

# “Smart” Vacuum Cleaners

An Audit Into The Security and Integrity of IoT Systems

Andrew Wong | UNSW Sydney

# Today's Agenda

- Thesis B plan
- Thesis B review
- Thesis B retrospective
- Thesis C revised plan

# Statement

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

- Digital Privacy - Investigate the nature of network data (i.e. content, frequency, destination) and how the data is used.
- Product Security - Investigate potential security vulnerabilities and assess the effectiveness of current security fortifications.

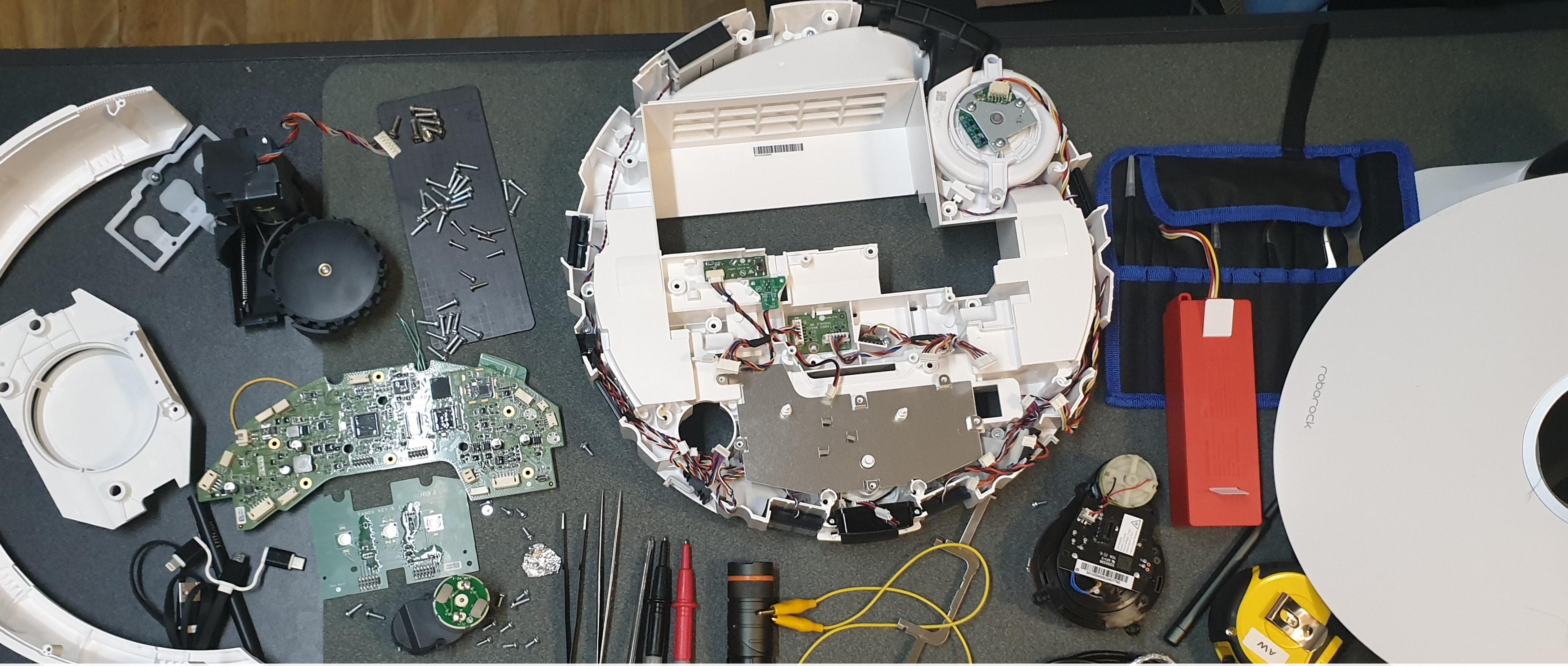
# Original Project Timeline

## Thesis B - Binary Assessment

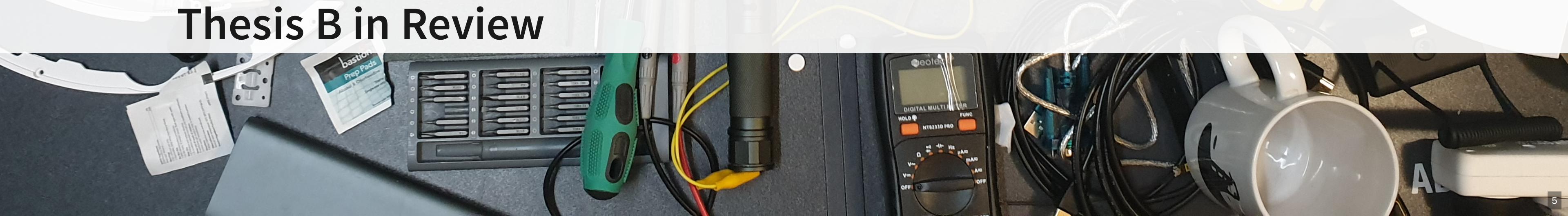
- Disassembly and analysis of firmware binaries to identify vulnerabilities
  - inc. ADB binary functionality
- Search for unsecured secrets, logs, configurations

## 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



## Thesis B in Review



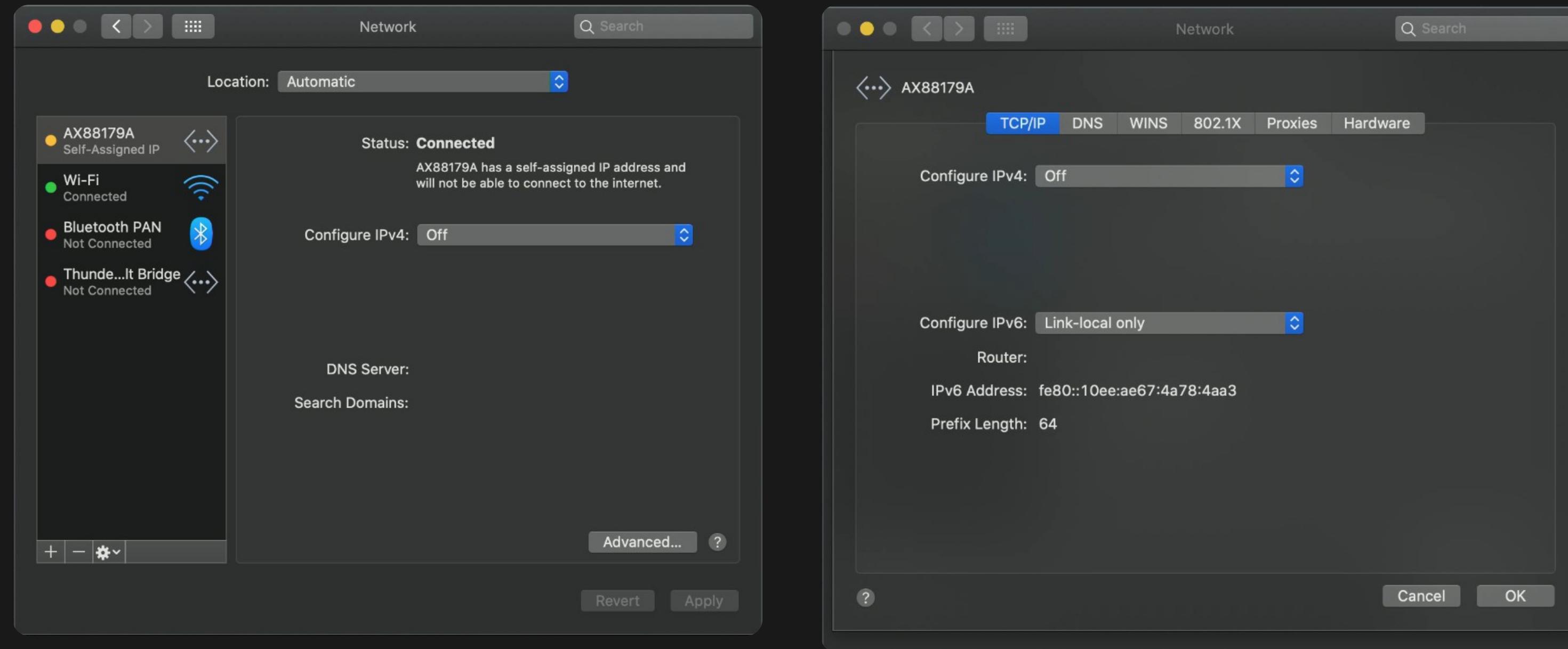
# More logging

Previously packet captures only logged WAN traffic...

- Now port mirroring from a switch ([TP-Link TL-SG105E](#))
- Now getting all LAN data too! (port mirrored from AP)

# More logging

Previously packet captures only logged WAN traffic...

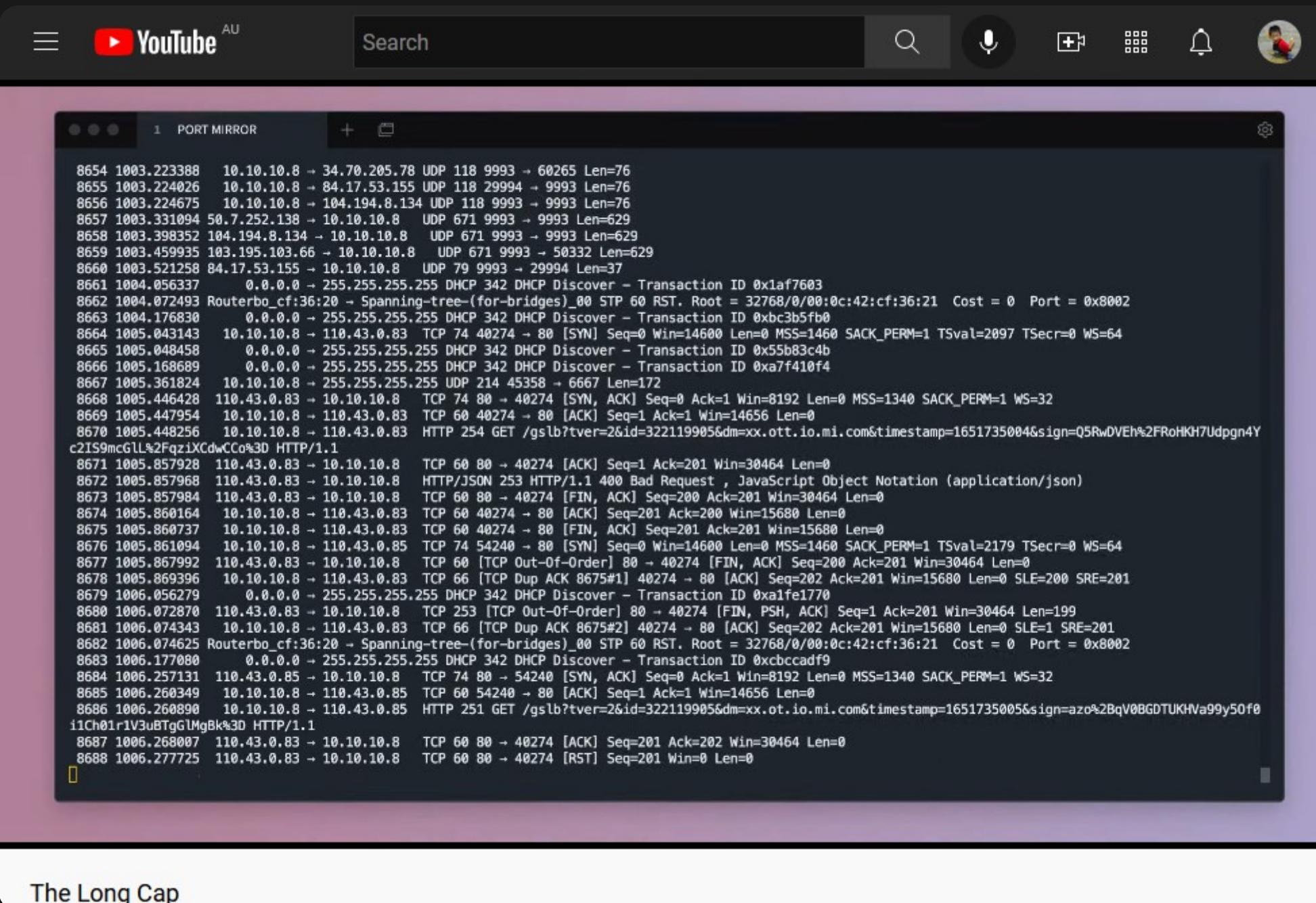


- The switch doesn't have true port mirroring - also seeing sink traffic
- Disabled IPv4 and (attempt to disable) IPv6 on the network adapter
- Can filter out irrelevant packets later

# More logging

Previously packet captures only logged WAN traffic...

*Will later use dumps to check frequency and access*

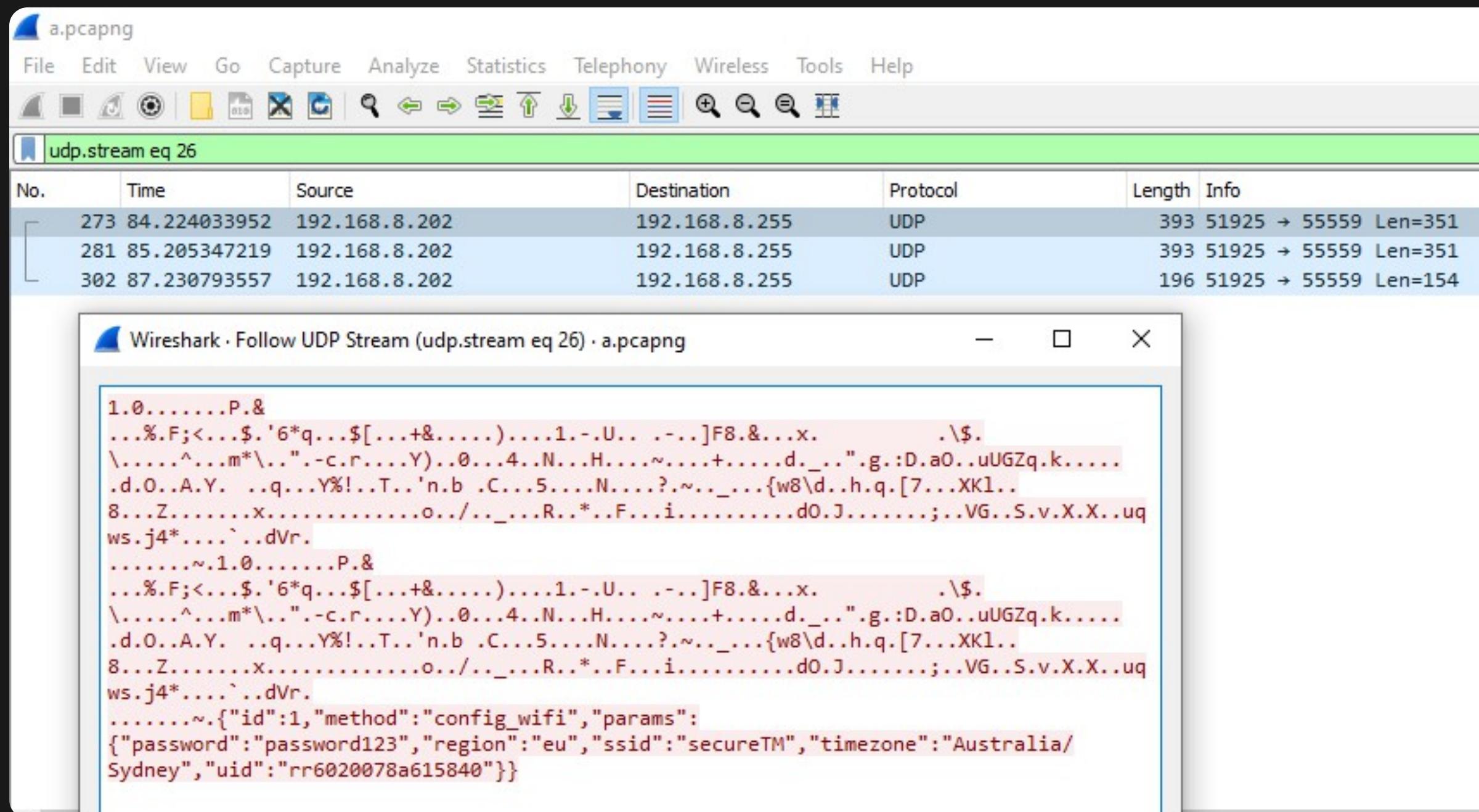


```
1 PORT MIRROR
8654 1003.223388 10.10.10.8 -> 34.70.205.78 UDP 118 9993 -> 60265 Len=76
8655 1003.224026 10.10.10.8 -> 84.17.53.155 UDP 118 29994 -> 9993 Len=76
8656 1003.224675 10.10.10.8 -> 104.194.8.134 UDP 118 9993 -> 9993 Len=76
8657 1003.331094 50.7.252.138 -> 10.10.10.8 UDP 671 9993 -> 9993 Len=629
8658 1003.398352 104.194.8.134 -> 10.10.10.8 UDP 671 9993 -> 9993 Len=629
8659 1003.459935 103.195.103.66 -> 10.10.10.8 UDP 671 9993 -> 50332 Len=629
8660 1003.521258 84.17.53.155 -> 10.10.10.8 UDP 79 9993 -> 29994 Len=37
8661 1004.056337 0.0.0.0 -> 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0x1af7603
8662 1004.072493 Routerbo_cf:36:20 - Spanning-tree-(for-bridges)_00 STP 60 RST. Root = 32768/0/00:0c:42:cf:36:21 Cost = 0 Port = 0x8002
8663 1004.176830 0.0.0.0 -> 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0xbc3b5fb0
8664 1005.043143 10.10.10.8 -> 110.43.0.83 TCP 74 40274 -> 80 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=2097 TSecr=0 WS=64
8665 1005.048458 0.0.0.0 -> 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0x55b83c4b
8666 1005.168689 0.0.0.0 -> 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0xa7f410f4
8667 1005.361824 10.10.10.8 -> 255.255.255.255 UDP 214 45358 -> 6667 Len=172
8668 1005.446428 110.43.0.83 -> 10.10.10.8 TCP 74 80 -> 40274 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1340 SACK_PERM=1 WS=32
8669 1005.447954 10.10.10.8 -> 110.43.0.83 TCP 60 40274 -> 80 [ACK] Seq=1 Ack=1 Win=14656 Len=0
8670 1005.448256 10.10.10.8 -> 110.43.0.83 HTTP 254 GET /gslb?tver=2&id=322119905&dm=xx.ott.io.mi.com&timestamp=1651735004&sign=Q5RwDVEh%2FRoHKH7Udpgn4Y
c2IS9mcGll%2FqziXCdwCCo3D HTTP/1.1
8671 1005.857928 110.43.0.83 -> 10.10.10.8 TCP 60 80 -> 40274 [ACK] Seq=1 Ack=201 Win=30464 Len=0
8672 1005.857968 110.43.0.83 -> 10.10.10.8 HTTP/JSON 253 HTTP/1.1 400 Bad Request , JavaScript Object Notation (application/json)
8673 1005.857984 110.43.0.83 -> 10.10.10.8 TCP 60 80 -> 40274 [FIN, ACK] Seq=200 Ack=201 Win=30464 Len=0
8674 1005.860164 10.10.10.8 -> 110.43.0.83 TCP 60 40274 -> 80 [ACK] Seq=201 Ack=200 Win=15680 Len=0
8675 1005.860737 10.10.10.8 -> 110.43.0.83 TCP 60 40274 -> 80 [FIN, ACK] Seq=201 Ack=201 Win=15680 Len=0
8676 1005.861094 10.10.10.8 -> 110.43.0.85 TCP 74 54240 -> 80 [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=2179 TSecr=0 WS=64
8677 1005.867992 110.43.0.83 -> 10.10.10.8 TCP 60 [TCP Out-Of-Order] 80 -> 40274 [FIN, ACK] Seq=200 Ack=201 Win=30464 Len=0
8678 1005.869396 10.10.10.8 -> 110.43.0.83 TCP 66 [TCP Dup ACK 8675#1] 40274 -> 80 [ACK] Seq=202 Ack=201 Win=15680 Len=0 SLE=200 SRE=201
8679 1006.056279 0.0.0.0 -> 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0xa1fe1770
8680 1006.072870 110.43.0.83 -> 10.10.10.8 TCP 253 [TCP Out-Of-Order] 80 -> 40274 [FIN, PSH, ACK] Seq=1 Ack=201 Win=30464 Len=199
8681 1006.074343 10.10.10.8 -> 110.43.0.83 TCP 66 [TCP Dup ACK 8675#2] 40274 -> 80 [ACK] Seq=202 Ack=201 Win=15680 Len=0 SLE=1 SRE=201
8682 1006.074625 Routerbo_cf:36:20 - Spanning-tree-(for-bridges)_00 STP 60 RST. Root = 32768/0/00:0c:42:cf:36:21 Cost = 0 Port = 0x8002
8683 1006.177080 0.0.0.0 -> 255.255.255.255 DHCP 342 DHCP Discover - Transaction ID 0xcbccadef9
8684 1006.257131 110.43.0.85 -> 10.10.10.8 TCP 74 80 -> 54240 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1340 SACK_PERM=1 WS=32
8685 1006.260349 10.10.10.8 -> 110.43.0.85 TCP 60 54240 -> 80 [ACK] Seq=1 Ack=1 Win=14656 Len=0
8686 1006.268890 10.10.10.8 -> 110.43.0.85 HTTP 251 GET /gslb?tver=2&id=322119905&dm=xx.ott.io.mi.com&timestamp=1651735005&sign=azo%2BqV0BGDTUKHVa99y50f0
i1Ch01r1V3uBTgGMgBk%3D HTTP/1.1
8687 1006.268007 110.43.0.83 -> 10.10.10.8 TCP 60 80 -> 40274 [ACK] Seq=201 Ack=202 Win=30464 Len=0
8688 1006.277725 110.43.0.83 -> 10.10.10.8 TCP 60 80 -> 40274 [RST] Seq=201 Win=0 Len=0
```

The Long Cap

# Speaking of packets...

*WiFi credentials in plain text during setup*



- Minor issue, only exploitable during time of setup

# Fingerprinting

## System

```
[ 0.340]U-Boot 2011.09-rc1-dirty (Mar 25 2020 - 20:45:43) Allwinner Technology
[ 0.000000] Linux version 3.4.39 (rockrobo@apimg) (gcc version 4.8.4 (Ubuntu/Linaro 4.8
[ 0.000000] CPU: ARMv7 Processor [410fc075] revision 5 (ARMv7), cr=10c5387d
[ 0.000000] Machine: sun8i
...

```

CPU: Allwinner R16 (ARM Cortex-A7) - ARMv7l / armhf

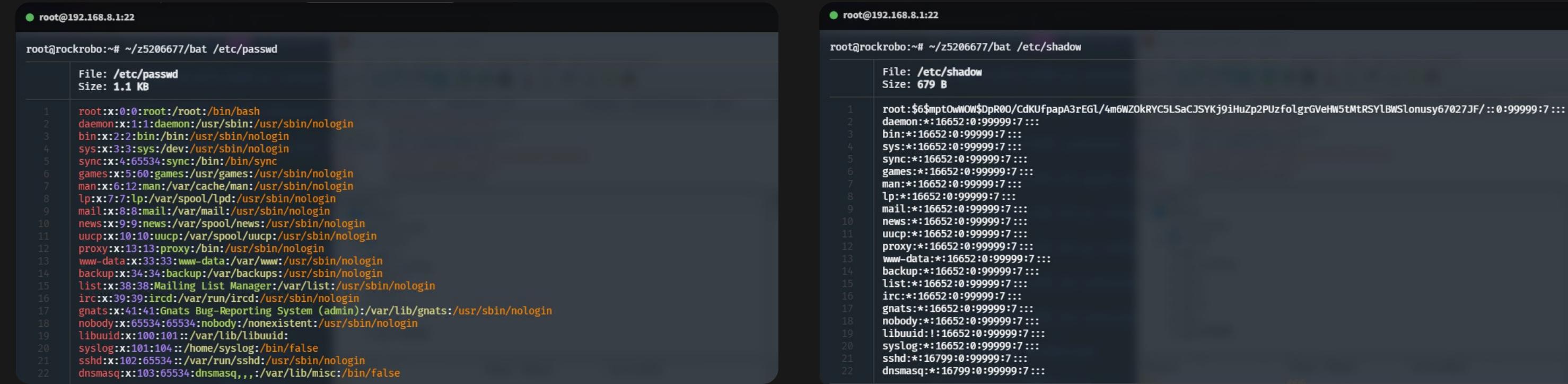
ACU: STM32F103VCT6 (ARM Cortex-M3)

Roborock Firmware version: 3.5.4\_1558

Operating system: Ubuntu 14.04.3 LTS

# Fingerprinting

## Users



The image shows two terminal windows side-by-side. Both are running on a host with IP 192.168.8.1:22. The left window displays the contents of the /etc/passwd file, and the right window displays the contents of the /etc/shadow file. Both files show a list of users and their corresponding information.

**/etc/passwd Content:**

	File: /etc/passwd	Size: 1.1 KB
1	root:x:0:0:root:/root:/bin/bash	
2	daemon:x:1:1:daemon:/usr/sbin/nologin	
3	bin:x:2:2:bin:/usr/sbin/nologin	
4	sys:x:3:sys:/dev:/usr/sbin/nologin	
5	sync:x:4:65534:sync:/bin:/sync	
6	games:x:5:160:games:/usr/games:/usr/sbin/nologin	
7	man:x:6:12:man:/var/cache/man:/usr/sbin/nologin	
8	lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin	
9	mail:x:8:8:mail:/var/mail:/usr/sbin/nologin	
10	news:x:9:9:news:/var/spool/news:/usr/sbin/nologin	
11	uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin	
12	proxy:x:13:13:proxy:/usr/sbin/nologin	
13	www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin	
14	backup:x:34:34:backup:/var/backups:/usr/sbin/nologin	
15	list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin	
16	irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin	
17	gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin	
18	nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin	
19	libuuid:x:100:101::/var/lib/libuuid:	
20	syslog:x:101:104::/home/syslog:/bin/false	
21	sshd:x:102:65534::/var/run/sshd:/usr/sbin/nologin	
22	dnsmasq:x:103:65534:dnsmasq,,,:/var/lib/misc:/bin/false	

**/etc/shadow Content:**

	File: /etc/shadow	Size: 679 B
1	root:\$6\$mp0OWW\$DpR00/CdkUfpapA3rEGL/4m6WZ0kRYC5LSaCJSYKj9iHuZp2PUzfolgrGVeHW5tMtRSYLBWSlonusy67027JF/:0:99999:7:::	
2	daemon:*:16652:0:99999:7:::	
3	bin:**:16652:0:99999:7:::	
4	sys:**:16652:0:99999:7:::	
5	sync:**:16652:0:99999:7:::	
6	games:**:16652:0:99999:7:::	
7	man:**:16652:0:99999:7:::	
8	lp:**:16652:0:99999:7:::	
9	mail:**:16652:0:99999:7:::	
10	news:**:16652:0:99999:7:::	
11	uucp:**:16652:0:99999:7:::	
12	proxy:**:16652:0:99999:7:::	
13	www-data:**:16652:0:99999:7:::	
14	backup:**:16652:0:99999:7:::	
15	list:**:16652:0:99999:7:::	
16	irc:**:16652:0:99999:7:::	
17	gnats:**:16652:0:99999:7:::	
18	nobody:**:16652:0:99999:7:::	
19	libuuid:**:16652:0:99999:7:::	
20	syslog:**:16652:0:99999:7:::	
21	sshd:**:16799:0:99999:7:::	
22	dnsmasq:**:16799:0:99999:7:::	

*No additional users*

```
root@rockrobo:~# ls /home  
ruby
```

/home/ruby exists but no user ruby, though exists in /etc/passwd~

# Fingerprinting

## Processes

Everything is running as root

# Fingerprinting

## Ports

Proto	Recv-Q	Send-Q	Local Address	Foreign Address	State	PID/Program
tcp	0	0	127.0.0.1:54322	0.0.0.0:*	LISTEN	991/mio_c
tcp	0	0	127.0.0.1:54323	0.0.0.0:*	LISTEN	991/mio_c
tcp	0	0	0.0.0.0:22	0.0.0.0:*	LISTEN	1644/sshd
tcp	0	0	127.0.0.1:55551	0.0.0.0:*	LISTEN	998/rriot_
tcp	0	0	0.0.0.0:6668	0.0.0.0:*	LISTEN	998/rriot_
tcp6	0	0	:::22	:::*	LISTEN	1644/sshd

tcp/22 and tcp/6668 are exposed

# Fingerprinting

## Firewall

At least port 22 is blocked by iptables

```
root@rockrobo:~# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source          destination
DROP       udp  --  anywhere        anywhere        udp  dpt:6665
DROP       tcp  --  anywhere        anywhere        tcp  dpt:6665
DROP       tcp  --  anywhere        anywhere        tcp  dpt:ssh

Chain FORWARD (policy ACCEPT)
target     prot opt source          destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source          destination
```

- What runs on port 6665
  - player
  - What about file-based IPC?

# Fingerprinting

```
root@rockrobo:~# ip6tables -L
Chain INPUT (policy ACCEPT)
target    prot opt source          destination
Chain FORWARD (policy ACCEPT)
target    prot opt source          destination
Chain OUTPUT (policy ACCEPT)
target    prot opt source          destination
```

... except IPv6 isn't..

*Future work: Test IPv6 lease*

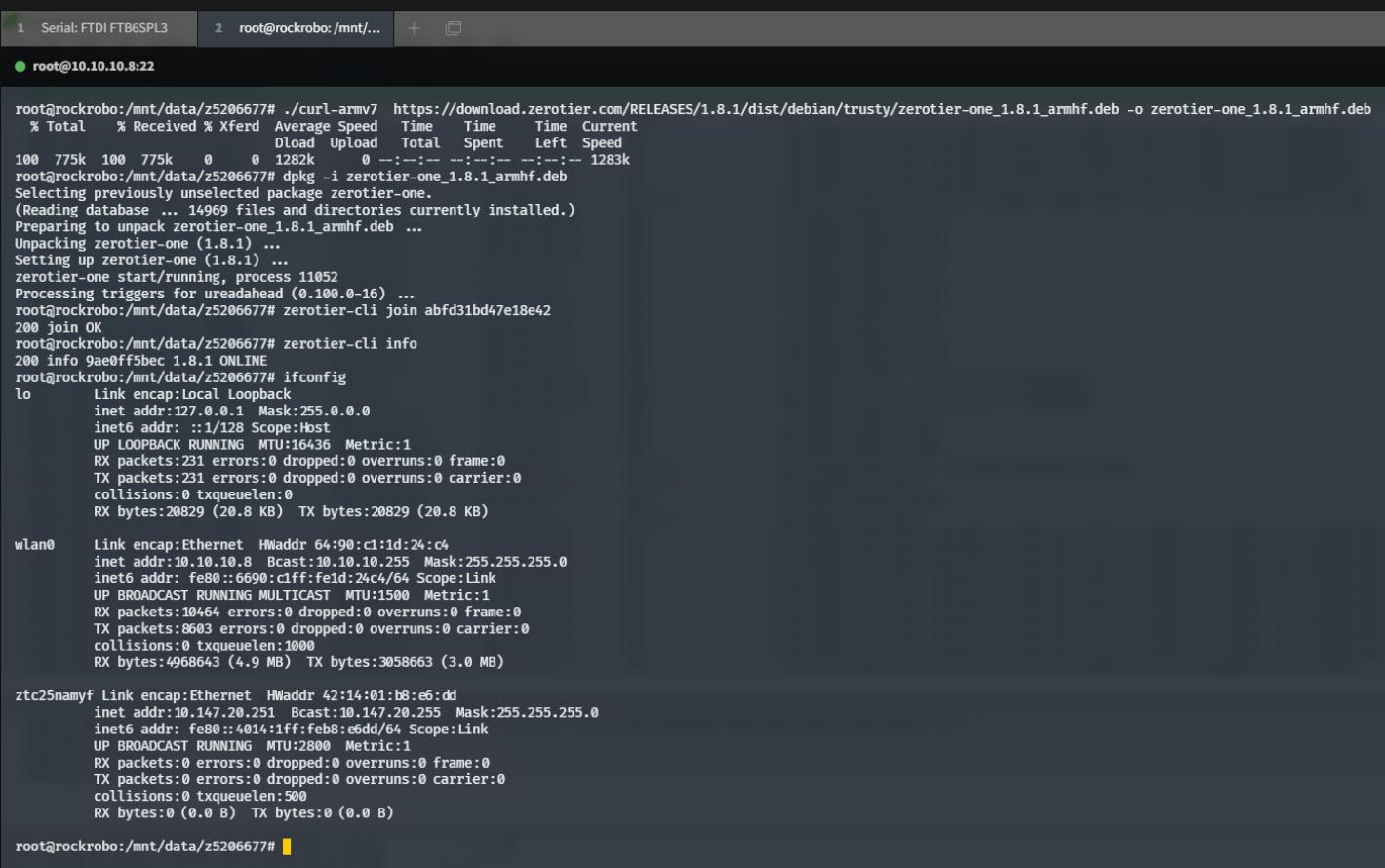
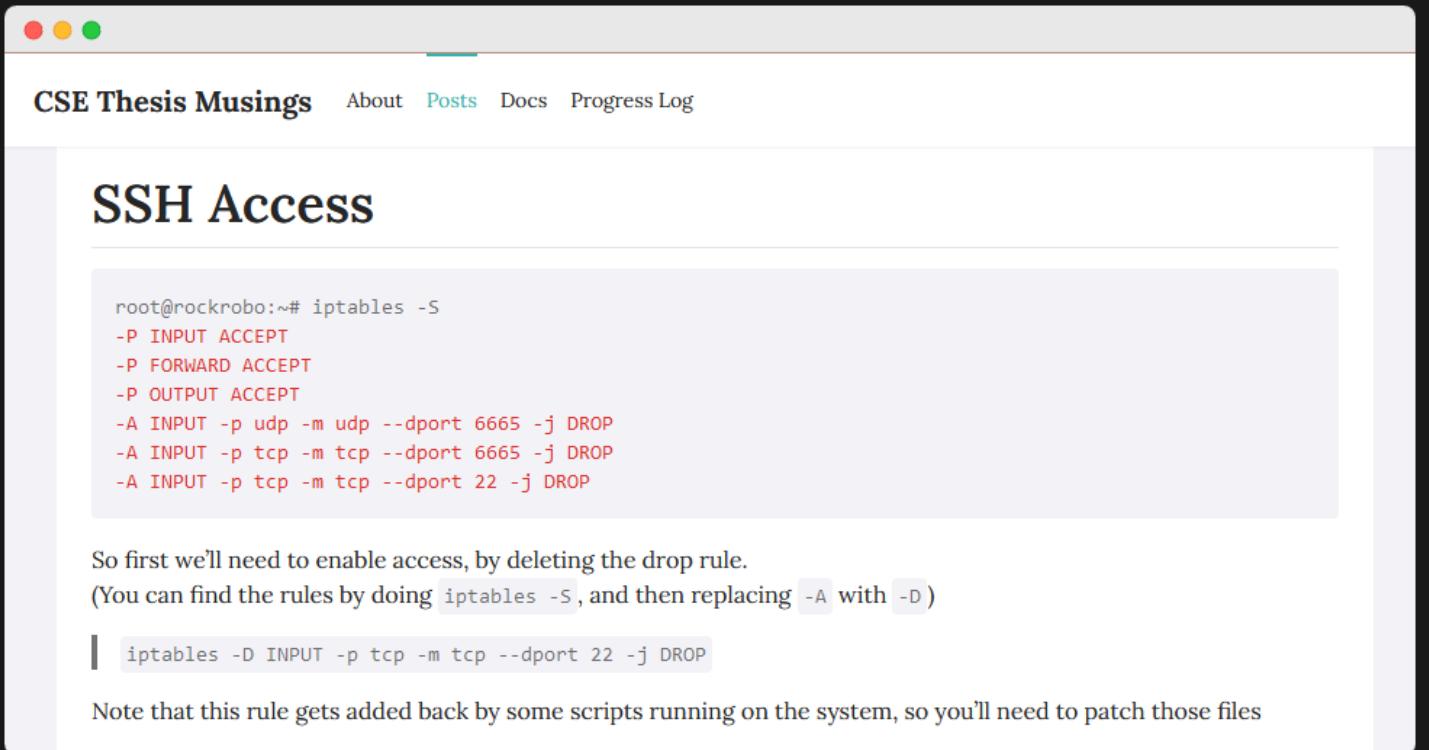
# Fingerprinting

## Other small tests

- Can I ping the internet / make outbound connections?
  - Yes
- Can I run my own software
  - Yes (armhf architecture)

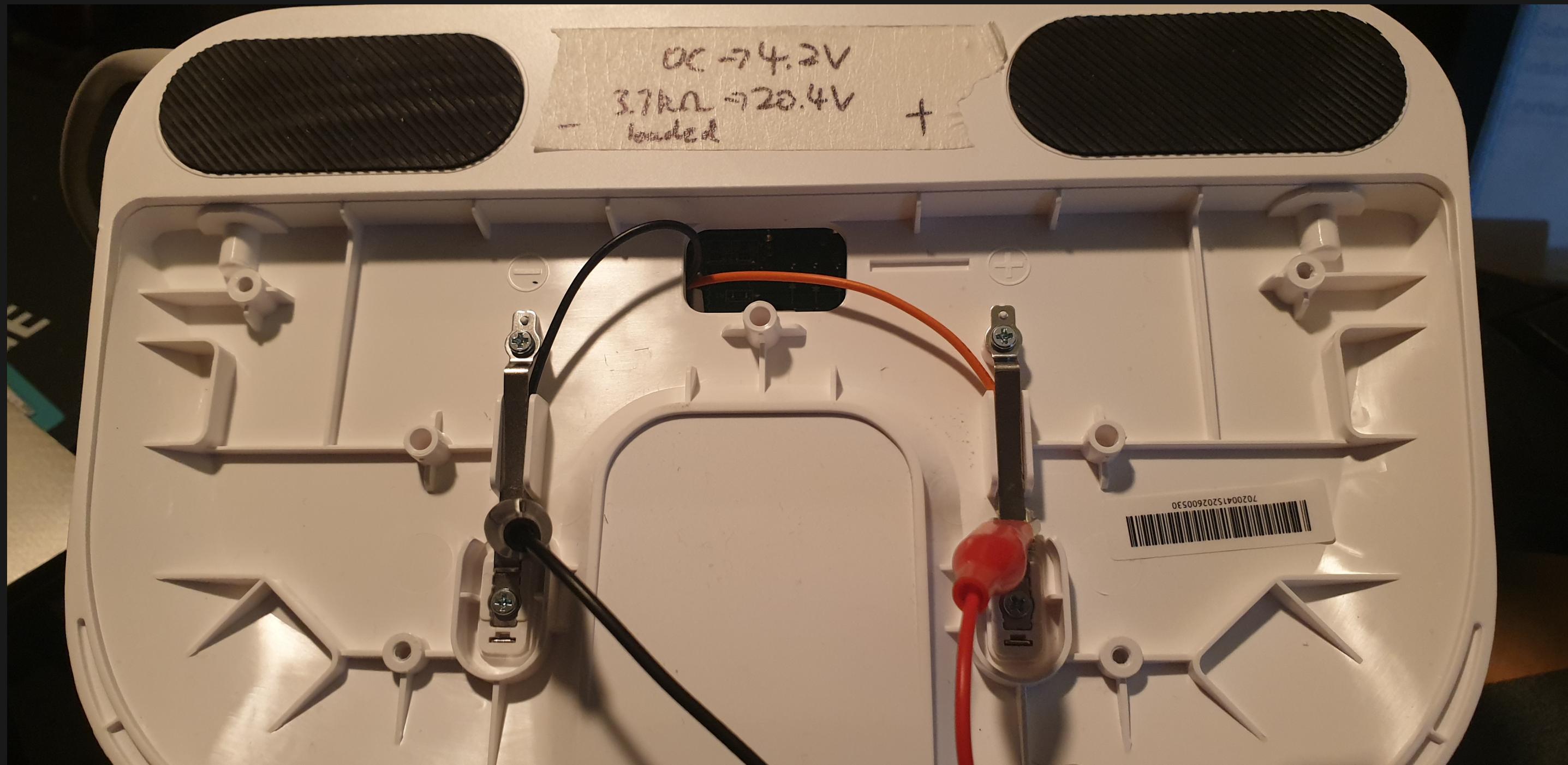
# Going wireless - establishing SSH

- Remove iptables rule to gain access
    - (and so could an attacker)
  - Can I add persistent access?
    - Yes, modify  
`rr watchdog.conf`
  - Can also add remote access
    - e.g. ZeroTier



# Trivial Power Analysis

*Batteries don't last forever!*



## *Test: What if I unplug the battery?*

- No change in output during boot
- But device will turn off after around 20 seconds

```
Ubuntu 14.04.3 LTS rockrobo ttyS0

rockrobo login: ###### Usual login prompt
wait-for-state stop/waiting
haveged: haveged Stopping due to signal 15 ###### Shutdown SIGTERM

        * Stopping rsync daemon rsync [ OK ]
        * (not running)
        * Asking all remaining processes to terminate... [ OK ]
        * All processes ended within 1 seconds... [ OK ]

umount: /tmp: device is busy.
        (In some cases useful info about processes that use
         the device is found by lsof(8) or fuser(1))
        * Unmounting temporary filesystems... [fail]
        * Deactivating swap... [ OK ]
        * Unmounting local filesystems... [ OK ]
        * Will now halt

[ 26.948171] [MCU_UART] sent ap poweroff event to mcu ###### Device turns off
```

See 2-wire log, 4-wire log

# File System Imaging

The eMMC only has 4GB of storage, so we can't (also shouldn't) image the flash onto itself... but we can image it remotely!

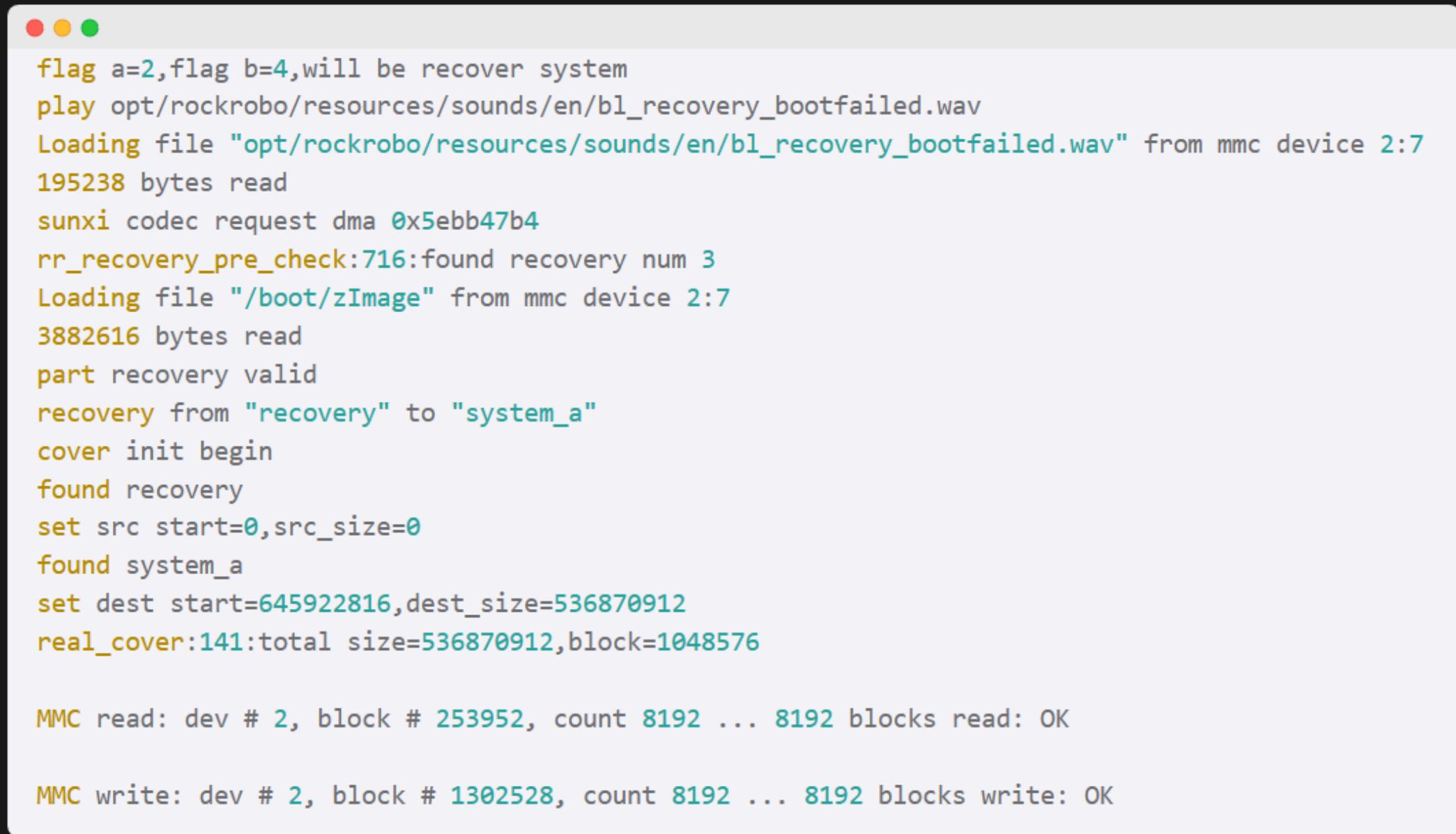
```
IP=10.10.10.8
for partition in `ssh root@$IP "ls /dev/mmcblk0?* -1"`
do
    ssh root@$IP "sudo dd if=$partition bs=1M" | dd of=$(basename $partition).img
done
```

# File System Structure

partition	label	size	description
mmcblk0p1	UDISK	1.5 GB	user data
mmcblk0p2	boot-res	8 MB	bootloader stuff
mmcblk0p5	env	16 MB	
mmcblk0p6	app (RO)	64 MB	device data
mmcblk0p7	recovery	512 MB	stock firmware
mmcblk0p8	system_a	512 MB	Main OS (boot)
mmcblk0p9	system_b	512 MB	Backup OS
mmcblk0p10	Download	528 MB	Update temp
mmcblk0p11	reserve	16 MB	blackbox???

# Recovery Reset

Recovery supposedly resets system\_a, system\_b, UDISK and Download

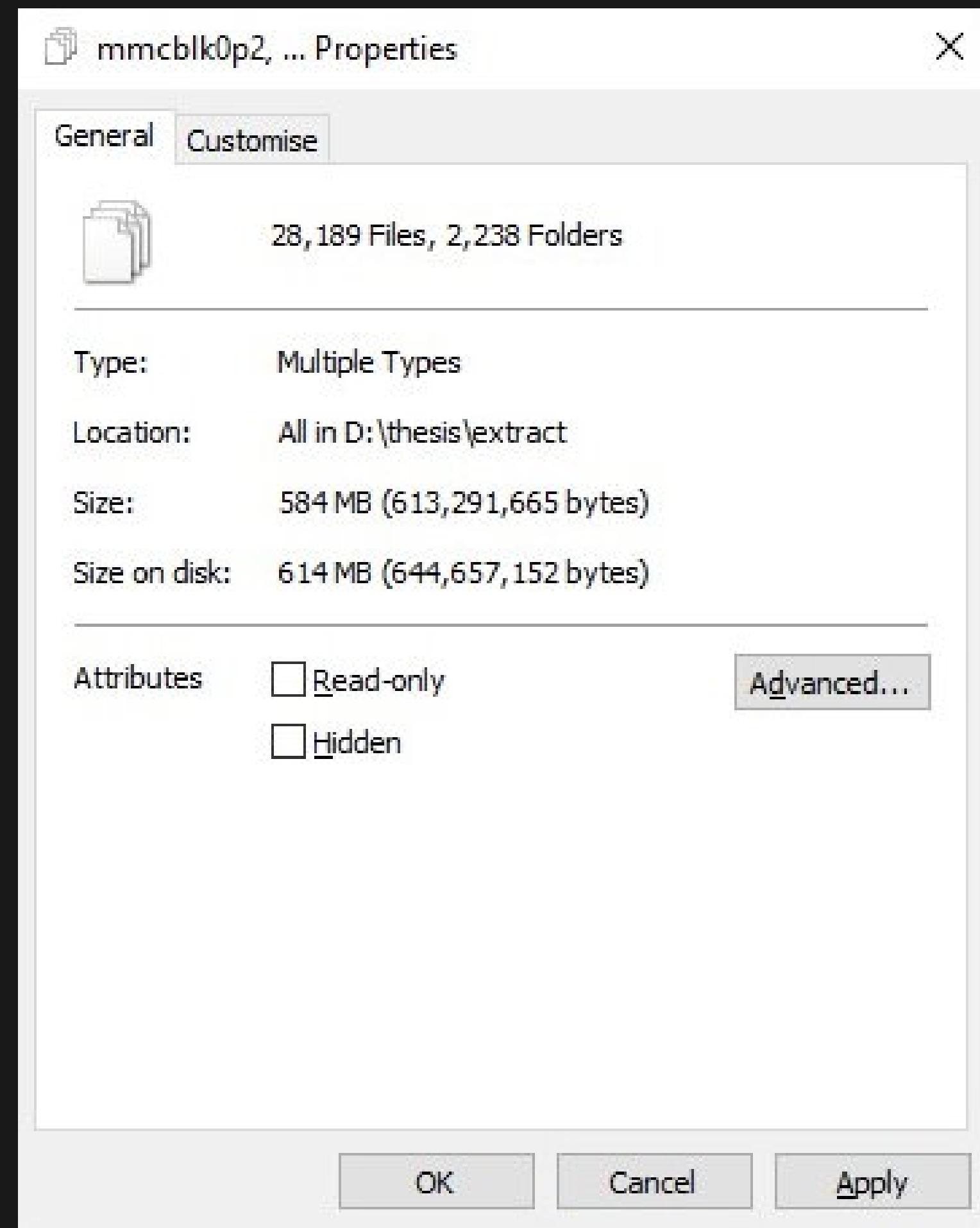
A terminal window showing the log output of a recovery process. The log includes messages about recovering the system, playing recovery boot failed sound, loading zImage from mmc device 2:7, and performing a sector copy operation from offset 645922816 to 1302528, size 536870912 bytes.

```
flag a=2,flag b=4,will be recover system
play opt/rockrobo/resources/sounds/en/bl_recovery_bootfailed.wav
Loading file "opt/rockrobo/resources/sounds/en/bl_recovery_bootfailed.wav" from mmc device 2:7
195238 bytes read
sunxi codec request dma 0x5ebb47b4
rr_recovery_pre_check:716:found recovery num 3
Loading file "/boot/zImage" from mmc device 2:7
3882616 bytes read
part recovery valid
recovery from "recovery" to "system_a"
cover init begin
found recovery
set src start=0,src_size=0
found system_a
set dest start=645922816,dest_size=536870912
real_cover:141:total size=536870912,block=1048576

MMC read: dev # 2, block # 253952, count 8192 ... 8192 blocks read: OK

MMC write: dev # 2, block # 1302528, count 8192 ... 8192 blocks write: OK
```

- What about the other partitions?
- Can we install software in the recovery partition? A: Yes



28,189 files...

*Well there's for sure a lot  
of files to look at...*

# I did a thing - Commentree

*Plain-text annotation / commentary tool*

# (some) Interesting Files

## The Search

- Looked for any passwords, secrets, keys, IDs, function calls, logs, ...
- Find changed files (\*)
- See where they are used
- See how they are used
- Anything of general interest

# (some) Interesting Files

- mmcblk0p1
  - miio/device.token
  - miio/device.uid
  - rockrobo/
  - rockrobo/rrlog/ (logs are encrypted!)
- mmcblk0p8/opt/rockrobo
  - Binaries
  - scripts/pipes.sh
  - rrlog/misc.sh
- mmcblk0p11/endpoint.bin - AWS address + key?

# (some) Interesting Files

## *mmcblk0p8/opt/rockrobo/rrlog/misc.sh*

```
...  
  
#echo "=====device.conf======" >> /dev/shm/misc.log  
#cat /mnt/default/device.conf >> /dev/shm/misc.log  
  
...
```

## *mmcblk0p6/device.conf*

```
did=DDDDDDDDDD          # (9 digits)  
key=XXXXXXXXXXXXXXXXXX  # (16 alpha-num, case-sensitive)  
mac=64:90:C1:1D:24:C4  
vendor=roborock  
model=roborock.vacuum.s6
```

# (some) Interesting Files

*Calls for system*

# (some) Interesting Files

## */var/log/apt/history.log*

Installed packages that are not part of the base system

```
Start-Date: 2016-01-25 11:18:05
Commandline: /usr/bin/apt-get install rsync
Install: rsync:armhf (3.1.0-2ubuntu0.2)
End-Date: 2016-01-25 11:18:11
```

```
Start-Date: 2016-04-05 12:30:59
Commandline: /usr/bin/apt-get install ccrypt
Install: ccrypt:armhf (1.10-4)
End-Date: 2016-04-05 12:31:01
```

```
Start-Date: 2016-04-25 09:58:29
Commandline: /usr/bin/apt-get install tcpdump
Install: tcpdump:armhf (4.5.1-2ubuntu1.2), libpcap0.8:armhf (1.5.3-2, automatic)
End-Date: 2016-04-25 09:58:33
```

- Why does a vacuum cleaner need rsync or tcpdump?
- No usage calls found yet

# (some) Interesting Files

*mmcblk0p7/usr/sbin/tcpdump*

- External but unmodified binary
- Only hub traffic visible (wireless)
- (not really that interesting)

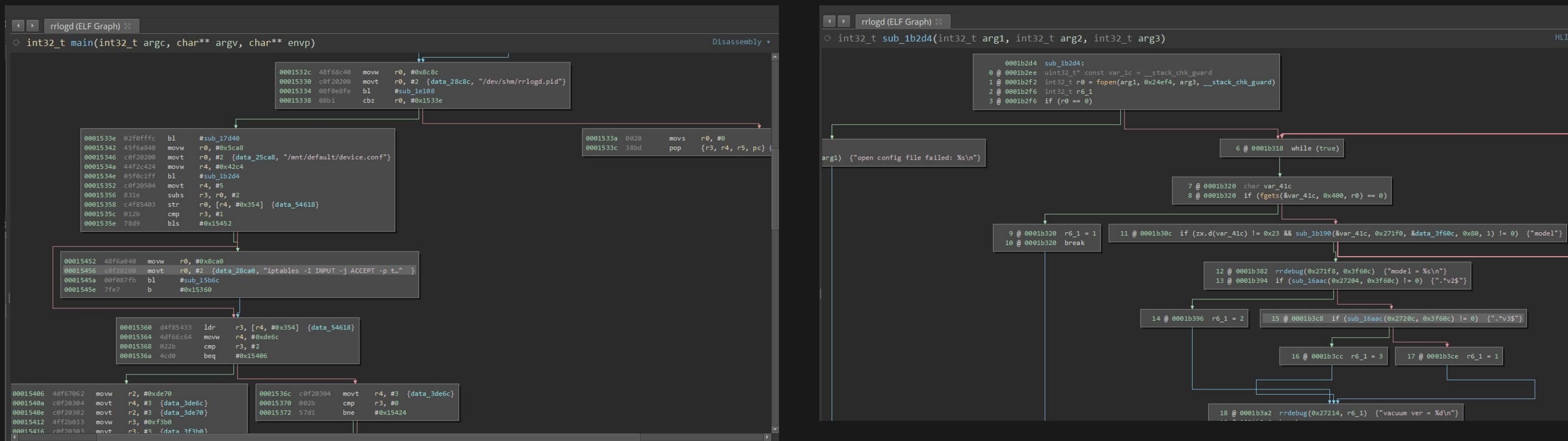
# (some) Interesting Files

*mmcblk0p8/opt/rockrobo/rrlog/rrlogd*

Logs are encrypted at rest (after being packed)

Originally used to be a symmetric key, now using a public key

⌚ Logging program has the functionality to unblock port 22?



`iptables -I INPUT -j ACCEPT -p tcp --dport 22`

# (some) Interesting Files

*mmcblk0p6/vinda*

Previously... XOR this file to get the root password

## File References

Name	Located In	File Extension	Date Modified	Size	Count
Journal	D:\thesis\extract\mmcblk0p6\[SYS]		1/01/1970 11:00:14 A...	4.00 MB	5 / 5
SysUpdate	D:\thesis\extract\mmcblk0p7\opt\rockrobo\cleaner\bin		25/03/2020 11:40:13 ...	250 KB	2 / 1
adbd	D:\thesis\extract\mmcblk0p7\usr\bin		25/03/2020 11:48:31 ...	482 KB	1 / 1
SysUpdate	D:\thesis\extract\mmcblk0p8\opt\rockrobo\cleaner\bin		25/03/2020 11:40:13 ...	250 KB	2 / 1
adbd	D:\thesis\extract\mmcblk0p8\usr\bin		25/03/2020 11:48:31 ...	482 KB	1 / 1

**AstroGrep Search**

Search Path: D:\thesis\extract

File Types: \*\*

Search Text: vinda

Results:

```
2904 NUL NUL 1 use default adb_passwd: %s
2905 NUL NUL NUL 2 use default adb_passwd: %s
2906 NUL NUL NUL /mnt/default/vinda NUL NUL rockroboproduct@NUL NUL NUL NUL use default sys_passwd: %s
2907 NUL /proc/cmdline NUL NUL NUL boot_reason=NUL NUL NUL NUL adb get boot_reason: %x
2908 NUL NUL NUL NUL adb can't find the valid boot_reason
```

Search Cancel

# (some) Interesting Files

## *mmcblk0p7/usr/bin/adbd*

- Custom ADB binary
- Had a brief look ([more](#))

```
locksec_init_key: can not find the prefix str from adb conf file, use default
locksec_init_key: can not find the suffix str from adb conf file, use default
locksec_init_serial: adb read 465 bytes from /proc/cpuinfo
locksec_init_key: locksec_init_key, rockrobo%()+-[]_8a80ab8936d76c118000:;<=>?@{}rubyde
locksec_apply_key: locksec_apply_key, erI09cyW%()+-[]_8a80ab8936d76c118000:;<=>?@{}CzD2
locksec_apply_passwd: adb source str: erI09cyW%()+-[]_8a80ab8936d76c118000:;<=>?@{}CzD2
locksec_apply_passwd: locksec_apply_passwd, passwd: 0y[ad8@w
```

## Related files

- mmcblk0p6/vinda
- mmcblk0p6/adb.conf
- mmcblk0p8/var/log/upstart/adbd.log

# (some) Interesting Files

## Future: the other programs

- cleaner
- miio
- rockrobo
- rrlog
- rriot

# Issues, thoughts & discussions

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

*Wireless credentials are stored in plain text*

- Anyone with physical access to the machine can gain wireless credentials
- However, takes a lot of effort to open up the device
- Why? `wpa_supplicant` is part of the underlying Linux framework

# Issues, thoughts & discussions

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

⚠ *SSH server exposed on tcp/22*

- Why does this server exist?
- When / where is it used?
  - Allow rule inside the `rrlogd` binary
- Roborock has made an attempt to protect their product with `iptables`
- But did not fully protect their product against access via IPv6

# Issues, thoughts & discussions

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

*Processes are running as root*

- Any vulnerability in any of the programs can result in elevated access
  - Dropping of iptables restrictions
  - Persistence planting
  - System takeover
- Should run as a de-privileged user
- Why? Compatibility, perhaps ease of development
  - i.e. udev rules

# Issues, thoughts & discussions

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

## *Recovery partition is modifiable*

- Can be modified to contain malicious software that persists a factory reset
- Mountable - mount `/dev/mmcblk0p7` . . .
- Why? Allows easy updates of the ‘factory image’
- But the partition could somehow be encrypted

# Issues, thoughts & discussions

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

## A note on hardware and software

*access to the hardware = game over?*

- Are there tamper-proof / tamper-evident design possibilities?
- What about some sort of “Secure Element”
- Or read protection?
- Choice of OS
- Choice of auth implementation (e.g. vinda)
- Limitation on what programs are allowed to execute?

# Issues, thoughts & discussions

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

## The Good Things

- An effort to restrict SSH access via iptables
- AuthN / AuthZ is present within interfaces to the device
- UART shell requires a password
- Logs are encrypted locally

# Current Challenges

## *Intercepting encrypted data / TLS traffic*

- Ubuntu 14.04 has some issues (?)
  - PolarProxy is too new (libc requirements)
  - apt update doesn't work with socks5:// or http proxies properly???
- Routing?
- Hook into the encryption/decryption process somehow?
  - Use Frida?
  - Or look at the data communicated by the smartphone app?
    - Objection tool didn't work with the RoboRock app

# Current Challenges

*Electricity is funny.*

Using my main personal computer is not a good idea for a test-bench...  
Thank you Gigabyte for having ESD-protected USB ports

# Current Challenges

*Still a lot of files to look at*

Need to figure out which files are worthwhile to inspect..

# Current Challenges

## File Inspection Approach 1 - Filter by date modified

*Ubuntu 14.04.3 LTS was released back in 2014, any changes would have a later timestamp (hopefully)*

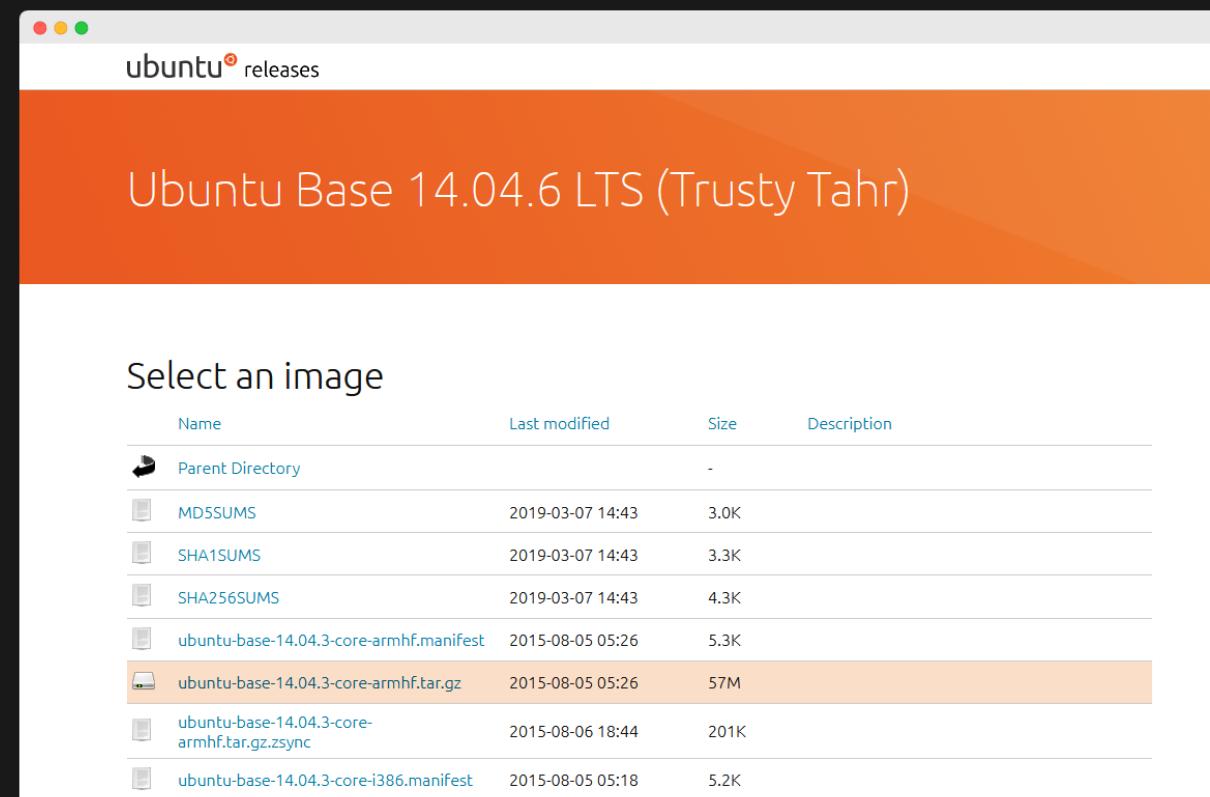


Google search results for "ubuntu 14 release date". The search bar shows the query. Below it, there are filters for All, Images, News, Videos, Shopping, More, and Tools. It displays "About 64,200,000 results (0.93 seconds)". A snippet of text states: "Ubuntu 14.04 LTS / Initial release date" followed by "17 April 2014". A screenshot of the Ubuntu desktop environment is shown below the text. At the bottom, a snippet from Wikipedia says: "On 18 October 2013, it was announced that Ubuntu 14.04 would be dubbed "Trusty Tahr". This version was released on **17 April 2014**, and is the 20th release of Ubuntu." A link to "Ubuntu version history - Wikipedia" is provided.

Name	Date modified	Type
nologin	18/03/2022 10:10 PM	.symlink
blkid.tab	18/03/2022 10:10 PM	.symlink
vtrgb	18/03/2022 10:10 PM	.symlink
ld.so.cache	25/03/2020 11:48 PM	CACHE File
OS_VERSION	25/03/2020 11:48 PM	File
subgid	25/03/2020 11:48 PM	File
group	25/03/2020 11:48 PM	File
gshadow	25/03/2020 11:48 PM	File
subuid	25/03/2020 11:48 PM	File
passwd	25/03/2020 11:48 PM	File
os-release	25/03/2020 11:45 PM	File
fstab	25/03/2020 11:44 PM	File
modules	25/03/2020 11:44 PM	File
rc.local	25/03/2020 11:44 PM	LOCAL File
toprc	23/01/2016 5:08 PM	File
mailcap	4/01/2016 5:03 PM	File
dnsmasq.conf	30/12/2015 1:02 PM	CONF File

# Current Challenges

## File Inspection Approach 2 - Binary Comparisons



Compare executable files and find differences in binary function

**bindiff, binwalk, ssdeep, sdhash**

As seen in [A Large-Scale Analysis of the Security of Embedded Firmwares](#) - Andrei C, Jonas Z, Aur'elien F, Davide B

# Thesis B Retrospective

- Time management - could have done more work
  - Busy / other commitments
  - Hardware work restricts me to only working at home
- Project breadth / depth / scope
  - Binary analysis takes a lot of time

## Response

- Schedule more focus times
- Hardware work pretty much completed - likely able to work remotely now
- Restrict binary analysis to the most likely binaries
  - May consequently miss something

## Thesis B Completion

- Analysis of firmware binaries to identify vulnerabilities
  - Still in progress
- Search for unsecured secrets, logs, configurations
  - Completed (excluding encrypted rrlog files)

## Revised Thesis C Plan

- (priority) Inspection of outbound WAN traffic - security, PII, etc
- ~~Inspection of LAN traffic~~ rather, see if it is stored
- ~~Inspection of interaction with nearby devices~~
- ~~Protocol analysis~~
- Update to a newer firmware version and look at changes
- Check what files gets cleared during a format
- Binary assessment
- Verify IPv6 SSH access

# Incoming Timeline

- 22T2 W1 - IPv6 SSH verification, continue binary assessment
- 22T2 W2 - WAN traffic analysis
  - Look at network behaviour
  - Try view WAN data pre-encryption / post-decryption
- 22T2 W4 - Update to latest version (and hope we don't get locked out)
  - Do another vacuum clean, reimagine, compare binaries
- 22T2 W5 - Factory reset device, check for remnant files
- 22T2 W8 - Demo submission
- 22T2 W11 - Report submission

# Thank You

---

Andrew Wong

w: [featherbear.cc/UNSW-CSE-Thesis](http://featherbear.cc/UNSW-CSE-Thesis)

e: [andrew.j.wong@student.unsw.edu.au](mailto:andrew.j.wong@student.unsw.edu.au)