"Smart" Vacuum Cleaners

An Audit Into The Security and Integrity of IoT Systems

Andrew Wong | UNSW Sydney

Introduction

Internet of Things (IoT) and Smart Home devices are everywhere.

Q: Can we completely trust a device's {security, privacy}?

A: no

- Developers are humans.
  - Humans make mistakes.
    - Developers make ~~mistakes~~ bugs
- Or maybe secret company agendas?

We should always verify and test things where possible!

About Me

Andrew Wong

Computer Engineering @ UNSW Sydney

e: andrew.j.wong@student.unsw.edu.au

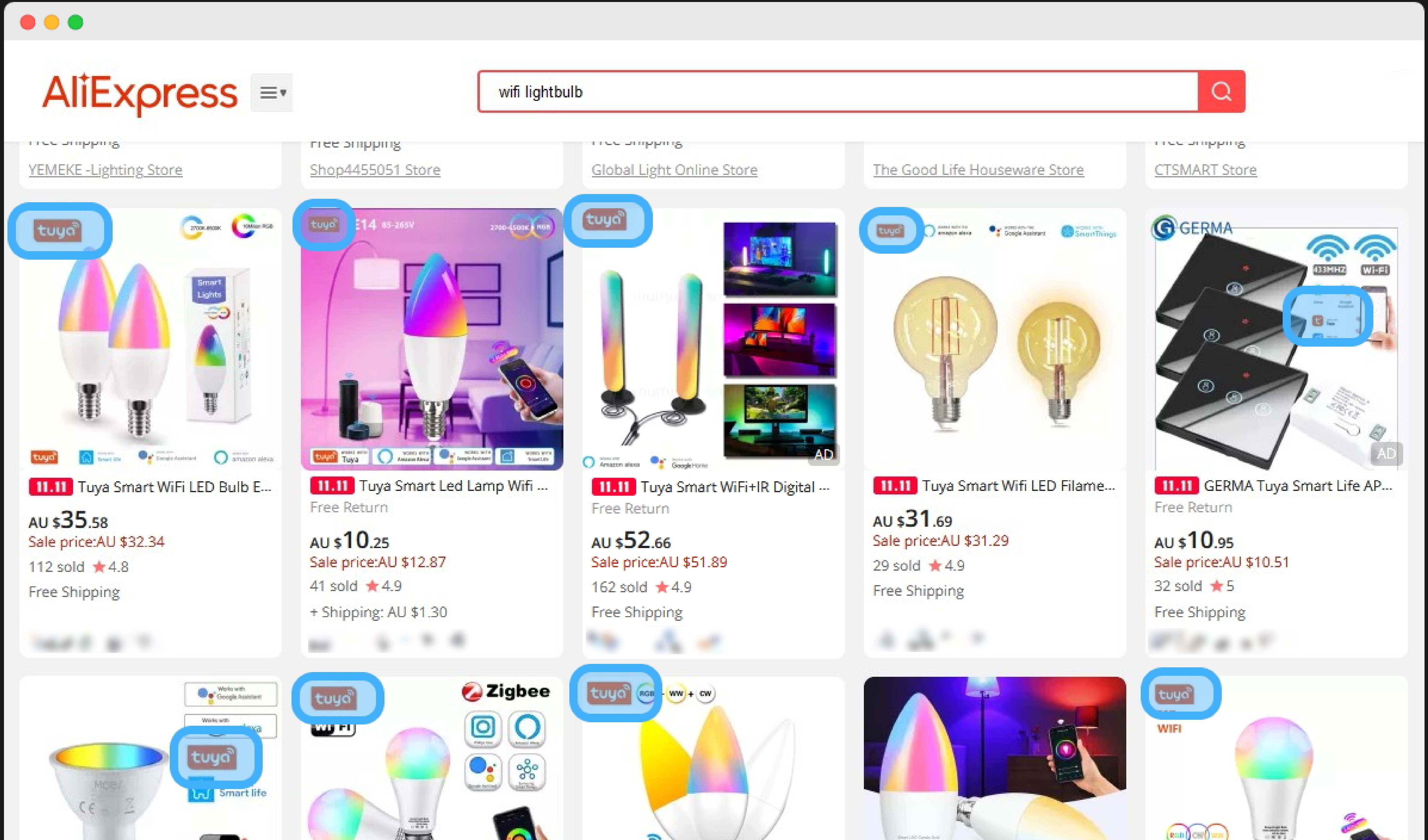
Interests

Making things, breaking things… mainly the latter

Background Information

Background Information

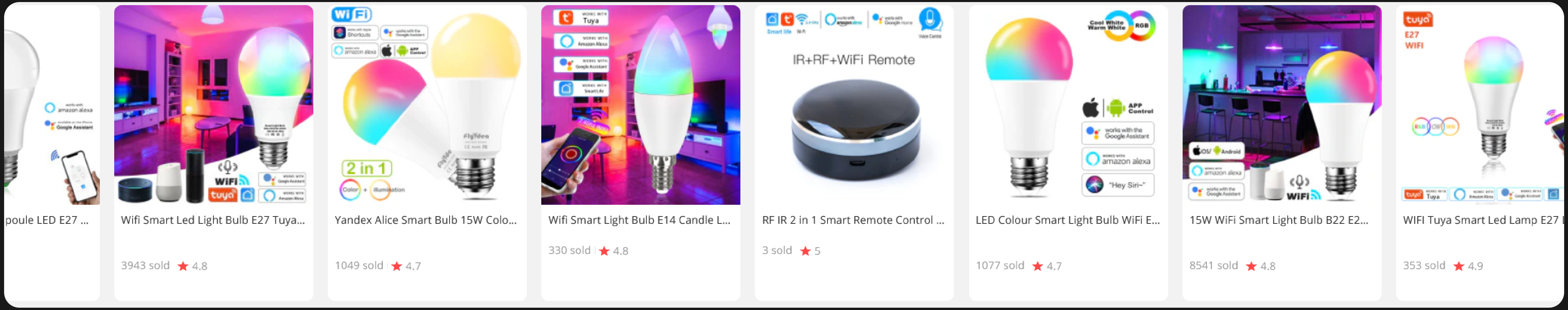
Widespread availability of IoT brands



Background Information

Widespread availability of IoT brands

- IoT manufacturers sell their products to vendors
  - The product itself
  - Cloud infrastructure
  - Smartphone application
- White-label vendors buy a generic product
  - Rebrand and sell products under their name



Background Information

Widespread availability of IoT brands

Vulnerabilities in IoT infrastructure

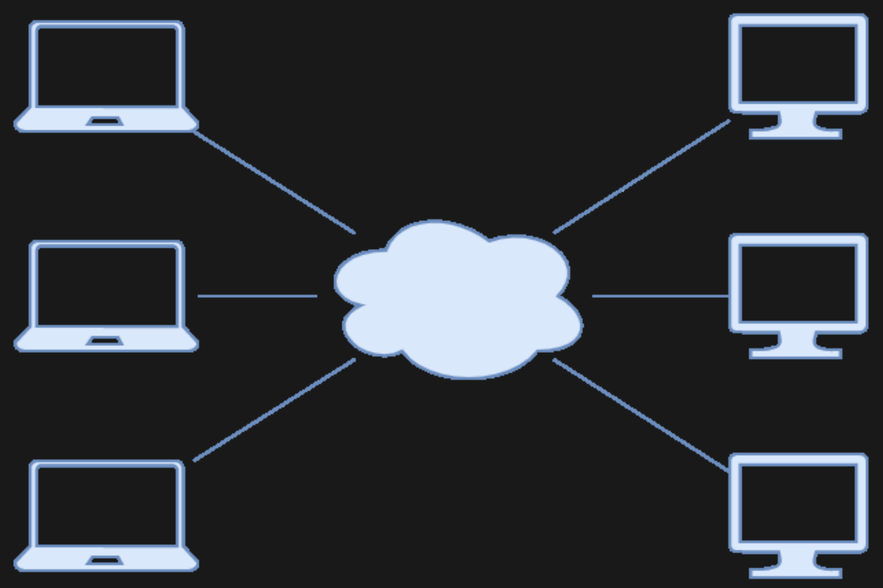
=

Vulnerability in all white-label products

Background Information

Centralisation and IoT Manufacturers as “Data Giants”

- Same IoT cloud infrastructure used by white-label vendors
- Data and network activity is all centralised / standardised
- Privacy concerns - Who, What, Where, When, Why?
- Infrastructure outage = really *really* big outage..



Background Information

Centralisation and IoT Manufacturers as “Data Giants”

- Reverse engineering of cloud communications protocols / API
  - e.g. MiIO protocol (link)
- Decoupling of devices from the necessity of internet / IoT cloud
  - HomeAssistant - Home Automation (link)
  - OpenHAB - Home Automation (link)
  - Valetudo - Cloud-less vacuum cleaner control interface (link)
  - DustCloud - Xiaomi Cloud Emulation (link)
  - MiCloudFaker - Xiaomi Cloud Emulation (link)
  - tuya-convert - Flash Tuya devices to custom firmware (link)

About The Company

About The Company

roborock

- Robotic home cleaning appliances
- Founded in July 2014, Beijing
- Partnered with Xiaomi in September 2014
  - Investments + Partnership

About The Company

- September 2016 - Mi Home Robotic Vaccuum Cleaner
  - Very first product!
- ⋮ Roborock S5, E2, E3
- June 2019 - Roborock S6
- ⋮ Roborock S5 Max, S4, S6 Pure, S6 MaxV, E4, S4 Max
- January 2021 - S7

About The Device

Roborock S6 Vacuum Cleaner

About The Device

Roborock S6 Vacuum Cleaner

Specifications

- CPU: Allwinner R16 Quad-core ARMv7
- ACU: STM32F103VC
- RAM: 512 MB
- Flash: 4 GB eMMC
- Wireless: RTL8189ETV (802.11 b/g/n)
- Cloud: Tuya / Xiaomi
- OS: Ubuntu 14.04

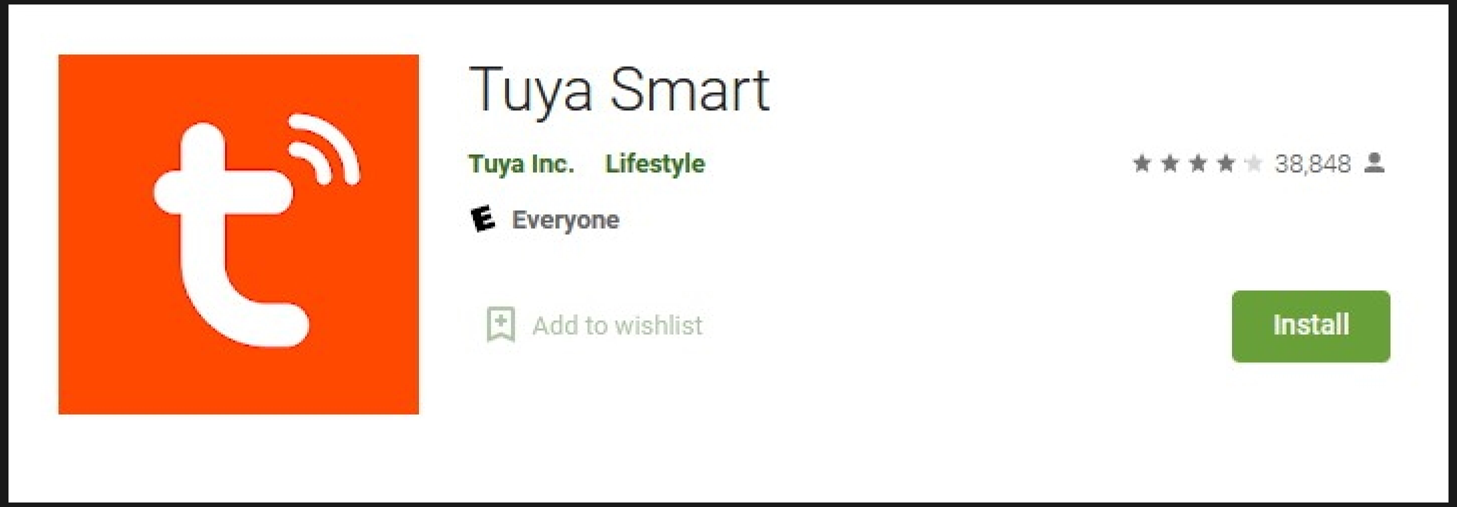
About The Device

Roborock S6 Vacuum Cleaner

Cloud Capability

Roborock (Xiaomi Cloud)

Tuya Cloud



About The Device

Roborock S6 Vacuum Cleaner

IoT infrastructure vulnerability (15/09/2021)

https://global.roborock.com/pages/disclosure-security-vulnerability-on-tuya-iot-cloud

Disclosure: Security Vulnerability on Tuya IoT Cloud (Resolved)

Sep 15, 2021

Overview

Roborock vacuum cleaners (i.e. devices) connect to either Tuya IoT cloud or Roborock IoT cloud depending on the version of the firmware and Roborock app. For those devices connect to Tuya IoT cloud, the device side library uses an insecure random number generator when negotiating communication channel with the Tuya IoT cloud. This vulnerability affects a portion of Roborock product models globally. Those devices connected to Roborock IoT cloud are not affected by this vulnerability.

Threat

This issue undermines the security of the user data transmitted on the channel between the device and Tuya IoT cloud, including device info, cleaning data, maps, robot settings and customization options.

Affected Models

This issue affects the following products

- Roborock S6
- Roborock S5 Max
- Roborock S6 Pure

Statement

How have manufacturers of IoT / smart home devices addressed the increasing concerns of digital privacy and product security?

Statement

How have manufacturers of IoT / smart home devices addressed the increasing concerns of digital privacy and product security?

Rationale

Security is important!

Check things for yourself!

Proposal

Digital Privacy

Investigate the nature of network data (i.e. content, frequency, destination) from the Roborock S6, and how the data is used.

Product Security

Investigate potential security vulnerabilities of the Roborock S6, and assess the effectiveness of current security fortifications.

Literature Review

Existing Works and Papers

Literature Review

IoT ☁

The majority of hardware hacks / custom firmwares have originated from the desire to decouple hardware from cloud services

Literature Review

IoT | 2018 - Michael Steigerwald (VTRUST)

Talk: Smart home - Smart hack

- Products from different manufacturers used the same cloud infrastructure each with their own ‘customised’ (white-label) smartphone apps
  - Supposed ‘military-grade security’
- Used the Espressif ESP8266 chip
  - WiFi-enabled SoC with Arduino support
  - Often used by tinkerers and enthusiasts
- Anyone can become an ‘IoT company’ regardless of “having in-depth technical knowledge of IoT or IT security.”
  -

Literature Review

IoT | 2018 - Michael Steigerwald (VTRUST)

Talk: Smart home - Smart hack

“The analysis of the ‘smart’ devices using this basic platform is generally frightening […] serious […] shortcomings”

- Insecure transmission of encryption keys, serial number, etc…
- Insecure transmission of wireless credentials during pairing
- Ease of white-labelling and starting your own IoT business
  - Ease of selling malicious devices

Literature Review

IoT | 2018 - Michael Steigerwald (VTRUST)

Talk: Smart home - Smart hack

ct-Open-Source/tuya-convert

A collection of scripts to flash Tuya IoT devices to alternative firmwares

Python ★ 3.6k 417

Automated flashing tool `tuya-convert` created that exploited prior vulnerabilities to flash custom decoupled firmware
(i.e. ESPhome, Tasmota, etc…)

Literature Review

IoT | 2018 - Michael Steigerwald (VTRUST)

Tuya's Response

- 28th January 2019 - patch released *(later subverted)*
  - TLS encrypted firmware update procedure
  - Encryption of flash memory
- 3rd January 2020 - new patch released
  - unbreakable™
- 23rd April 2020 - Switched from the ESP8266 to a custom SoC
  - Tuya WB3S
- 16th June 2021 - Announced official support for HomeAssistant

Literature Review

IoT | 2017/2018 - Xiaomi Dafang Hacks

EliasKotlyar/Xiaomi-Dafang-Hacks

Shell ★ 3.7k 948

samtap/fang-hacks

Collection of modifications for the XiaoFang WiFi Camera

Shell ★ 1.6k 347

- Cheap WiFi camera that can be made to boot off a microSD card
- Circuit board exposed UART (`baud_rate=115200`) pins that allowed interaction with U-Boot bootloader
- Modification of boot environment to start `/bin/sh` ([link])
- Gain root shell access
- Dump firmware
- Analyse, modify and package updated firmware

Literature Review

Access and Control

Gaining access to a shell / stored data / things we shouldn’t.

Literature Review

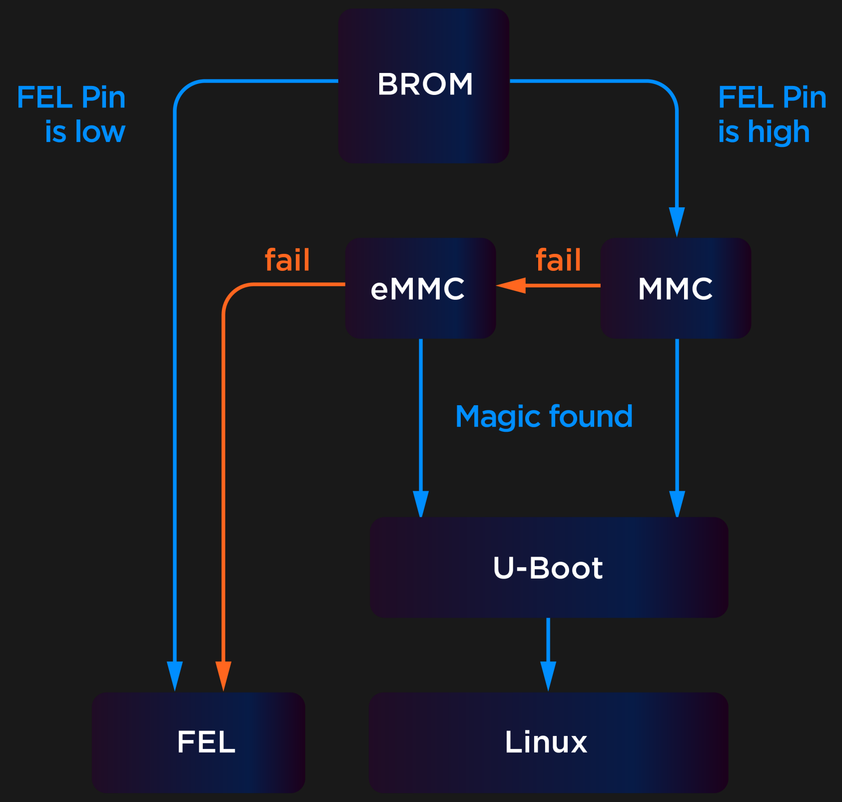
Flash IC Dumping

- May require a proprietary flash programmer (above: US$3655)
- Budget solution for common flash types: Raspberry Pi (AU$100)
- Some flash chips (depending on form factor) may require to be desoldered
  - Possibly a destructive process
- Open-source software: `flashrom`

Source: J. Jimenez - Practical Reverse Engineering

Literature Review

BGA shorting to gain access to FEL



- FEL mode is a “fallback” system on Allwinner SoCs
- Allows the flashing and reprogramming of the SoC
- Generally triggered by pulling FEL pin (`LRADC0`) LOW during boot
- FEL mode can also be entered if the bootloader fails to load

Literature Review

BGA shorting to gain access to FEL

- On the Allwinner R16 (BGA package) FEL pin located on ball location L14
  - Not located on package edge the chip so desoldering required

> *Enter FEL mode by preventing (e)MMC load?*

- SoC has a solder plane height of around 0.3mm
- Too shallow for a wire, but tall enough for aluminium foil…

Literature Review

BGA shorting to gain access to FEL

Aluminium Foil

- Thickness: ~0.02mm (… 0.02mm « 0.3mm)
- Conductive: Yep!
- $$$

Documented: `SEEMOO-MSC-0142`

Literature Review

BGA shorting to gain access to FEL | Aside (2021)

> *On later versions (post 2020), U-Boot shell access was patched, so shell access via UART was mitigated*

Pin TPA17 on the Roborock S7 circuit board was discovered to connect to ball location L14 on the SoC.

Therefore by pulling TPA17 / L14 / LRADC0 LOW (i.e connect to GND), FEL mode can be entered

Literature Review

Vacuums in the Cloud: Analyzing Security in a Hardened IoT Ecosystem

Presentation: USENIX WOOT 19

- Security analysis performed on a Neato BotVac Connected robot vacuum cleaner (popular in the US)
- AM335x Microprocessor (ARM Cortex-A8)
- Cold-boot attack allowed RAM to be dumped over serial
  - Cold-boot attack - restarting the system whilst keeping memory modules powered on, keeping memory (mostly) in-tact
  - USB + Serial communication allowed boot into custom image that could then dump the memory for later triage

Literature Review

Vacuums in the Cloud: Analyzing Security in a Hardened IoT Ecosystem

- Memory dumps contained confidential keys
  - Auth/Authz to the robot
  - Auth/Authz to the cloud infrastructure
- Logs and coredumps were encrypted… but keys hardcoded
- Secret key RNG algorithm determined to be weak
  - Small keyspace given known data = bruteforceable
- RSA key was shared with all devices
  - Identity impersonation

> *Also discovered buffer overflow vulnerability in an unauthenticated stage.*

Literature Review

2014 - Firmware Analysis

> *Paper: A Large-Scale Analysis of the Security of Embedded Firmwares*

- Broad analysis of a large number of firmware images
- Discovered 38 new vulnerabilities over 693 images
- Similarities in vulnerabilities
- Static analysis and extraction of keys, credentials, configurations and other ‘tells’

Literature Review

2014 - Firmware Analysis

- Source code changes largely remain the same
- But binary files change ‘arbitrarily’
- Difficult to compare binary files
- Calculate fuzzy hashes instead to compare similarity

e.g. binwalk, ssdeep, sdhash

Literature Review

Client-Side and Infrastructural Security

> *iOS application of a smart doorlock was analysed to (in)validate claims made by the device company*

Findings

- Lock events and other sensitive information were being logged independent of locking functionality
- Access to lock settings were purely client-side UI checks
- Certificate pinning bypass-able

Source: Backdooring the Frontdoor

Literature Review

LIDAR - Acoustic Eavesdropping

LIDAR - Light Detection and Ranging

- Uses laser lights to sense distance
- Side-channel also exposes intensity (on some units)
  - Can use to detect minute vibrations induced by audio sources

Literature Review

LIDAR - Acoustic Eavesdropping

- Vibrations are extracted and turned back into sound waves
  - Extraction of sensitive data (i.e. credit card digits)
  - Achieved 91% classification accuracy

2015: Acoustic Eavesdropping through Wireless Vibrometry

2020: LidarPhone: acoustic eavesdropping using a lidar sensor

Literature Review

Xiaomi Ecosystem | 2017-2019 - Dennis Giese

> *Paper: `SEEMOO-MSC-0142` (July 10, 2019)*

- Research available: dontvacuum.me
- Performed security analysis of a range of Xiaomi products
- Found ways to root the Mi Home Robotic Vacuum Cleaner and the Roborock S6
  - UART, hardware fault injection, etc…
- Developed cloud emulation software (DustCloud)
- Research led to development of 3rd party software (i.e. Valeduto)

Literature Review

Xiaomi Ecosystem | 2017-2019 - Dennis Giese

> *“How secure is the implementation of the ecosystem of the IoT market leader Xiaomi?”*

Conclusions

- The company quickly responds to security concerns
- Many exposed endpoints of deprecated APIs
- Many devices do not enforce proper HTTPS checks
- Difficult to enforce security for plugins (vendor-provided)
- CIA principles generally kept

Literature Review

Xiaomi Ecosystem | 2017-2019 - Dennis Giese

> *More to be done*

Extrapolation

Previous Achievements

- Smartphone application reverse engineering
- Device firmware interception
- Device hardware and component identification
- Network traffic analysis
- Storage analysis

Unaddressed Areas

- Post-2019 replication study
- In-depth firmware analysis

Plan

Plan

- Research
- Get the Roborock S6 vacuum cleaner
- Acquisition and capture of network activity
- Find a way in (it runs Linux!)
- Image the system for offline analysis
- Reverse engineering and binary analysis of firmware and software
    - Look through binaries for security vulnerabilities and fortifications

Plan

Considerations

- I’m just a fourth year!
    - Limited skills
    - i.e. microsoldering for flash chip extraction and dumping
- Access to equipment and facilities are limited (COVID?)
- Only have one device to test on

Plan

Contingency

If we can’t get into the device?

- Option 1 - Protocol analysis (network traffic)
  - i.e. Inspect the data and its nature
    - Content, Frequency, Destination
- Option 2 - Investigate the Xiaomi Home smartphone application (used to communicate with the device)
  - i.e. Decompile the Android APK file and look for security vulnerabilities and fortifications

Plan

Future Plans

- See what the sensors see
- Circuit board decomposition
- Analyse the custom ADB binary serving the USB port

Research, Upskill, Tooling

Research areas as of initial exploration

- How to capture network activity without compromising my home network?
- Interfacing with JTAG / UART / Serial
- Linux filesystem / system forensics
- Learn the ARM Instruction Set (ISA)
  - Processor Modes, Protection Rings?
- Learn about other hardware protections
  - Secure Boot, RPBM, SELinux, LUKS, OPTEE, TrustZone, etc…
- Acquisition of hardware
  - Serial adapters?
  - Network switch?
  - etc…

Project Timeline

Thesis A

- Initial research and research environment setup
- Teardown and initial hands-on of Roborock S6

Thesis B - Binary Assessment

- Disassembly and analysis of firmware binaries to identify vulnerabilities
    - inc. ADB binary functionality
- Search for unsecured secrets, logs, configurations

Thesis C - Connectivity Assessment

- Inspection of outbound internet traffic - security, PII, etc
- Inspection of local network traffic
- Inspection of interaction with nearby devices
- Protocol analysis

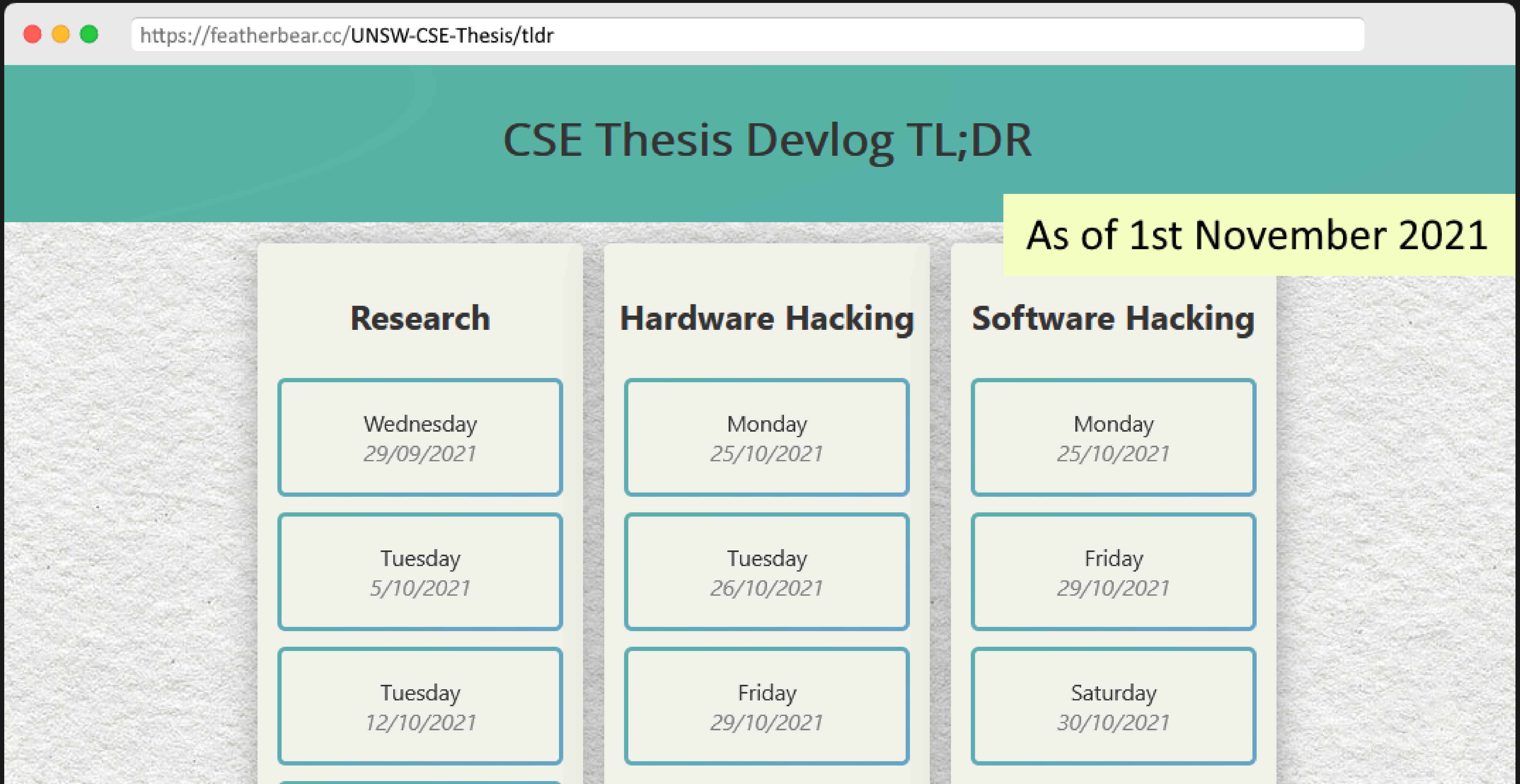
Current Progress

Rolling Research

featherbear.cc/UNSW-CSE-Thesis

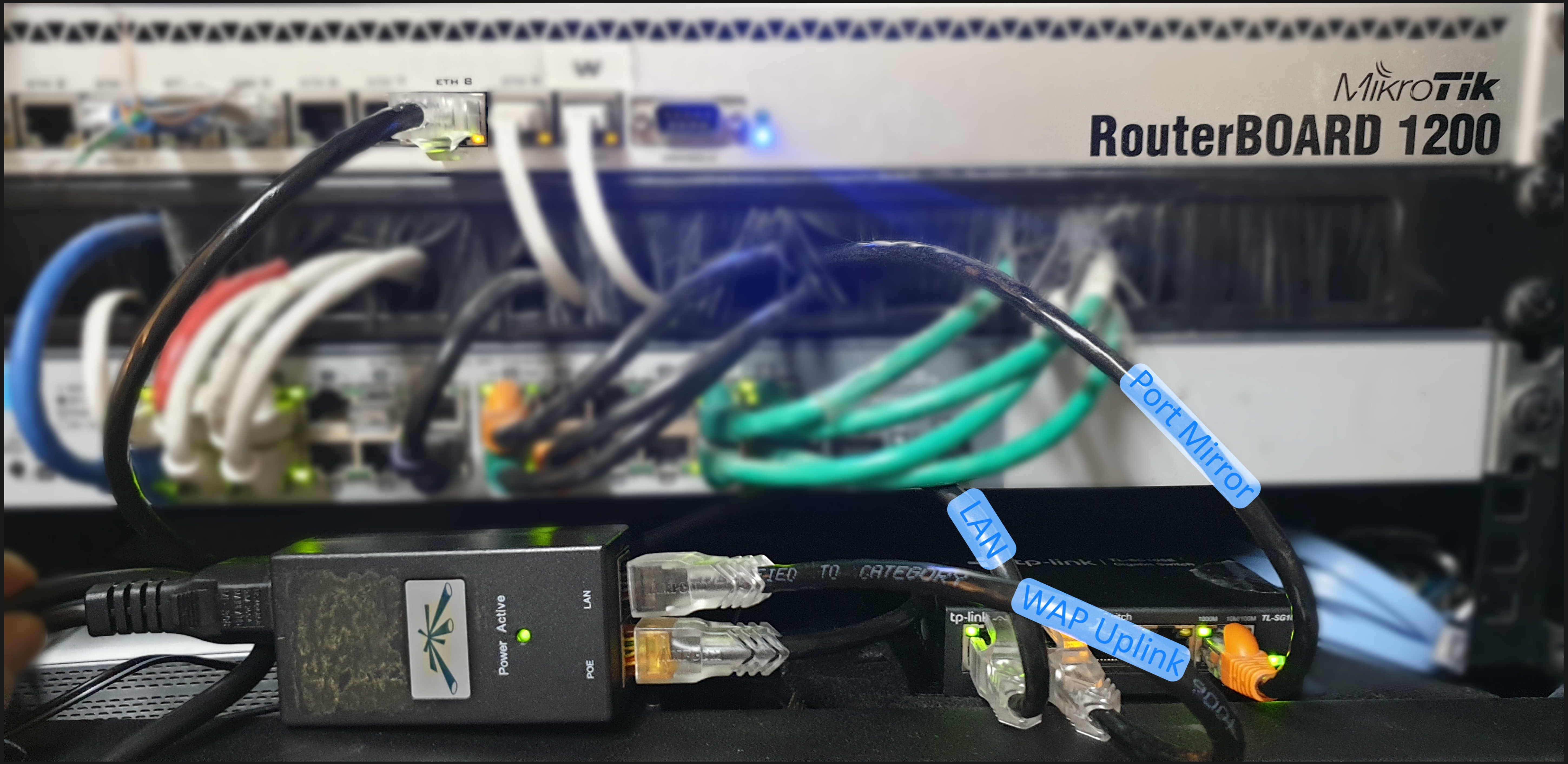
Rolling Research

Summary



Preliminary Results

Network Setup



Preliminary Results

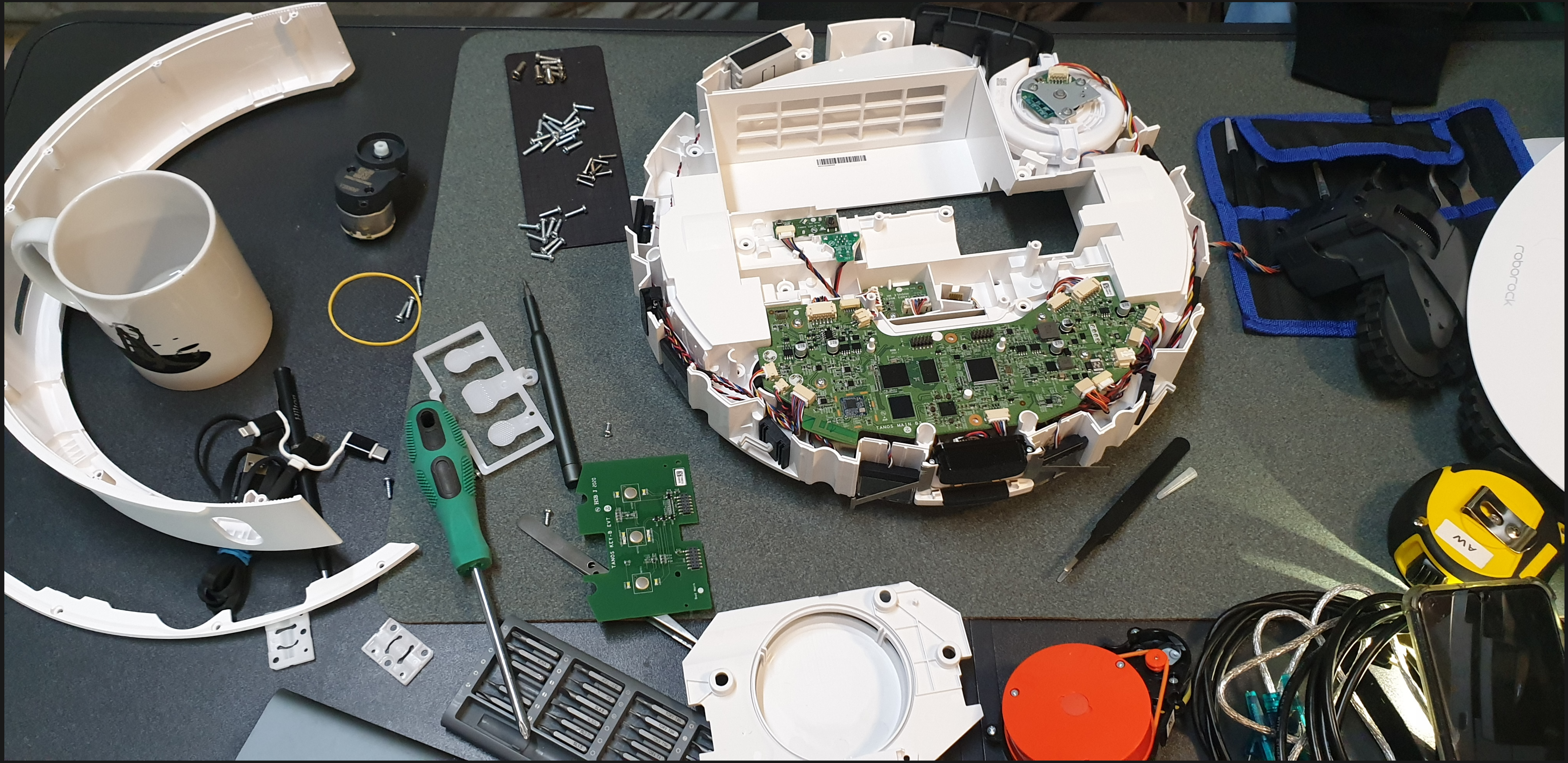
[Initial] Packet Capture

No.	Time	Source	Destination	Protocol	Length	Info
1291…	4315.804111	BeijingX_1d:24:c4	Broadcast	ARP	107	Who has 10.10.10.1? Tell 10.10.10.8
1291…	4315.804161	Routerbo_cf:36:21	BeijingX_1d:24:c4	ARP	89	10.10.10.1 is at 00:0c:42:cf:36:21
1291…	4315.805362	10.10.10.8	110.43.0.85	TCP	121	41134 → 80 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=4294939886 TSecr=0 WS=64
1291…	4316.203673	110.43.0.85	10.10.10.8	TCP	121	80 → 41134 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1340 SACK_PERM=1 WS=32
1291…	4316.205277	10.10.10.8	110.43.0.85	TCP	107	41134 → 80 [ACK] Seq=1 Ack=1 Win=14656 Len=0
1291…	4316.205876	10.10.10.8	110.43.0.85	HTTP	299	GET /gslb?tver=2&id=322119905&dm=sg.ott.io.mi.com&timestamp=1635171460&sign=Lc9j28ajJwk7nMufGq2APVGiElKwzagUsZ%2FyuIRj79Q%3D HTTP/1.1
1292…	4316.505288	0.0.0.0	255.255.255.255	DHCP	389	DHCP Discover - Transaction ID 0x731c1bc8
1292…	4316.515522	0.0.0.0	255.255.255.255	DHCP	389	DHCP Discover - Transaction ID 0x7448626d
1292…	4316.605227	110.43.0.85	10.10.10.8	TCP	101	80 → 41134 [ACK] Seq=1 Ack=199 Win=30336 Len=0
1292…	4316.605288	110.43.0.85	10.10.10.8	HTTP/JSON	300	HTTP/1.1 400 Bad Request , JavaScript Object Notation (application/json)
1292…	4316.606968	10.10.10.8	110.43.0.85	TCP	107	41134 → 80 [ACK] Seq=199 Ack=200 Win=15680 Len=0
1292…	4316.606968	110.43.0.85	10.10.10.8	TCP	101	80 → 41134 [FIN, ACK] Seq=200 Ack=199 Win=30336 Len=0
1292…	4316.607530	10.10.10.8	110.43.0.85	TCP	107	41134 → 80 [FIN, ACK] Seq=199 Ack=200 Win=15680 Len=0
1292…	4316.608113	10.10.10.8	110.43.0.83	TCP	121	55090 → 80 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=4294939967 TSecr=0 WS=64
1292…	4316.608683	10.10.10.8	110.43.0.85	TCP	107	41134 → 80 [ACK] Seq=200 Ack=201 Win=15680 Len=0
1292…	4316.625602	110.43.0.85	10.10.10.8	TCP	101	[TCP Out-Of-Order] 80 → 41134 [FIN, ACK] Seq=200 Ack=199 Win=30336 Len=0
1292…	4316.628140	10.10.10.8	110.43.0.85	TCP	113	[TCP Dup ACK 129208#1] 41134 → 80 [ACK] Seq=200 Ack=201 Win=15680 Len=0 SLE=200 SRE=201
1292…	4316.802243	10.10.10.7	10.10.10.1	DNS	122	Standard query 0xed04 A eas.outlook.com
1292…	4316.815908	110.43.0.85	10.10.10.8	TCP	300	[TCP Out-Of-Order] 80 → 41134 [FIN, PSH, ACK] Seq=1 Ack=199 Win=30336 Len=199
1292…	4316.817453	10.10.10.8	110.43.0.85	TCP	113	[TCP Dup ACK 129208#2] 41134 → 80 [ACK] Seq=200 Ack=201 Win=15680 Len=0 SLE=1 SRE=201
1292…	4316.971851	10.10.10.7	10.10.10.1	DNS	125	Standard query 0x02dc A account.xiaomi.com
1292…	4317.006784	110.43.0.85	10.10.10.8	TCP	101	80 → 41134 [ACK] Seq=201 Ack=200 Win=30336 Len=0
1292…	4317.027181	110.43.0.85	10.10.10.8	TCP	101	80 → 41134 [RST] Seq=201 Win=0 Len=0
1292…	4317.047587	110.43.0.83	10.10.10.8	TCP	121	80 → 55090 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1340 SACK_PERM=1 WS=32
1292…	4317.049209	10.10.10.8	110.43.0.83	TCP	107	55090 → 80 [ACK] Seq=1 Ack=1 Win=14656 Len=0
1292…	4317.049348	10.10.10.8	110.43.0.83	HTTP	300	GET /gslb?tver=2&id=322119905&dm=sg.ot.io.mi.com&timestamp=1635171461&sign=y4ipkGw7yjTyoKEoXTTQF0D2IsRB5T2O%2BDrkec5%2FhHg%3D HTTP/1.1
1292…	4317.216362	110.43.0.85	10.10.10.8	TCP	101	80 → 41134 [RST] Seq=201 Win=0 Len=0
1292…	4317.483104	110.43.0.83	10.10.10.8	TCP	101	80 → 55090 [ACK] Seq=1 Ack=200 Win=30464 Len=0
1292…	4317.483104	110.43.0.83	10.10.10.8	HTTP/JSON	300	HTTP/1.1 400 Bad Request , JavaScript Object Notation (application/json)
1292…	4317.483135	110.43.0.83	10.10.10.8	TCP	101	80 → 55090 [FIN, ACK] Seq=200 Ack=200 Win=30464 Len=0
1292…	4317.485013	10.10.10.8	110.43.0.83	TCP	107	55090 → 80 [ACK] Seq=200 Ack=200 Win=15680 Len=0
1292…	4317.485075	10.10.10.8	110.43.0.83	TCP	107	55090 → 80 [FIN, ACK] Seq=200 Ack=201 Win=15680 Len=0
1292…	4317.486424	10.10.10.8	10.10.10.1	DNS	122	Standard query 0x8180 A sg.ot.io.mi.com
1292…	4317.489390	110.43.0.83	10.10.10.8	TCP	101	[TCP Out-Of-Order] 80 → 55090 [FIN, ACK] Seq=200 Ack=200 Win=30464 Len=0
1292…	4317.490689	10.10.10.8	110.43.0.83	TCP	113	[TCP Dup ACK 129225#1] 55090 → 80 [ACK] Seq=201 Ack=201 Win=15680 Len=0 SLE=200 SRE=201
1292…	4317.507735	0.0.0.0	255.255.255.255	DHCP	389	DHCP Discover - Transaction ID 0x57bd3221
1292…	4317.517990	0.0.0.0	255.255.255.255	DHCP	389	DHCP Discover - Transaction ID 0x3260a72
1292…	4317.702426	110.43.0.83	10.10.10.8	TCP	300	[TCP Out-Of-Order] 80 → 55090 [FIN, PSH, ACK] Seq=1 Ack=200 Win=30464 Len=199
1292…	4317.703797	10.10.10.8	110.43.0.83	TCP	113	[TCP Dup ACK 129225#2] 55090 → 80 [ACK] Seq=201 Ack=201 Win=15680 Len=0 SLE=1 SRE=201
1292…	4317.901091	10.10.10.7	10.10.10.1	DNS	121	Standard query 0x83d2 A www.google.com
1292…	4317.911588	110.43.0.83	10.10.10.8	TCP	101	80 → 55090 [ACK] Seq=201 Ack=201 Win=30464 Len=0

- No LAN-LAN packets???
- incomplete test - misconfigured packet capture setup

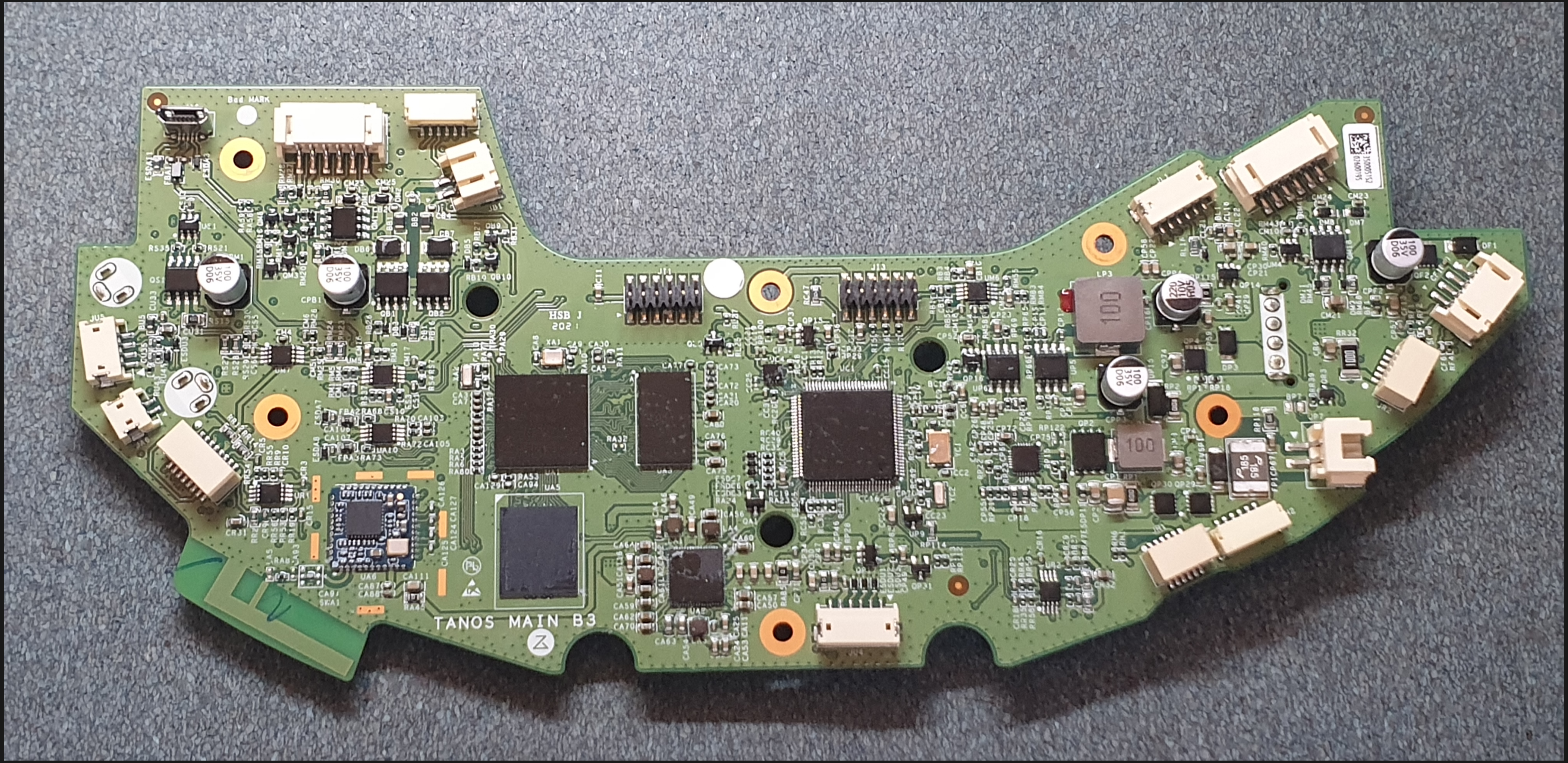
Preliminary Results

Teardown



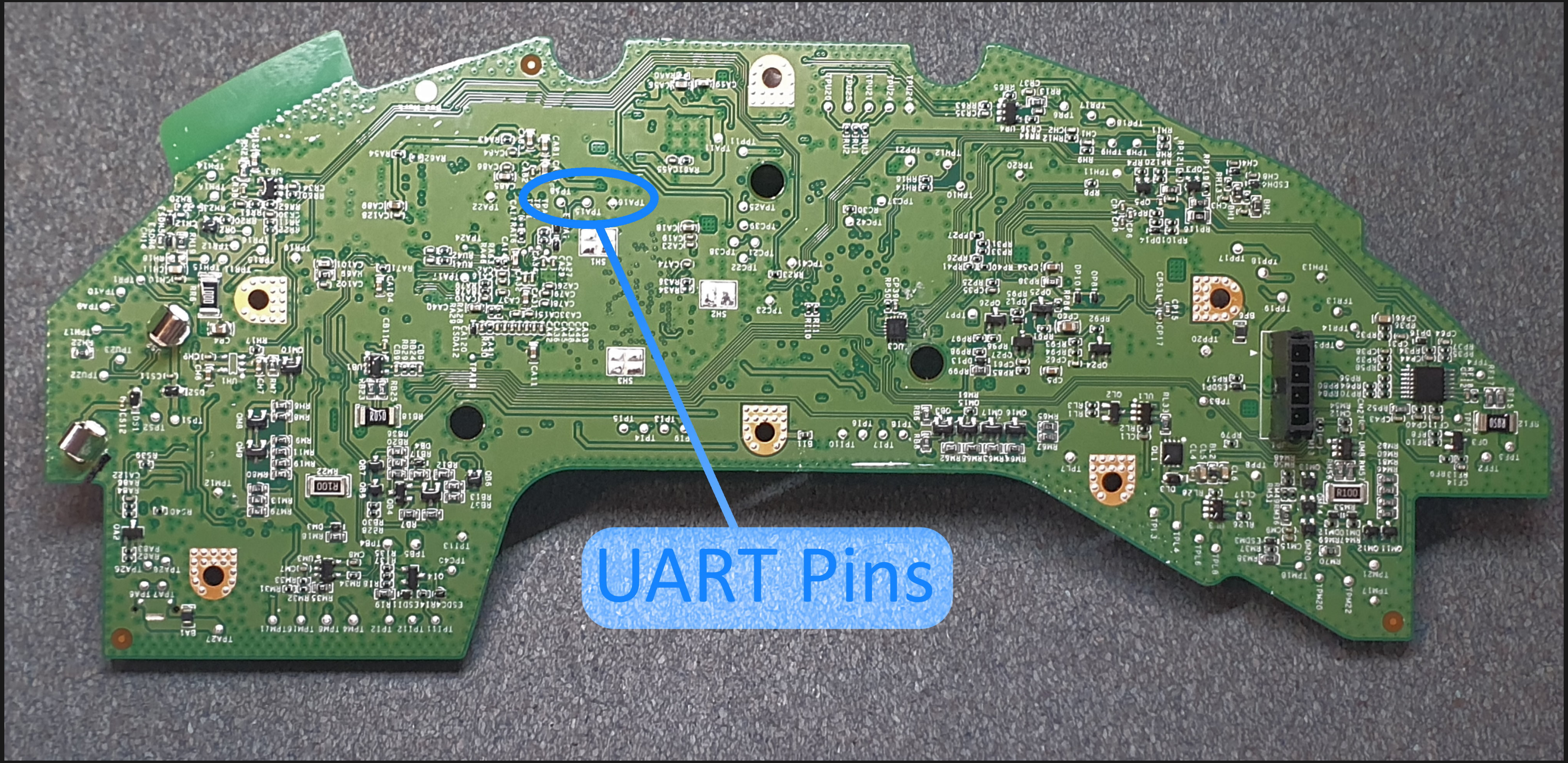
Preliminary Results

Initial Breakdown and Pinout (where needed)



Preliminary Results

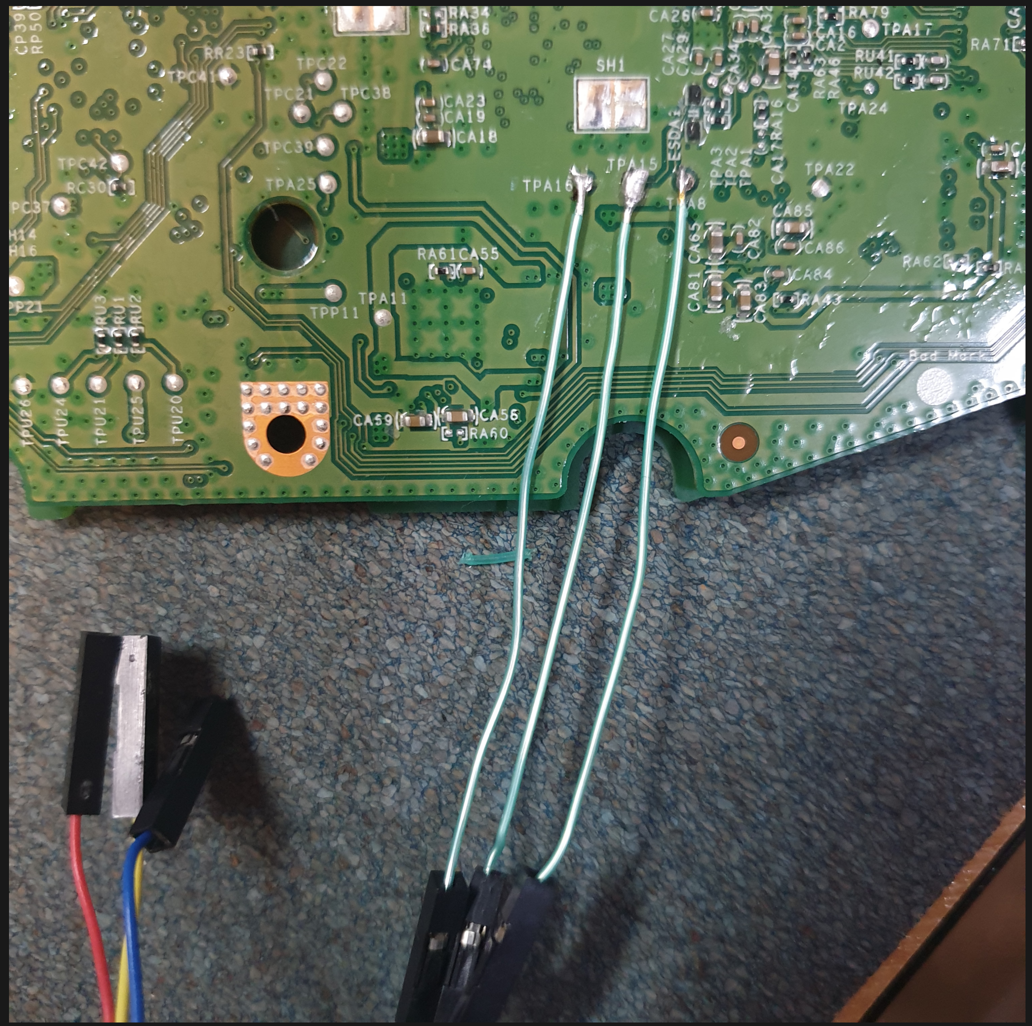
Identification of the UART pins

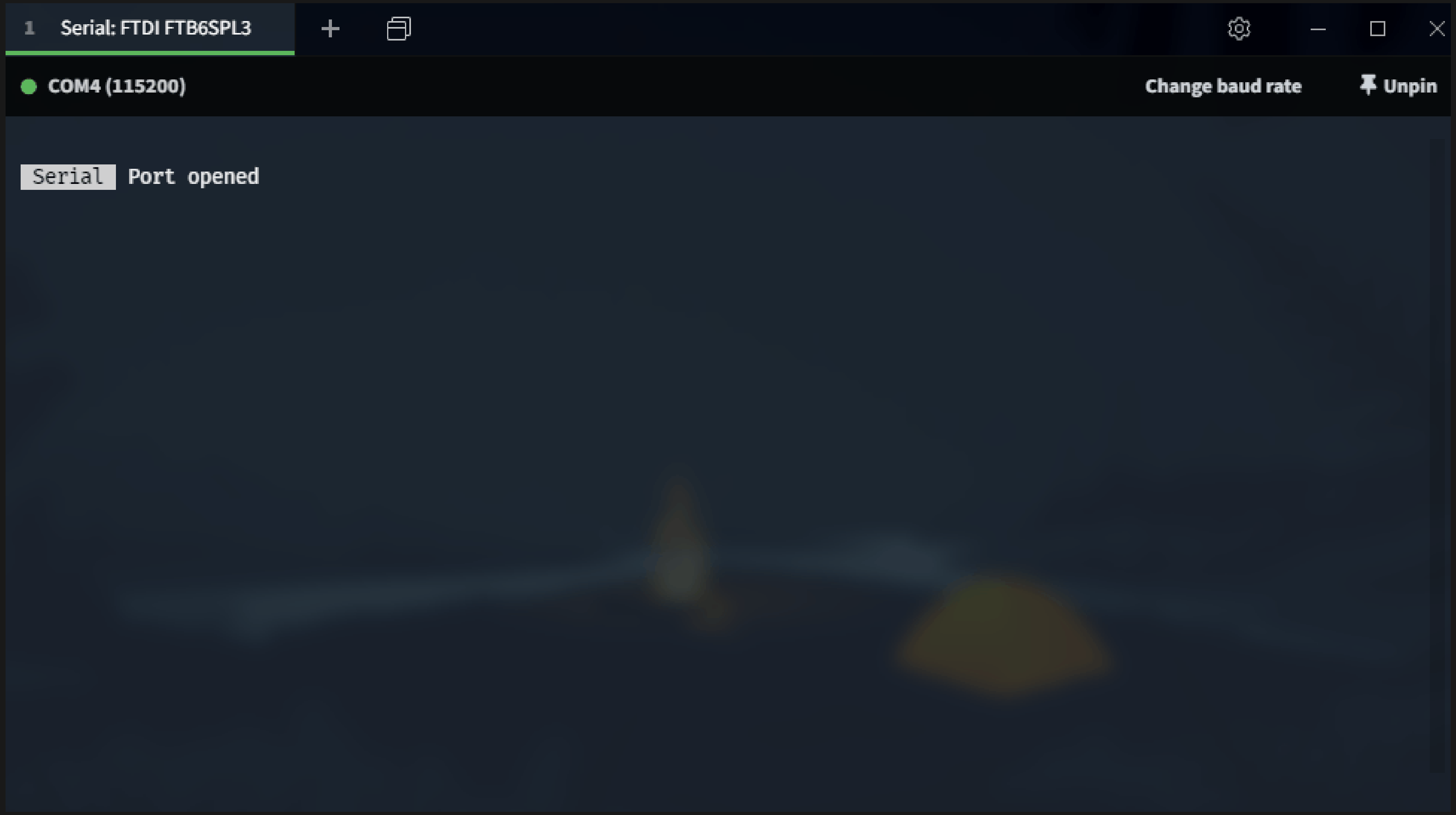


Preliminary Results

Serial Access

- Serial (baud=115200) gives us a shell!





Need a root password though…

Preliminary Results

U-Boot Bootloader

https://github.com/allwinner-zh/bootloader/blob/master/u-boot-2011.09/board/sunxi/board_common.c#L843-L847

```c
		}
		else if(c == 's')			//shell mode
		{
			gd->force_shell = 1;
			gd->debug_mode = 1;
		}
		return 0;
}
```

- Able to enter the bootloader shell if `s` is pressed during init

```
flag_bootA:0x1
upgrade stage:0x63
No upgradeover system found, check if has normal system, pmu: 0x0
board_common.c:check_android_misc:will be boot A system
to be run cmd=run setargs_mmc boot_normal
boot A system
WORK_MODE_BOOT
[      1.996]Hit any key to stop autoboot:  0
sunxi#
```

Preliminary Results

Root!

```
sunxi#ext4load
ext4load - load binary file from a Ext4 filesystem

Usage:
ext4load <interface> <dev[:part]> [addr] [filename] [bytes]
          - load binary file 'filename' from 'dev' on 'interface'
                 to address 'addr' from ext4 filesystem
sunxi#ext4load mmc 2:6 0 vinda
Loading file "vinda" from mmc device 2:6
16 bytes read
sunxi#md 0 4
00000000: 5b415243 51454346 54505042 525f5655    CRA[FCEQBPPTUV_R
```

```
rockrobo login: root
Password:
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.4.39 armv7l)

 * Documentation:  https://help.ubuntu.com/

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@rockrobo:~#
```

Next Steps

- Dump the firmware and begin RE / forensics
- Redo (and further investigate) live system analysis
  - i.e. Properly capture *all* network traffic

Any Questions?

Andrew Wong

w: featherbear.cc/UNSW-CSE-Thesis

e: andrew.j.wong@student.unsw.edu.au