

“Smart” Vacuum Cleaners

An Audit Into The Security and Integrity of IoT Systems

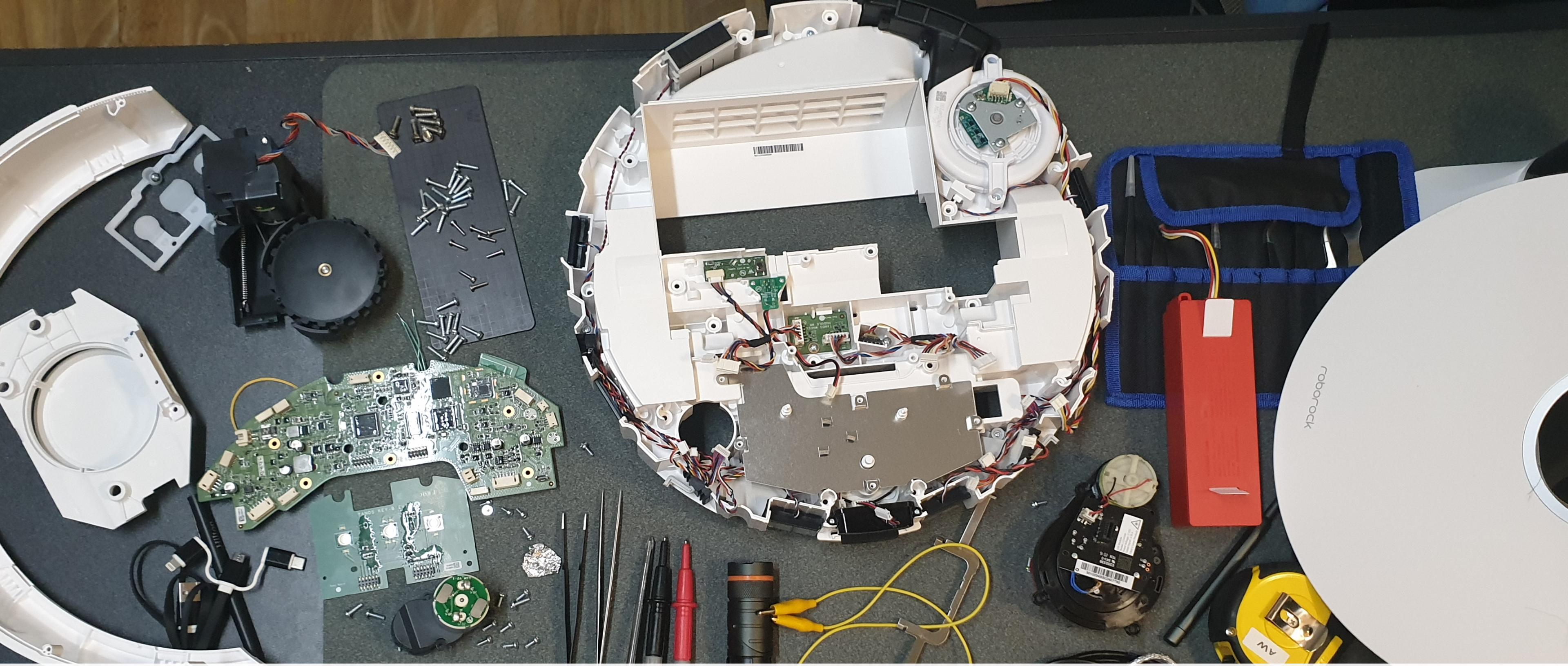
Andrew Wong | UNSW Sydney

Today's Agenda

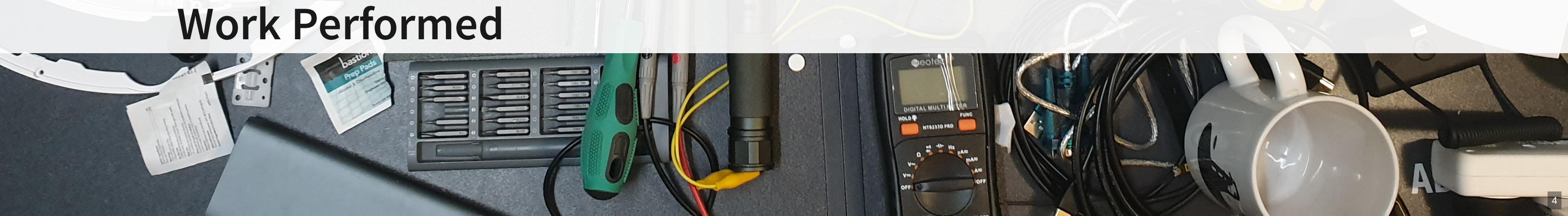
Statement

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.



Work Performed



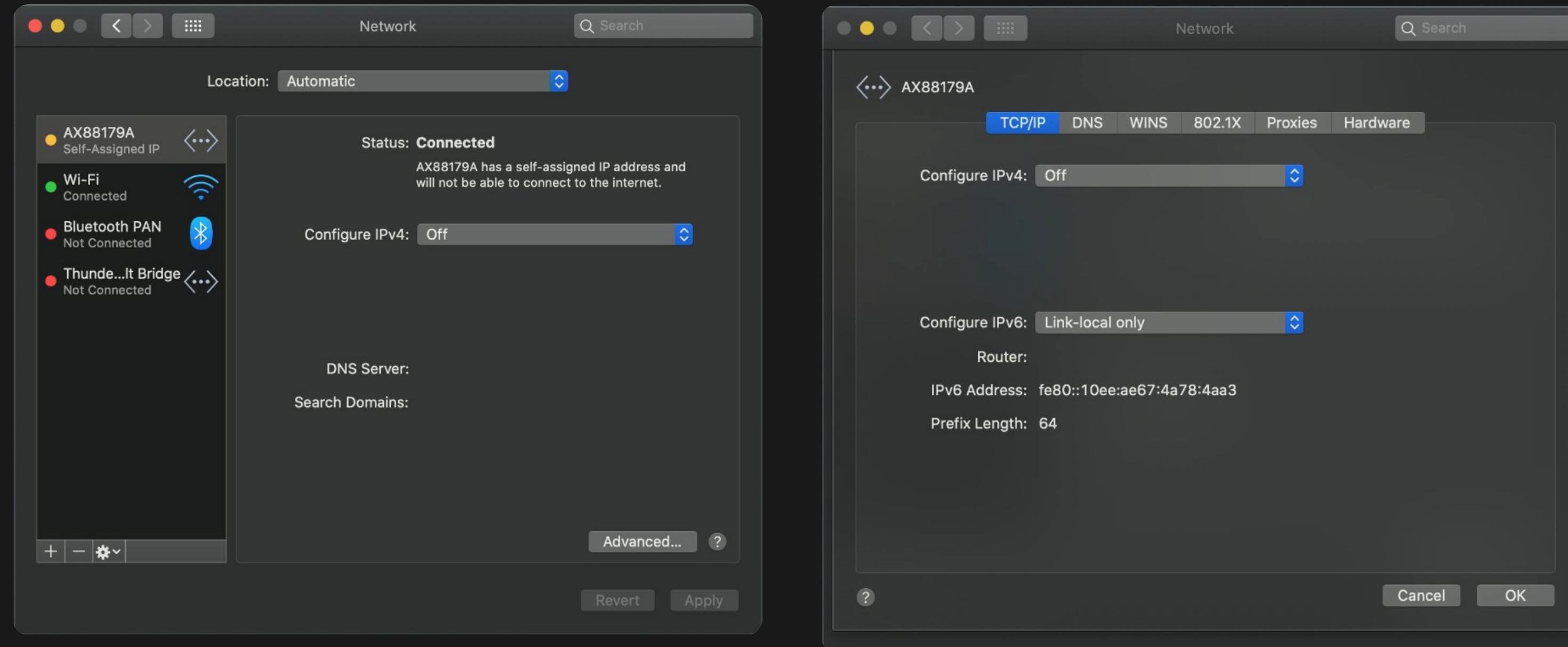
More logging

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... now port mirroring from a switch (TP-Link TL-SG105E)



- The switch doesn't offer true port mirroring - so also seeing sink data
- Disabled IPv4 and (attempt to disable) IPv6 on the network adapter

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

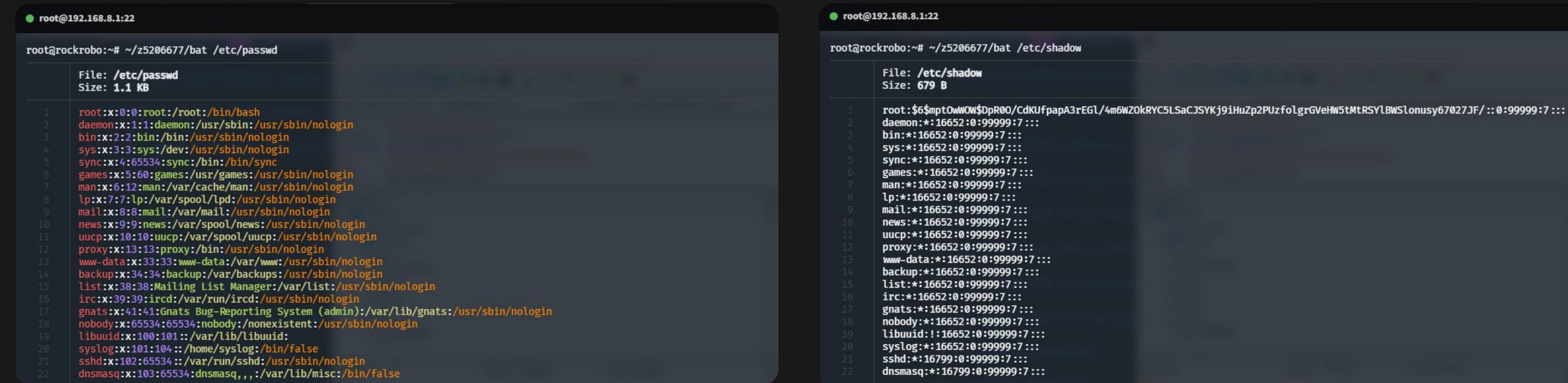
ACU: STM32F103VCT6 (ARM Cortex-M3)

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

```
root@rockrobo:~# netstat -nltp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
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
 - Why not 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

Fingerprinting

Other small tests

- Can I ping the internet?
 - Yes
- Can I run my own software
 - Yes (armhf)

Going wireless - establishing SSH

The screenshot shows a blog post titled "SSH Access" from "CSE Thesis Musings". The post contains a terminal session showing iptables rules:

```
root@rockrobo:~# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p udp -m udp --dport 6665 -j DROP
-A INPUT -p tcp -m tcp --dport 6665 -j DROP
-A INPUT -p tcp -m tcp --dport 22 -j DROP
```

Text below the terminal session:

So first we'll need to enable access, by deleting the drop rule.
(You can find the rules by doing `iptables -S`, and then replacing `-A` with `-D`)

```
| iptables -D INPUT -p tcp -m tcp --dport 22 -j DROP
```

Note that this rule gets added back by some scripts running on the system, so you'll need to patch those files

Below the blog post is a screenshot of a network configuration interface for a virtual interface named "abit_suspicious_dont_you_think". The interface details are as follows:

ID	abfd/[REDACTED]	Managed IPs
Name	abit_suspicious_dont_you_think	10.147.20.87/24
Type	PRIVATE	
Status	OK	
Ethernet MAC	42:53:[REDACTED]	
Virtual NIC Device	ethernet_32784	
Virtual NIC MTU	2800	
Ethernet Broadcast	enabled	
Ethernet Bridging	prohibited	
DNS Domain	(not configured)	
DNS Servers	(none)	
Allow Managed IPs	<input checked="" type="checkbox"/>	
Allow Global Internet IPs	<input type="checkbox"/>	
Allow Default Route Override	<input type="checkbox"/>	
Allow DNS Configuration	<input type="checkbox"/>	

Buttons at the bottom right: "Disconnect"

- 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!

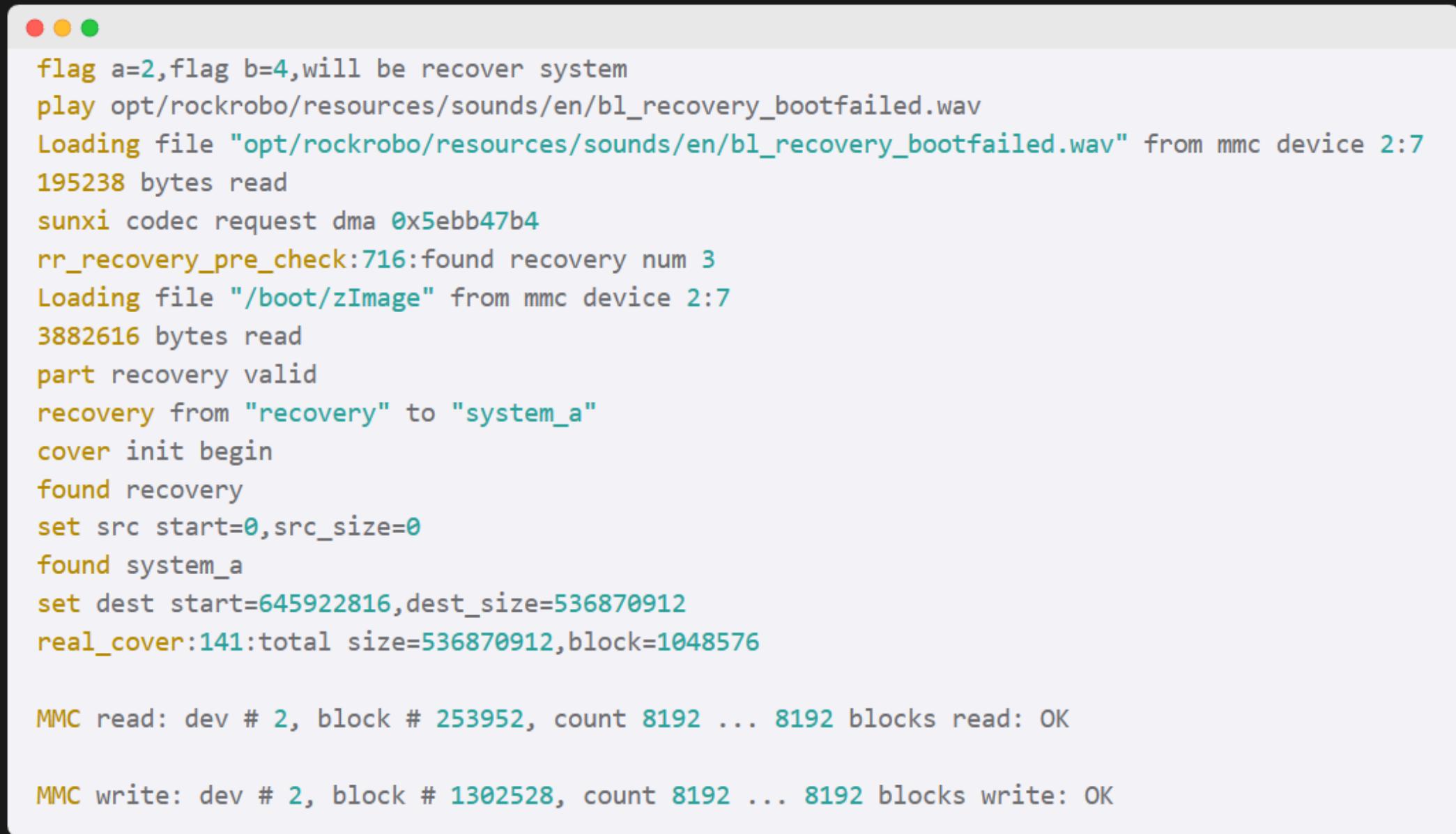
```
IP=10.10.10.8
for partition in `ssh root@$IP "ls /dev/mmcblk0?* -1"`
do
    ssh root@$IP "sudo dd if=$partition bs=1M" | dd of=$(basename $partition).img
done
```

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

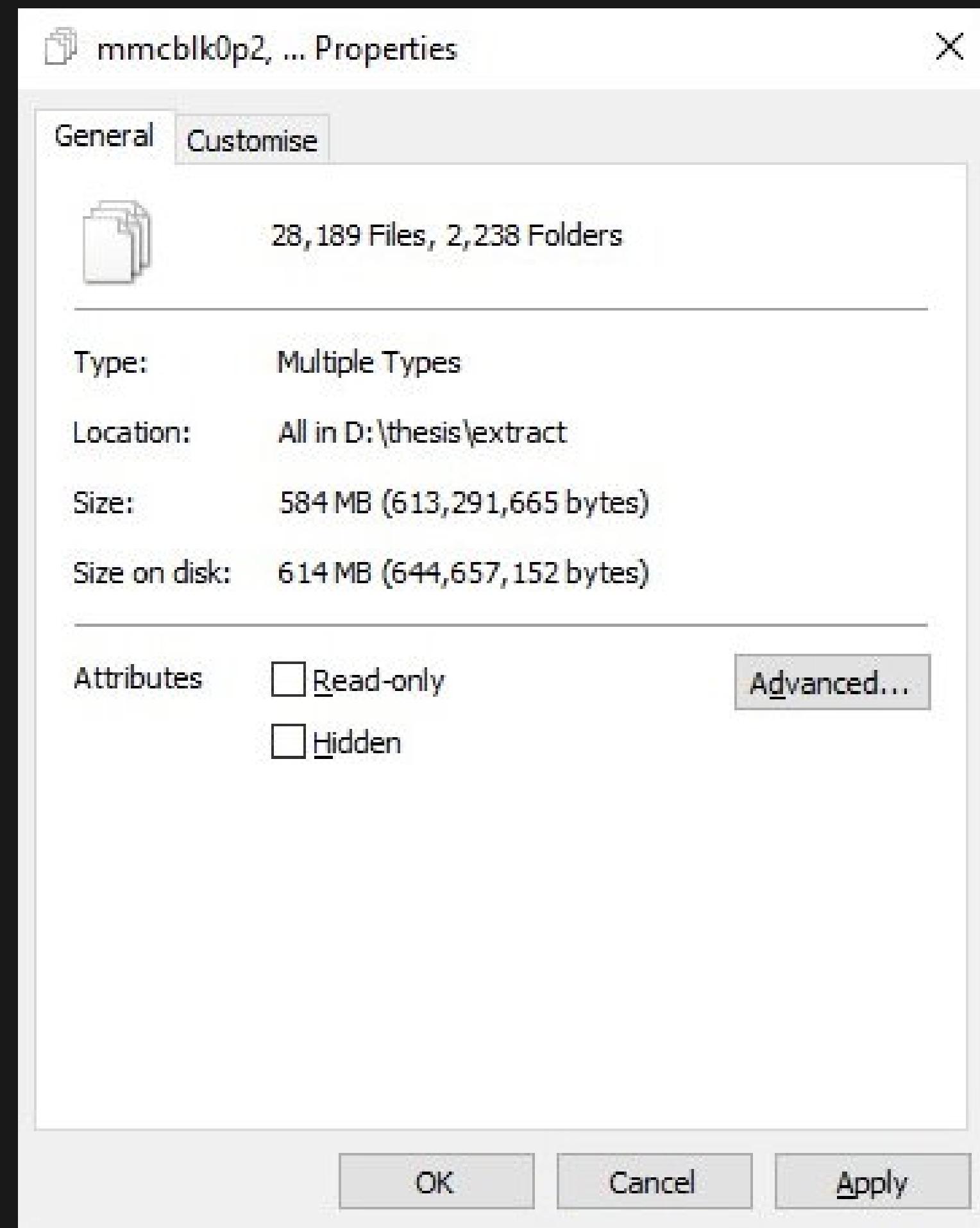
A terminal window showing the log output of a recovery process. The log includes messages about recovering the system, loading files from mmc device 2:7, and performing writes to mmc device 2.

```
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 plant malicious software in recovery? 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

Interesting Files

- mmcblk0p1/miio/device.token=utnevRELra5sqef3
- mmcblk0p1/miio/device.uid=1738271950
- mmcblk0p1/rockrobo/
- mmcblk0p11/endpoint.bin - AWS address + key?
- mmcblk0p7/boot/zImage - bootloader

vinda usage

passwords syslogs

Look for any emails

Look for IPs, emails, host/domains, passwords, keys

Check where DID and UID is used

Dummy data to check if it's logged

What other files were changed?

compare against base ubuntu system?

Logs

/var/log/apt/history.log

```
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
```

tcpdump

/usr/sbin/tcpdump

```
root@rockrobo:/var/log# cat dpkg.log | grep "install "
2016-04-25 09:58:30 install libpcap0.8:armhf <none> 1.5.3-2
2016-04-25 09:58:31 install tcpdump:armhf <none> 4.5.1-2ubuntu1.2
2016-05-03 12:06:01 install pigz:armhf <none> 2.3-2
2016-05-23 09:41:25 install lsof:armhf <none> 4.86+dfsg-1ubuntu2
2016-06-23 15:02:26 install gcc-6-base:armhf <none> 6.1.1-3ubuntu11~14.04.1
2016-06-23 15:06:01 install gcc-5-base:armhf <none> 5.3.0-3ubuntu1~14.04
root@rockrobo:/var/log#
```

miio mmcblk0p7\opt\rockrobo rrlog rriot

ADB

adb

- Custom ADB binary
- Had a brief look ([more](#))
- mmcblk0p6\adb.conf
- mmcblk0p8\var\log\upstart\adbd.log

rsync and tcpdump

Static binaries

```
./htop --sort-key=PID -C
```

Compare against standards (i.e. Xiaomi's standard)

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

“Once you have access to the hardware, it’s game over”

- netcat, tcpdump, ccrypt?

How easy is it for someone to attack the system?

What's Good

- iptables
- some logs are encrypted locally
- Hands-on access a system = game over
 - But should it be?

Are there any backdoors? outbound requests persistent software (install to recovery + system_a)



```
root@rockrobo:/proc# cat misc
48 android_adb
49 mali
50 network_throughput
51 network_latency
52 cpu_dma_latency
53 xt_qtaguid
54 leds
236 device-mapper
130 watchdog
200 tun
55 lds_motor
56 jiffies
57 uart_lds
58 uart_mcu
237 loop-control
59 sw_sync
60 cuse
229 fuse
61 sunxi-reg
62 cachefiles
63 ion
```

- Can I plant software (y)

- Why is netcat installed but not curl, wget?
- TODO: Check what gets cleared during a format / update

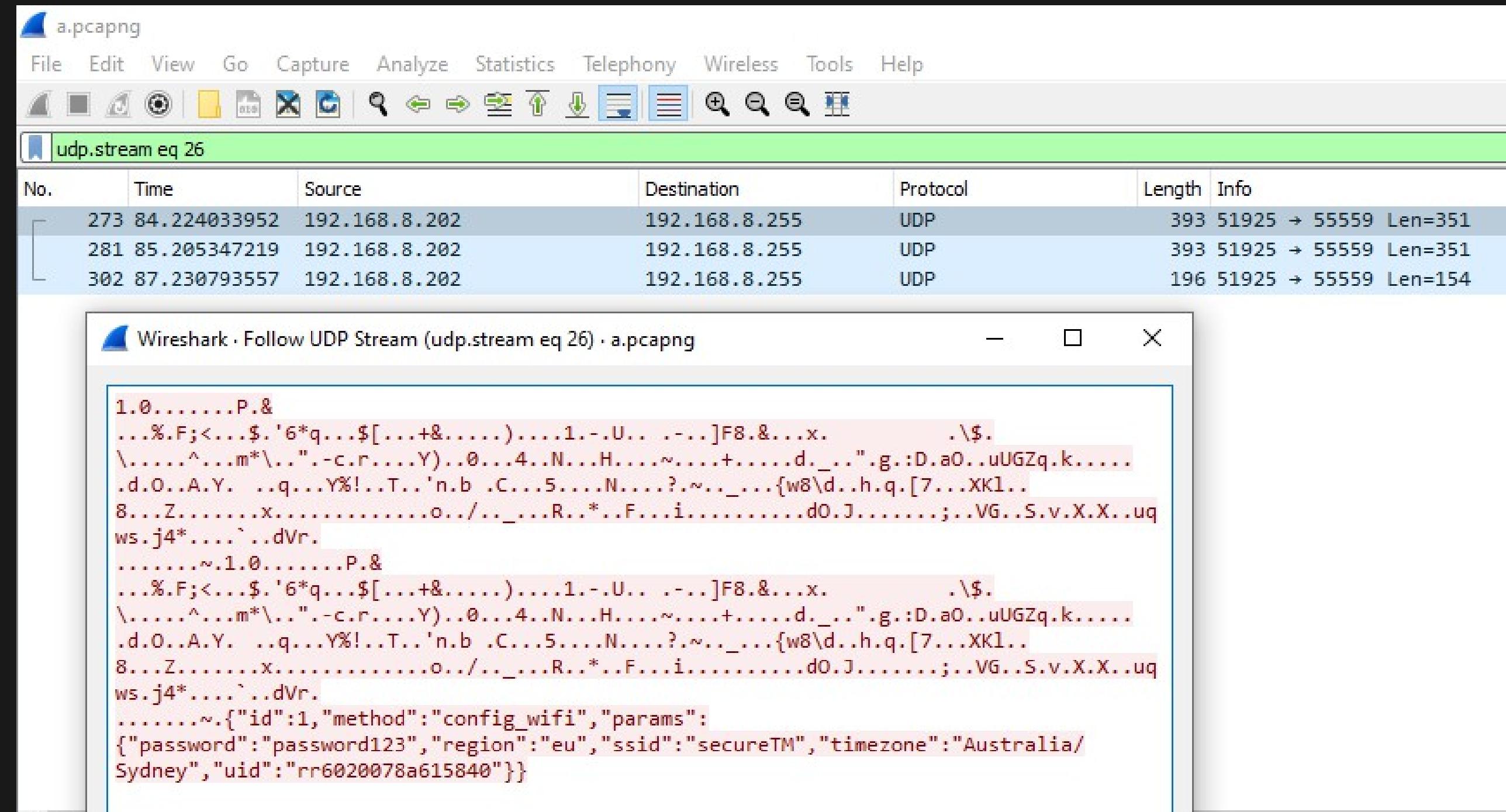
Capturing network data during device registration

- Have shell access but
- PolarProxy is too new for Ubuntu 14.04
- apt update doesn't work with socks5:// or http proxies properly

On the smart app side

Frida nope windows env nope <https://github.com/NickstaDB/patch-apk>
<https://blog.silentsignal.eu/2020/05/04/decrypting-and-analyzing-https-traffic-without-mitm/> RoboRock app

during pairing the password is transmitted in plaintext



Current Challenges - Equipment

- Electricity is dangerous
- Using personal equipment is not a good idea for a test-bench
- Thank you Gigabyte for having ESD-protected USB ports

rrlogd

Logs are encrypted at rest (after being packed)

rrlogd

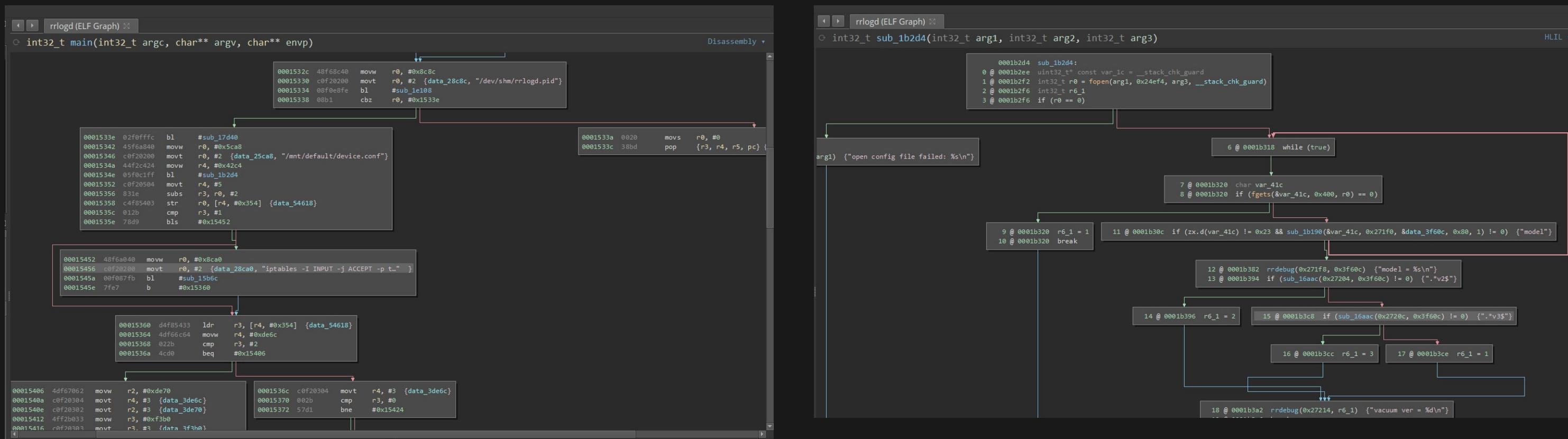
Possible functionality to perform any arbitrary command?

```
int32_t __saved_r6 {Frame offset -8}
int32_t var_4 {Frame offset -4}
int32_t arg1 {Register r0}

sub_15b6c:
00015b6c 70b5      push   {r4, r5, r6, lr} {var_4} {__saved_r6} {__saved_r5} {__saved_r4}
00015b6e 0021      movs   r1, #0
00015b70 0546      mov    r5, r0
00015b72 1120      movs   r0, #0x11
00015b74 fff7a0e8  blx    #signal
00015b78 2946      mov    r1, r5
00015b7a 0646      mov    r6, r0
00015b7c 46f23800  movw   r0, #0x6038
00015b80 c0f20200  movt   r0, #2 {data_2601c[0x1c], "%s\n"}
00015b84 08f074fa  bl    #rrdebug
00015b88 2846      mov    r0, r5
00015b8a fff7aee8  blx   #system
00015b8e 3146      mov    r1, r6
00015b90 0446      mov    r4, r0
00015b92 1120      movs   r0, #0x11
00015b94 fff790e8  blx   #signal
00015b98 621c      adds   r2, r4, #1
00015b9a 1fd0      beq   #0x15bdc

00015b9c 6306      lsls   r3, r4, #0x19
00015b9e c4f30721  ubfx   r1, r4, #8, #8
00015ba2 09d1      bne   #0x15bb8
```

rrlogd



Logging program has the potential to unblock port 22?

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


Unfinished Work

- Still a lot of files to look at
- Need to figure out which files are worthwhile to inspect

Unfinished Work

- Still a lot of files to look at
- Need to figure out which files are worthwhile to inspect

Approach 1 - Filter by date modified

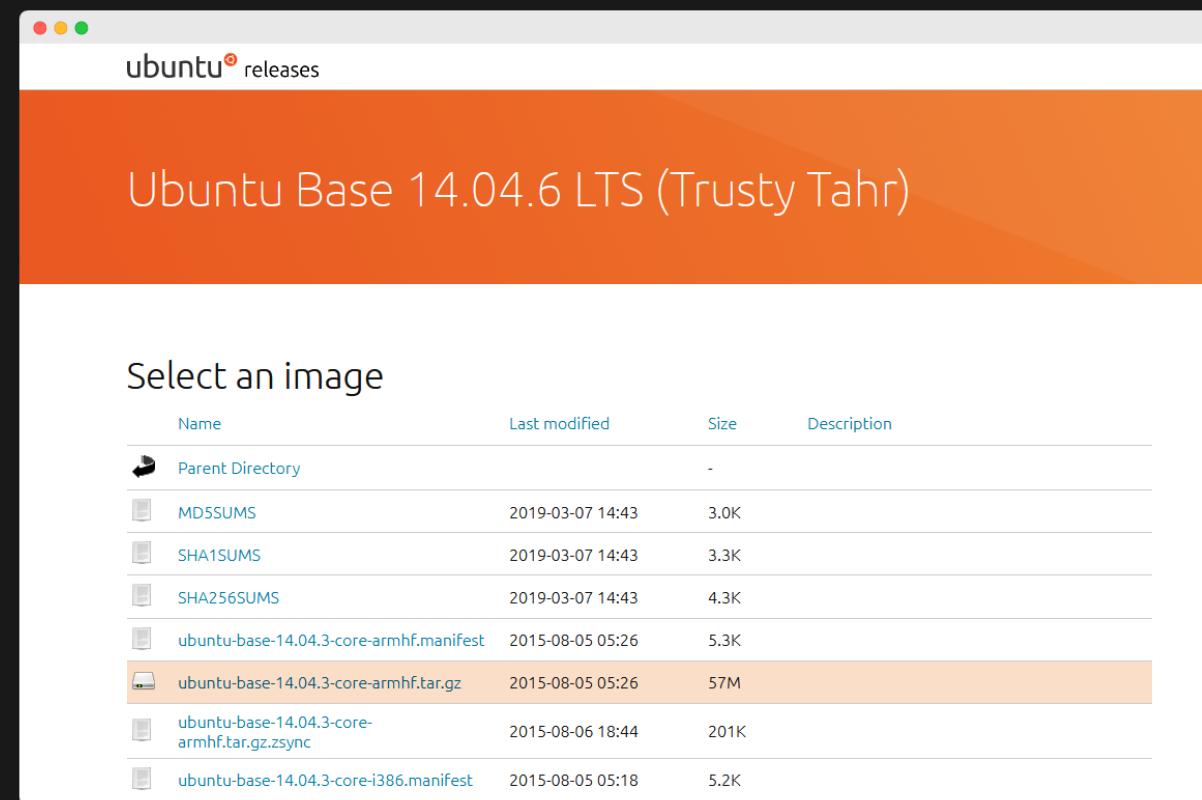
Ubuntu 14.04.3 LTS was released back in 2014, any changes would have a later timestamp

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

Unfinished Work

- Still a lot of files to look at
- Need to figure out which files are worthwhile to inspect

Approach 2 - File 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

Retrospective

- Time management / busy / other work
- Could have done more work

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

[Static] Binary assessment

In the mean time

Findings

Project Plan

Revised Project Plan

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