Buzzing Smart Devices

Smart Watch Hacking



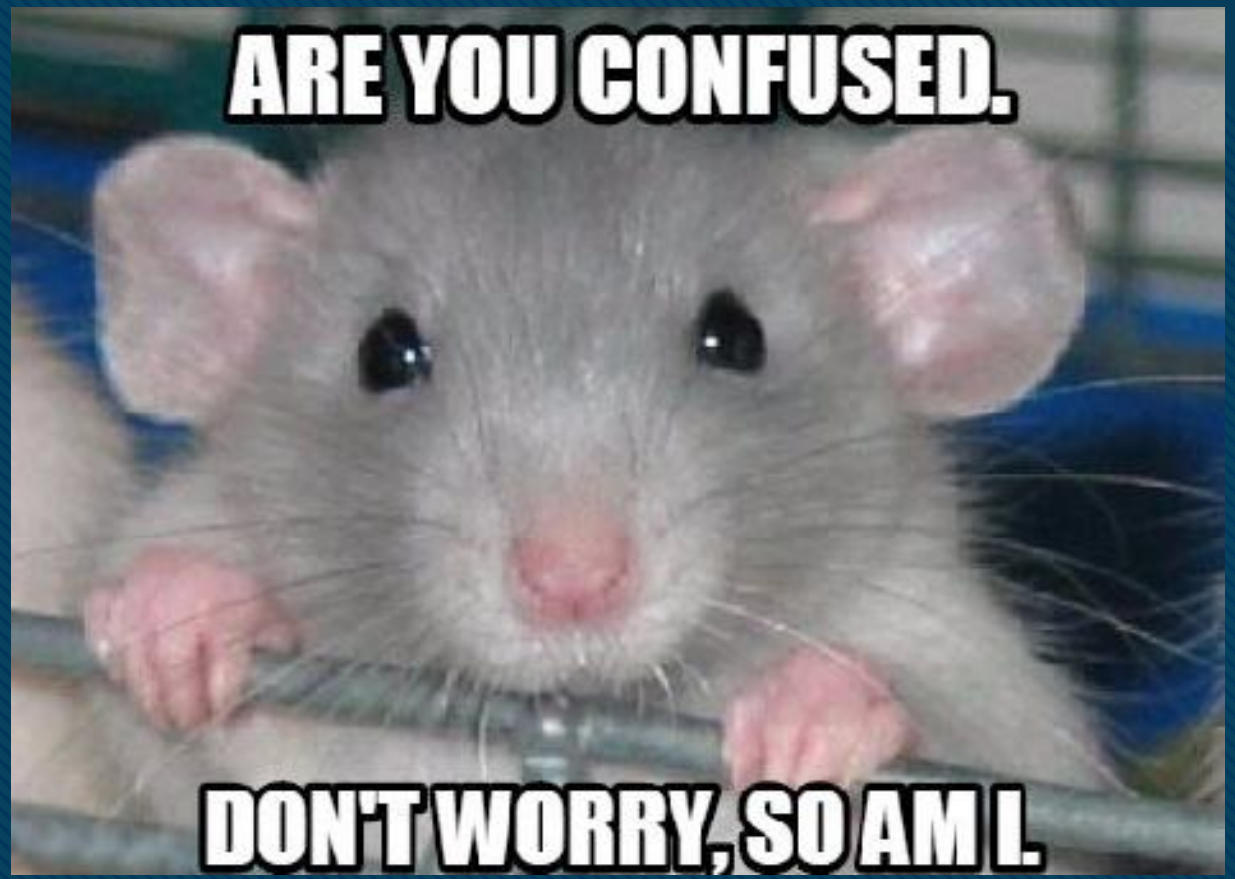
I Have A Question. “?”

Why CATS Everywhere??????



Cats are Evil

Era of Mouse



Source: memegenerator

About me

- Anti-Cat Person
- Security Researcher at Payatu Software Labs
- Hardware Maker
- Intel Software Innovator
- Occasional Artist

Agenda

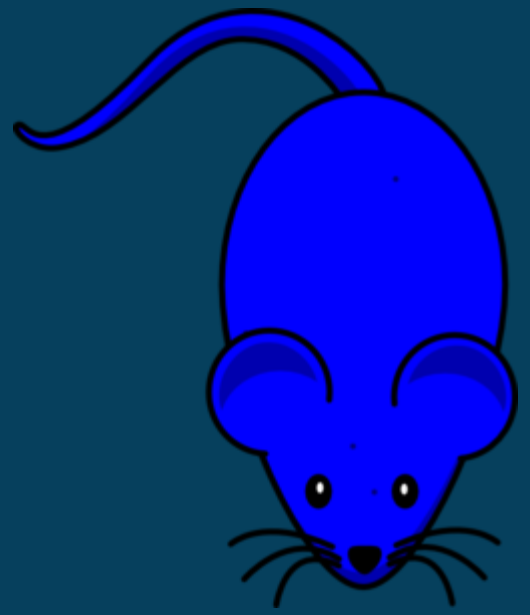
- Introduction to Bluetooth
- Bluetooth Attacks
  - Passive and Active Analysis
  - Man-In-The-Middle
  - Fuzzing
- Mobile Attacks
  - Static Analysis
  - Active Analysis
- Hardware Attacks
  - Hardware Analysis
  - Debug interface
  - Flash protection bypass

Bluetooth Low Energy

Payatu Software Labs

What is Bluetooth Low Energy:

- Bluetooth is a most commonly used wireless communication protocol, especially in mobile phones, smart devices and many more
- As the name says, It consumes Low power and resource for it’s operations.
- It is completely different from your traditional Bluetooth Classic.
- Since it is Low power, devices can operate without battery for years and forever with energy harvesters
- Since most devices has Bluetooth, It is Interoperable.



Specifications of Bluetooth LE

- Low latency connection (3ms)
- Low power (15ma peak transmit, 1uA sleep)
- Designed to send small packet of data (opposed to streaming)
- 128bit AES CCM
- GFSK @ 2.4 GHz
- Adaptive Frequency Hopping
- 37+3 Data + Broadcast Channels

Bluetooth LE Stack

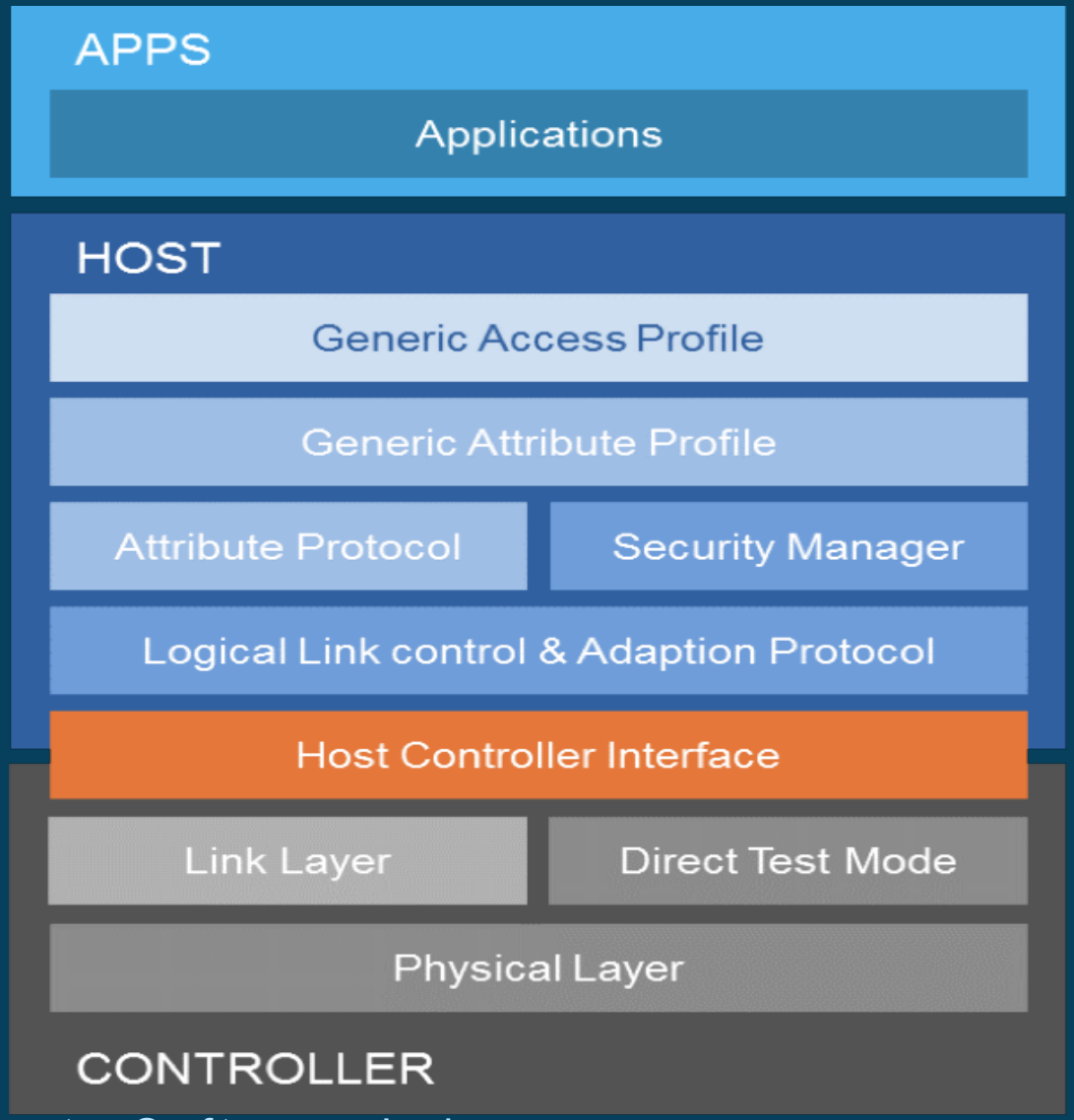


Image Source: Bluetooth Specification

How Bluetooth LE connects?

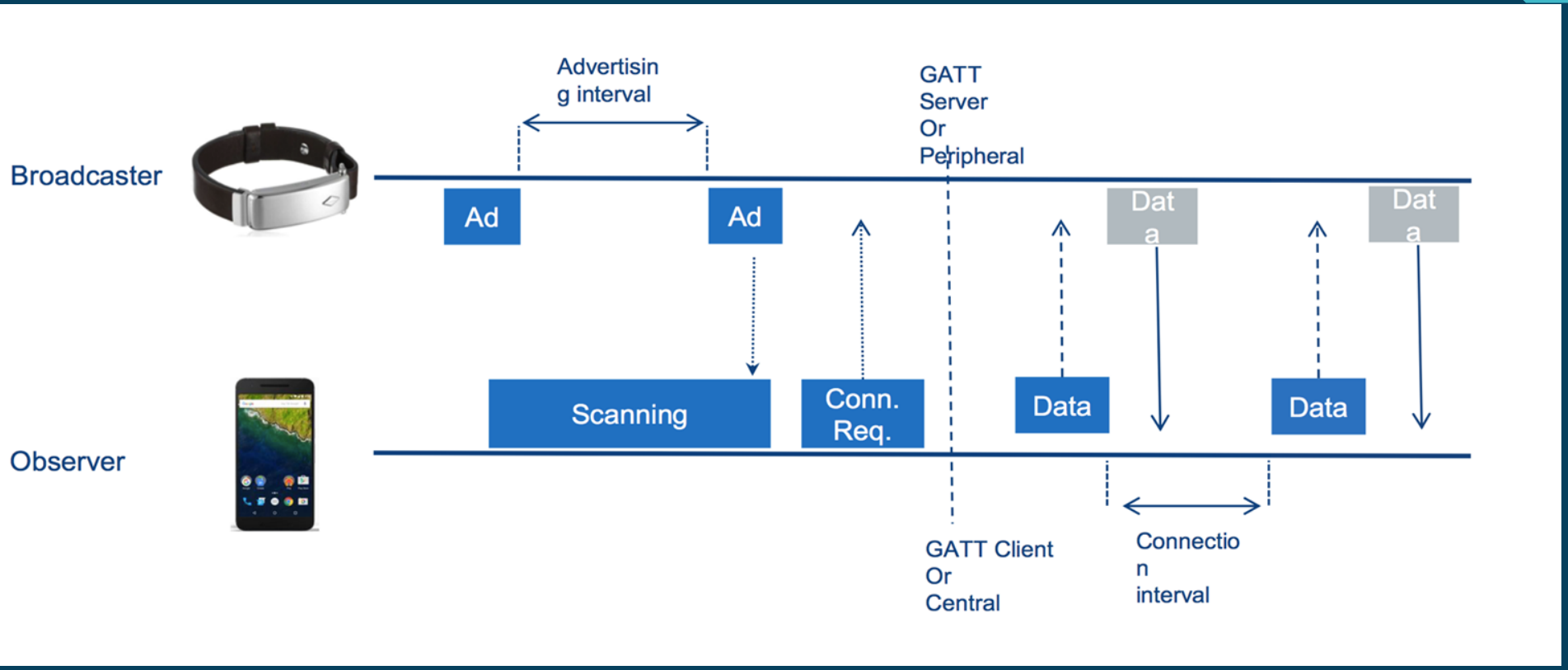


Image source: When Encryption is Not Enough

Roles in Bluetooth LE

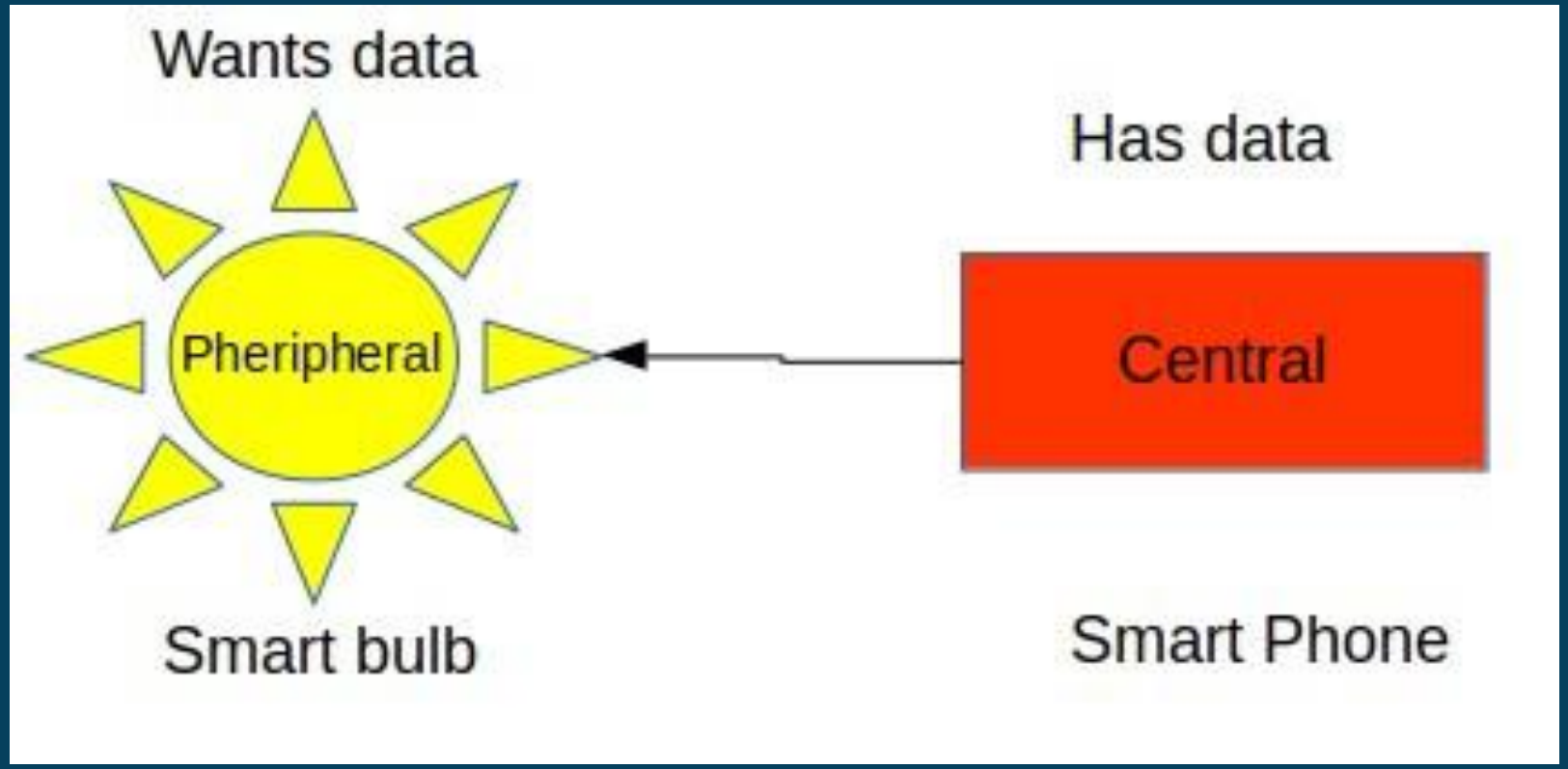


Image source: When Encryption is Not Enough

GAP and GATT

Payatu Software Labs

Generic Access Profile

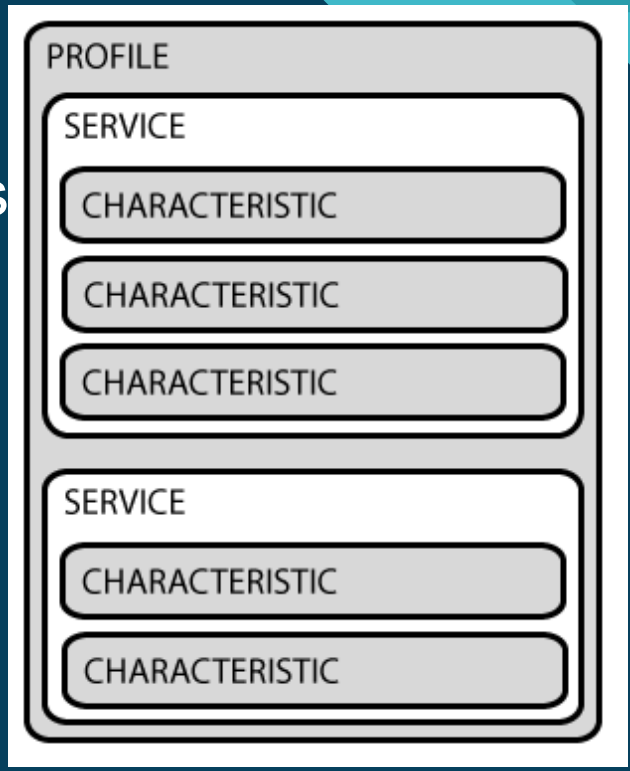
- GAP controls connections and advertising
- Two ways to send advertising
  - Advertising Data Payload – Mandatory and periodically transmitted out by peripheral
  - Scan Response Payload – An optional secondary payload that can be requested by the central. It usually contains little more data than advertising packet
- The peripheral stops advertising as and when the connection is established

Generic Attribute Profile

- GATT
- Defines the communication semantics between the client and the server
- Comes into play when the connection is established
- Uses concepts called Profiles, Services and Characteristics
- Uses ATT (Attribute) Protocol
  - Stores services, characteristics in lookup table using 16-bit IDs for each entry

Profiles, Services and Characteristics

- Profile – Predefined collection of services compiled by either
  - Bluetooth SIG
  - Peripheral designers
- Service – May contain one or more characteristics
  - Used to break up data in different entities
  - Identified by 16-bit or 128-bit UUID
- Characteristic – Encapsulates single data point
  - Identified by 16-bit or 128-bit UUID



Previous Attacks

Security Researchers Hacked a Bluetooth-Enabled Butt Plug

The quest for a secure wireless sex toy continues.

SMART HOME

Have a smart lock? Yeah, it can probably be hacked

Two security experts show just how easy it is to hack certain smart locks.

BY **MEGAN WOLLERTON** / AUGUST 9, 2016 3:43 PM PDT



LILY HAY NEWMAN SECURITY 07.19.17 09:00 AM

WATCH HACKERS TAKE OVER A SEGWAY WITH SOMEONE ON IT

Even More Bluetooth Smart Bulb Hacking

Broken Bulb Leads to Reverse Engineering Adventures!

One morning, my Magic Blue bulb fell and broke. Normally, this would be annoying, if not upsetting, but I was happy about it—I've wanted to break it

Bluetooth LE Security

Payatu Software Labs

Security in LE

- Security of Bluetooth LE resides in GAT and GAP layer of the protocol
- Security is enclosed in these three methods
  - **Connecting**
  - **Pairing**
  - **Bonding**

Security in LE

Connecting is the act of establishing a communication link. No pairing or bonding is required to communicate over Bluetooth LE

Security in LE

Pairing is the act of exchanging keys after connection, typically to set up and maintain an encrypted connection

Security in LE

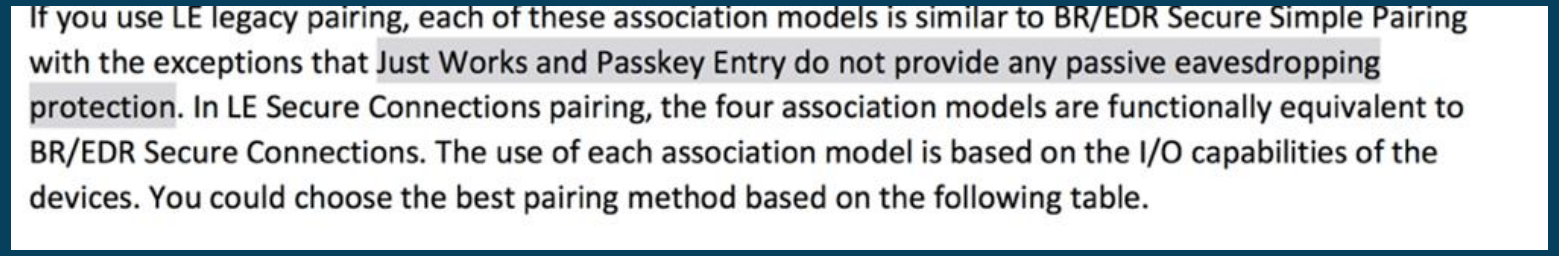
Bonding is the act of storing the exchanged keys after pairing, typically to re-establish an encrypted connection without needing to exchange these keys again.

Connect in LE

- Most smart devices works till connecting and doesn’t provide pairing/bonding because of so many external reasons.
- Like a Light bulb cannot have a keypad or display to enter keys.

Pairing in Bluetooth LE

- These are different methods by which pairing can be estabilished
  - Just Works (000000)
  - Numeric Comparison (Yes/No)
  - Passkey (6-digit)
  - Out-of-Band



If you use LE legacy pairing, each of these association models is similar to BR/EDR Secure Simple Pairing with the exceptions that Just Works and Passkey Entry do not provide any passive eavesdropping protection. In LE Secure Connections pairing, the four association models are functionally equivalent to BR/EDR Secure Connections. The use of each association model is based on the I/O capabilities of the devices. You could choose the best pairing method based on the following table.

Image Source: Bluetooth Specifications

Key Exchange in Bluetooth LE

- Pairing involves a series of key exchanges to encrypt the pairing process and ultimately all communication. The keys include:
  - **Short Term Key (STK),** used to initially encrypt the connection for further key exchange. An algorithm with numerous inputs, including a 128-bit *TK,* is used to generate the same STK on both central and peripheral devices; the STK in its entirety is never exchanged.
  - **Long Term Key (LTK),** used to encrypt the connection after pairing, and is stored to encrypt all future connections between bonded devices. It is only exchanged after initial encryption using the STK.

Tools of Trade

Ubertooth One

- Bluetooth Sniffer and injector
- 2.4 GHz transmit and receive capabilities
- Open source
- By Great Scott Gadgets
- Easily integrates with Wireshark

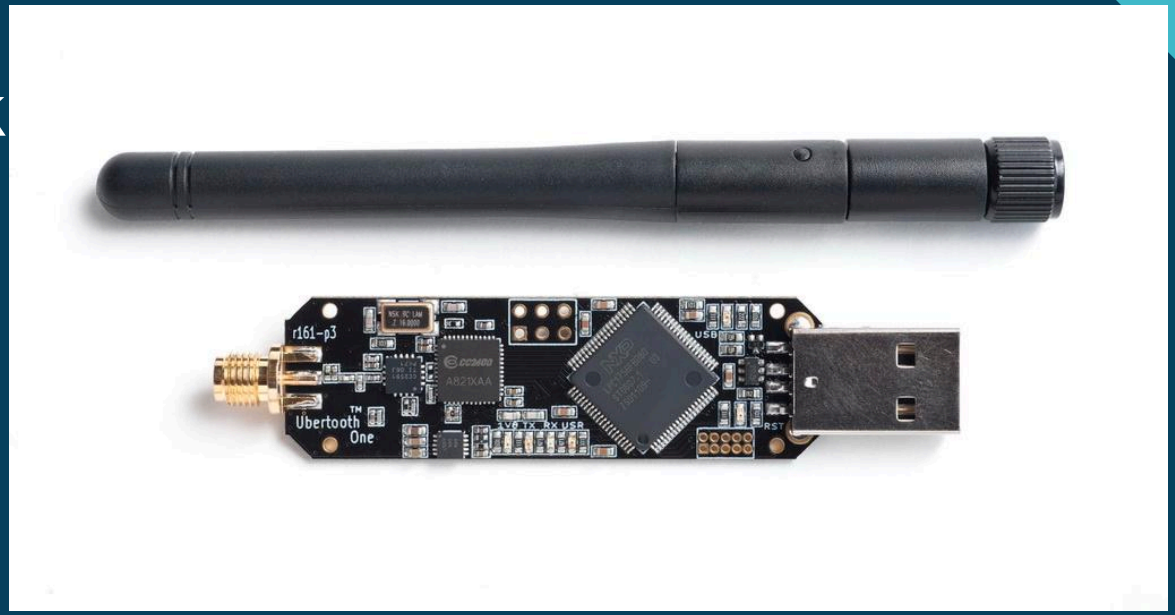


Image Source: ////////////////

CSR 4.0 Dongle

- Bluetooth Adapter
- 2.4 GHz transmit and receive capabilities
- Supports BLE 4.0
- Freely Available on online shopping sites
- Low Cost effective, Comes in 4-8$
- Easily integrates with Wireshark



Image Source: ////////////////

Soft-Tools

- GATTTOOL
- GATTACKER
- BTLEJUICE
- BLUEDIVING
- BLUEAH

Image Source: ////////////////

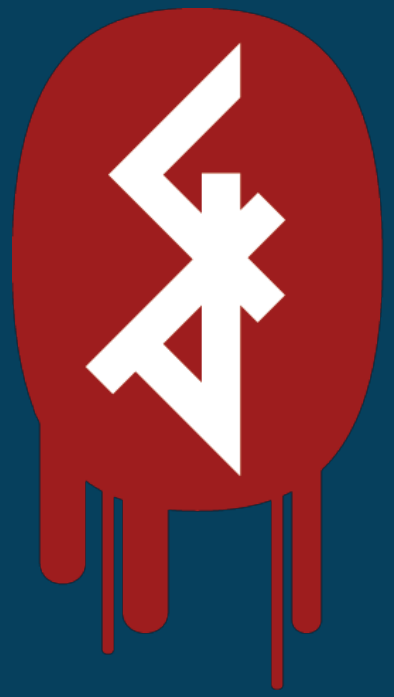
GATTTOOL

- It is a part of bluez framework
- It can connect to a Bluetooth device and can read/write to the characteristics
- It has interactive and non-interactive mode

- Command: gatttool –I –b <MAC>
- Other commands like characteristics, char-read-hnd, char-write-req are available.

GATTACKER

- This is my favorite Bluetooth MiTM tool
- It needs two Bluetooth interface and act as proxy



GATTacker

OUTSMART THE THINGS

Image Source: https://github.com/securing/gattacker

Labs

Bluetooth - Labs

Step 1 : RECON

- Get use to hcitool and hciconfig to gain more information about the device

- Commands:
  - hciconfig hci0 reset
  - hcitool lescan
  - hcitool leinfo <MAC>

Step 2 : Connect

- Use GATTTOOL to connect to the device and understand how to read and write to it

- Commands:
  - gatttool –I –b <mac>
  - characteristics
  - char-read-hnd <handle>
  - char-write-req <handle> <data>

Step 3 : MiTM

- You can use either gattacker or btlejuice
- Configure two machine with Bluetooth Dongle
- Connect your app and the device to the proxy and perform some operation to log the data

Step 4 : Replay

- Now with the data you analyzed, perform a replay/relay attack from gatttool
- Or alternatively you can also use a app called as nrf connect.

Step 5 : Fuzz

- Since you know the format of the data which controls the watch
- Use exploit framework to fuzz the handle and see if you can see any reactions in the watch.
- Find all the vulnerable data.

Mobile App - Labs

Payatu Software Labs

Step 1 : Disassemble

- Use jadx in the VM to disassemble the apk.
- Look for information like characteristics, writedata and other gatt related class
- Understand how data are being transmitted

Step 2 : Active Analysis

- Use things like logcat and internal memory to gain some meaningful information
- Use Frida to hook to the class which is responsible for the GATT write and read operation

Step 3 : Look for Firmware update

- Repeat the previous steps for information related to firmware update and the mechanism on which it is done

Hardware - Demo Labs

Step 1 : Disassemble

- Open the smart watch.
- Identify various parts in it.
- Look for any test pads/pins which is labeled as TX/RX or SWDIO/SWCLK
- It could be labelled in different names too.
- These are the debug and log ports for the watch

Step 2 : Access the debug ports

- Connect the Debug port of the watch to the debugger.
- Try to read the internal memory or read other information of the device.

Step 3 : Extract the firmware

- If the flash memory is locked, search for any exploits. just like you would for a software exploits.
- Run the Flash protection bypass to extract the firmware and analyze

Step 3 : Play with the firmware

- Use visual inspection and figure out how different parts are connected to it.
- Try to port your own code.
- You can make it as a HiD device or program your own watch.

Conclusion

- Start with basic recon on the mobile app or the Bluetooth dump.
- Perform MiTM to understand the data that goes to the device
- Fuzz it
- Open the hardware, check for hardware ports and extract the firmware.

Reference

- https://learn.adafruit.com/introduction-to-bluetooth-low-energy/introduction
- http://research.worksap.com/research/ble-1/
- https://en.wikipedia.org/wiki/Bluetooth_low_energy
- https://developer.bluetooth.org/TechnologyOverview/Pages/Profiles.aspx#GATT
- https://github.com/greatscottgadgets/ubertooth/wiki/Capturing-BLE-in-Wireshark
- http://lacklustre.net/projects/crackle/
- https://github.com/securing/gattacker
- https://github.com/DigitalSecurity/btlejuice



Mouse life matters too”

- Arun Magesh

Thank You

Contact:

arun.m@payatu.com

@marunmagesh