Bluetooth, Suffering, and You
Part I: Into The Bluetooth-verse

Paul A. Wortman, PhD
Mauddib28

January 14, 2025

Table of contents

Who the BLEEP are you?

Yeah, what the title says!

- What will today be about? Bluetooth Biznatches!
- What should I learn? Basics of Bluetooth and Limitations
- Why pay attention? Basically wireless wires and Attacks!

Whoami

- PhD
- Bluetooth Security Researcher
- Research Scientist

Whatisthis?

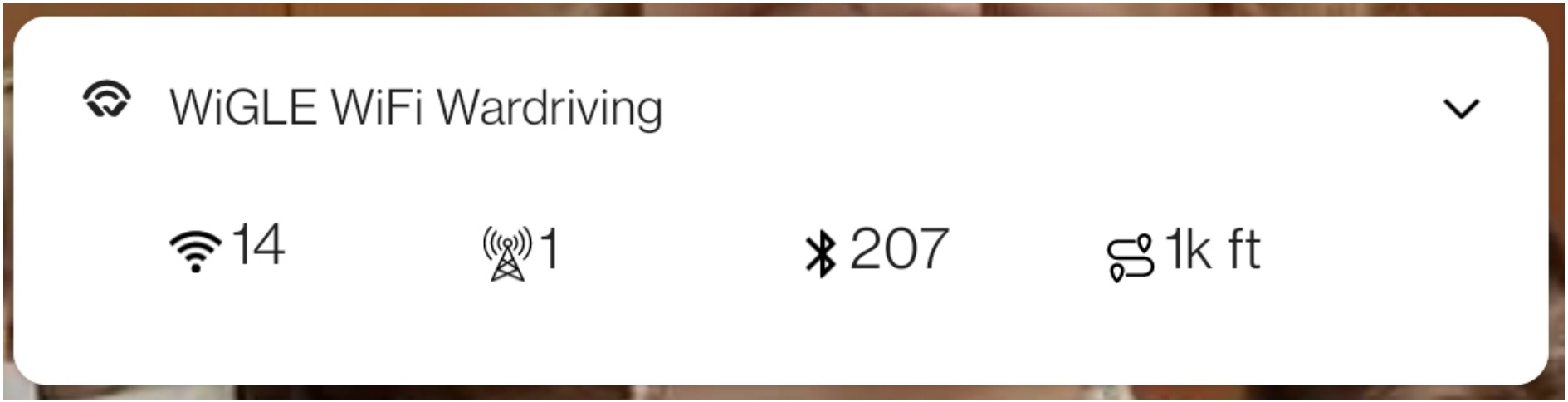


Figure 1: Wiggle War Drive - Landing Strip

“Lightning” Headphones That Require Bluetooth

Josh Whiton:

> A crazy experience — I lost my earbuds in a remote town in Chile, so tried buying a new pair at the airport before flying out. But the new wired, iPhone, lightning-cable headphones didn’t work. Strange.

Why is Bluetooth Important?

- Ubiquity of devices in the wild
- What is Bluetooth purposed for?
  - Replace Wires... Wirelessly...
- Embedded Systems? Internet of Things!
  - Ex: Medical, Home, Space, Towns, Roads, Sensors, etm.

What is Bluetooth (A History)

- Who pioneered a key component of Bluetooth???

What is Bluetooth (A History)

- Hedy Lamarr: Secret runaway, Movie star, scientific leader, all around bad-ass

What is Bluetooth (A History)

- Designs the first modern airplane wing
- Discovered how to manipulate radio frequencies (called frequency hopping) to form an unbreakable code to protect classified messages from being intercepted by the German military
  - Technology (e.g. spread spectrum) that made wireless phones, GPS systems, and many other devices of our day possible
- Her work was a forerunner to modern day Bluetooth technology used in mobile phones and Wi-Fi systems
- She was marketed as the most beautiful woman in the world by Hollywood film makers

BR/ED vs BLE

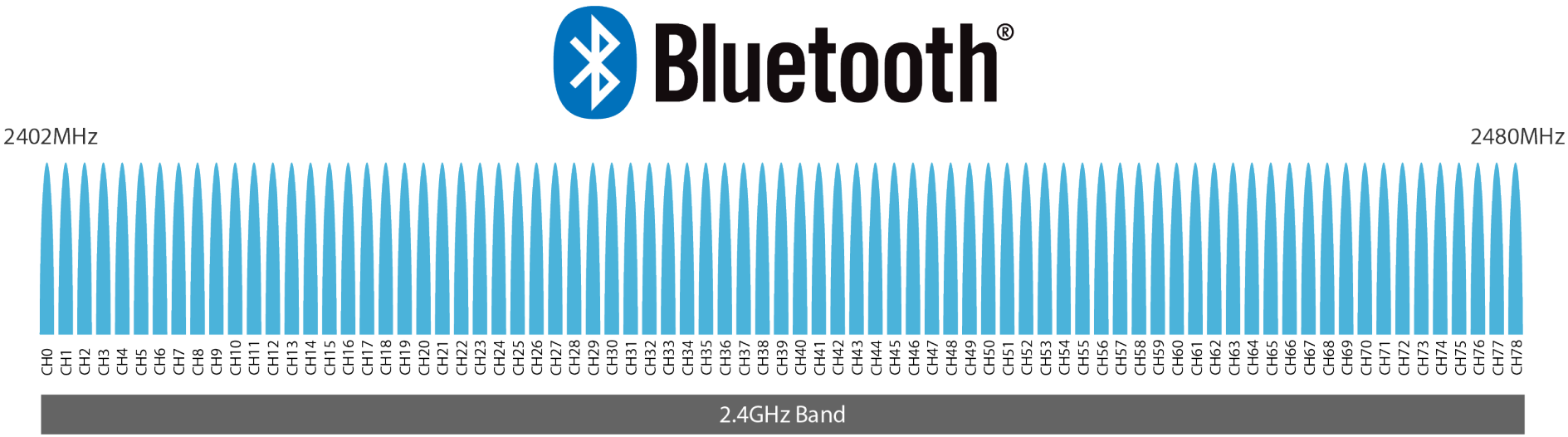


Figure 3: Bluetooth Classic Spectrum

- 79 number of (pairing) channels
- Freq. hoping to minimize interference
  - Defense in depth
- Larger power requirements minimize effectiveness for embedded systems

BR/ED vs BLE

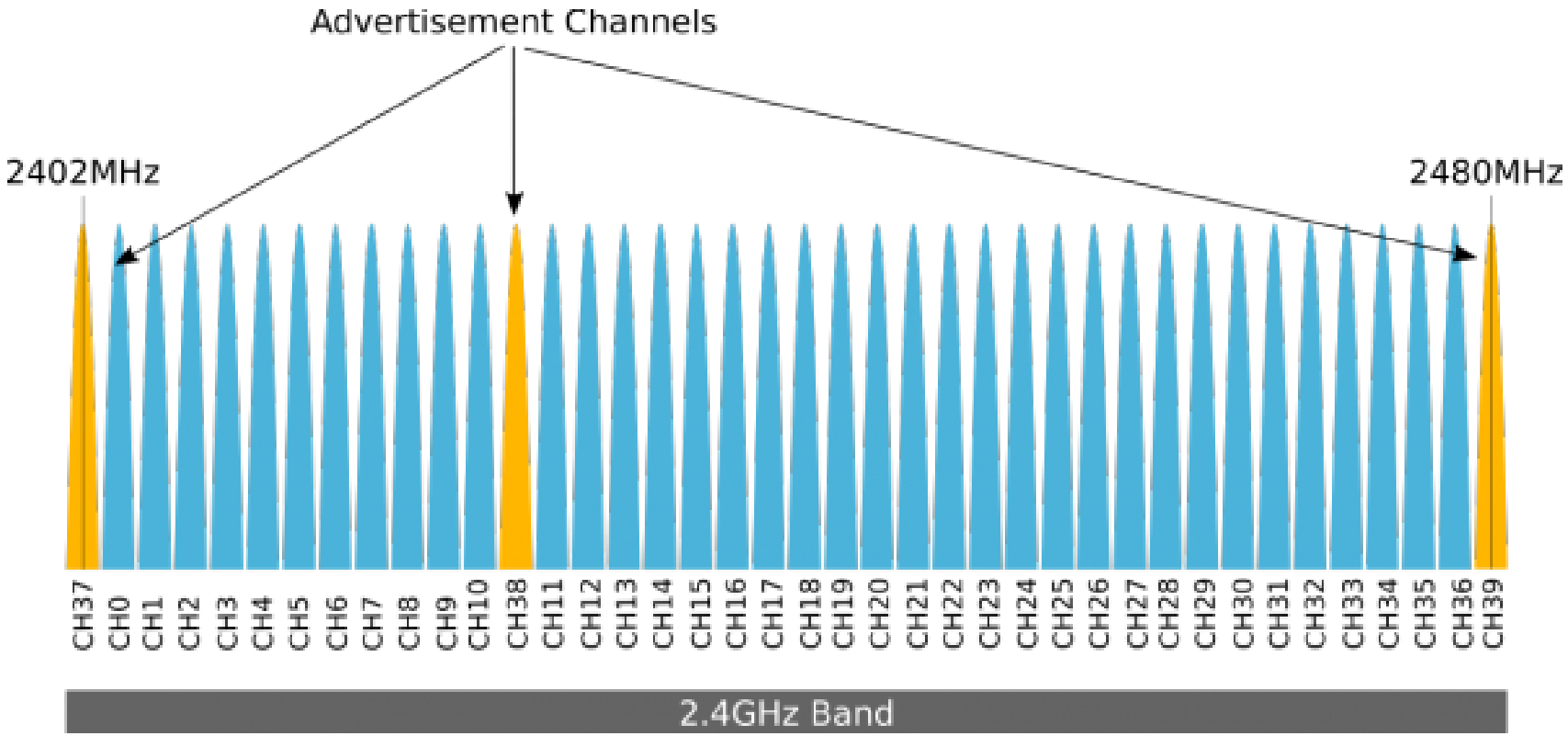


Figure 4: Bluetooth Low Energy Spectrum

- Three pairing channels; simplified sniffing of pairing
- Post connection return to classic frequency hopping
  - Capture of handshake provides easier eavesdropping
- Lower power requirements; ideal for embedded systems

BLE - Must Know Basics

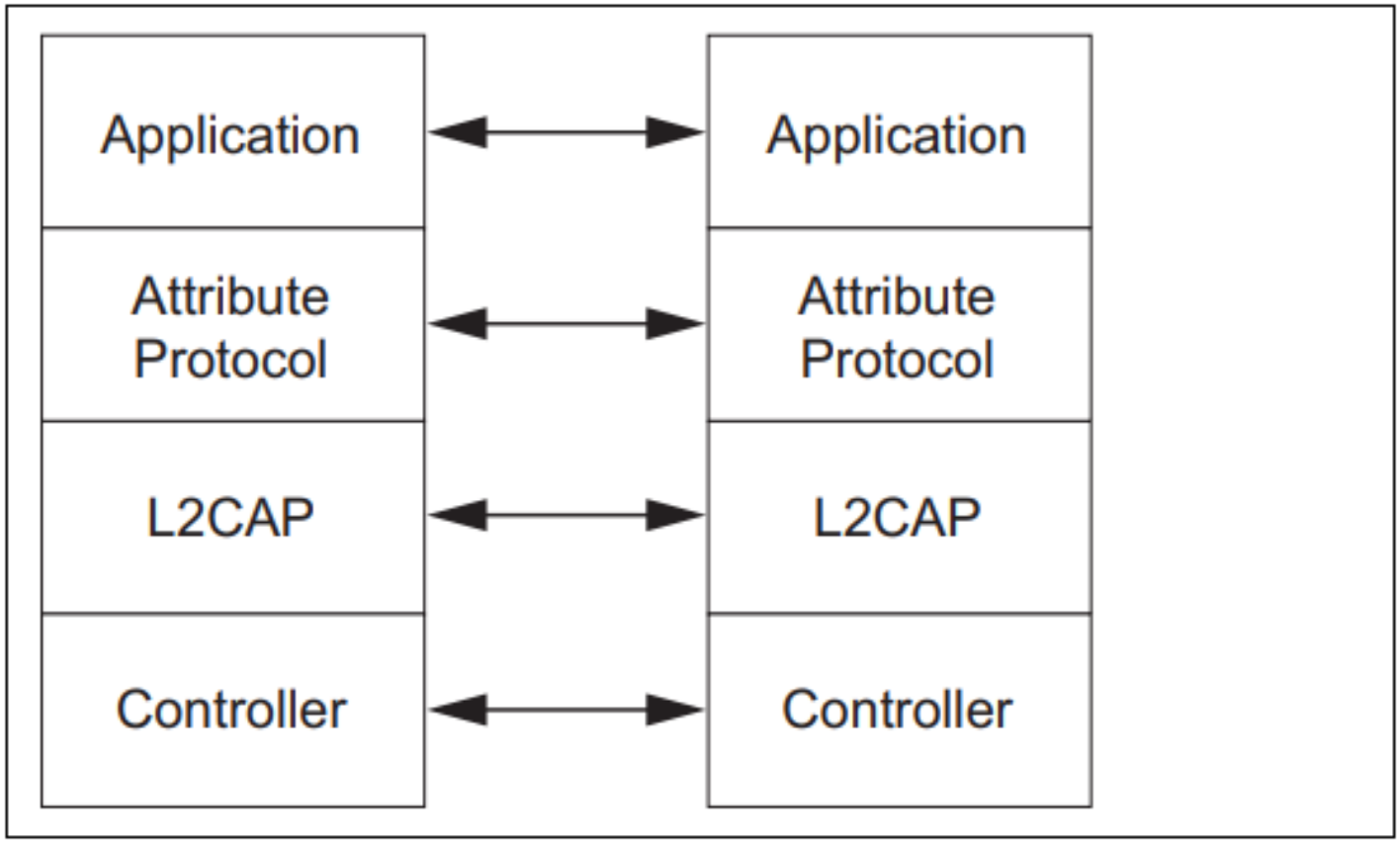


Figure 5: Protocol model [14]

- Everything goes through the HCI Layer; bottom most layer
  - HW-2-SW translation

BLE - Must Know Basics - High-level BLE Structure

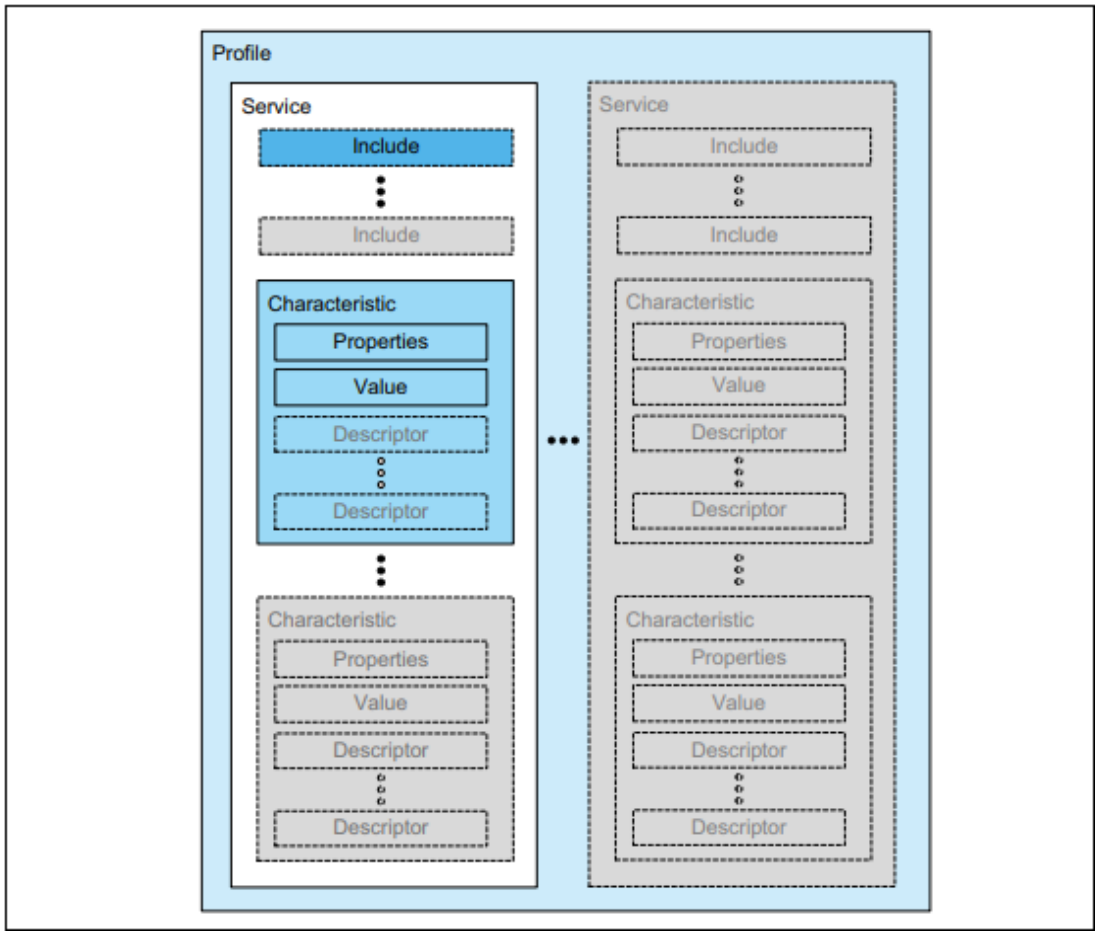


Figure 6: GATT-Based Profile hierarchy [14]

- Device, Services, Characteristics, Descriptors
- Characteristics have to be read once to populate data
  - Second read allows determining the data

What Hardware Exists

Bluetooth Hardware:

- Ubertooth One – Bluetooth Classic listener
- UD100-G03 Bluetooth Adapter – USB Bluetooth device effective for Bluetooth sniffing and interaction
- HackRF – Software-Defined Radio (SDR) that can be tuned to capture Bluetooth Radio Frequency (RF) spectrum transmissions
- Micro:Bit – USB Bluetooth Low Energy development device; very inexpensive
- Embedded Systems - ESP32, RTL8720DN/BW16 (Arduino IDE Codable)

What Code Libraries Exist

What libraries are out there you say?

- btzen
- μpython bluetooth / aioble
- bleak
- pybluez
- simplepyble
- NoBLE
- BLEno
- PyBT
- PyGATT

Note: A generally found reference is ukBaz Bluetooth Writings

What about Reference Material?

- Bluetooth Special Interest Group Specs
- BlueZ Reference Documentation

What about the tools?

Tools in the Ether

- gatttool
  - Useful, mainly depreciated
- bettercap
  - Go framework meant as an all-in-one solution; incorporated in older *bleah* toolset
- Blue Hydra
  - Bluetooth device sniffing and tracking
- hcitool
  - Command Line tool for configuring Bluetooth
- HCIDump
  - Tool similar to TCPDump to capture Bluetooth traffic on the host
- Bluetoothctl
  - Interactive Bluetooth control tool
- BlueSoleil
  - Windows-based software for service enumeration and interaction with Bluetooth devices

Sanity Check via Tools

- Learning the basics with tools
  - dbus-send, dbus-monitor, gdbus, busctl
  - btmon, btmgmt, bluetoothctl

Sanity Check via Tools - bluetoothctl

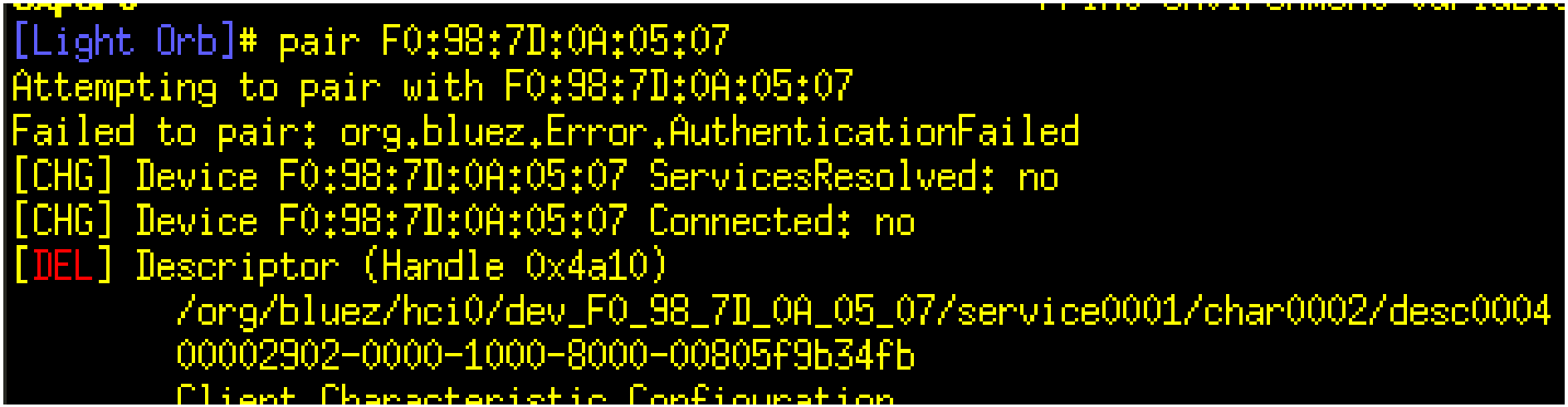


Figure 7: bluetoothctl - Light Orb Test

Cool... - Now What?

What D-Bus is this?

- Linux specific orientation to work
- Not as familiar with Windows layer for Bluetooth API/Interaction
- From Armis 2023 White Paper:
  - "The first significant stack is Linux's BlueZ stack, which was used by early Android versions, and is still in use by Linux and other OSs derived from it, such as Samsung's Tizen OS. Later on, Android developed its own stack, called Bluedroid or Fluoride, used in all Android devices from version 4.2 onwards. Windows has its own Bluetooth stack, since Windows XP, and Apple created two variations of the Bluetooth stack, one for iOS, and the other for OSX."

Basics

- D-Bus developed and maintained by freedesktop.org
  - URL: www.freedesktop.org/wiki/Software/dbus
- D-Bus uses code agnostic low-level API reference implementation that works/portable to any Linux or UNIX flavor
  - Make use of Python API calls with tool
- D-Bus is a message bus system; providing a simple way for applications to talk to one another
  - Provides a system daemon and a per-user-login-session daemon [1]

D-Bus Interfaces - Basics

- D-Bus interaction and enumeration is done via a series of interfaces
- Interfaces exist for each “layer” of interactivity with the GATT Server

D-Bus Interfaces - Little Bit Deeper Now

- Several types of D-Bus interfaces; each with separate purposes
  - Methods = Functions
  - Signals =?? Periodic Communications ?? __(o.O)__/
  - Properties = Properties
- Examples of interfaces:
  - Device - org.bluez.Device1
  - GATT Service - org.bluez.GattService1
  - Properties - org.freedesktop.DBus.Properties
  - Introspection - org.freedesktop.DBus.Introspectable

D-Bus Interfaces - busctl

```
┌──(user  kali)-[~/Documents/BTInteract]
└─$ busctl tree org.bluez
└─/org
  └─/org/bluez
    └─/org/bluez/hci0
      ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6
      │ ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0001
      │ │ └─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0001/char0002
      │ │   └─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0001/char0002/desc0004
      │ └─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0029
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char002b
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char002d
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char002f
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0031
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0033
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0035
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0037
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0039
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char003b
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char003d
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char003f
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0041
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0043
      │   ├─/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0028/char0045
```

D-Bus Interfaces - busctl - Introspection

```
┌──(user  kali)-[~/Documents/BTInteract]
└─$ busctl introspect org.bluez /org/bluez/hci0/dev_CC_50_E3_B6_BC_A6
NAME                                TYPE      SIGNATURE RESULT/VALUE                                FLAGS
org.bluez.Device1                   interface -         -                                           -
.CancelPairing                      method    -         -                                           -
.Connect                            method    -         -                                           -
.ConnectProfile                     method    s         -                                           -
.Disconnect                         method    -         -                                           -
.DisconnectProfile                  method    s         -                                           -
.Pair                               method    -         -                                           -
.Adapter                            property  o         "/org/bluez/hci0"                           emits-change
.Address                            property  s         "CC:50:E3:B6:BC:A6"                         emits-change
.AddressType                        property  s         "public"                                    emits-change
.Alias                              property  s         "2b00042f7481c7b056c4b410d28f33cf"          emits-change writable
.Appearance                         property  q         -                                           emits-change
.Blocked                            property  b         false                                       emits-change writable
.Class                              property  u         -                                           emits-change
.Connected                          property  b         true                                        emits-change
.Icon                               property  s         -                                           emits-change
.LegacyPairing                      property  b         false                                       emits-change
.ManufacturerData                   property  a{qv}     -                                           emits-change
.Modalias                           property  s         -                                           emits-change
.Name                               property  s         "2b00042f7481c7b056c4b410d28f33cf"          emits-change
.Paired                             property  b         false                                       emits-change
.RSSI                               property  n         -                                           emits-change
.ServiceData                        property  a{sv}     -                                           emits-change
.ServicesResolved                   property  b         true                                        emits-change
.Trusted                            property  b         false                                       emits-change writable
.TxPower                            property  n         -                                           emits-change
.UUIDs                              property  as        3 "000000ff-0000-1000-8000-00805f9b34fb"    emits-change
.WakeAllowed                        property  b         -                                           emits-change writable
org.freedesktop.DBus.Introspectable interface -         -                                           -
.Introspect                         method    -         s                                           -
org.freedesktop.DBus.Properties     interface -         -                                           -
.Get                                method    ss        v                                           -
.GetAll                             method    s         a{sv}                                       -
.Set                                method    ssv       -                                           -
```

D-Bus Interfaces - Introspection

- Compare returned data against 'busctl' visibility
- Create rough structures
  - Enough to proceed with enumeration
- Understanding: Each interface is explored via an **introspection** interface
  - Using the introspection interface's introspect method **(-_-;)**

Abstracting Introspection

- Produces an XML map of the introspected interface's information
  - Contains (1) [Sub-]Interfaces and (2) [Sub-]Nodes associated to the introspected interface

```
[!] Introspection E-Tree Information:          <Element 'node' at 0x74b486f0c2c0>
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.freedesktop.DBus.Introspectable'} ]
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.bluez.GattService1'} ]
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.freedesktop.DBus.Properties'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char002f'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char0031'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char0034'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char0037'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char003a'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char003d'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char003f'} ]
        Child Tag:      [ node ]               -              Child Attrib:   [ {'name': 'char0042'} ]
[!] Introspection E-Tree Information:          <Element 'node' at 0x74b486f0cc70>
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.freedesktop.DBus.Introspectable'} ]
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.bluez.GattCharacteristic1'} ]
        Child Tag:      [ interface ]          -              Child Attrib:   [ {'name': 'org.freedesktop.DBus.Properties'} ]
```

Figure 10: Etree XML Example - Light Orb

Signature-ing a Time Sink - Characteristic

```
┌──(user  kali)-[~/Documents/BTInteract]
└─$ busctl introspect org.bluez /org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/service0001/char0002
NAME                                TYPE      SIGNATURE RESULT/VALUE                              FLAGS
org.bluez.GattCharacteristic1       interface -         -                                         -
.AcquireNotify                      method    a{sv}     hq                                        -
.AcquireWrite                       method    a{sv}     hq                                        -
.ReadValue                          method    a{sv}     ay                                        -
.StartNotify                        method    -         -                                         -
.StopNotify                         method    -         -                                         -
.WriteValue                         method    aya{sv}   -                                         -
.Flags                              property  as        1 "indicate"                              emits-change
.MTU                                property  q         500                                       emits-change
.NotifyAcquired                     property  b         -                                         emits-change
.Notifying                          property  b         false                                     emits-change
.Service                            property  o         "/org/bluez/hci0/dev_CC_50_E3_B6_BC_A6/   emits-change
.UUID                               property  s         "00002a05-0000-1000-8000-00805f9b34fb"    emits-change
.Value                              property  ay        0                                         emits-change
.WriteAcquired                      property  b         -                                         emits-change
org.freedesktop.DBus.Introspectable interface -         -                                         -
.Introspect                         method    -         s                                         -
org.freedesktop.DBus.Properties     interface -         -                                         -
.Get                                method    ss        v                                         -
.GetAll                             method    s         a{sv}                                     -
.Set                                method    ssv       -                                         -
.PropertiesChanged                  signal    sa{sv}as  -                                         -
```

Figure 11: busctl - Inspecting BlueZ Characteristic

Reads + Writes - Read Method I

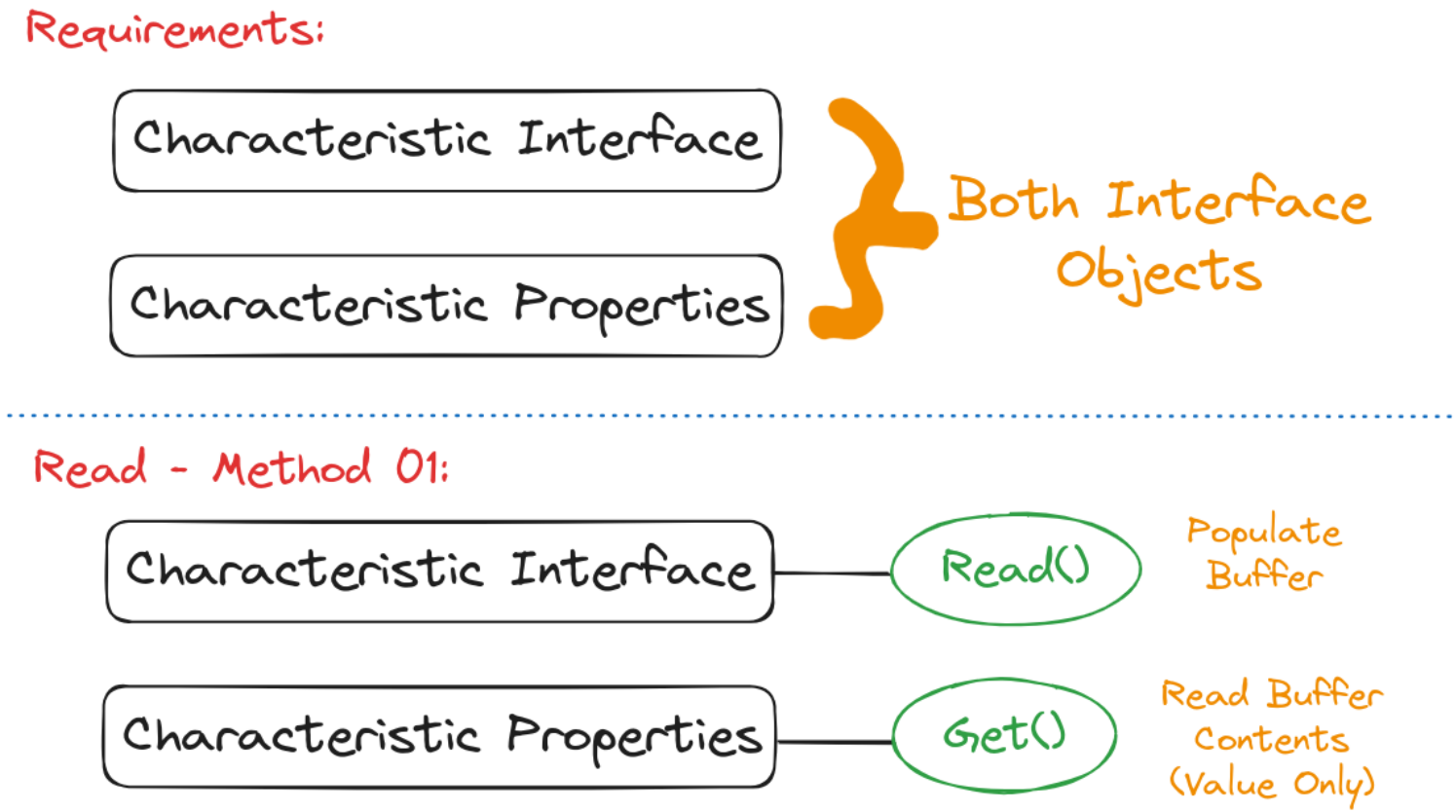


Figure 12: I/O - Read Method 01

Reads + Writes - Read Method II

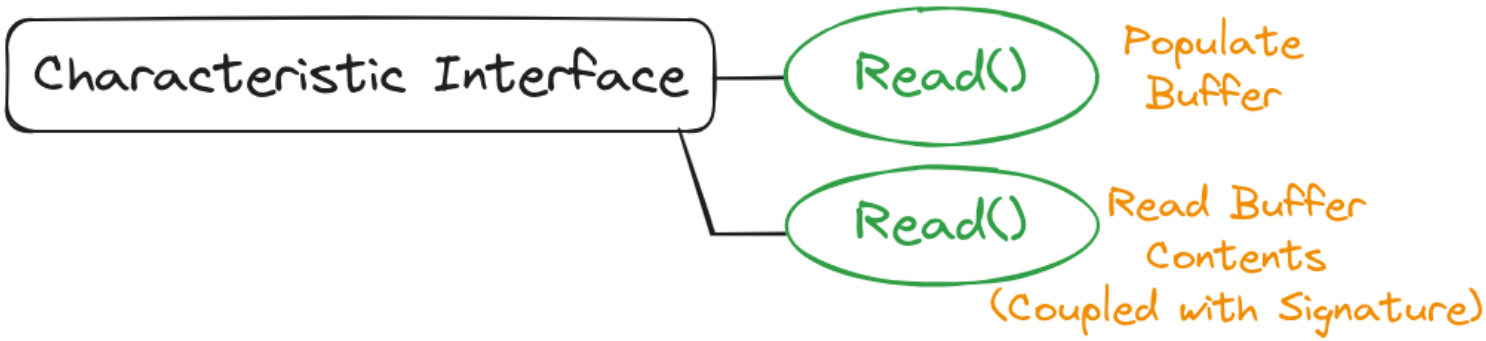


Figure 13: I/O - Read Method 02

Reads + Writes - Write Method

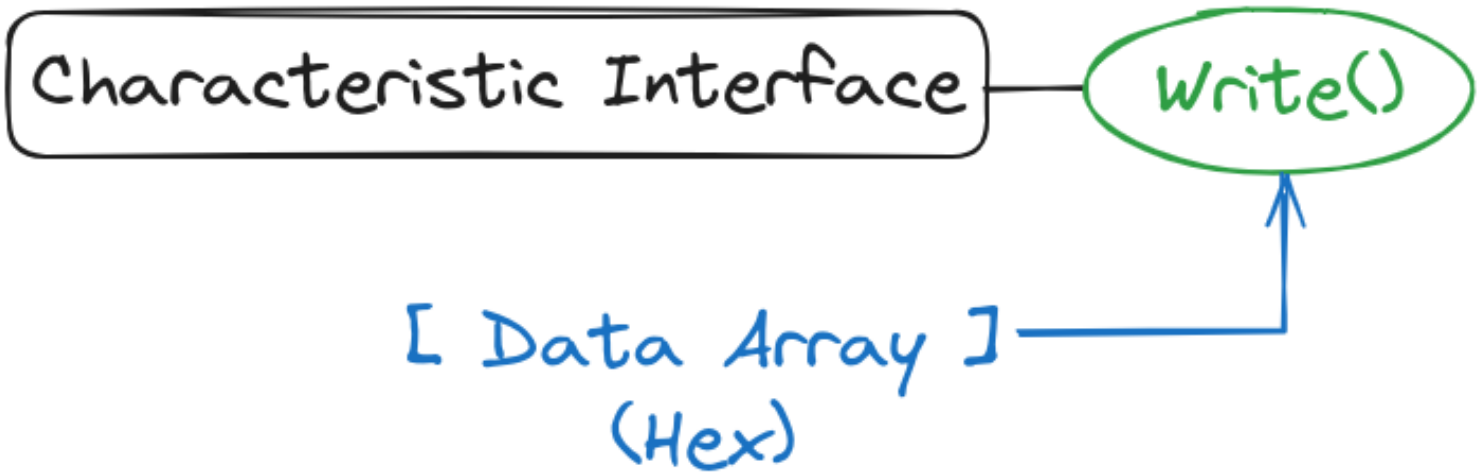
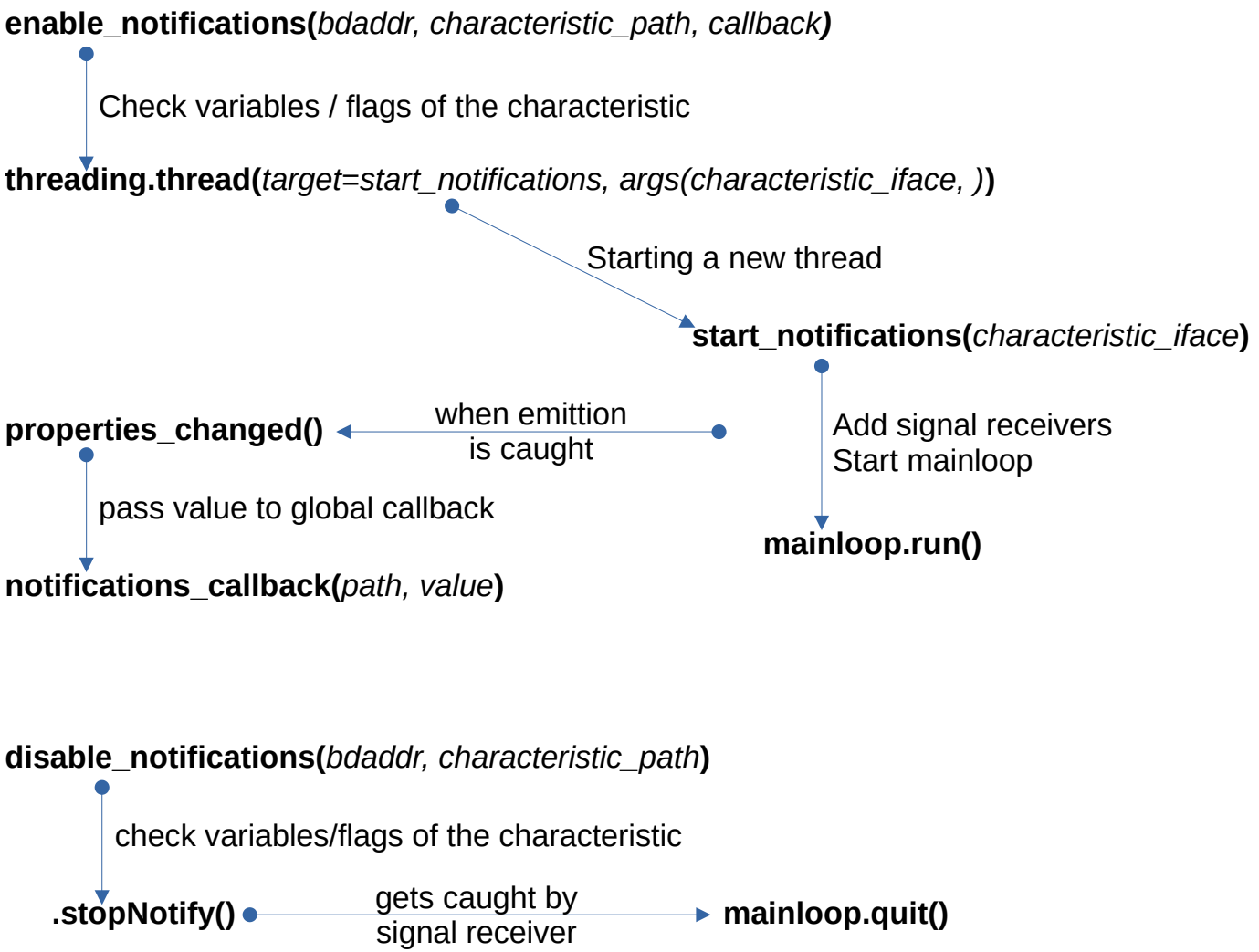


Figure 14: I/O - Write Method

Emissions - Signals + Notifications

- What are signals/emissions?
  - Messages which might be received asynchronously [11]
  - Application must register interest in the signal
  - Provide callback function to process the signal and any arguments

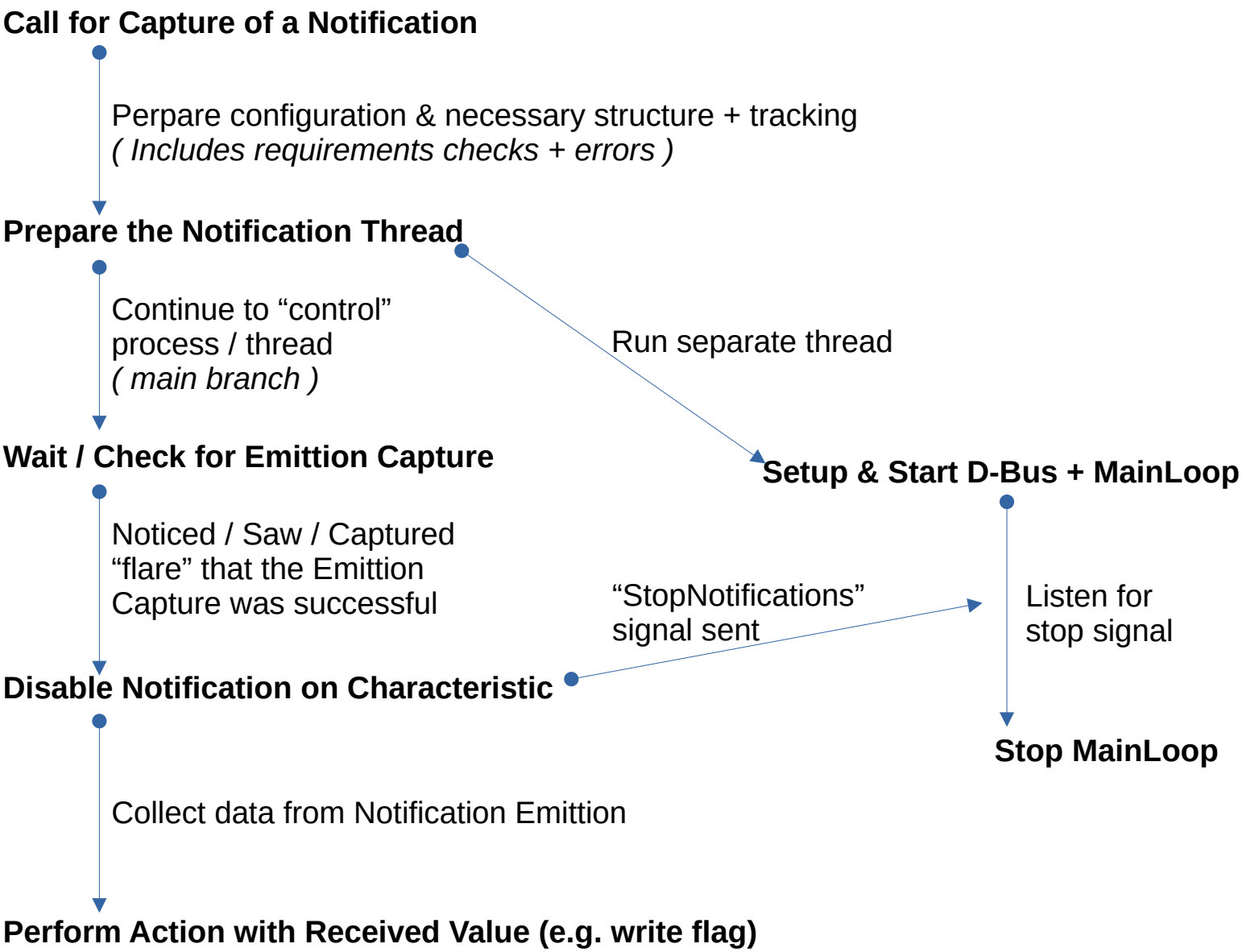
Emissions - Signals + Notifications - BT SIG Example Flow



Emissions - Signals + Notifications - Pseudo Code

- Functionality Requirements:
  - Capture and parse “PropertiesChanged” Emittion
  - Stop the GLib MainLoop (Engine)
  - Configure D-Bus & GLib MainLoop
  - Validate the passed characteristic
  - Configure & run the thread
  - Act on Emission’s data

Emissions - Signals + Notifications - Developed Flow



What about Bluetooth Battle?

Known Custom Tools

- BLEJuice
  - Man-in-the-Middle tool for Bluetooth Low Energy
  - Creates a fake device as a clone
- BLEJack
  - Bluetooth Low Energy connection Hijacking tool
  - Requires time to map the hopping pattern variables
- GATTacker
  - Node.js package for BLE
  - Focused on Man-in-the-Middle and other attacks

Moar Custom Tools

- Bluesnarfer
  - Bluetooth attacking tool
- Bluediving
  - Bluetooth penetration testing suite
- Spooftooph
  - Tool for automating masquerading and spoofing Bluetooth device information
- blefuzzV21.sh
  - Bluetooth Low Energy Fuzzer Version 21

Moar Custom Tools

- Bluetooth Stack Smasher (bss)
  - Performs several L2CAP checks sending malicious packets
- gr-Bluetooth
  - Tool written to sniff Bluetooth using a HackRF One
- ubertooth-btle
  - Bluetooth Low Energy (BLE) sniffing and more
- BlueDucky
  - Bluetooth Rubber Ducky

Online search can show sets for Classic and Low Energy; Black Arch Bluetooth

Bluetooth CTF/Training

Bluetooth CTFs:

- GitHub - hackgnar/ble_ctf: A Bluetooth low energy capture the flag
- GitHub - hackgnar/ble_ctf_infinity: Advanced: Bluetooth low energy capture the flag

Known Vulnerabilities

- Braktooth
- Blueborne
- KNOB Attack
- BLUFFS attack
- SweynTooth
- CVE-2023-45866 (Android Keyboard; JustWorks)

KNOB Vulnerability

Key Negotiation of Bluetooth (KNOB)

- Attack on the encryption key negotiation protocol of Bluetooth BR/EDR
  - Allows an attacker to make two+ victims agree on an encryption key with only 1 byte (8 bits) of entropy
  - Can then brute force the negotiated encryption keys, decrypt eavesdropped ciphertext, inject valid encrypted messages (in real-time)

BrakTooth

- Targets the Bluetooth Link Manager
- Impacts categorized into crashes and deadlocks
  - Majority can be viewed as DoS
- One vulnerabilities reported that allows Arbitrary Code Execution in IoTs
  - Requires knowledge of firmware layout on target device

SweynTooth

- 12 Bluetooth based vulnerabilities in BLE software development kits
- Three types of flaw results: Crash, Deadlock, Security Bypass
  - Security Bypass due to out-of-order request during Secure Connections pairing
  - Allows arbitrary read/write

BLUFFS Vulnerability

BLUFFS

- Six “novel” attacks that break Bluetooth session forward security and future security
  - Exploits two novel vulnerabilities related to unilateral and repeatable session key derivation; at architectural level
- Low-cost tool kit to perform and automatically check effectiveness of the attacks
- Enables device impersonation and machine-in-the-middle across sessions via comprise of a single key

Blueborne

- The attack does not require the targeted device to be set on discoverable mode or to be paired to the attacker's device
- The targeted user is not required to authorize or authenticate the connection to the attacker's device
- Armis Labs identified eight vulnerabilities which can be used as part of the attack vector
  - "The vulnerabilities described above, and the related exploitation techniques are not very complex. They demonstrate how protocols which are difficult to implement are susceptible to bug"

CVE-2023-45866

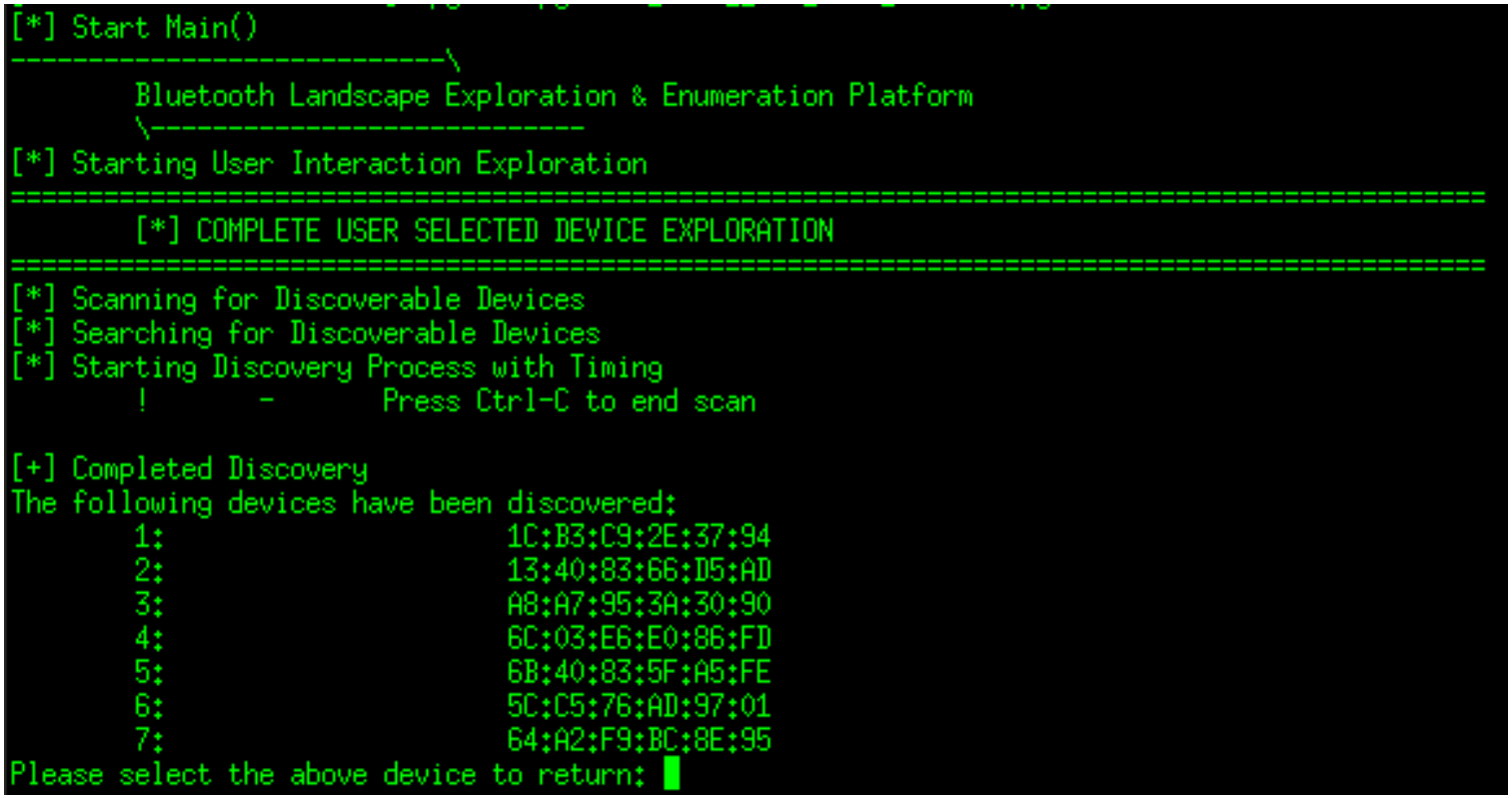
- “Bluetooth HID Hosts in BlueZ may permit an unauthenticated Peripheral role HID Device to initiate and establish an encrypted connection, and accept HID keyboard reports, potentially permitting injection of HID messages when no user interaction has occurred in the Central role to authorize such access”
- Essentially a “JustWorks” scenario for a headless keyboard

Questions

Questions?

Research Details

- Git Repo: https://github.com/Mauddib28/bleep-tool



Check Point - Lessons / Observations [End?]

- Make, take, and create clear documentation
  - Can you understand it in a week? month? year?
- Odd Behavior
  - Double read the GATT, Initialized GATT cannot be altered
  - D-Bus willingly forgets
  - W to Arduino Descriptors = panic crash
  - Default buffer character limit
- How to create and customize a basic BLE server
  - Remove blind testing and finger-crossing
  - **Note:** Not all BLE are the same!!

Signature-ing a Time Sink

- BlueZ git.kernel repository documentation [16]
  - API documentation for various components (e.g. Device, Service, Characteristic, Descriptor)
  - *Note:* Documentation is explanation of C/C++ code being abstracted via Python
- Read / Write interaction via D-Bus structures more complex than original PyBlueZ research variant
  - Defined interface (e.g. characteristic_interface) to access D-Bus methods
  - R+W **must** have "dict_options" passed; even if empty ({})

Signature-ing a Time Sink

- Reading: straight forward and easy
  - Ex: `characteristic_interface.ReadValue({})`
  - Returns: `dbus.Array([dbus.Byte(0)], signature=dbus.signature('y'))`

Signature-ing a Time Sink

- Writing: more complex with lots of trail/error
  - Ex: characteristic_interface.WriteValue("1", {})
  - Returns: `ERROR: dbus.connection: Unable to set arguments ('1', {}) according to signature 'aya{sv}': < class 'TypeError' >: an integer is required (got type str)`
- Requirement was to pass the value as an array
  - Ex: `.WriteValue([1], {})`
  - *Note:* Information that can be decoded from the signature

Emissions - Signals + Notifications - Back to the Research

- "... need details of the signal to be received so [one] can specify these details when registering with the system bus" [11]
  - Ex: `bus.add_signal_reciever(greeting_signal_received, dbus_interface = ‘‘com.example.greeting’’, signal_name = ‘‘Greeting signal’’)`
- Learned via Pain:
  - *Every* `add_signal_reciever()` generated gets called **whenever** a corresponding signal event occurs
  - Object and Interface aspects are **not** interchangeable; Characteristic Object has no StartNotify

Signal-ing Madness Herein

- Begin w/ attempt to **only** monitor the hci0 BlueZ device
  - `sudo dbus-monitor --system ‘‘path=‘/org/bluez/hci0’ ’’`
- Return of the GLib...
  - While notificaiton signals (e.g. PropertiesChanged) arrive on D-Bus after .StartNotify() method call, signal receiver not actively get data until GLib MainLoop running
  - **MAJOR** pain point in getting signals off the ground
- The use of ‘path_keyword=“path” ’ within add_signal_receiver() **sets** what the name of the variable that will contain the path information

Bibliography References I

Freedesktop

What is D-Bus, Published 2022

https://www.freedesktop.org/wiki/Software/dbus/

Last Accessed: 2024-03-28 22:43:19 EST

GNU

Knowing the Details of D-Bus Services

https://www.gnu.org/software/emacs/manual/html_node/dbus/Introspection.html

Last Accessed: 2024-04-01 19:48:34 EST

Programiz

Python Decorators

https://www.programiz.com/python-programming/decorator

Last Accessed: 2024-04-01 19:52:34 EST

Bibliography References II

hbldh

characteristic.py

https://github.com/hbldh/bleak/blob/63adefa24cb6ed11c8cf154fa41f51ecff1df98c/bleak/backends/characteristic.py

Last Accessed: 2024-04-01 19:56:34 EST

elsampsa

Bluetooth + DBus + gobject demo

https://github.com/elsampsa/btdemo

Last Accessed: 2024-04-01 19:54:52 EST

Freedesktop

DbusTools, Published May 07 2021

https://www.freedesktop.org/wiki/Software/DbusTools/

Last Accessed: 2024-03-28 22:57:29 EST

Bibliography References III


Bluetooth SIG
assigned_numbers
https://bitbucket.org/bluetooth-SIG/public/src/main/assigned_numbers/
Last Accessed: 2024-04-01 21:00:29 EST


Bluetooth SIG
public bitbucket
https://bitbucket.org/bluetooth-SIG/public/src/main/
Last Accessed: 2024-04-02 15:13:33 EST


Archlinux Forum
Activation via systemd failed for unit dbus-org.bluez.service
https://bbs.archlinux.org/viewtopic.php?id=155714
Last Accessed: 2024-04-02 15:09:42 EST

Bibliography References IV

oscaracena

gattlib.h

https://github.com/oscaracena/pygattlib/blob/7d08c0805313201b2ab12628e19544bb180218a8/src/gattlib.h

Last Accessed: 2024-04-02 15:09:42 EST

Bluetooth SIG

Bluetooth for Linux Developers Study Guide - Versions 1.0, 1.0.1

https://www.bluetooth.com/bluetooth-resources/bluetooth-for-linux/

Last Accessed: 2021-12-29 10:26:27 EST, 2022-10-18 18:54:02 EST

Bibliography References V

Bluetooth SIG

Developer Study Guide Bluetooth Internet Gateways - Version 2.0.0

https://www.bluetooth.com/blog/the-bluetooth-internet-gateway-study-guide/

Last Accessed: 2021-07-21 14:46:56 EST

Bluetooth SIG

Bluetooth LE Developer Study Guide - Version 5.2.0

https://www.bluetooth.com/bluetooth-resources/bluetooth-le-developer-starter-kit/

Last Accessed: 2023-02-16 11:36:57 EST

Bibliography References VI

Bluetooth SIG

Bluetooth Core Specification - Versions 5.3, 5.4

https://www.bluetooth.com/specifications/specs/core-specification-5-[3—4]/

Last Accessed: 2023-12-19 13:24:22 EST, 2023-12-16 11:33:37 EST

Bluetooth SIG

Generic Attribute Profile (GATT)

https://www.bluetooth.com/wp-content/uploads/Files/Specification/HTML/Core-54/out/en/host/generic-attribute-profile–gatt-.html

Last Accessed: 2023-12-16 11:22:03 EST

Marcel Holtmann, Maxim Krasnyansky, Qualcomm

BlueZ - Bluetooth protocol stack for Linux

https://git.kernel.org/pub/scm/bluetooth/bluez.git/tree/doc

Last Accessed: 2024-04-11 10:39:23 EST

Bibliography References VII


Daniele Antonioli
BLUFFS: Bluetooth Forward and Future Secrecy Attacks and Defenses
https://dl.acm.org/doi/abs/10.1145/3576915.3623066


seemoo-lab
InternalBlue
https://github.com/seemoo-lab/internalblue


Daniele Antonioli
Post on BLUFFS: Bluetooth Forward and Future Secrecy Attacks and Defenses
https://francozappa.github.io/post/2023/bluffs-ccs23/
Last Accessed: 2024-012-16 19:25:38 EST