet at any time. One device (known as the master) acts as the overall controller of the network, while the others (known as slaves) obey its instructions. Two or more separate piconets can also join up and share information forming what's called a scatternet.

Bluetooth Protocol Stack

Bluetooth network technology connects mobile devices wirelessly over a short-range to form a personal area network (PAN). The Bluetooth architecture has its own independent model with a stack of protocols, instead of following the standard OSI model or TCP/IP model. Another unique feature is that it is not mandatory for all the devices in the Bluetooth system to use all the protocols in the stack. This is because Bluetooth is designed to be used by myriad applications and the application designates which part of the protocol stack is to be used.

The heart of this specification is the protocol stack, which is used to define how Bluetooth works. The Bluetooth protocol stack is a set of layered programs. Each layer in a protocol stack talks to the layer above it and to the layer below it.



Bluetooth protocol stack consists of a three-layer

Lower Stack Layers

The lower layers are the basic core specifications that describe how Bluetooth works. The base of the Bluetooth protocol stack is the radio layer, or module. The radio layer describes the physical characteristics of the transceiver. It is responsible for modulation/demodulation of data for transmitting or receiving over radio frequencies in the 2.4 GHz band. This is the physical wireless connection. It splits the transmission band into 79 channels and performs fast frequency hopping (1600 hops/sec) for security.

Above the radio layer is the baseband and link controller/link manager protocol (LMP). Perhaps the best way to think of these layers is that the baseband is responsible for properly formatting data for transmission to and from the radio. It defines the timing, framing, packets, and flow control on the link. The link manager controller translates the host controller interface (HCI) commands from the upper stack, and establishes and maintains the link. It is responsible for managing the connection, enforcing fairness among slaves in the piconet, and provides for power management.

Upper Stack Layers

The upper stack layers consist of profile specifications that focus on how to build devices that will communicate with each other, using the core technology. The host controller interface (HCI) serves as the interface between the software part of the system and the hardware (i.e., the device driver).The L2CAP (logical link control and adaptation protocol) layer is above the HCI in the upper stack. Among other functions, it plays a central role in communication between the upper and lower layers of the Bluetooth stack. It keeps track of where data packets come from and where they should go. It is a required part of every Bluetooth system.

Above the L2CAP layer, the protocol stack is not as linearly ordered. Still, the service discovery protocol (SDP) is important to mention because it exists independently of other higher-level protocol layers.

It provides the interface to the link controller and allows for interoperability between Bluetooth devices.

Bluetooth Protocol Profiles

A Bluetooth profile is a set of instructions for using the protocol stack in a certain way. Many different profiles exist, depending on the types of devices connected and their purposes. For example, a cell phone might implement the Headset Profile (HSP), while a FAX machine might implement the FAX Profile.

A profile is a complete definition of how a product manufacturer can implement Bluetooth wireless technology for a particular usage – how to accomplish specific tasks. In other words, to use Bluetooth wireless technology, a device must be able to interpret certain Bluetooth profiles, of which there are many, depending on the activity desired. Profiles are definitions of possible applications, and specify the behaviors that Bluetooth-enabled devices use to communicate with each other. A non-technical way to describe profiles is to think of them as settings, or instructions to be followed for communication to occur. All devices in the connection must be compatible with the subset of Bluetooth profiles necessary to use the desired services.
At a minimum, each Bluetooth profile contains information on the following:

- Dependencies on other profiles/protocols
- Suggested user interface formats
- Specific parts of the Bluetooth protocol stack used by the profile.

To perform its task, each Bluetooth profile uses specific options and parameters at each layer of the stack. As an example, suppose the desire is to stream high-quality Audio. To accomplish this, the A2DP (Advanced Audio Distribution Profile) uses the LMP and L2CAP protocols that control the Baseband Radio interface to transmit the desired digital audio stream that would be used. On the receiving end, the process would flow in reverse to yield the desired audio data.

Many Bluetooth protocols are implemented in software – too many and diverse to describe in this post. The L2CAP provides the interface to the link controller, and allows for interoperability between Bluetooth devices.

Examples of a few of the many profiles are:

- A2DP – Advanced Audio Distribution Profile
- AVRCP – Audio/Video Remote Control Profile
- GAVDP – General Audio/Video Distribution Profile
- PAN – Personal Area Networking
- HFP – Hands-Free Profile
- HSP – Headset Profile
- CTP – Cordless Telephony Profile
- VDP – Video Distribution Profile
- FTP – File Transfer Profile
- RFCOMM – Radio Frequency Communications
- TCS – Telephony Control Protocol
- WAP – Wireless Application Protocol
- SDP – Service Discovery Protocol
- TCP/IP – Transmission Control Protocol/Internet Protocol

Protocols in the Bluetooth Protocol Stack

- **Core protocols** – This includes Bluetooth radio, Baseband, Link Manager Protocol (LMP), Logical Link Control and Adaptation Protocol (L2CAP), and Service Discovery Protocol (SDP).

- **Cable Replacement Protocol** – This includes Radio Frequency Communications (RFComm) protocol. It is short for Radio Frontend Component. It provides a serial interface with WAP.

- **Adopted Protocols** – These are the protocols that are adopted from standard models. The commonly adopted protocols used in Bluetooth are Point-to-Point Protocol (PPP), Internet Protocol (IP), User Datagram Protocol (UDP), Transmission Control Protocol (TCP), and Wireless Application Protocol (WAP).

- **AT Commands** – ATtention command set.

Layers in the Bluetooth Protocol Stack are :

1. Radio (RF) layer
2. Baseband Link layer
3. Link Manager protocol layer
4. Logical Link Control and Adaption protocol layer
5. SDP layer
6. RF comm layer
7. OBEX
8. WAP
9. TCS
10. Application layer

Communication Principle

The Bluetooth standard is based upon a master/slave operational mode. The term piconet is used to refer to the network formed by one device and all devices found within its range. Up to 10 piconets can coexist within a single coverage area. A master can simultaneously connect to up to 7 active slave devices (255 when in parked mode). Devices in a piconet have a logical address of 3 bits, for a maximum of 8 devices. Devices in parked mode are synchronised, but do not have their own physical address in the piconet. In reality, at a given moment, the master device can only be connected to a single slave at once. Therefore, it quickly switches between slaves in order to make it seem as if it is simultaneously connected to all the slave devices. Bluetooth enables two piconets to be linked to one another in order to form a wider network, called a scatternet, using certain devices which act as a bridge between the two piconets.

Piconet :

It is a type of bluetooth network that contains one primary node called master node and seven active secondary nodes called slave nodes. Thus, we can say that there are total of 8 active nodes which are present at a distance of 10 meters. The communication between the primary and secondary node can be one-to-one or one-to-many. Possible communication is only between the master and slave; Slave-slave communication is not possible. It also have 255 parked nodes, these are secondary nodes and cannot take participation in communication unless it get converted to the active state.

Scatternet:

It is formed by using various piconets. A slave that is present in one piconet can be act as master or we can say primary in other piconet. This kind of node can receive message from master in one piconet and deliver the message to its slave into the other piconet where it is acting as a slave. This type of node is refer as bridge node. A station cannot be master in two piconets.

Establishing connections

Establishing a connection begins with a phase called inquiry, during which the master device sends an inquiry request to all devices found within its range, called access points. All devices that receive the query reply with their address. Then, the master device chooses an address and synchronises with the access point using a technique called paging.This primarily involves synchronising its clock and frequency with the access point.
After, a link with the access point is established, allowing the master device to enter an access point service discovery phase, using a protocol called Service Discovery Protocol (SDP).

At the end of this service discovery phase, the master device is ready to create a communication channel with the access point, using the protocol L2CAP.
Depending on the service's needs, an additional channel (called RFCOMM and operating over the L2CAP channel) may be established in order to provide a virtual serial port.

Some applications have been designed to connect to a standard port, independent of the hardware used. For example, certain highway navigation programs have been designed to connect to any GPS Bluetooth device. Today, there are also more and more bluetooth headphones.

The access point may include a security mechanism called pairing that restricts the access to authorised users only, in order to give the piconet a certain measure of protection. Pairing is done with an encryption key commonly known as a PIN (Personal Information Number). To do so, the access point sends a pairing request to the master device. Most of the time, this may prompt the user to enter the access point's PIN. If the PIN received is correct, the connection is made.

In secure mode, the PIN will be sent encrypted, using a second key, in order to prevent the signal from being compromised.

When the pairing becomes active, the master device is free to use the communication channel thereby established.

EXISTING REPORTS OF BLUETOOTH THREATS

The problems regarding Bluetooth security have been reported since its inception. But, it has not been considered as a significant problem until its adaptation into mobile devices. A brief overview of some of the real incidents is listed below:

- In 2004, the first Bluetooth virus was reported in the literatures as a 'proof-of-concept'. It was proved as a potential threat to the Bluetooth technology .
- In January 2005, a mobile malware called 'Lasco' was detected. Lasco was a selfreplicating worm, which was successful in rendering a mobile device unstable before infecting another device .
- In August 2005, Bluetooth enabled phones were used to track other mobile device left inside of cars .
- In October 2007, Kevin Finistere and Thierry Zoller demonstrated the first Bluetooth and link key cracking technique at a conference. A remote root shell via Bluetooth on Mac OS X v10.3.9 and v10.4 was used in that demonstration.

BLUETOOTH NETWORK VULNERABILITIES

Bluetooth devices are exposed to malicious intervention during the process of pairing with another device. These weaknesses are primarily due to flaws in the link key establishment protocol, which is required for devices to pair, and the fact that the encryption of a session is optional and created at the end of the pairing process. It means that the various types of attacks can be performed well before pairing is complete. Even after the pairing is complete, the attackers can still sniff the airwaves to gain enough information to steal link keys so that they can deceptively authenticate or perform Man-in-the-Middle (MITM) attacks to impersonate other devices. Some of the attacks are-

1. Blue-jacking:

Blue-jacking is used for sending unauthorised 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 blue-jacker and his Bluetooth enabled in his device, does blue-jacking happen. Blue-jacking involves sending unsolicited business cards, messages, or pictures. The blue-jacker 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.

The messages are anonymous to the recipient as only the mobile name and model number of the blue-jacker'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.

Steps To Blue-jack a Device

- Blue-jacker 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.

2. Blue-bugging

An extraordinarily powerful attack mechanism, blue-bugging allows an attacker to take control of a victim's phone using the AT command parser. Blue-bug allows an attacker to access a victim's phone in order to make phone calls, send short message service (SMS) messages, read SMS messages stored on the phone, read and write contact list entries, alter phone service parameters, connect to the Internet, set call forwarding, and more.

It is a hacking technique that allows individuals to access a device with a discoverable Bluetooth connection. Once the target device accesses a rigged link, the attacker can take full control of it. The hacker can read and send messages, access the victim’s phonebook, and initiate or eavesdrop on phone calls. Initially, blue-bugging focused on eavesdropping or bugging a computer with Bluetooth capability. With the increasing use of smartphones, cybercriminals shifted to hacking mobile phones. This attack is often limited due to the range of Bluetooth connections, which goes up to only 10 meters. Some attackers use booster antennas to widen their attack range.
It’s not much different from bugging a landline phone, except it can be done without gaining access to the physical device.

3. Blue-Snarfing

Attacks allowing for the theft of information from a Bluetooth device using the OBEX Push Profile. The attacker needs only find a phone that has Bluetooth in discoverable mode. Blue-snarf works by a connection to most of the Object Push Profile services and the attacker retrieves the file names of known files from the Infrared Mobile Communications (IrMC) list instead of sending vCard information as expected. With these attacks the hacker can retrieve items such as the phonebook, calendar, and other personal information. With Blue-snarf++, the attacker has full read and write access to the file system of the phone. When an attacker is connected via the OBEX Push Profile, he/she has full access to the victim's phone without having to pair the two devices. The biggest risk with this function is that an attacker can delete crucial file system files, rendering the victim's device useless. In addition, the attacker can access any memory cards that are attached to the device.

Blue-snarfing is information theft that occurs over a Bluetooth connection. Bluetooth is a wireless connection technology that provides high-speed access between various devices over short distances. Theft of this type exploits a weakness in some mobile Bluetooth implementations and allows unauthorised access to personal information.

4. Blue-Smacking

Blue-smacking is a cyber attack done on bluetooth enabled devices. 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.

Procedure For The Attack

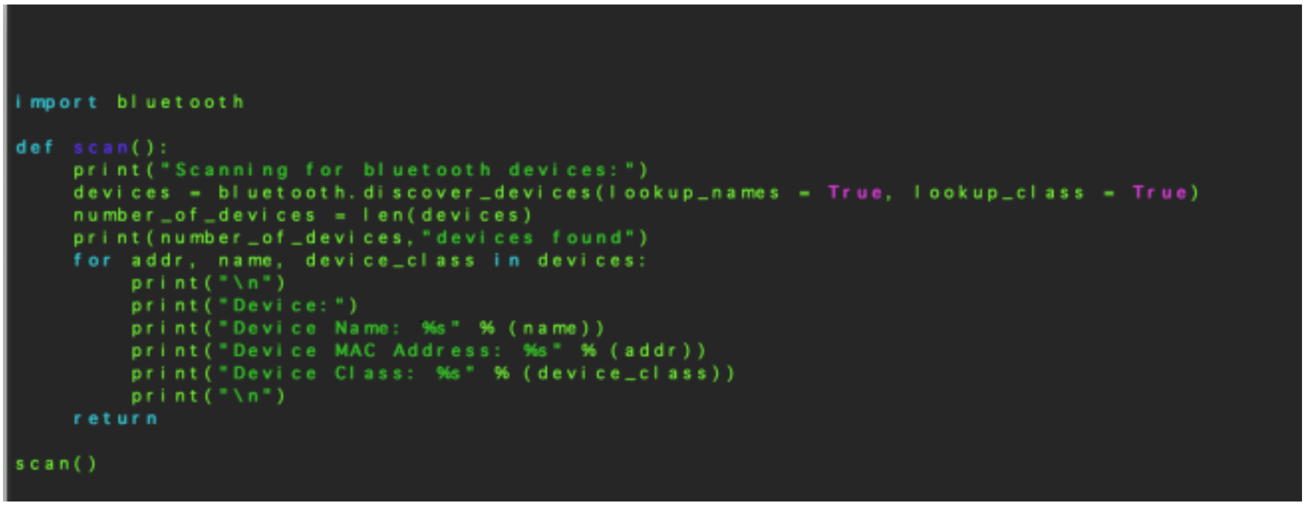
- The hacker first uses the standard tools such as l2ping that come with Linux Bluex utils package.
- The l2ping tool further allows a hacker to specify the packet length with some commands. Due to this, the Bluetooth enabled devices are overwhelmed by the malicious requests from the hacker, causing the device to be inoperable by the victim.
- The attack atlast affects the regular operation of the victim device and can even degrades the performance of the device.

Software/Tools Used

- **BlueSpam**
- **Meeting Point**
- **Freejack**
- **Easyjacking (ejack)**
- **Super Bluetooth Hack 1.08**
- **Blue Scanner**
- **Blue Sniff**
- **BlueBugger**
- **BTBrowser**
- **BTCrawler**
- **BlueSnarfing**
- **l2ping**

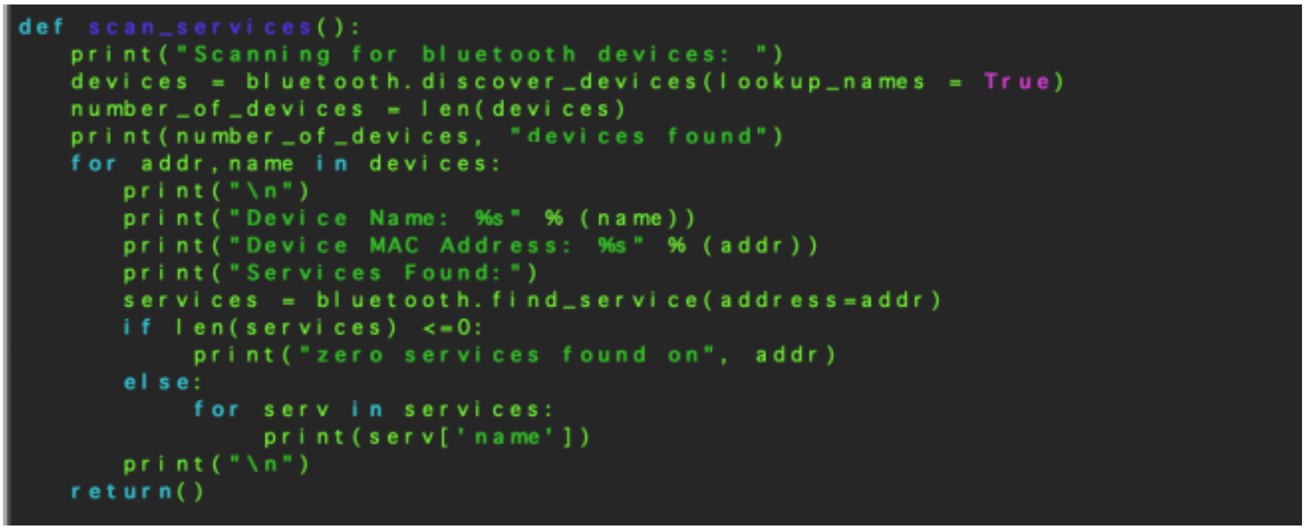
Code

```
import bluetooth
def scan():
    print("Scanning for bluetooth devices:")
    devices = bluetooth.discover_devices(lookup_names = True, lookup_class = True)
    number_of_devices = len(devices)
    print(number_of_devices,"devices found")
    for addr, name, device_class in devices:
        print("\n")
        print("Device:")
        print("Device Name: %s" % (name))
        print("Device MAC Address: %s" % (addr)
        print("Device Class: %s" % (device_class))
        print("\n")
    return()
scan()
```



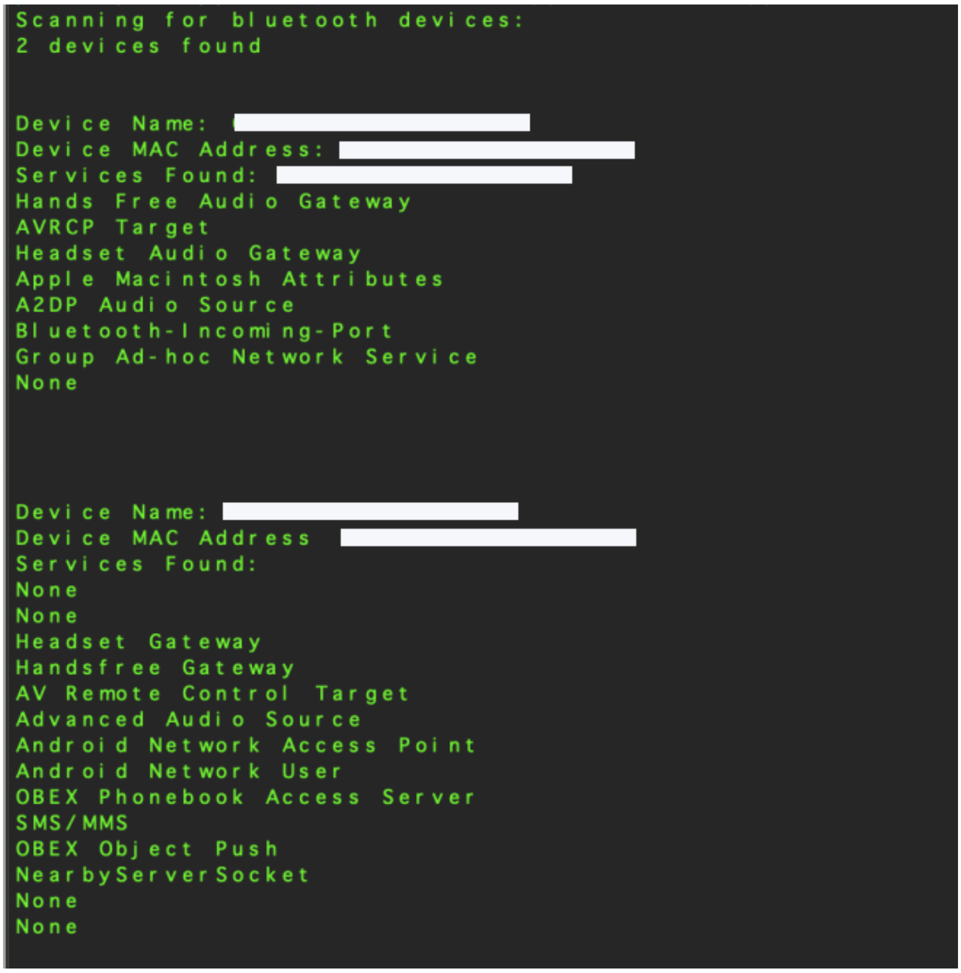
```
import bluetooth

def scan():
    print("Scanning for bluetooth devices:")
    devices = bluetooth.discover_devices(lookup_names = True, lookup_class = True)
    number_of_devices = len(devices)
    print(number_of_devices,"devices found")
    for addr, name, device_class in devices:
        print("\n")
        print("Device:")
        print("Device Name: %s" % (name))
        print("Device MAC Address: %s" % (addr))
        print("Device Class: %s" % (device_class))
        print("\n")
    return

scan()
```

```
def scan_services():
  print("Scanning for bluetooth devices: ")
  devices = bluetooth.discover_devices(lookup_names = True)
  number_of_devices = len(devices)
  print(number_of_devices, "devices found")
  for addr,name in devices:
    print("\n")
    print("Device Name: %s" % (name))
    print("Device MAC Address: %s" % (addr))
    print("Services Found:")
    services = bluetooth.find_service(address=addr)
    if len(services) <=0:
       print("zero services found on", addr)
    else:
       for serv in services:
          print(serv['name'])
    print("\n")
  return()
```



```
def scan_services():
    print("Scanning for bluetooth devices: ")
    devices = bluetooth.discover_devices(lookup_names = True)
    number_of_devices = len(devices)
    print(number_of_devices, "devices found")
    for addr,name in devices:
        print("\n")
        print("Device Name: %s" % (name))
        print("Device MAC Address: %s" % (addr))
        print("Services Found:")
        services = bluetooth.find_service(address=addr)
        if len(services) <=0:
            print("zero services found on", addr)
        else:
            for serv in services:
                print(serv['name'])
        print("\n")
    return()
```

Implementation

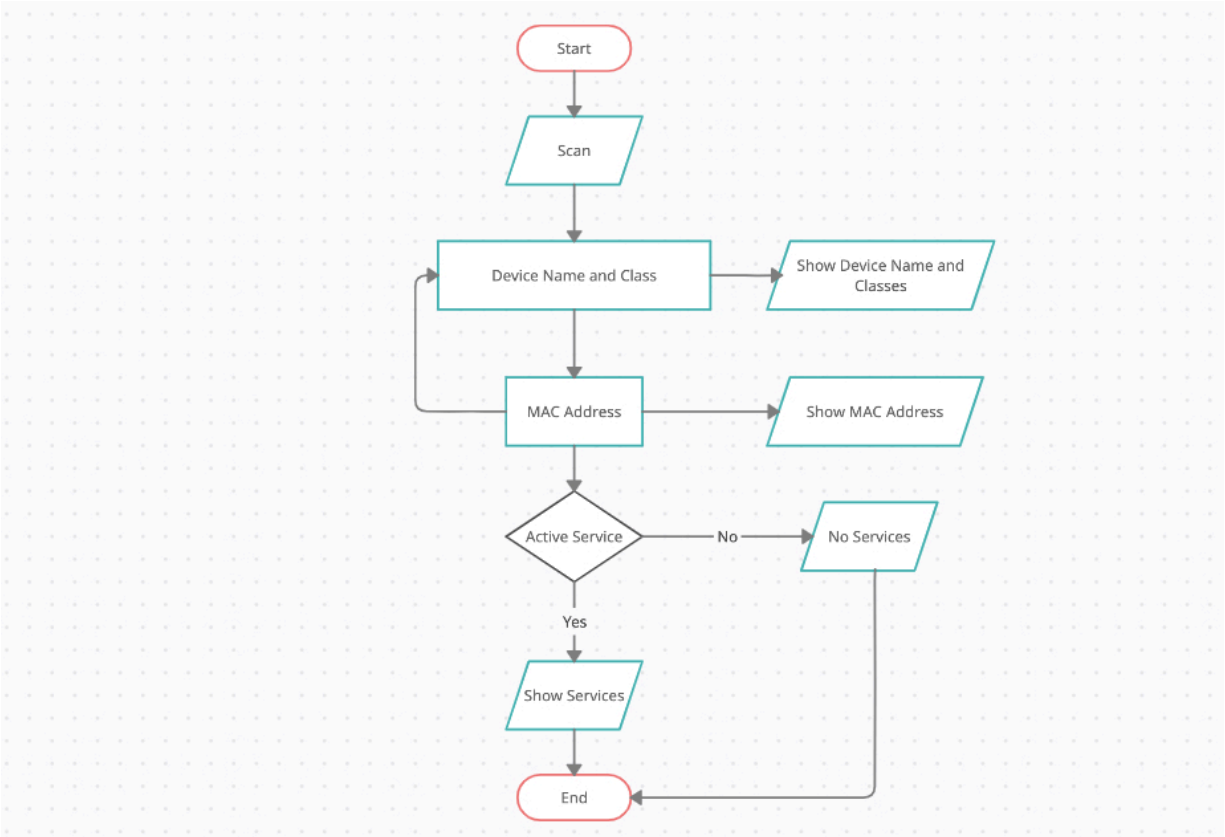


```
Scanning for bluetooth devices:
2 devices found


Device Name: [redacted]
Device MAC Address: [redacted]
Services Found: [redacted]
Hands Free Audio Gateway
AVRCP Target
Headset Audio Gateway
Apple Macintosh Attributes
A2DP Audio Source
Bluetooth-Incoming-Port
Group Ad-hoc Network Service
None



Device Name: [redacted]
Device MAC Address [redacted]
Services Found:
None
None
Headset Gateway
Handsfree Gateway
AV Remote Control Target
Advanced Audio Source
Android Network Access Point
Android Network User
OBEX Phonebook Access Server
SMS/MMS
OBEX Object Push
NearbyServerSocket
None
None
```

Code Explanation

The code imports the bluetooth library and scan for nearby devices using scan function which finds their names and class and finds the number of devices active. After exploring all the devices the program access the loop of all the active addresses for their MAC address and class. As soon as it gets the relevant address it returns the informations.
The second function work similar with that of first one but in addition it returns the information of active services on specific addresses.
If no services are running it simply returns “No service found”, else it returns the services.

Flowchart



Conclusion

Bluetooth devices are exposed to malicious intervention during the process of pairing with another device. These weaknesses are primarily due to flaws in the link key establishment protocol, which is required for devices to pair and hence are key vulnerabilities in the process. Keeping the pairing process transparent and more understandable to the audience is the key of reducing the terror of compromising the bluetooth connections and devices. Understanding the services and its functioning can also reduce improper authentication and can prevent intended and unintended compromising of the device and connections .

Reference

- https://www.researchgate.net/publication/314233155_Bluejacking_Technology_A_Review

- https://www.govinfo.gov/content/pkg/GOVPUB-C13-c528fe2437b557e63cc73e6b431be093/pdf/GOVPUB-C13-c528fe2437b557e63cc73e6b431be093.pdf

- http://index-of.co.uk/Hacking-Coleccion/Bluetooth%20Security.pdf

- https://ccm.net/contents/69-bluetooth-how-it-works

- https://commons.erau.edu/cgi/viewcontent.cgi?article=1025&context=db-security-studies
- https://hearinghealthmatters.org/waynesworld/2014/bluetooth-101-part-vi/