Introduction to Bluetooth Low Energy Exploitation

Maxine Filcher
Security Consultant

IOActive®

Whoami

- Maxine Filcher
- Security Consultant with IOActive
- US Army Veteran
- B.S. Info Assurance & Cybersecurity
  - Minor: Law & Policy
- SANS Women's Academy 2018 Cohort
- GSEC, GCIH, GPEN

  - maxine.filcher@ioactive.com
  - @FreqyXin

Disclaimer:

I am not a Bluetooth Developer
This is not a comprehensive dive on Bluetooth

IOActive.

What is RF Security?

- "…the prevention of unauthorized access or damage to computers or data using wireless networks." – Wikipedia

- "As the number and availability of wireless-enabled devices continues to increase, it is important for organizations to actively test and secure their enterprise wireless environments. Wireless scans can help organizations determine corrective actions to mitigate risks posed by wireless enabled technologies." – NIST SPUB 800-115

The Full Spectrum Wireless Attack Surface

Many Vectors

- Bluetooth, Wi-Fi, Zigbee, RFID, NFC, Cell, SATCOMM

Large Surface

- Medical, IoT, Industry 4.0, Mobile, Auto, Wearable

Bluetooth

2.4 – 2.485 GHz

Bluetooth Classic BR/EDR

Point-to-Point

Data transmission / Continuous Connection

Bluetooth Low Energy (BLE)

Point-to-Point

Low energy consumption

Bluetooth Mesh

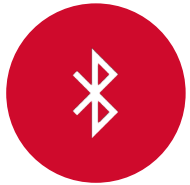
Many-to-Many

Supports 32,767 nodes per mesh network

Ubiquity & Growth

2.3 BILLION MOBILE DEVICES & PC'S IN 2018

780 MILLION CONNECTED BLE IOT DEVICES IN 2018



86% OF NEW VEHICLES INCLUDE BLUETOOTH (2018)

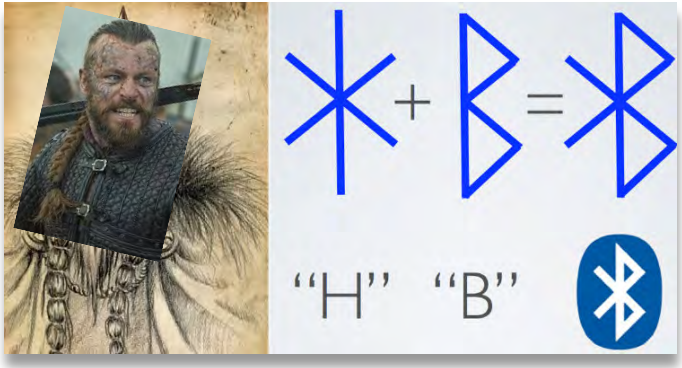
10X BLUETOOTH INDUSTRIAL IOT DEVICES BY 2022

815 MILLION SMART BUILDING DEVICES BY 2022

Bluetooth SIG – www.Bluetooth.com

A Quick History

- Herald “Bluetooth”
  - King of Norway and Denmark
- Hedy Lamarr
  - 1914-2000
  - Frequency Hopping Spread Spectrum (FHSS)
  - Radio Controlled Torpedoes
  - George Antheil & Self-playing Piano’s
  - Bombshell: The Hedy Lamarr Story



Terminology

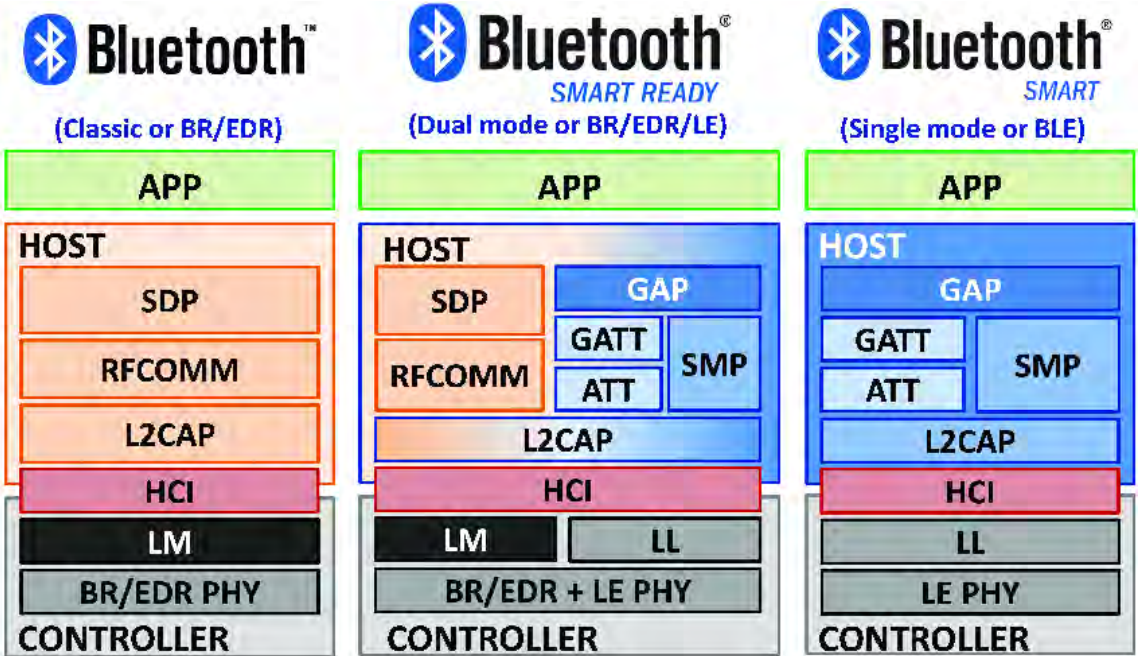
- Devices:
  - Central
    - Connection Initiator
    - Controls timing and data exchange
  - Peripheral
    - Advertises
    - Accepts incoming connections

Connections

- Advertising
  - 3 Channels for advertisement 37, 38, 39
  - 4 Advertising PDU Types
- Connecting
  - 36 Channels (US)
  - 2 MHz separation
- Pairing
  - Temporary security key until bonded
- Bonding
  - Create a trusted connection with a device via LTK exchange

Bluetooth Protocol Stack

- Generic Access Profile
  - Advertising and connections
- GATT Generic Attribute Profile
  - Profiles
  - Services
  - UUIDS
- ATT
  - Opcodes
- Security Management Profile
  - Controls Pairing and Bonding sessions



BLE GATT

- Service – (Full UUID)
  - Characteristics – (Full UUID)
    - Properties – (Read/Write/Notify)
    - Descriptors –(Short UUID)
    - Value – The data that affects device operation
- Catalog of services on a device

GATT “Hacking”

- Abuse Read Privileges
  - Device Details – (iOS: Battery level, User name, OS version)
- Abuse Write Privileges
  - Simple GATT Value Examples
    - 0x08 – Write New Pin
    - 0x01 – Initialize OTA Firmware Update
    - 0x02 – Start Heating Cycle

BLE Security

- BLE 4.0 – Introduced Encrypted Sessions with 4.2
  - MITM
  - Eavesdropping
  - Authentication via Pairing/Bonding
- BLE 5.0
  - Strengthens “Just-Works” pairing by introducing nonce keying

Bluetooth Vulnerabilities

- Blueborne
  - Armis, 2017
  - Windows, Android, iOS
  - https://www.armis.com/blueborne/
- Bleedingbit
  - Armis, 2018
  - Aruba, Cisco
  - https://www.armis.com/bleedingbit/
- SweynTooth
  - Feb 2020
  - Singapore University of Technology and Design
  - 12 vulnerabilities (Crash, Deadlock, Security Bypass)
  - 6+ Vendors
- KNOB (BR/EDR)
  - 2019
  - Key Negotiation on Bluetooth
  - Currently Affects Broadcom and Cypress Chips
  - https://knobattack.com/
- BIAS (BR/EDR)
  - May 2020
  - Impersonation attacks, LTK compromise
  - Multiple Vendors

Tools of The Trade

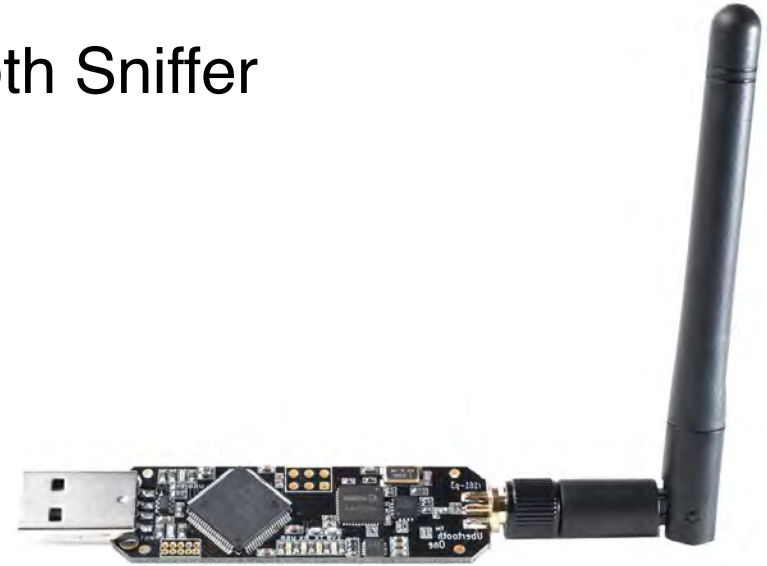
IOActive®

Bluetooth Tools

- nRF Connect
- nRF Sniffer
- Bettercap
- Scapy
- adb (Android Debug Bridge)
- InternalBlue
- Apple_bleee
- Blueborne Scanner
- Wireshark
- Kismet

Ubertooth One

- Still most accessible Bluetooth Sniffer
- Can follow BLE connections



GreatFet One—Quince

- GreatScott Gadgets
- “Neighbor” for GreatFet
- Will enable full spectrum sniffing for Bluetooth/BLE
  - Watch all the advertisement channels at once!
- No idea about price or release date yet

Elisys

- Starts ~$17k
- Uses SDR
- Forward compatible

nRF Connect

- Free application for iOS/Android/PC
- Intended for BLE debugging
- Great for BLE “Hacking”!

  *IMHO: Gold Standard for BLE research
- Creates log files that can be exported
- Records Macro functions for basic scripting capabilities

nRF Sniffer

- Wireshark ‘plugin’
- Supports various nRF DK’s and Dongles
- Does not support latest Wireshark yet
- Difficult to setup

Bettercap

- Has a UI much like Kismet
- Can be difficult to install
- Has ability to send custom GATT request/command

https://www.bettercap.org/

Scapy

- CLI for creating BLE packets
- Library for crafting BLE packets with python

  https://scapy.readthedocs.io/en/latest/layers/bluetooth.html

adb (Android Debug Bridge)

- PC interface to Android phone
- Enables sniffing of BLE traffic higher up the stack
- Opcodes will be unencrypted

*Recommend using Linux

  - sudo apt install android-tools-adb
  - sudo adb start-server
  - Open Wireshark
  - Select Interface

Wireshark

- Btatt.opcode == 0x52
  - Looks for ATT Write Commands

Kismet

- All purpose Sniffer
- Supports BLE

  *Limited interaction capabilities

Apple_bleee

- PoC for security issues in Apple's BLE stack
- Spoof Airpods
- Retrieve phone number from AirDrop transfer

https://github.com/hexway/apple_bleee

Blueborne Scanner

- Deprecated

  https://github.com/terry2012/BlueBorneVulnScanner

  – Emulator version

Hands On

IOActive®

Challenges for Researchers

- Most modern BLE connections will be encrypted
- In most cases Ubertooth can follow connections, but will not break the encryption
- Crackle is largely ineffective on current BLE pairings
- Commercial Sniffers are $$$$$+
- Most cheaper alternatives will not outperform Ubertooth

Approaches to BLE Security Testing

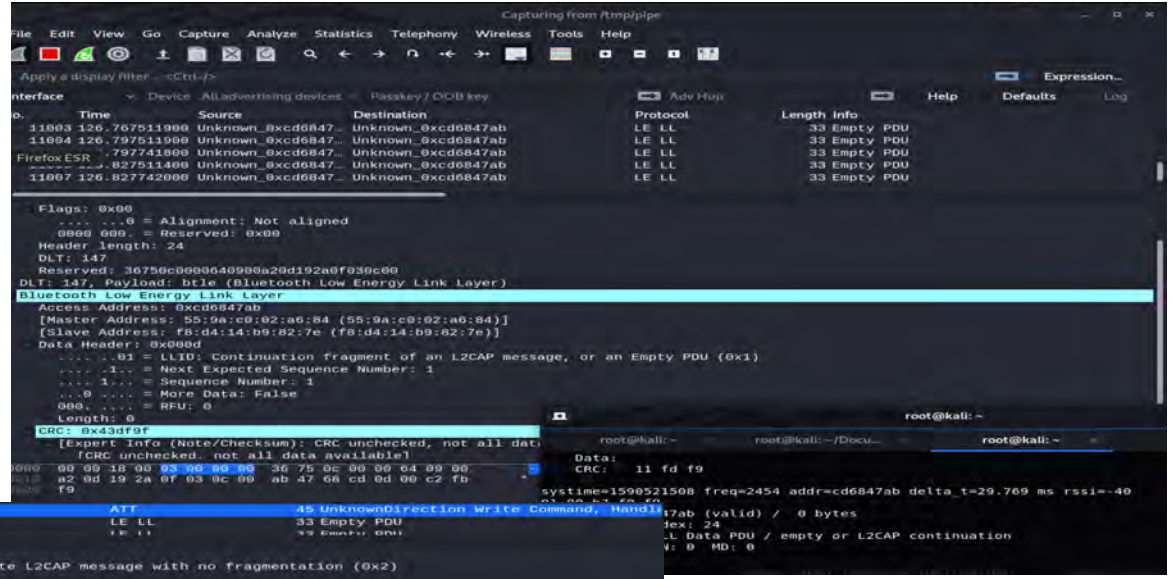
- 1. Sniff broadcast traffic between devices
  - Is it encrypted?
  - What can you see from the outside looking in?
- 2. Android Debug Bridge
  - Use Wireshark to sniff BLE traffic from app to device
- 3. Attempt Out-of-App Connection with Devices
  - Can a connection be established?
  - Can a bonded session be initialized?
  - GATT Fuzzing
- 4. Spoof Device
  - Does the companion application work with the spoof?
  - What GATT services does the app interact with?
  - Are any Opcodes written?

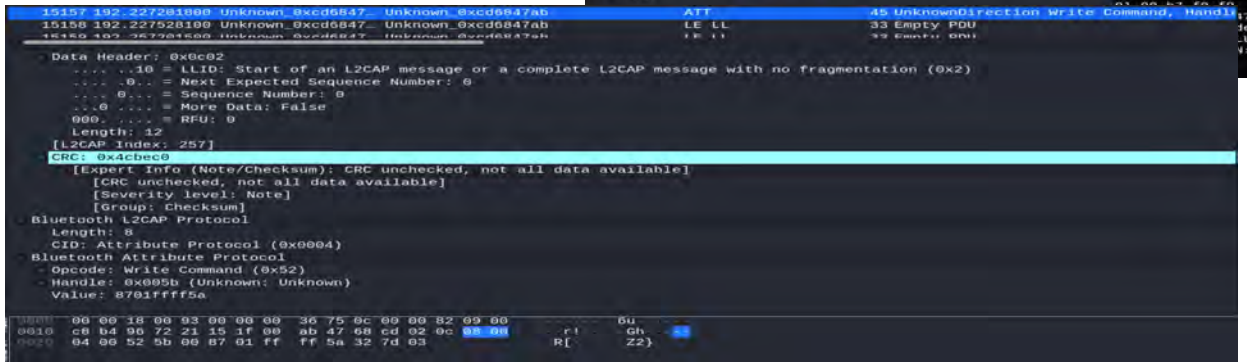
List of Equipment & Tools

- Nordic Thingy 52 – Target
- Adafruit Bluefruit LE
- LG M150 – Sniffer
- Samsung S20 – Victim
- Samsung S8 – Attacker
- Nordic nRF Connect
- Nordic Thingy App
- Adafruit Bluefruit Connect App
- Adb
- Wireshark

External Scan

- Ubertooth-btle
  - FiFo pipe to Wireshark
  - Run ubertooth-btle
  - Hops across advertainment channels
  - Default cmd follows first observed pairing
  - Usually encrypted*

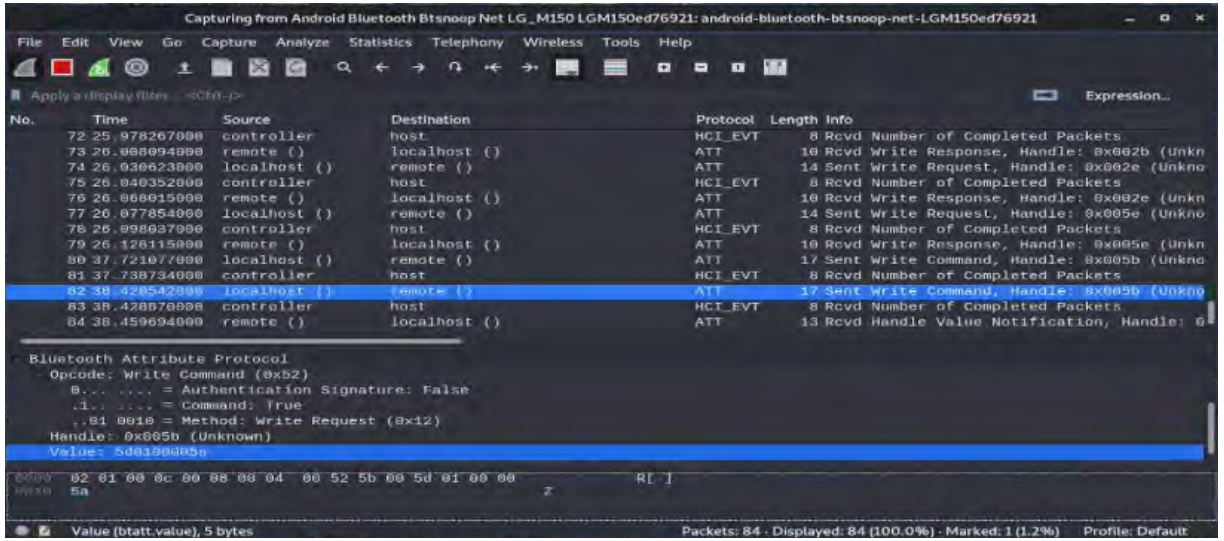




IOActive.

Internal Scan

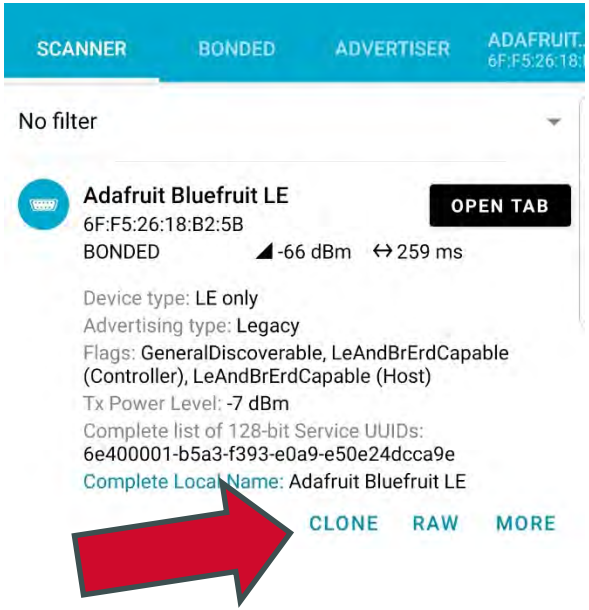
- BTSnoop
  - Retrievable file
  - Some devices support live capture
- Live Scan
  - adb
  - Wireshark

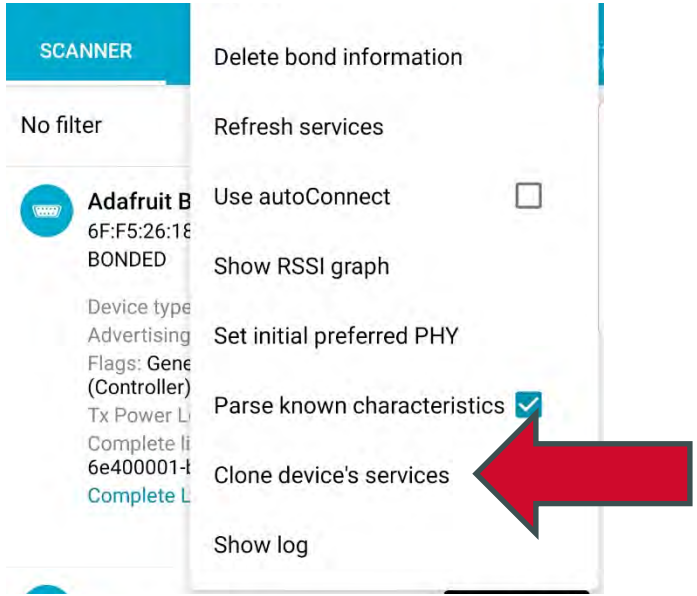


IOActive.

Cloning BLE Services

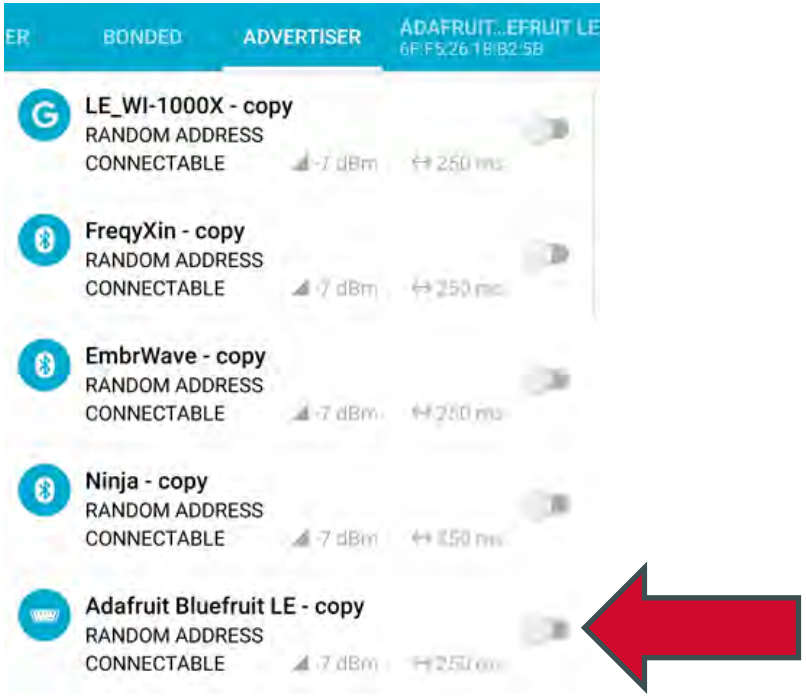
- Target: Adafruit Bluefruit LE
  - Spoofing BLE device to trick mobile application with nRF Connect





Spoofing BLE Device

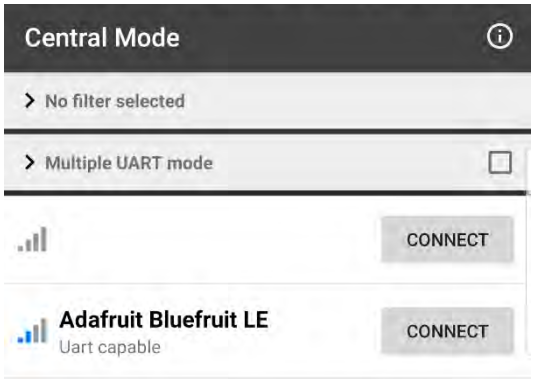
- Use cloned device details
- Change device name
- Wait for connection



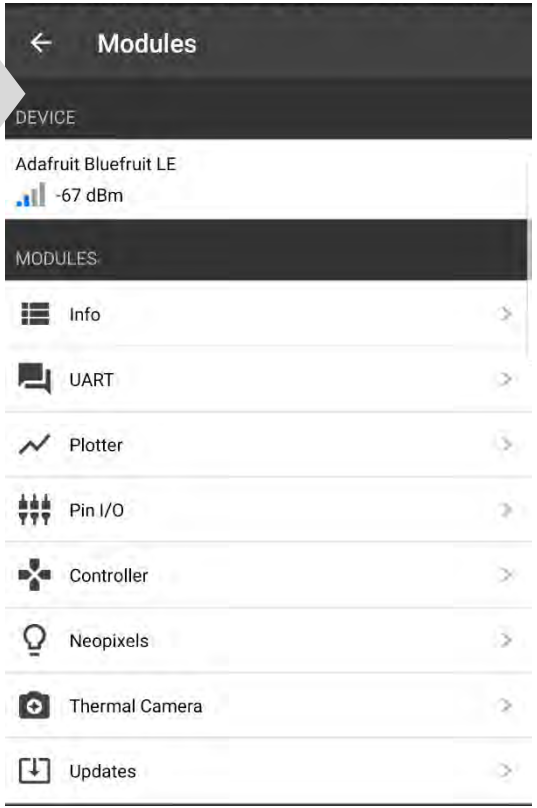
IOActive.

Connection to BLE Spoof

- View from victim
  - Adafruit Bluefruit Connect App



Success!



IOActive.

Hackgnar's BLE CTF

1. Plug in esp32 to power source
2. Use nRF connect application to create a connection with your device
3. Follow the guide at https://github.com/hackgnar/ble_ctf
4. Ask for help if you get stuck

Recommendations for the Individual

- Keep Bluetooth turned off, unless needed
- Choose low traffic, or private areas to pair with devices
- Be aware, some apps don’t have spoofing protections
- Audit your devices

Recommendations for the Developer

- Out-of-band (OOB) pairing options like NFC may reduce the exposure of security exchanges
- Don’t focus solely on device security
  - Does your companion app have strong posturing against spoofing attacks?
- A method to authenticate device specific keys with remote server to verify advertisements prior to connection would be ideal

Further Study

- Twitter
  - Joshua Wright
  - @naehrdine
- Books
  - Hacking Exposed Wireless
- YouTube
- Devices
  - Adafruit Bluerfruit LE
  - Thingy52
  - ESP32

Questions?

Thank You

IOActive®