The IoT Hacker's Toolkit

David Tomaschik (@Matir)
https://systemoverlord.com

Disclaimer

This presentation is being done in a personal capacity and does not reflect the views or opinions of my employer.

Disclaimer #2

Please be careful with devices powered from “mains” power. These may contain hazardous voltages. Avoid working on such devices unless you know what you’re doing.

Obligatory About Me

- Security Engineer
  - Blackbox Testing
  - Red Teaming
- Hacker
  - Embedded Devices
  - IoT
  - Webapps
- Maker
  - Electronics
  - Unofficial DEF CON Badge

What is IoT?

The Internet of things (IoT) is the network of physical devices, vehicles, home appliances and other items embedded with electronics, software, sensors, actuators, and connectivity which enables these objects to connect and exchange data. Each thing is uniquely identifiable through its embedded computing system but is able to inter-operate within the existing Internet infrastructure.

https://en.wikipedia.org/wiki/Internet_of_things

What is IoT?

I know it when I see it.

-- Supreme Court Justice Potter Stewart

What is IoT?

What is IoT?

What is IoT?

Mixabest on Wikimedia

Goals of IoT Security Researcher

- Understand IoT Devices
  - Functions
  - Security Properties
- Find Vulnerabilities
  - Improve Security
  - CVEs
  - Experience

- Hack All the Things
  - “Hacking in the 90s”
  - “Shooting Fish in a Barrel”
  - “A CTF on the Whole Internet”

Challenges of IoT Devices

- Architectures
  - ARM
  - MIPS
- Interfaces
  - No keyboard/mouse here!
- Protocols
  - HTTP
  - MQTT

Interfaces

Common

- WiFi
- Ethernet
- Bluetooth
- USB

Interfaces

Common

- WiFi
- Ethernet
- Bluetooth
- USB

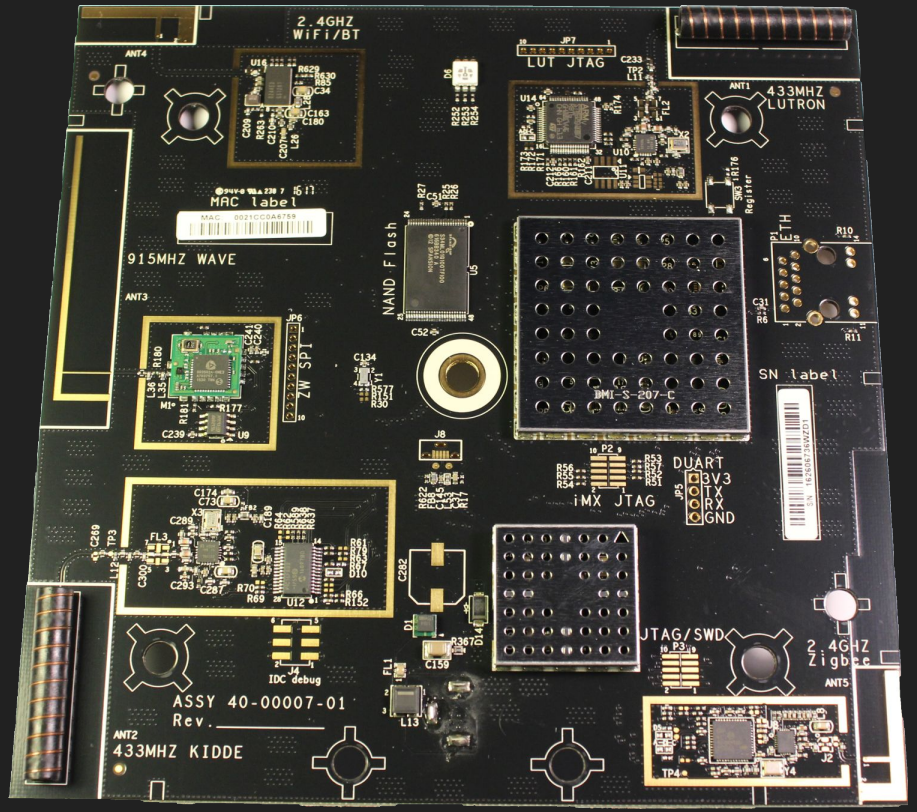
Uncommon

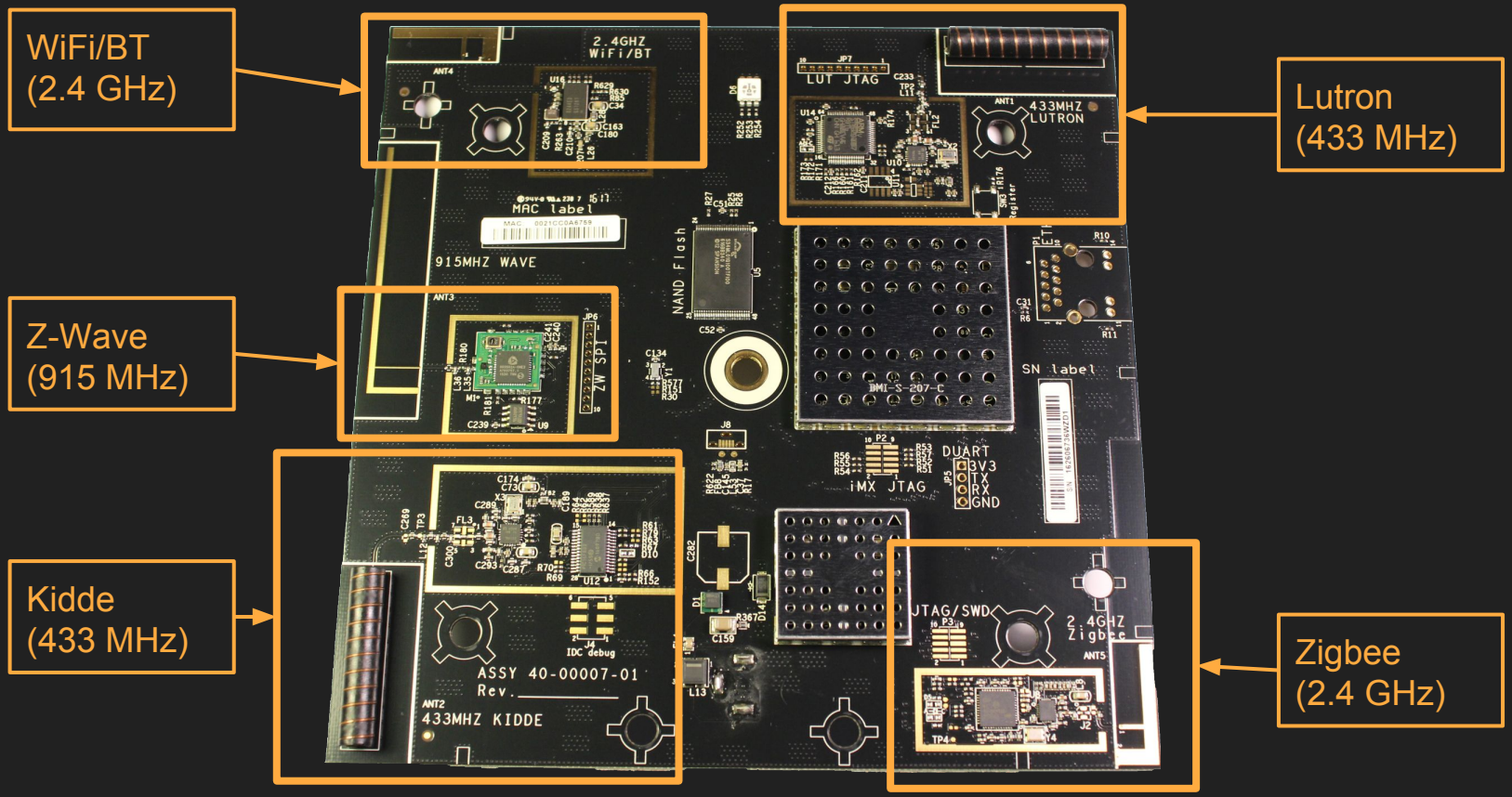
- Zigbee
- Z-Wave
- Cellular
- Proprietary

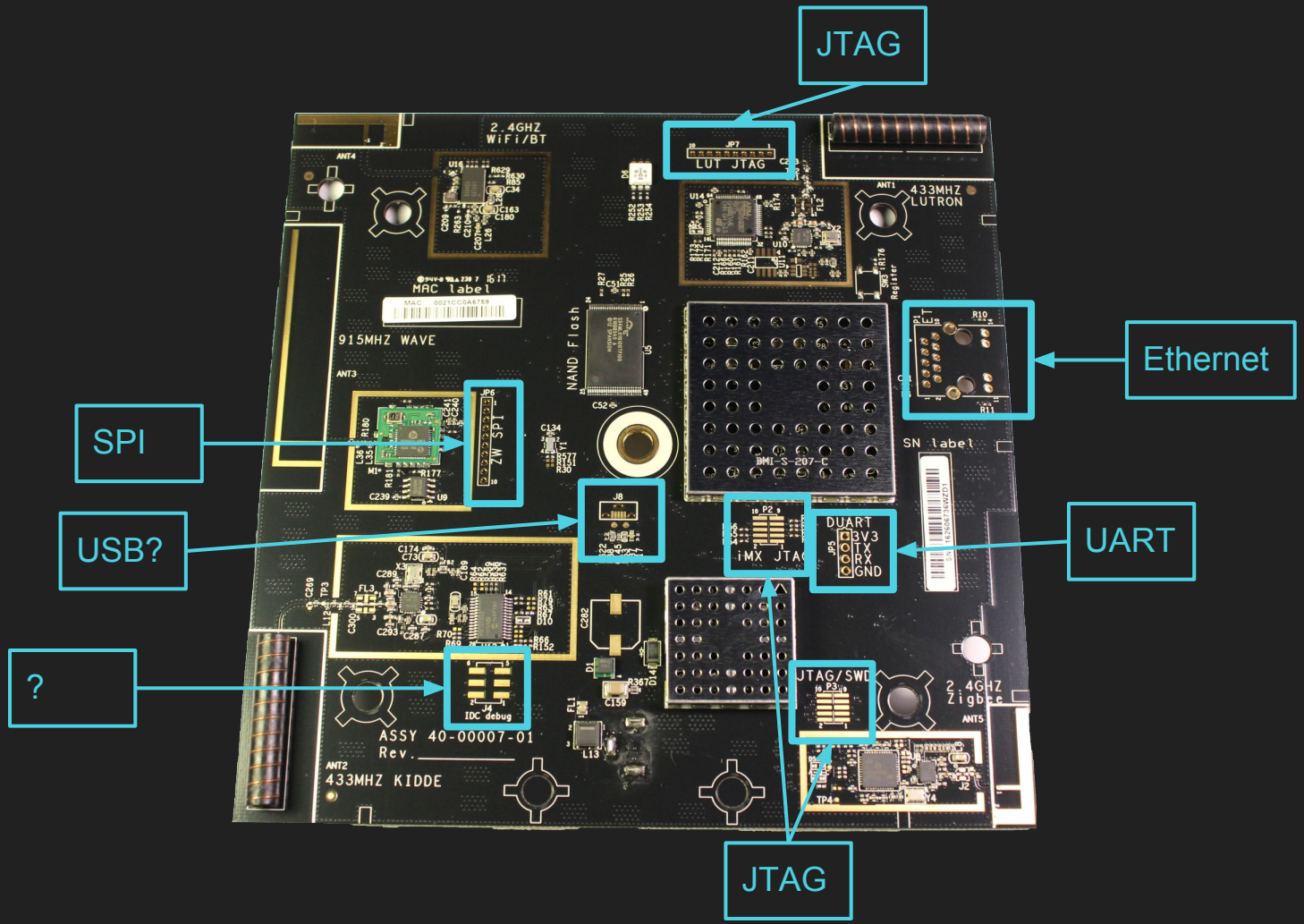
Internal

- UART
- JTAG
- SWD
- SPI

Interfaces


2.4GHZ
WiFi/BT
LUT JTAG
433MHZ
LUTRON
MAC label
915MHZ WAVE
NAND Flash
SN label
DMI-S-207-C
DUART
3V3
TX
RX
GND
iMX JTAG
JTAG/SWD
2.4GHZ
Zigbee
IDC debug
ASSY 40-00007-01
Rev.
433MHZ KIDDE


WiFi/BT
(2.4 GHz)
Z-Wave
(915 MHz)
Kidde
(433 MHz)
Lutron
(433 MHz)
Zigbee
(2.4 GHz)
2.4GHZ
WiFi/BT
LUT JTAG
433MHZ
LUTRON
MAC label
915MHZ WAVE
NAND Flash
ZW SPI
DMI-S-207-C
SN label
iMX JTAG
DUART
3V3
TX
RX
GND
JTAG/SWD
2.4GHZ
Zigbee
IDC debug
ASSY 40-00007-01
Rev.
433MHZ KIDDE


JTAG
Ethernet
SPI
USB?
UART
?
JTAG
2.4GHZ WiFi/BT
LUT JTAG
433MHZ LUTRON
MAC label
915MHZ WAVE
NAND Flash
SN label
DUART
3V3
TX
RX
GND
JTAG/SWD
2.4GHZ Zigbee
IDC debug
ASSY 40-00007-01
Rev.
433MHZ KIDDE

Basic Tools

Basic Tools

- Toolkit w/Security Bits
- Spudgers/Openers
- Multimeter
- Soldering Iron
- Wires & Jumpers

Basic Tools

- Toolkit w/Security Bits
- Spudgers/Openers
- Multimeter
- Soldering Iron
- Wires & Jumpers

Basic Tools

- Toolkit w/Security Bits
- Spudgers/Openers
- Multimeter
- Soldering Iron
- Wires & Jumpers

Think Capabilities, Not Toys

Missions

1. Man in the Middle Wireless Communications
2. Get a Local Console
3. Dump Firmware From Device
4. Find Vulnerabilities in Firmware
5. Observe Bluetooth Communications
6. Observe & Replay Proprietary Wireless

Mission: Wireless Man In The Middle

Tools Required:

- Wireless Interface
- ARP Spoofing
  - Bettercap
  - Ettercap
- Intercepting Proxy
  - Burp Suite
  - OWASP Zap
  - mitmproxy

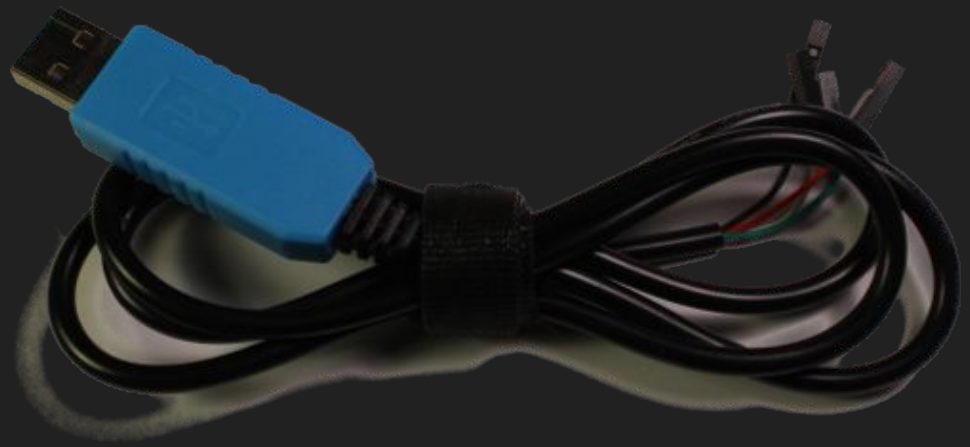
Steps:

1. Join same WiFi
2. ARP Spoof to Gain MITM Position
3. iptables magic to redirect HTTP(S) traffic
4. Intercepting proxy to play with traffic

Mission: Getting a Local Shell

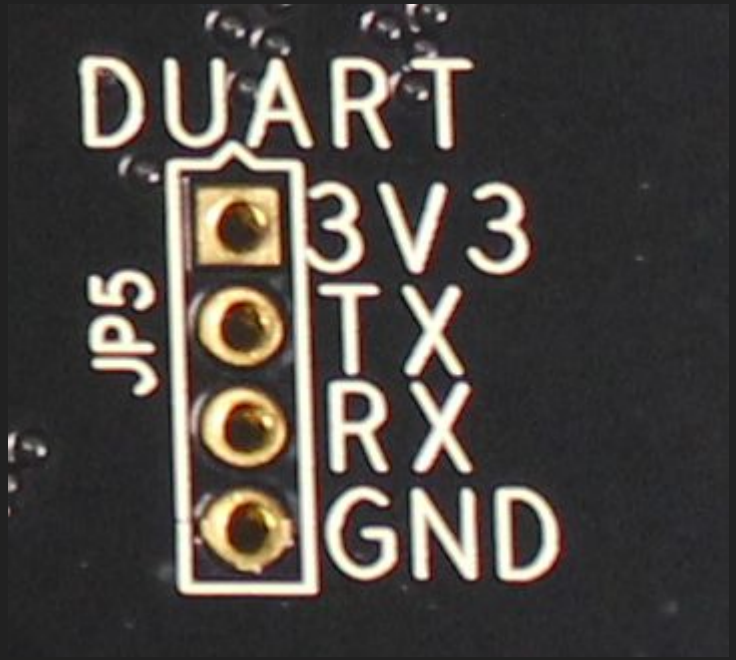
Tools Required:

- Screwdriver
- Soldering Iron (Maybe)
- Multimeter
- UART (Serial) Cable



Mission: Getting a Local Shell

1. (Maybe) Solder in a header
2. (Maybe) Use multimeter to identify pins
3. Connect UART Cable
4. Determine Settings
    a. 115200 8n1
    b. 9600 8n1
    c. Logic Analyzer/Oscilloscope
5. Use Console?



Mission: Dump Firmware

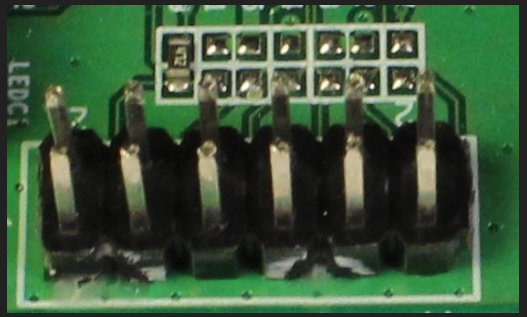
Many approaches:

- UART (Serial)
- JTAG
- Flash Chips

Mission: Dump Firmware

JTAG

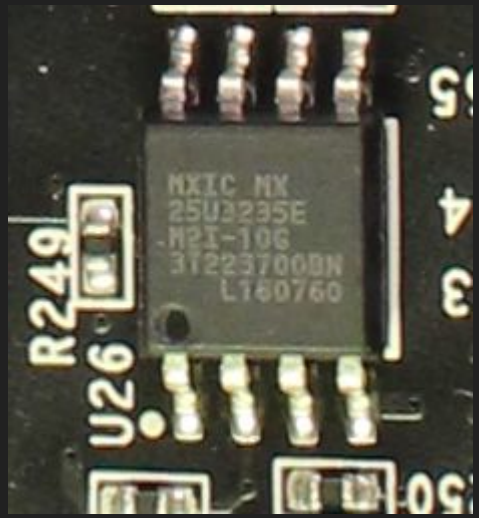
- JTAG
- SWD
- EJTAG



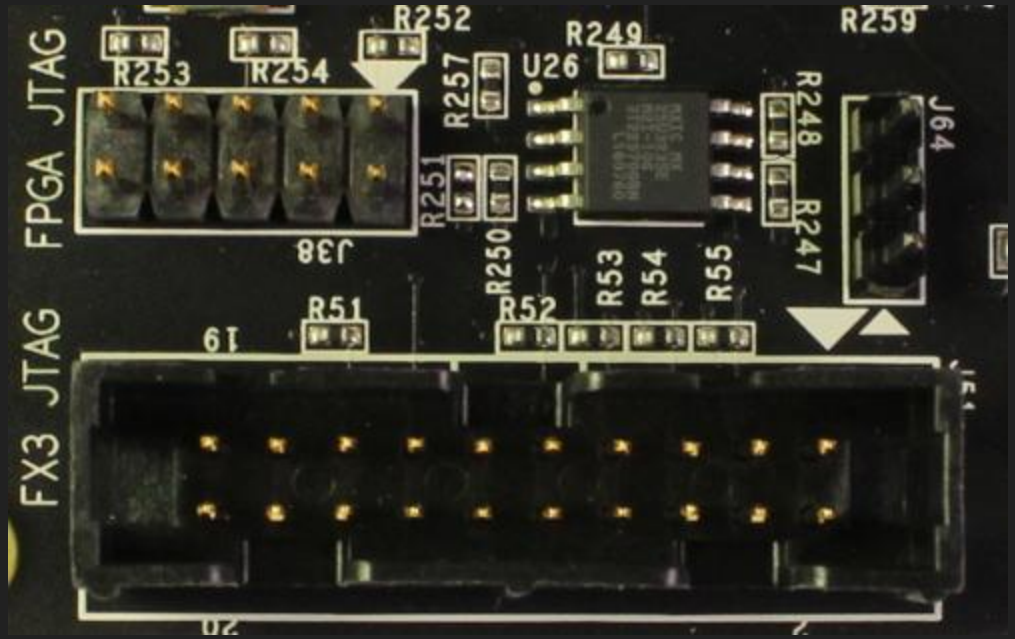
Mission: Dump Firmware

Flash

- SPI
- I2C
- eMMC
- Parallel

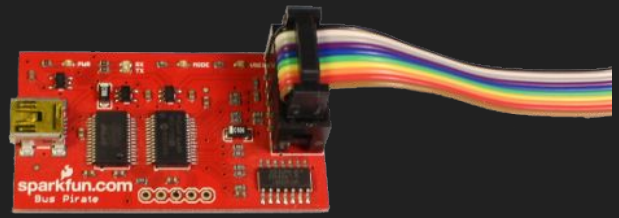


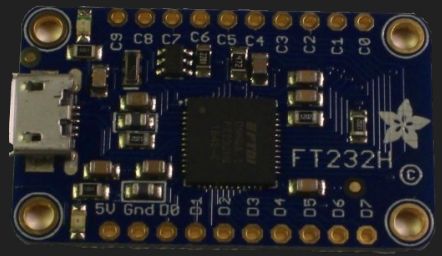
Mission: Dump Firmware



Mission: Dump Firmware (Interfaces)

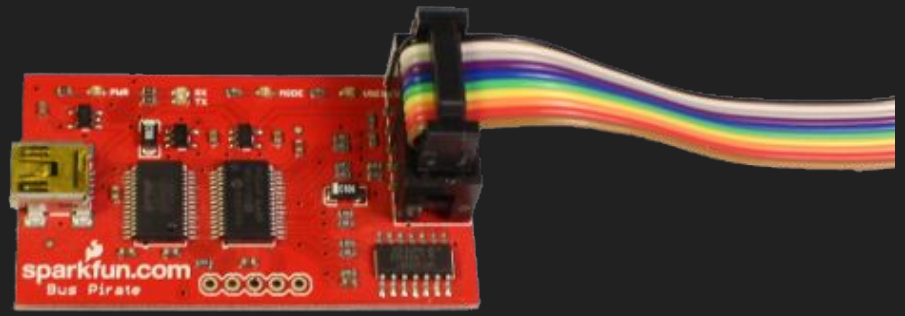
- Bus Pirate
- FT*232H
  - FT232H
  - FT2232H
  - FT4232H





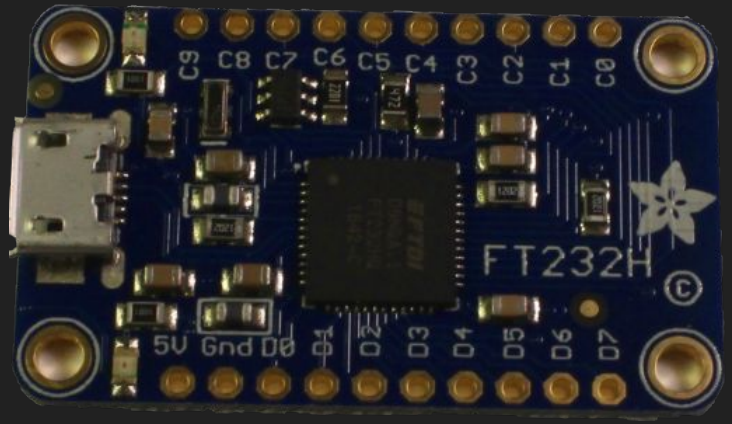
Mission: Dump Firmware (Bus Pirate)

- Bitbanging Microcontroller
- Many Protocols
  - 1-Wire
  - JTAG
  - UART
  - I2C
  - SPI
- Slow



Mission: Dump Firmware (FT*232H)

- Variety of Boards
  - TUMPA
  - Shikra
  - FTDI Breakout
  - Adafruit
  - FTDI Cable
- Variety of Protocols
  - UART
  - JTAG
  - Parallel Bit Bang
  - I2C
  - SPI



Mission: Dump Firmware

JTAG/SWD:

1. Connect via JTAG
2. Halt CPU
3. Dump Address Space
4. Carving Tools to Locate Flash

SPI/I2C Flash:

1. Connect to Flash with Target Unpowered or Halted
2. Dump Flash Memory on Host

NAND/NOR Flash:

1. Desolder Flash
2. Insert into reader
3. Dump flash

Mission: Find Vulnerabilities in Firmware

- Most Images
  - Compressed
  - Multiple Partitions
  - Some Raw
  - Some Filesystems
- Sometimes
  - “Encrypted” (Obfuscated)
  - Encrypted
  - Signed
- Binwalk is **the** tool of choice

```
DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
0             0x0             uImage header, header size: 64 bytes, header CRC:
0x589E011E, created: 2017-10-27 04:04:40, image size: 1209713 bytes, Data Addres
s: 0x80060000, Entry Point: 0x80060000, data CRC: 0x17CBA881, OS: Linux, CPU: MI
PS, image type: OS Kernel Image, compression type: lzma, image name: "MIPS OpenW
rt Linux-3.18.27"
64            0x40            LZMA compressed data, properties: 0x6D, dictionary
 size: 8388608 bytes, uncompressed size: 3648548 bytes
1245184       0x130000        Squashfs filesystem, little endian, version 4.0, c
ompression:xz, size: 7521138 bytes, 2419 inodes, blocksize: 262144 bytes, create
d: 2017-10-27 04:04:42
```

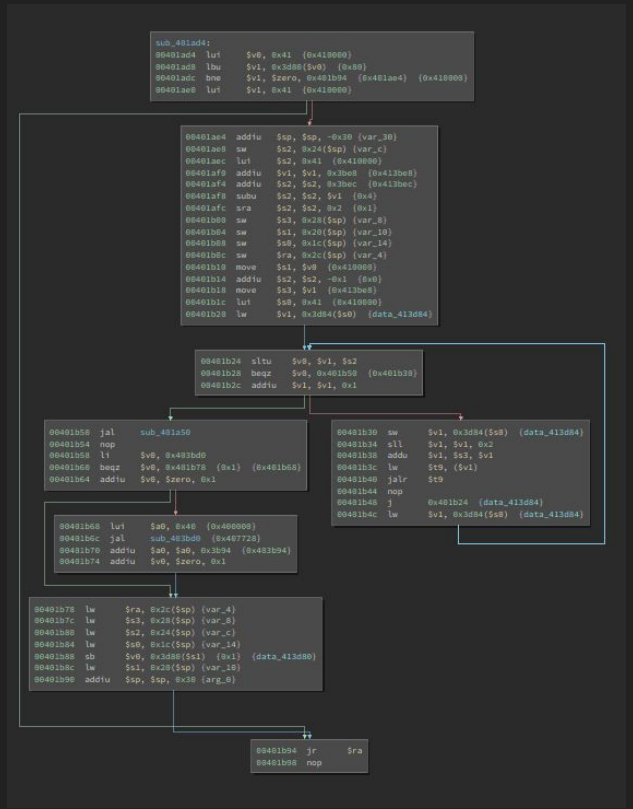
Mission: Find Vulnerabilities in Firmware

- OS Level Issues
  - Credentials
  - Service Configuration
- Web Interface
  - Scripting Language
  - Configuration

```
root:$1$$0xcwSNlRoTaNOIieiMl0v/:10933:0:99999:7:::
bin:*:10933:0:99999:7:::
daemon:*:10933:0:99999:7:::
adm:*:10933:0:99999:7:::
lp:*:10933:0:99999:7:::
sync:*:10933:0:99999:7:::
shutdown:*:10933:0:99999:7:::
halt:*:10933:0:99999:7:::
uucp:*:10933:0:99999:7:::
operator:*:10933:0:99999:7:::
nobody:*:10933:0:99999:7:::
default::10933:0:99999:7:::
```

Mission: Find Vulnerabilities in Firmware (Binaries)

- Static Analysis (Disassembler)
  - IDA - $$$
    - Expensive, most featureful
  - Binary Ninja - $
    - Upcoming alternative to IDA
  - radare2 - Free
    - Console interface, lots of architecture support
- Dynamic Analysis
  - User mode qemu

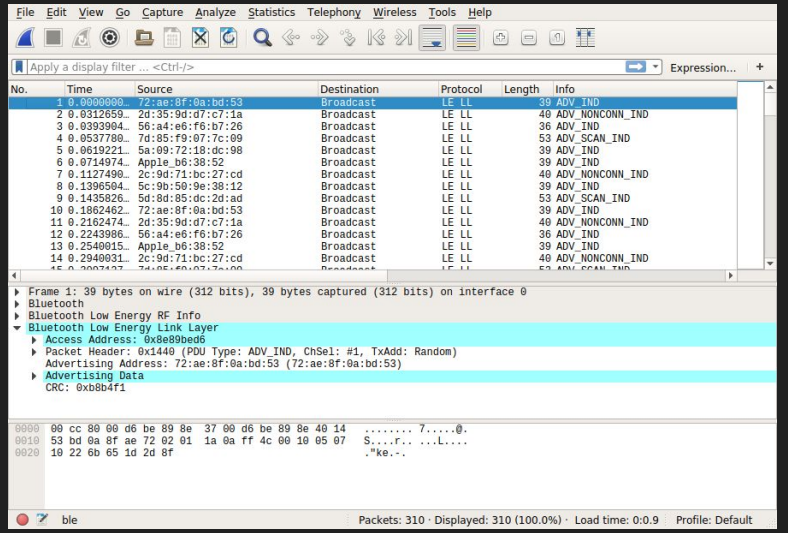


Mission: Observe Bluetooth Communications

- Cannot sniff bluetooth with traditional chipset
- Ubertooth One is designed to sniff
- Limited ability to replay
- Nordic Semiconductors also has sniffer firmware
- Does not work as well

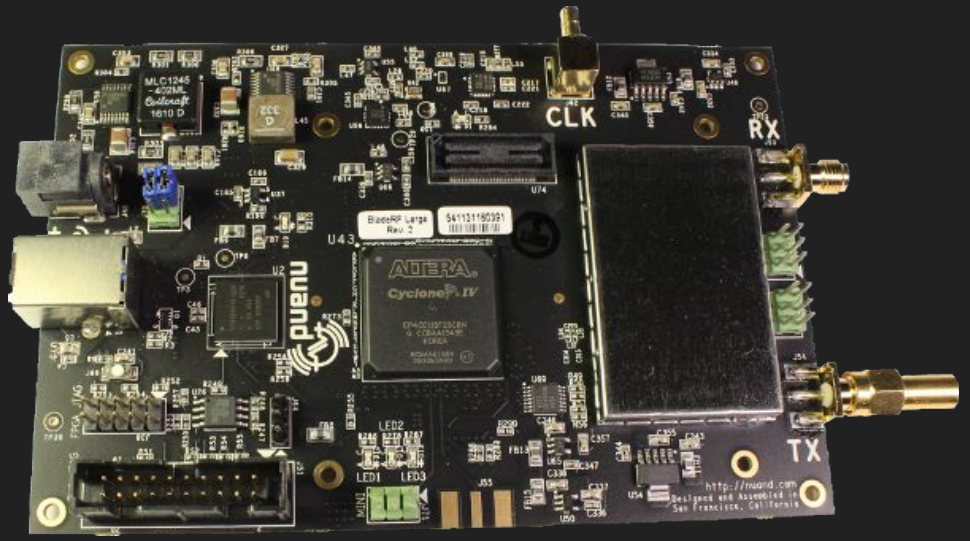
Mission: Observe Bluetooth Communications

1. Sniff advertising channels to find target
2. Set sniffer to follow target
3. Examine communications in Wireshark



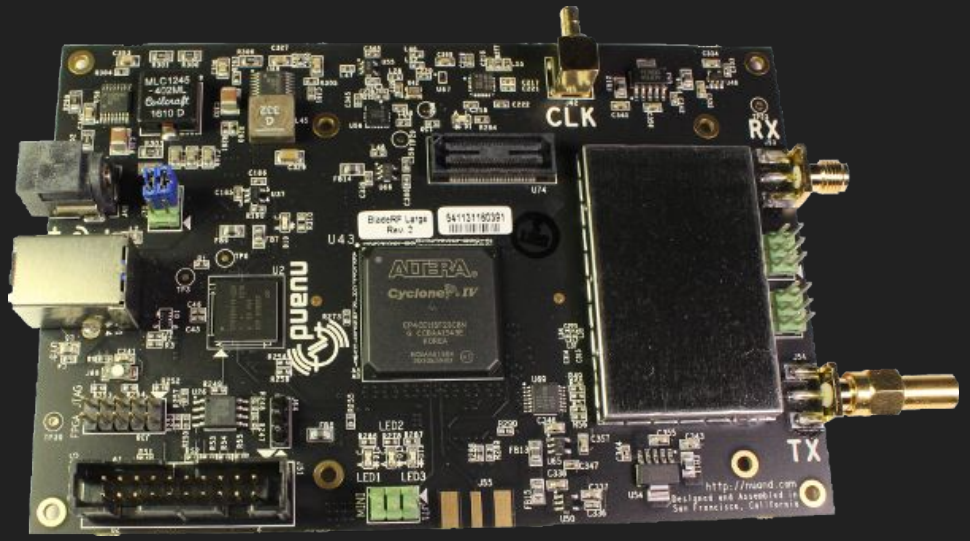
Mission: Observe & Replay Proprietary Wireless

- Software Designed Radio
  - Wideband Radio Chipset
  - Properties adjustable in Software
  - Processing Offloaded to FPGA
- Many open source building blocks
  - GNU Radio
  - Osmocom SDR
- Beyond that…
  - Prepare for Math and Physics
  - Need to understand modulations
  - Consider Amateur Radio License



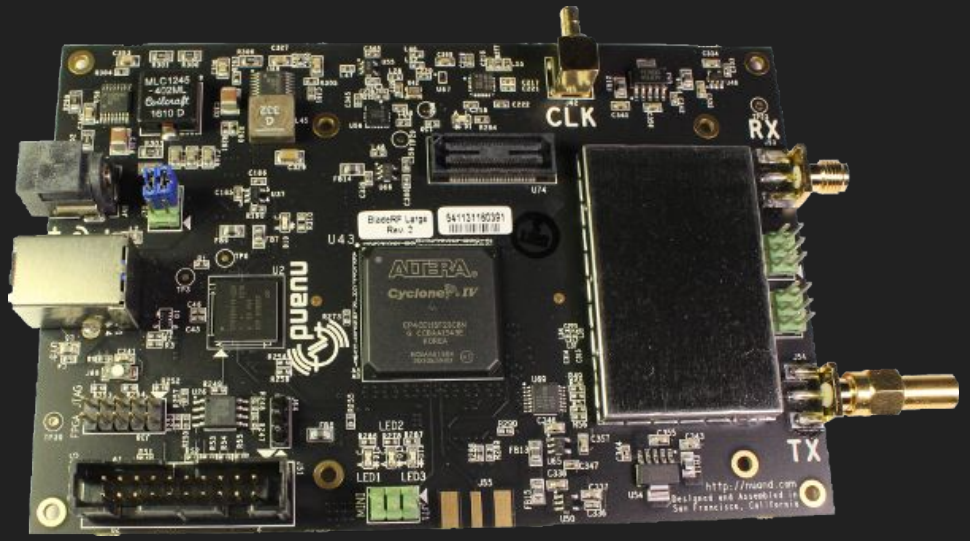
Mission: Observe & Replay Proprietary Wireless

1. Hook up SDR
    - Computer
    - Appropriate Antenna
2. Find signal of interest
    - Time correlation
    - Frequency known
    - Transmit strength
3. Explore demodulation
    - Analysis of exported data
    - Trial & Error



Mission: Observe & Replay Proprietary Wireless

4. Build demodulator (+ modulator?)
    - GNURadio
    - Modules in Python
5. Decode
    - Using GNURadio
6. Transmit
    - Be mindful of local regulations
    - SDRs are relatively weak transmitters
    - Use the right antenna



Other Tools

Logic Analyzer

- Tap for Physical Busses
  - UART
  - I2C
  - SPI
  - CAN
- Like Wireshark for Hardware

Logic Analyzer

- Identify
  - Pinouts
  - Protocols
  - Settings

Power Supply

- Avoid a mess of dedicated power supplies
- Power individual boards
- 5V, 12V, 3.3V are most useful
- Kits from old ATX PC PSUs
- Switching Power Supplies

Toolkit Summary

Software

Disassemblers/Decompilers

- IDA
- Binary Ninja
- Radare2

Firmware Reversing

- Binwalk
- Extraction Tools
    - Tar, Zip, etc.

Fuzzing/Testing

- QEMU
- American Fuzzy Lop (AFL)

Interfacing

- OpenOCD
- flashrom

Software

Networking

- MITM
  - Bettercap
  - Ettercap
- Packet Inspection
  - Wireshark
  - tcpdump
- HTTP Proxy
  - Burp Suite
  - OWASP Zap

RF

- Wireshark
- GNU Radio
- Osmocom SDR

Hardware

General Tools

- Screwdriver Set (w/Special Bits)
- Multimeter
- UART Cable
- Soldering Iron
- Jumper Wires

Interface Tools

- UART Cable
- Universal Interface
  - FTDI Breakout
  - Buspirate
- Flash Dumper
  - FTDI Breakout
  - Xeltec Dumper

Hardware

Networking

- Ethernet
- Wireless

Chip/Board-Level

- Logic Analyzer
- Oscilloscope
- Bench Power Supply

RF Tools

- Bluetooth Sniffing
  - Ubertooth One
  - Bluefruit/Nordic Sniffer
  - Commercial Sniffers $$$
- Software Defined Radio
  - RTL-SDR
  - HackRF
  - BladeRF
  - LimeSDR

References

- Hardware Hacking: Abusing the Things
- SecuringHardware.com
- HardwareSecurity.training

Questions?

- Slides & Blog Post: https://1337.fyi/toolkit
- Blog: https://systemoverlord.com
- Twitter: @Matir