Doing the IOT Penetration Testing – The right Way!

Yogesh Ojha

/Speaker/yogeshojha/KazHackStan> whoami

USER INFORMATION

Yogesh Ojha
From Nepal 🇳🇵
Cyber Security Analyst
Tata Consultancy Services India

Primary Research area includes
IoT Security, Hardware Hacking
and mobile application security

Medium
https://medium.com/@yogeshojha

Expectations/Agenda

- Understanding the basics of IoT Security
- Trends in IoT Security
- Attack Surface Mapping for IoT devices
- Understanding Effective penetration testing methodology for IoT
- Common vulnerabilities in IoT components
- Some Demos

Definition of IoT

Wikipedia:

system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human to human or human to computer interaction

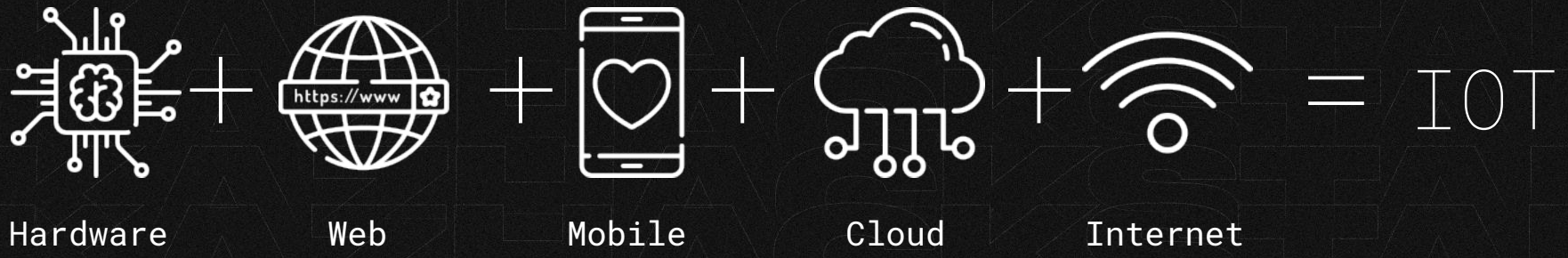
IBM:

concept of connecting any device to the Internet and to other connected devices. The IoT is a giant network of connected things and people — all of which collect and share data about the way they are used and about the environment around them

Gartner:

network of physical objects that contain embedded technology to communicate and sense or interact with their internal states or the external environment

Definition of IoT



Things in the Internet of Things

Current IoT security Problems

Current IoT security Problems

IoT Security ≠ Device Security

IoT at the moment

Home > News > Internet

Fancy Bear hackers used IoT devices to hack corporate networks

By Anthony Spadafora August 06, 2019 Internet

IoT devices provided an easy way int

Hack of Home Security System Highlights IoT Vulnera Issues

By Sue Walsh | September 19, 2019

ars TECHNICA SUBSCRIBE SIGN IN

STRONTIUM —

Microsoft catches Russian state hackers using IoT devices to breach networks

Fancy Bear servers are communicating with compro

IEEE SPECTRUM Topics

Tech Talk | Telecom | Internet

06 Sep 2019 | 19:10 GMT

IoT Security Risks: I Vibrators, and Kids' Toys Are Still Vulnerable to Hacking

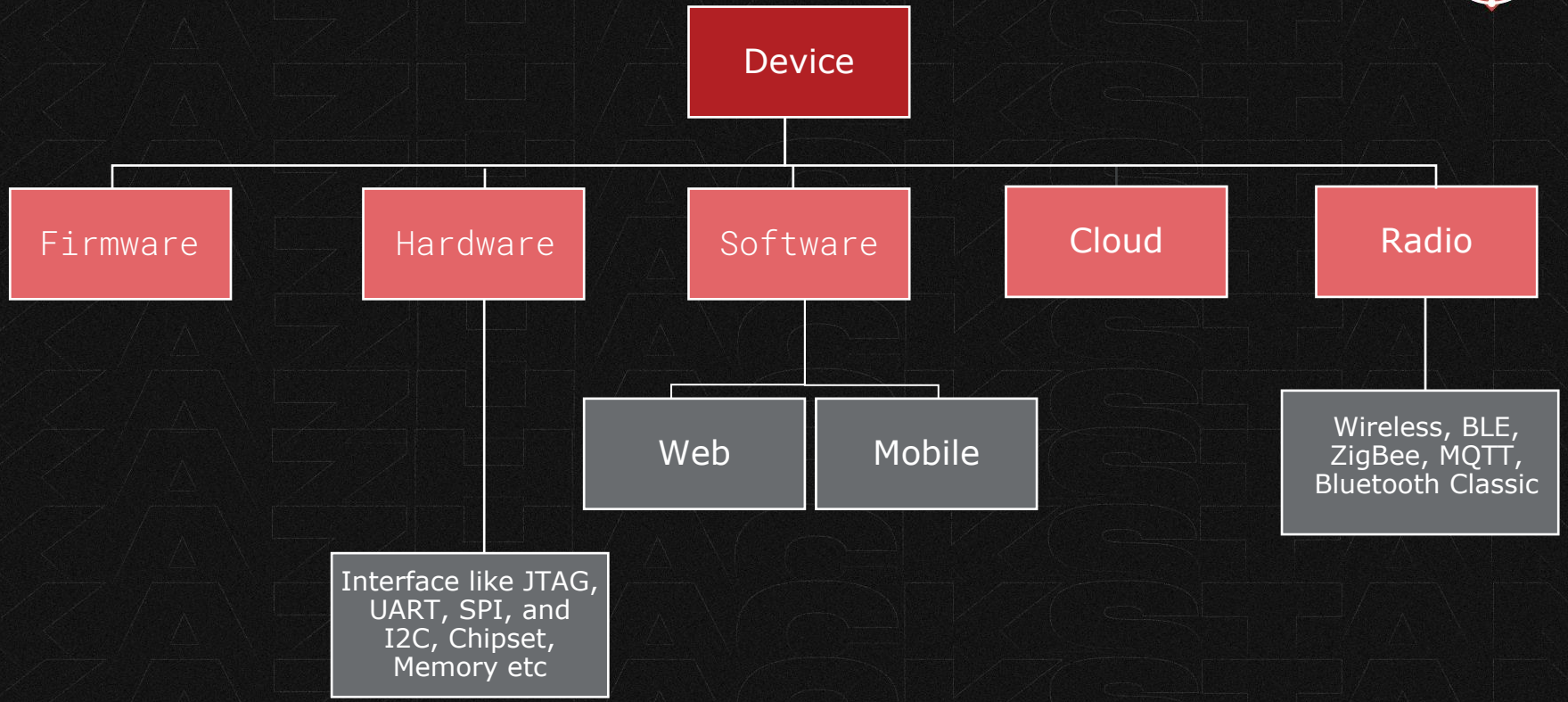
In a series of experiments, researchers showed how to intercept transmissions and hoist control of popular Internet of Things devices

IoTnews

Trend Micro blocked five million IoT camera hack attempts

Where to start?

Scope of IoT Testing



Current IoT security Problems

Firmware

Missing encryption, Missing Firmware validation, Hardcoded Sensitive information inside Firmware

Current IoT security Problems

Firmware	Missing encryption, Missing Firmware validation, Hardcoded Sensitive information inside Firmware
Hardware	Open debug ports, plain text communication in Bus, Insecure Storage

Current IoT security Problems

Firmware	Missing encryption, Missing Firmware validation, Hardcoded Sensitive information inside Firmware
Hardware	Open debug ports, plain text communication in Bus, Insecure Storage
Web	Good old XXE, XSS, CSRF etc

Current IoT security Problems

Firmware	Missing encryption, Missing Firmware validation, Hardcoded Sensitive information inside Firmware
Hardware	Open debug ports, plain text communication in Bus, Insecure Storage
Web	Good old XXE, XSS, CSRF etc
Mobile	Insecure API, Missing Authentication, Lack of Obfuscation

Current IoT security Problems

Firmware	Missing encryption, Missing Firmware validation, Hardcoded Sensitive information inside Firmware
Hardware	Open debug ports, plain text communication in Bus, Insecure Storage
Web	Good old XXE, XSS, CSRF etc
Mobile	Insecure API, Missing Authentication, Lack of Obfuscation

Current IoT security Problems

Firmware	Missing encryption, Missing Firmware validation, Hardcoded Sensitive information inside Firmware
Hardware	Open debug ports, plain text communication in Bus, Insecure Storage
Web	Good old XXE, XSS, CSRF etc
Mobile	Insecure API, Missing Authentication, Lack of Obfuscation
IoT	= Hardware + Software + Cloud + internet

Effective IoT Pentesting Methodology

- Evaluation
- Device Reconnaissance
  - Without Teardown
  - Teardown
- Mobile, Cloud & Web APIs
- Firmware reverse engineering
- Network
- Non-Invasive Hardware Attacks
- Radio (RF)

Evaluation & Device Reconnaissance

Evaluation

Understanding what the device does…

Any Visible ports? USB, UART, Anything else?

Find out the different components(Mobile, Web, Any Sensors, whatever component) and the communication medium they interact through (BLE, Internet, ZigBee, MQTT)

Are there any web end points? Your mobile app communicating to device via internet?

Map out features, functions, components, and communication path
Probably an architecture diagram?

Evaluation & Device Reconnaissance

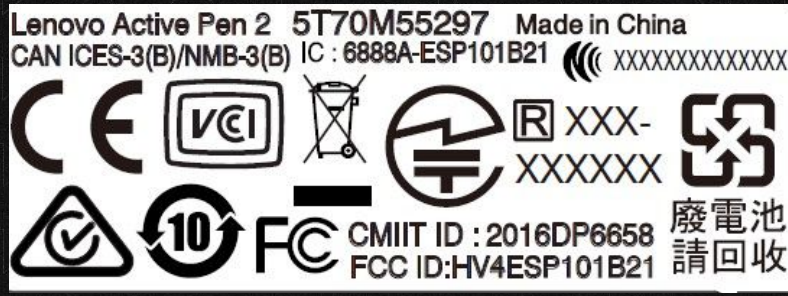
Device Reconnaissance without tearing up the device
Component version, Hardware version, Software version, Operating System Used(Mostly Linux)

Find out Chipset Used
Once you have the chipset name/number, look for the datasheet

FCC Data -> https://fccid.io/ many times, this will reveal wealth of information about the device

Circuitry connection
- UART
- JTAG
- SPI

#A quick Demo



Teardown

Get your screwdriver!
Look for the screws behind the rubber pads or labels

Have a look at the chipsets used, use phone’s flashlight to read the component’s name/number
Use google to find out more information on chipset used and look for datasheet

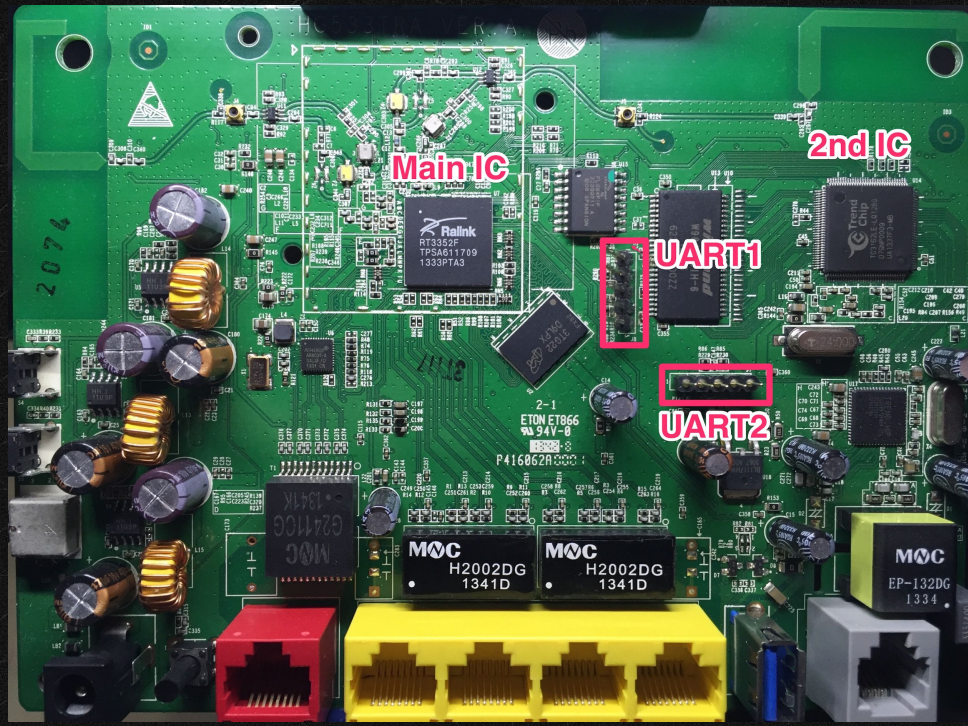
Evaluation & Device Reconnaissance

Device Reconnaissance after tearing up the device

Look for Physical Ports
- USB
- Serial
- Ethernet

Circuitry Connection
- UART - Usually 3-4 pins
- JTAG - 6,12,13,20 pin header
- SPI - indicates the presence of a flash chip

De-Solder the IC for extracting firmware



Firmware

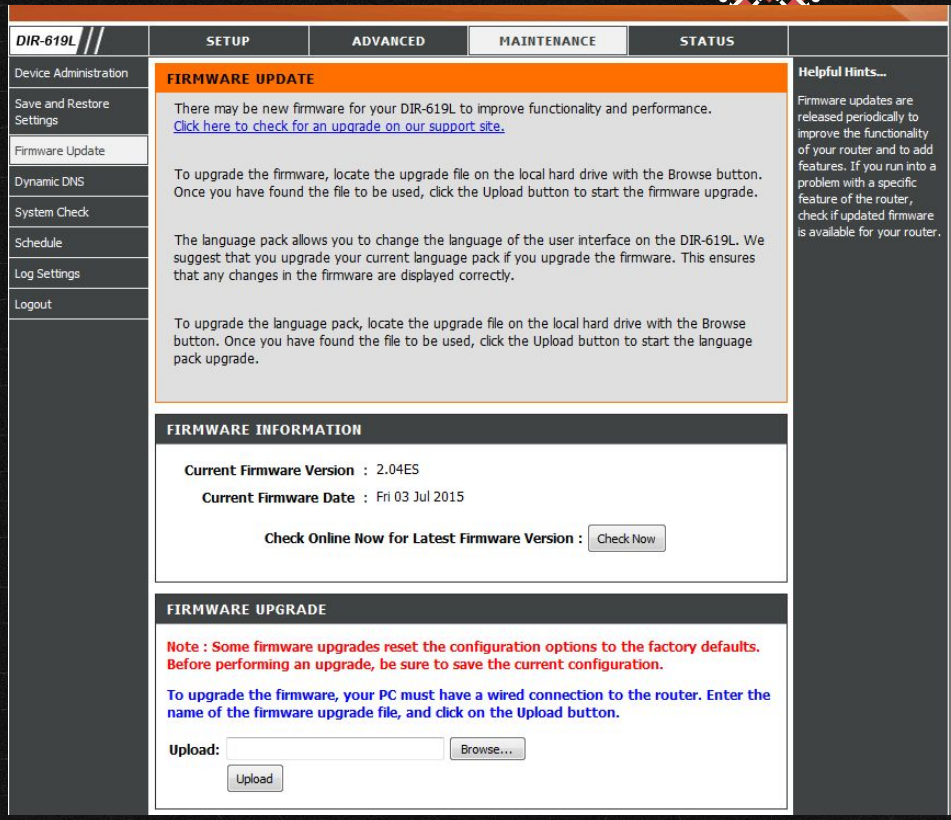
Firmware

Any software on your IoT device, responsible for running the IoT

Obtaining the firmware

- Dumping from Device
- Vendor's Website
- Support Groups & forums
- RE Mobile Application
- Download from vendor FTP server or search on ftp index sites
- Capture the firmware during update, traces of DFU from wireshark

Analysis of firmware before exploiting any hardware or software is important



What to look for in Firmware?

- Sensitive information about device
- Hardcoded SSIDs
- Hard-coded Passwords
- API tokens & endpoints
- Vulnerable services
- Firmware OTA update URLs
- Configuration files
- Source code
- Private keys
- Watch out for 3rd party libraries and SDKs

Firmware Analysis

Trying to identify as many security issues as possible

Firmware: bootloader + kernel + filesystem + additional resources

Find out the file system: $ hexdump - C firmware.XX | grep -i ‘hsqs’
hsqs is magic byte for squashfs

Use dd and unsquashfs to dump the contents of the firmware once squashfs is confirmed

Do this automatically using binwalk $ binwalk -e yourFirmware.bin

Firmware Analysis
Use Firmwalker : https://github.com/craigz28/firmwalker to look for interesting entries
Firmware Analysis Toolkit
From Attify https://github.com/attify/firmware-analysis-toolkit

Firmware Analysis FAQ

Can I emulate the firmware?
Yes, use Qemu and Chroot.
There are tools built on top of Qemu like firmadyne, FAT by attify that does almost everything like finding CPU architecture, running binwalk etc automatically for you.

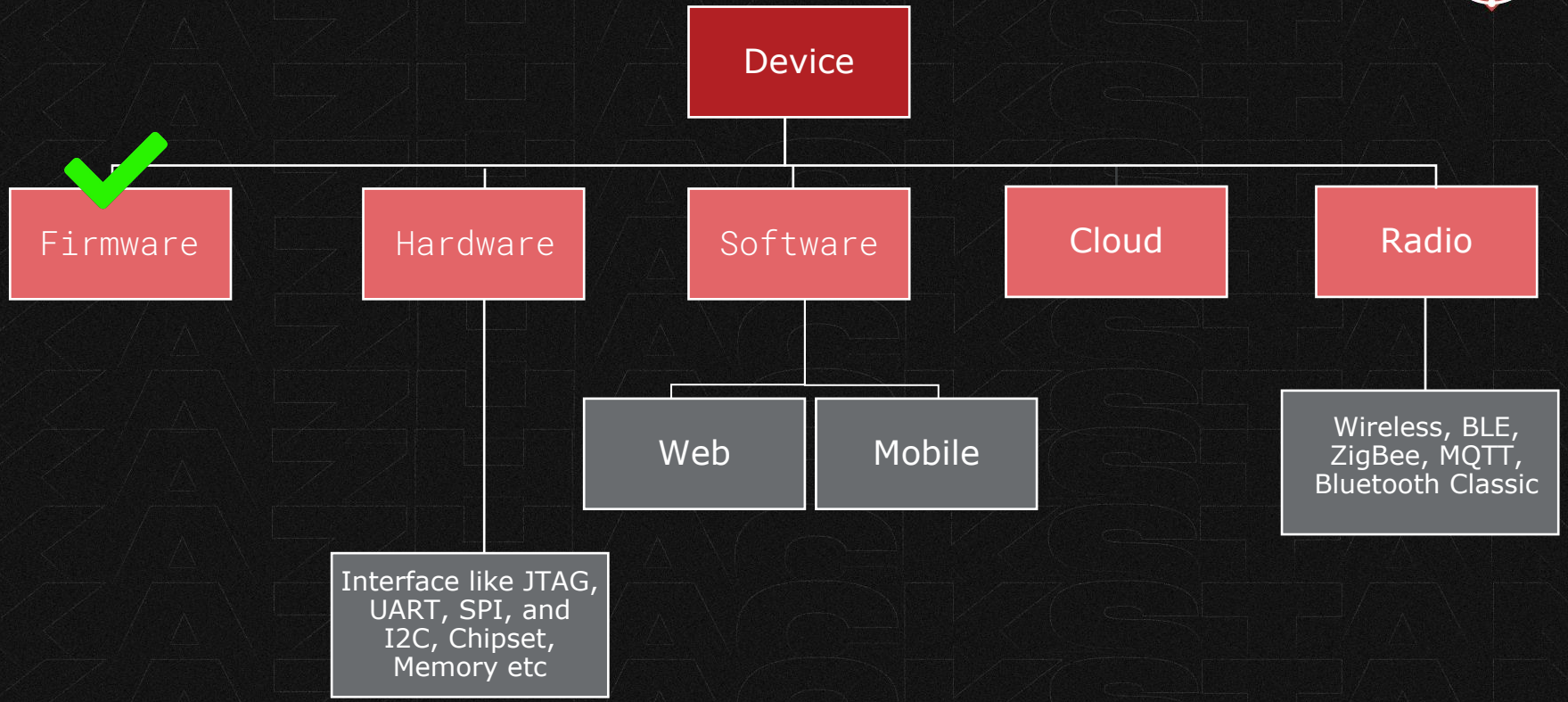
Can I modify the firmware?
Yes, use Firmware-Mod-Kit FMK

Find out if device detects firmware modifications?
If yes, it is missing firmware integrity verification

Firmware Emulation DEMO

Using Firmadyne and FAT.

Scope of IoT Testing



UART Identification

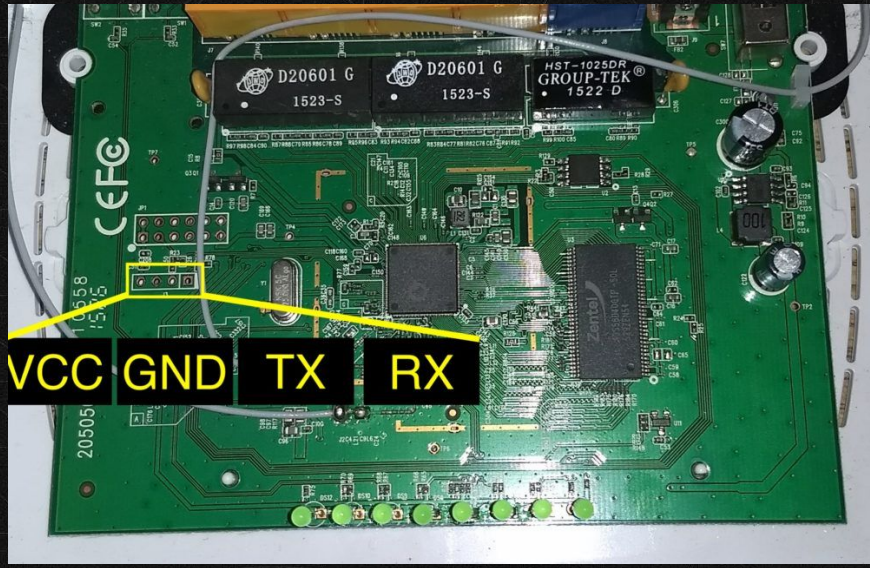
Actually being used by manufacturers for debugging/diagnostic purpose
UART - 3 or 4 pins VCC, GND, TX, RX
Goal is to Identify TX, RX, GND and VCC
GND and VCC are pretty easy to identify

Identifying TX
Get your multimeter

- One probe of your multimeter in the one of the pins and another probe in GND
- Reboot the device & measure the voltage between the remaining pins and GND (other than the Vcc and GND)
- Significant data transfer during bootup, notice the huge fluctuation in the voltage on one of the pins during boot process
  → TX

Identifying RX

- The remaining pin with lowest voltage fluctuation
  → RX



UART Exploitation

Once you have identified the pinouts for the Serial interface

- identify baud rate and use attify badge or any cheap usb2ttl
  Use Minicom to login to shell
- If you obtain the Shell
- Find out what all can be done from here
- Try dumping the firmware
- Try controlling the device components via the shell
- If the shell is authenticated, try brute forcing ;)

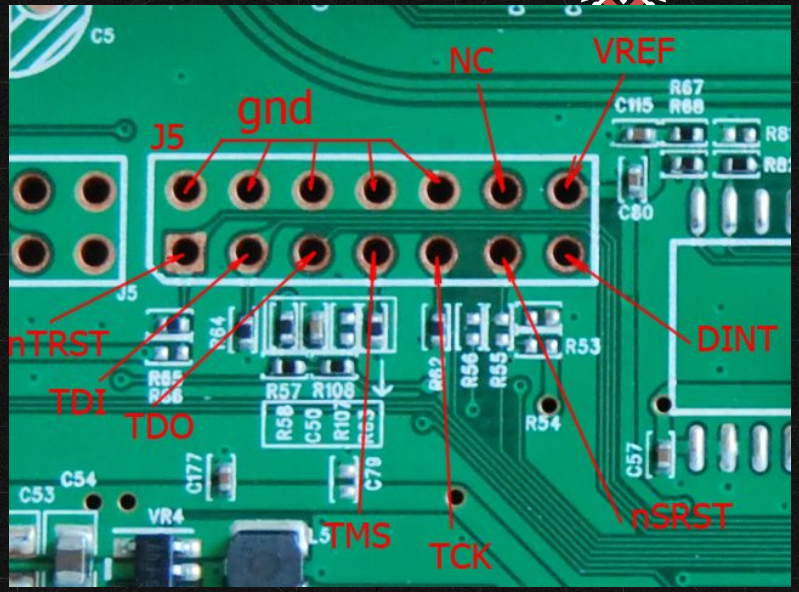
If UART is missing from PCB, look for the datasheet of the chipset used, trace the circuit, use multimeter to find TX and RX

JTAG Identification & Exploitation

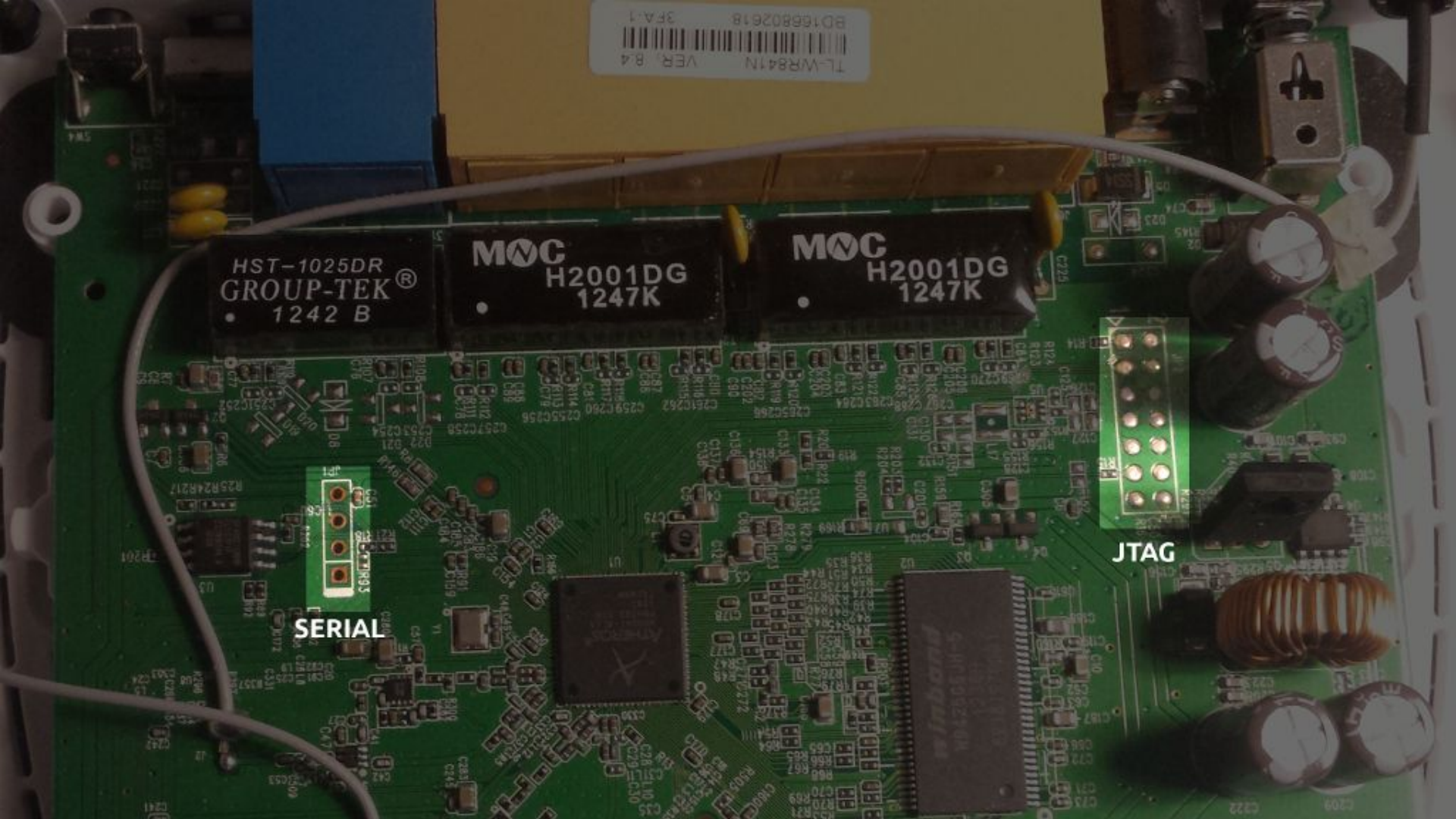
JTAG- 6,12,13,20 pin header
Use JTAGulator

Use it to dump firmware or write new firmware

Provides direct access to RAM and flash
Look for Test Data in (TDI),Test Data Out(TDO,
Test Clock (TTCK) and test mode select (TMS)




```
sudo screen /dev/ttyUSBX baudRate
```

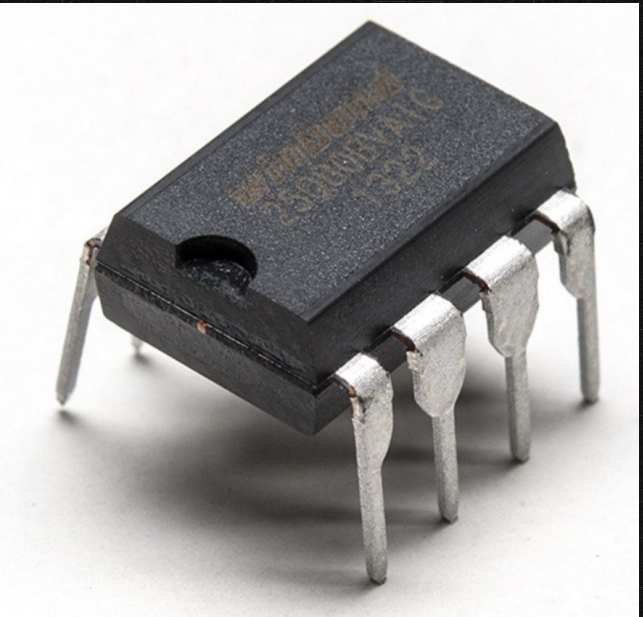

HST-1025DR
GROUP-TEK®
1242 B
MOC
H2001DG
1247K
MOC
H2001DG
1247K
SERIAL
JTAG

SPI Firmware/BIOS/Context Extraction

SPI and I2c falls under serial communication

Use flashrom and USB programmer to extract firmware or contents of SOIC8 SPI chip

```
sudo apt-get install flashrom
```



List possible chipset name

```
flashrom -p deviceXXXX
```

Extract Firmware/Contents

```
flashrom -p deviceXXX -c chipset
```

NAND Glitching

Used to bypass security measures (if no root shell on UART console)

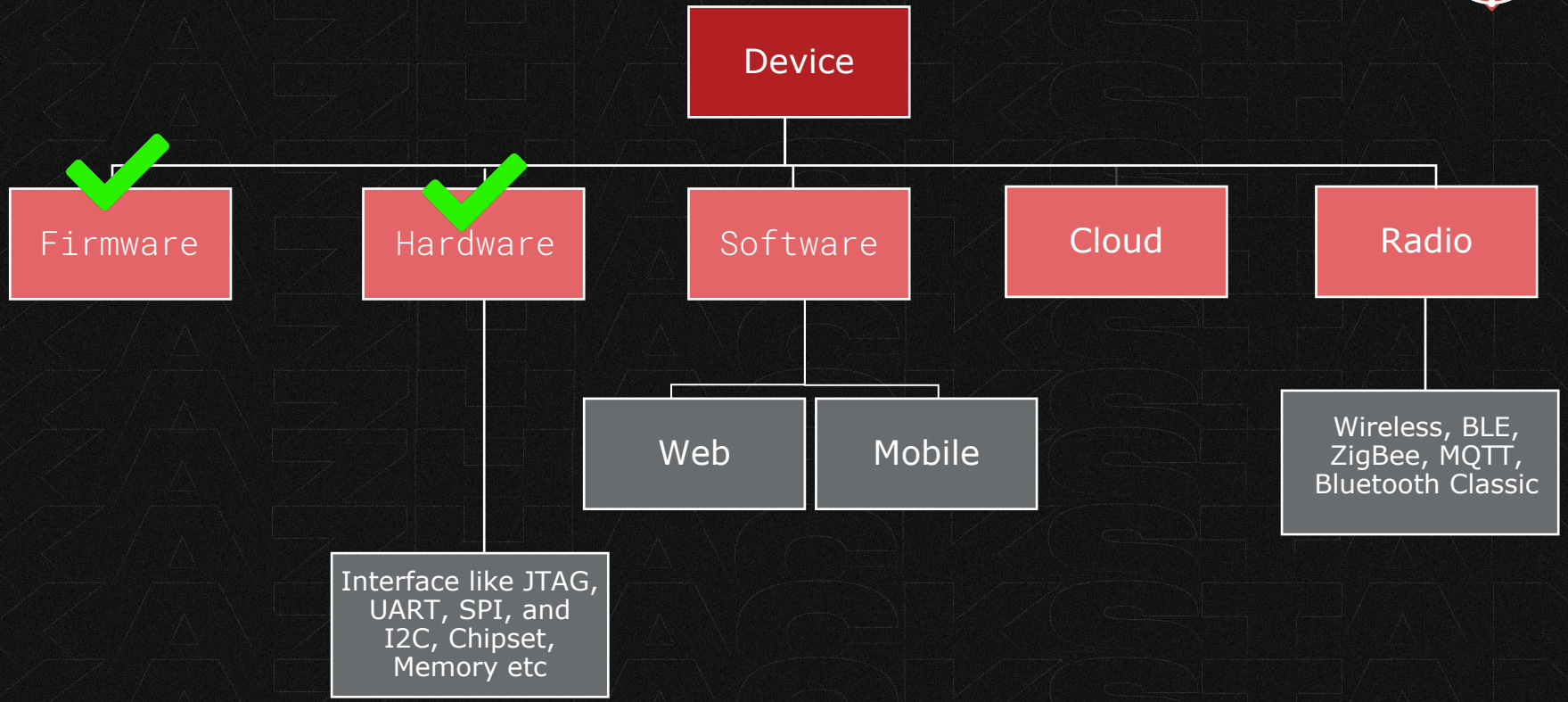
Short circuit one of the I/O pins of the device's NAND flash to a GND pin
This has to be performed as soon as the bootloader has booted and the kernel is about to boot up

If shorting works! kernel will fail to boot and thus causing you to drop to the default bootloader prompt

Further reading:

- https://www2.cs.arizona.edu/~collberg/Teaching/466-566/2012/Resources/presentations/2012/topic1-final/report.pdf
- https://www.cl.cam.ac.uk/~sps32/ECRYPT2011_1.pdf
- https://www.blackhat.com/docs/eu-15/materials/eu-15-Giller-Implementing-Electrical-Glitching-Attacks.pdf

Scope of IoT Testing



Identifying vulnerabilities in web console

Look for the good old bugs like XSS, SQLi, XXE, XSRF, IDOR etc
Use Burp Proxy to intercept, view and alter web traffic
Check for permission level bugs user, root, admin

Watch out for Command Injection

Identifying vulnerabilities in Mobile app

Mobile
Reverse engineer the mobile application, you may find entire logic on how device communicates with mobile app
Use jadx and apktool to RE mobile app
Use MobSF for static Analysis
Try Understanding the Java code

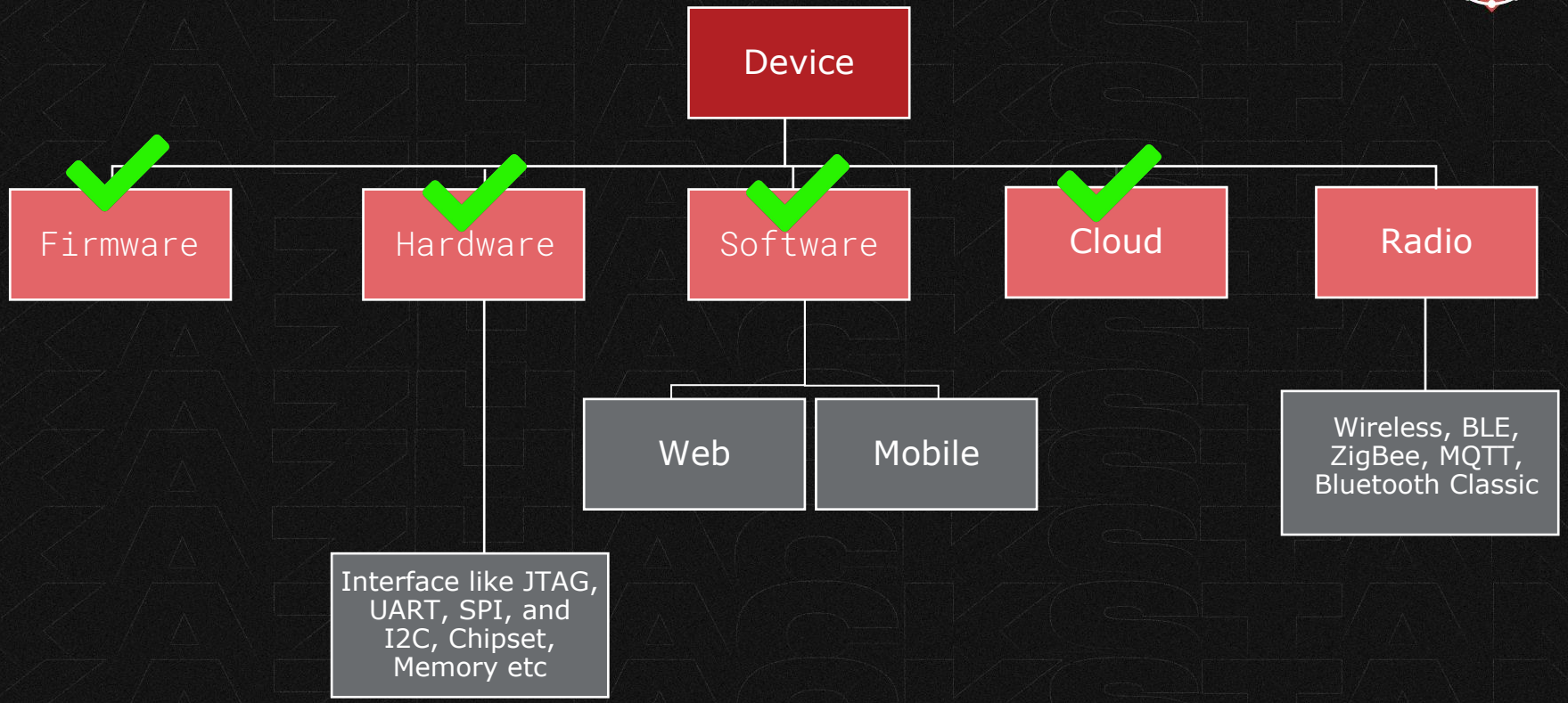
Common issues found in Mobile app

- hardcoded firmware download URLs
- Hardcoded SSIDs
- Hardcoded encryption keys
- Hardcoded username and password
- API URLS, port and much more

I would be surprised if you didn’t find anything useful after RE mobile app.

Many times, the mobile applications will have firmware required for DFU

Scope of IoT Testing



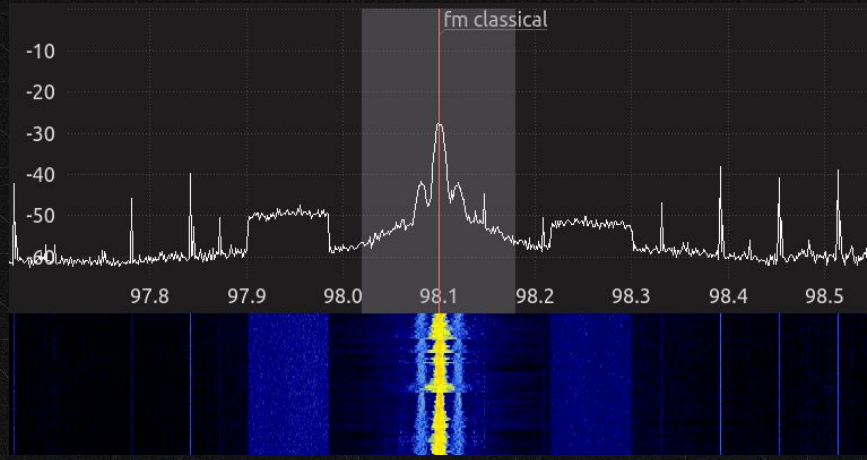
Identifying issues in Radio

Radio analysis requires special hardware and software
Different protocol require different h/w and s/w

Most commonly used are: BLE and ZigBee

What could be done with RF signals?

- Jamming based attacks
- Modifying and replay attack
- Sniffing the radio packets



Identifying issues in BLE

Straightforward process
Reverse Engineer the mobile app, this should give you enough information on which handle is data being written

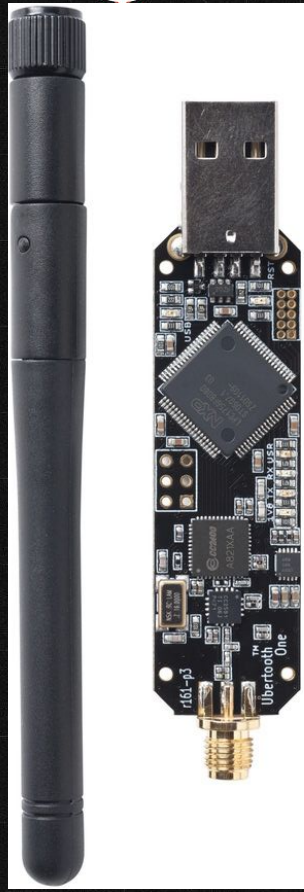
BLE Sniffer - Ubertooth $$$, Adafruit BLE Sniffer $$

Android HCIdump: $0

Use **gatttool** to rewrite those values on handles.

How I hacked Mi Band:
https://medium.com/@yogeshojha/i-hacked-xiaomi-miband-3-and-here-is-how-i-did-it-43d68c272391

Tools available: BTLeJuice, Gattacker



Identifying issues in BLE

Straightforward process
Reverse Engineer the mobile app, this should give you enough information on which handle is data being written

BLE Sniffer - Ubertooth $$$, Adafruit BLE Sniffer $$

Android HCIdump: $0

Use **gatttool** to rewrite those values on handles.

How I hacked Mi Band:
https://medium.com/@yogeshojha/i-hacked-xiaomi-miband-3-and-here-is-how-i-did-it-43d68c272391

Tools available: BTLeJuice, Gattacker

Identifying issues in BLE

Services: Set of provided features and associated behaviors to interact with the peripheral. Each service contains a collection of characteristics.
Characteristics:
Characteristics are defined attribute types that contain a single logical value.
You can use nrftool app to identify Services and Characteristics

Scan for LE

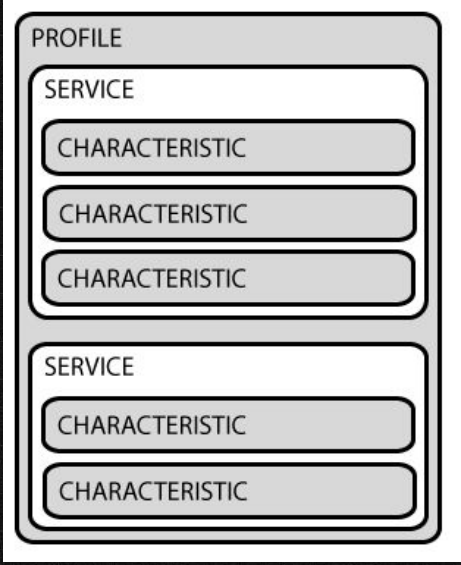
```
yogesh@yogesh:miBand3Hack$ sudo hcitool lescan
LE Scan ...
E1:E7:4E:DF:24:98 (unknown)
E1:E7:4E:DF:24:98 Mi Band 3
```

3 devices, out of 5 devices that I tested, did not have authentication!!!

Use 2 of these BLE 4.0 CSR Dongles with BTLEJuice to intercept BLE traffic

What to look for?

- Is replay possible?
- Is jamming possible?
- Is it possible to write in the handle using gatttool?
- Look for sensitive information being sent in clear text. (PIN Lock)



Identifying issues in Zigbee

2.4 GHz, 868 MHz(EU) or 944 MHz (US and AU)
Find ZigBee channel in which DUT is being operated

Use CC2532 $$ cheap ZigBee Sniffer
Also, Capture communication using zb_dump and analyze in Wireshark
Most of the times, communication could be encrypted
Capture the key exchange or find the key inside device or firmware
Try decrypting the communication

What to look for?
- Is replay possible?
- Sniff, MiTM possible?

Hardware: Atmel RzRaven USB Stick
KillerBee: Framework and Tools for Attacking ZigBee

https://github.com/riverloopsec/killerbee

Attacker Tools - Software

- Software Disassemblers
  - Ghidra
  - IDA
  - Binary Ninja
- Firmware Reverse Engineering
  - Binwalk
  - Any Extraction tools
- Fuzzing
  - QEMU
  - OpenOCD
  - Flashrom
- Minicom
- Protocol specific tools like can-utils
- Packet Inspection
  - Wireshark
- HTTP Proxy
  - Burp Suite - Yayy!!!
- Bluetooth
  - Bluehydra
  - Gattacker
  - BTLEJuice
- RF
  - Wireshark
  - GNU Radio
  - SDR
- Mobile reverse engineering
  - Apktool
  - jadx

Attacker Tools - Hardware

- General Toolkits
  - Screwdriver ;)
  - Multimeter
  - Soldering iron
  - Connectors/Cable/Wires
- Interface Tools
  - USB2UART
  - Flash Dumper
- RF Tools
  - Bluetooth Sniffing
    - Ubertooth One
    - Bluefruit/Nordic Sniffer
    - Commercial Sniffers $$$
  - Software Defined Radio
    - RTL-SDR
    - HackRF
    - BladeRF
  - Zigbee
    - CC2531 Sniffer

Conclusion

- Hardware Best Practices
  - Disable UART in production
    Case Study: One of the Xiaomi router enables the UART during the first boot after firmware is flashed, then completely disables it. Possible Solution
  - Disable JTAG in production
  - Encrypt firmware and data by using Trusted Platform module

- Software Best Practices
  - Data in transit must be encrypted end to end using SSL/TLS
  - Data in rest should be stored encrypted and stored in Tamper-resistant chips
  - Harden the RE process

Thanks

Further Reading

Follow these awesome people/talk/group/organization/blog/books for IoT Security

- Attify
- Pentesting Hardware And IoT by Mark Carney
- DEF CON 23 - IoT Village - Daniel Miessler - IoT Attack Surface Mapping
- IoT Penetration Testing by KreischerMiller
- IoT Penetration Testing Cookbook
- http://iotpentest.com/
- https://www.iotpentestingguide.com
- https://github.com/V33RU/IoTSecurity101
- https://www.exploitee.rs/

Practice on https://github.com/Vulcainreo/DVID