

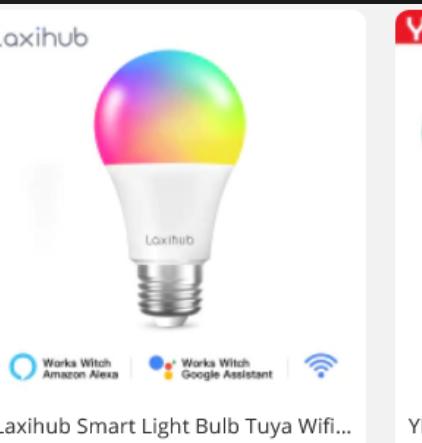
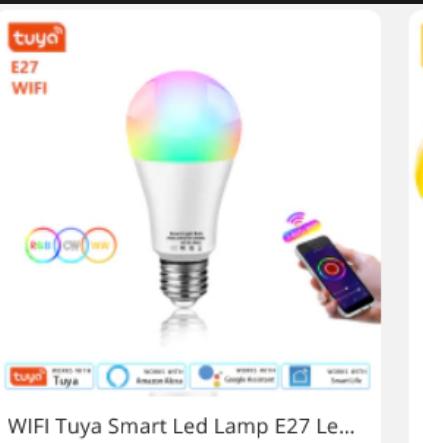
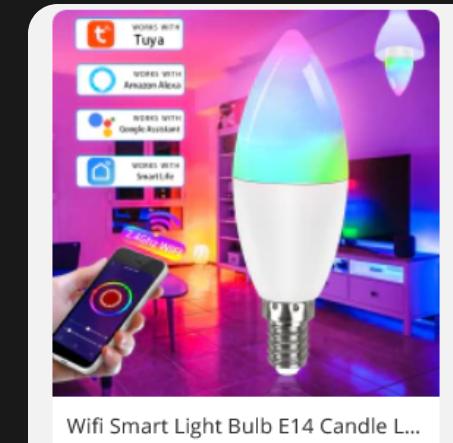
“Smart” Vacuum Cleaners

An Audit Into The Security and Integrity of IoT Systems

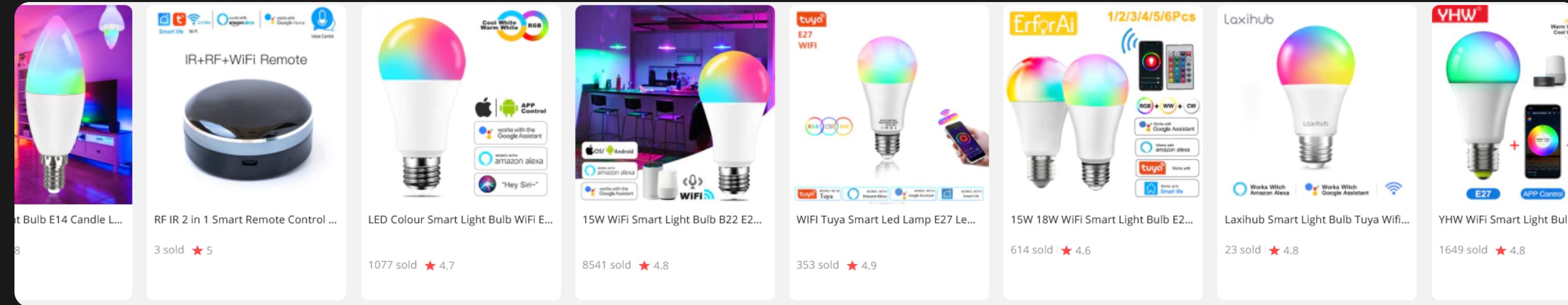
Andrew Wong | UNSW Sydney

Today's Agenda

- Topic Recap
 - The world of IOT devices
 - Thesis statement
- Where we left off
 - Previous results and findings
- New progress
- Discussion
- Conclusion



...so there are a lot of IOT devices and IOT brands out there...



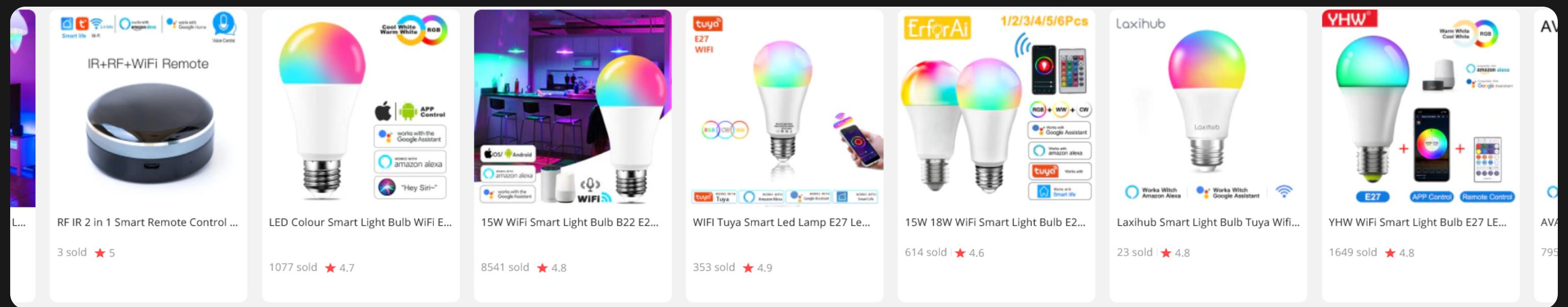
Are competing products looking suspiciously similar to you?
Most are white-labelled products, the biggest ecosystem vendor being **tuya**

Pros

- Use someone else's code
- Fast profit turnaround

Cons

- ⚠ Use someone else's code
- Potentially security vulnerabilities



IOT ecosystems often have a centralised system to manage their fleet (devices).

Pros

A centralised management is so much simpler/easier/faster/cheaper/‘better’ than a decentralised one.

Cons

- ⚠ Device functionality dependent on system availability
- ⚠ Little transparency about what/where/when/why data is transmitted

Statement

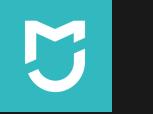
How have manufacturers of IoT / smart home devices addressed the increasing concerns of digital privacy and product security?

(Specifically Roborock / Xiaomi / Tuya)

- Digital Privacy
 - Investigate the nature of network data
 - i.e. content, frequency, destination, usage
- Product Security
 - Investigate security vulnerabilities
 - Assess the effectiveness of security fortifications

Statement

Our device in scope: Roborock S6 (2019)

A smart vacuum cleaner, with
integrations to both  and 
(depending on model)



*It works pretty (very) well, according to reviews.
But is it safe to connect to your home?*

Talk about thesis A talk about thesis b Talk about thesis c

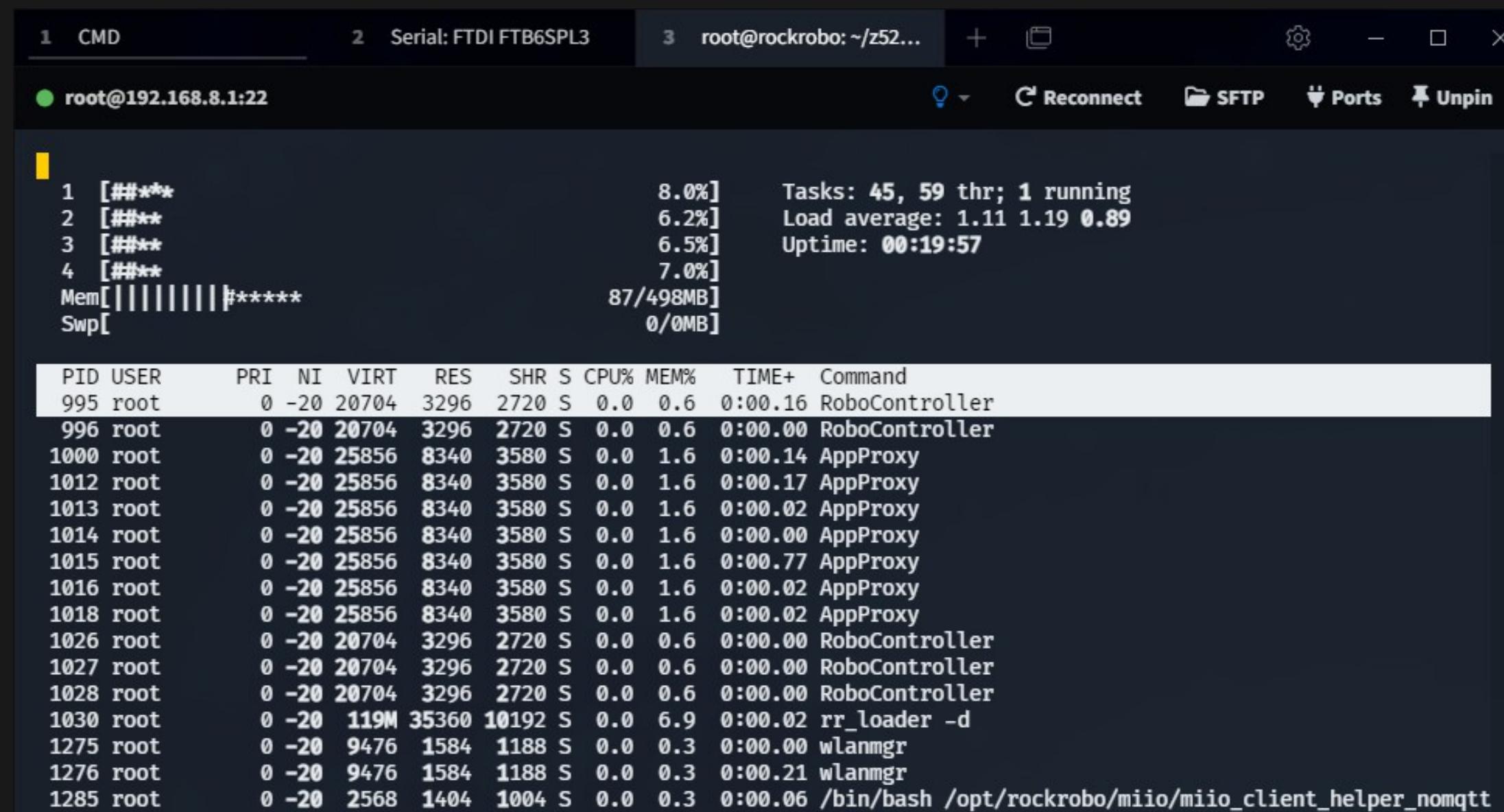
Thesis B | Results

Thesis B - Inspection of system (privileged processes)

Fingerprinting

Processes

Everything is running as root



The screenshot shows a terminal window with the following details:

- Tab 1: CMD
- Tab 2: Serial: FTDI FTB6SPL3
- Tab 3: root@rockrobo: ~/z52... (selected)
- Toolbar: Reconnect, SFTP, Ports, Unpin

System statistics:

```
1 [###**          8.0%] Tasks: 45, 59 thr; 1 running
2 [###**          6.2%] Load average: 1.11 1.19 0.89
3 [###**          6.5%] Uptime: 00:19:57
4 [###**          7.0%]
Mem[|||||] 87/498MB
Swp[          0/0MB]
```

Process list:

PID	USER	PRI	NI	VIRT	RES	SHR	S	CPU%	MEM%	TIME+	Command
995	root	0	-20	20704	3296	2720	S	0.0	0.6	0:00.16	RoboController
996	root	0	-20	20704	3296	2720	S	0.0	0.6	0:00.00	RoboController
1000	root	0	-20	25856	8340	3580	S	0.0	1.6	0:00.14	AppProxy
1012	root	0	-20	25856	8340	3580	S	0.0	1.6	0:00.17	AppProxy
1013	root	0	-20	25856	8340	3580	S	0.0	1.6	0:00.02	AppProxy
1014	root	0	-20	25856	8340	3580	S	0.0	1.6	0:00.00	AppProxy
1015	root	0	-20	25856	8340	3580	S	0.0	1.6	0:00.77	AppProxy
1016	root	0	-20	25856	8340	3580	S	0.0	1.6	0:00.02	AppProxy
1018	root	0	-20	25856	8340	3580	S	0.0	1.6	0:00.02	AppProxy
1026	root	0	-20	20704	3296	2720	S	0.0	0.6	0:00.00	RoboController
1027	root	0	-20	20704	3296	2720	S	0.0	0.6	0:00.00	RoboController
1028	root	0	-20	20704	3296	2720	S	0.0	0.6	0:00.00	RoboController
1030	root	0	-20	119M	35360	10192	S	0.0	6.9	0:00.02	rr_loader -d
1275	root	0	-20	9476	1584	1188	S	0.0	0.3	0:00.00	wlanmgr
1276	root	0	-20	9476	1584	1188	S	0.0	0.3	0:00.21	wlanmgr
1285	root	0	-20	2568	1404	1004	S	0.0	0.3	0:00.06	/bin/bash /opt/rockrobo/mio/mio_client_helper_nomqtt

Thesis B - Recovery partition manipulation (see [proof of concept](#))

Issues, thoughts & discussions

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

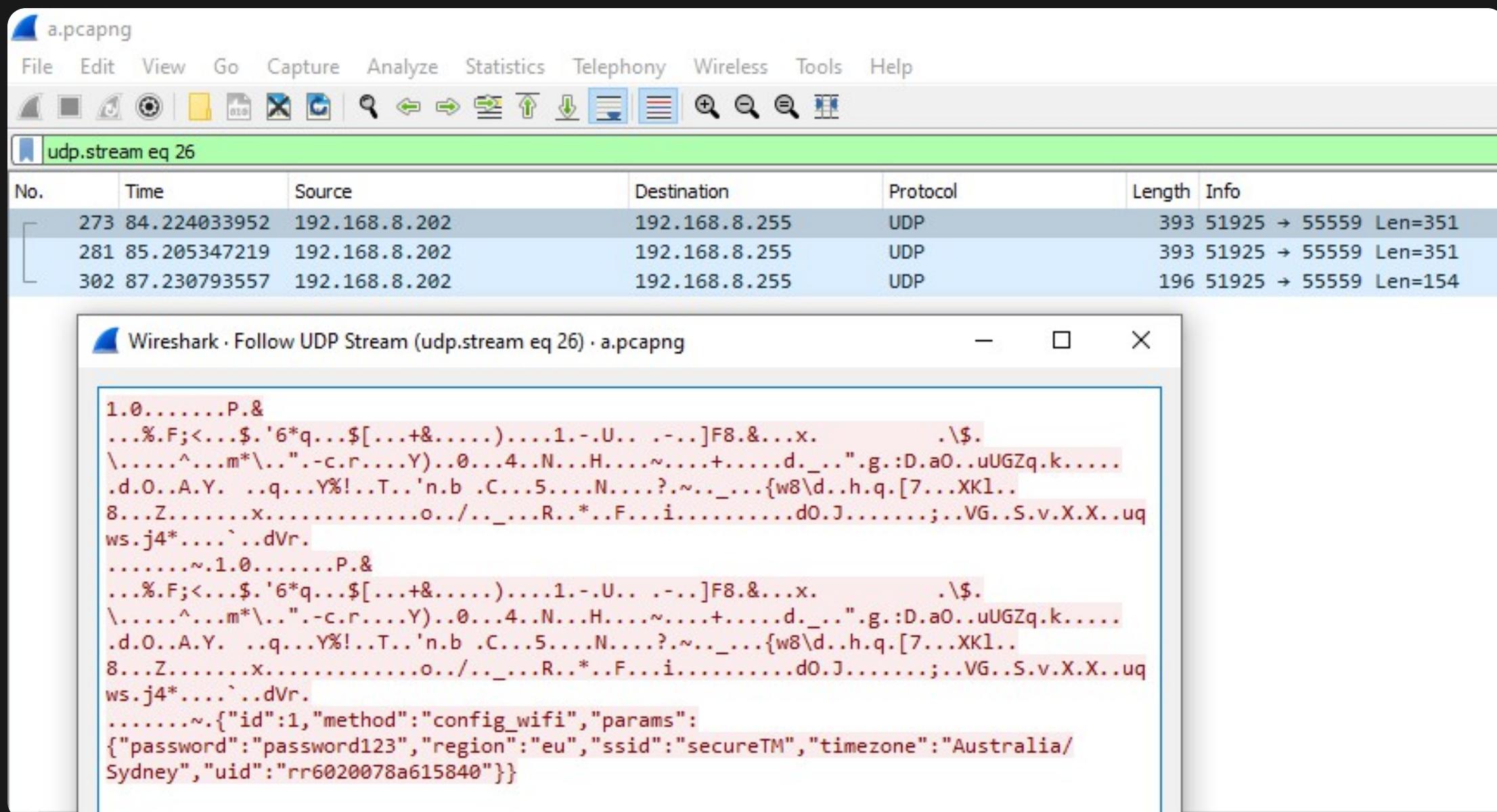
Recovery partition is modifiable

- 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

Thesis B - Capture of device traffic (port-mirroring)

Speaking of packets...

WiFi credentials in plain text during setup



Thesis B - Inspection of system services (netstat, iptables, ip6tables)

Fingerprinting

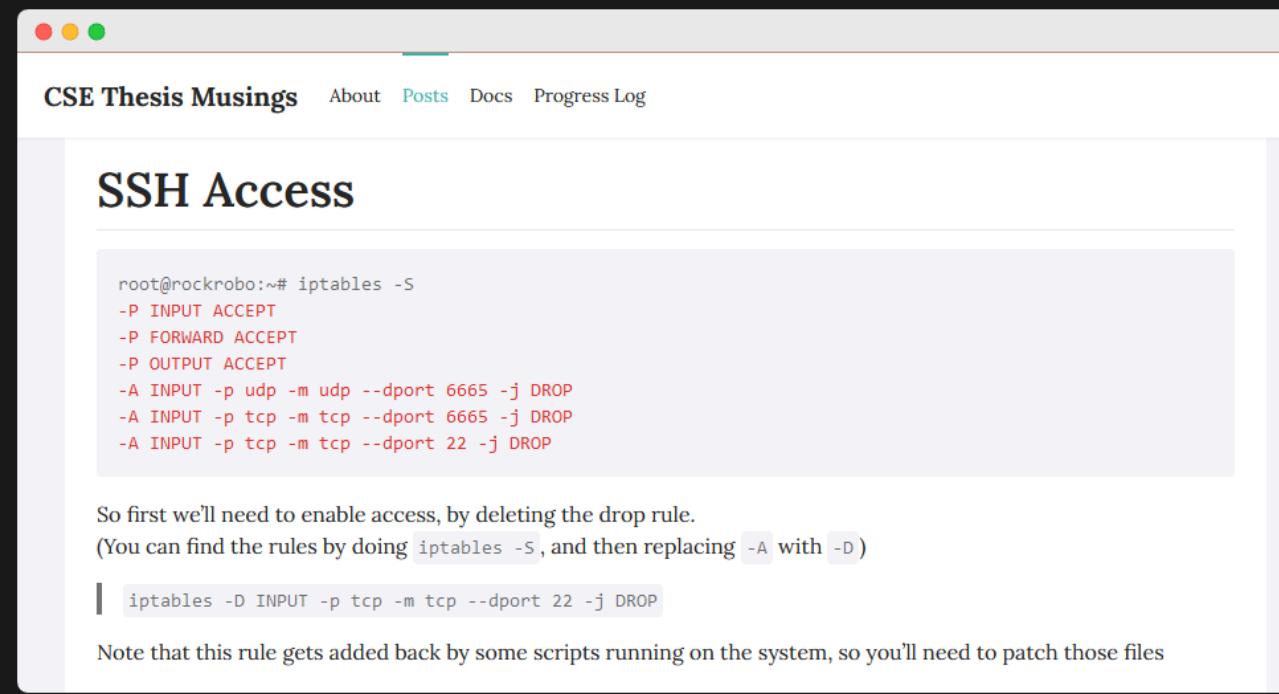
Ports

```
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

Thesis B - Remote access persistence (see proof of concept)

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:

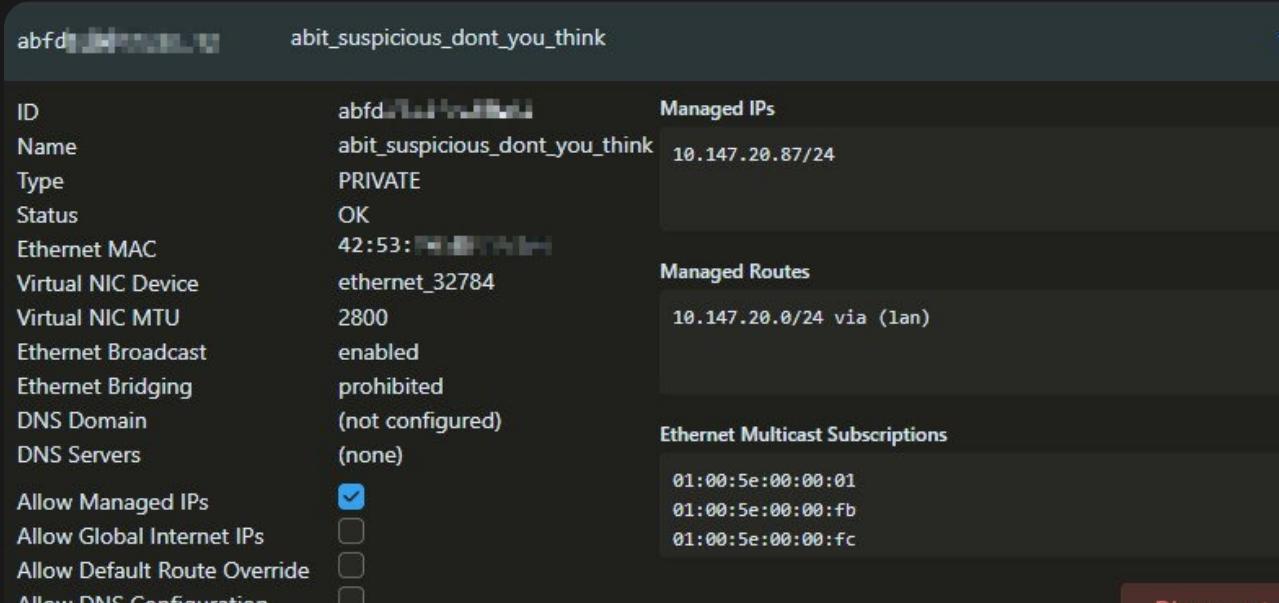
```
root@rockrobo:~# iptables -S
-P INPUT ACCEPT
-P FORWARD ACCEPT
-P OUTPUT ACCEPT
-A INPUT -p udp -m udp --dport 6665 -j DROP
-A INPUT -p tcp -m tcp --dport 6665 -j DROP
-A INPUT -p tcp -m tcp --dport 22 -j DROP
```

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



The screenshot shows the ZeroTier interface for a device named "abit_suspicious_dont_you_think". The device has the following configuration:

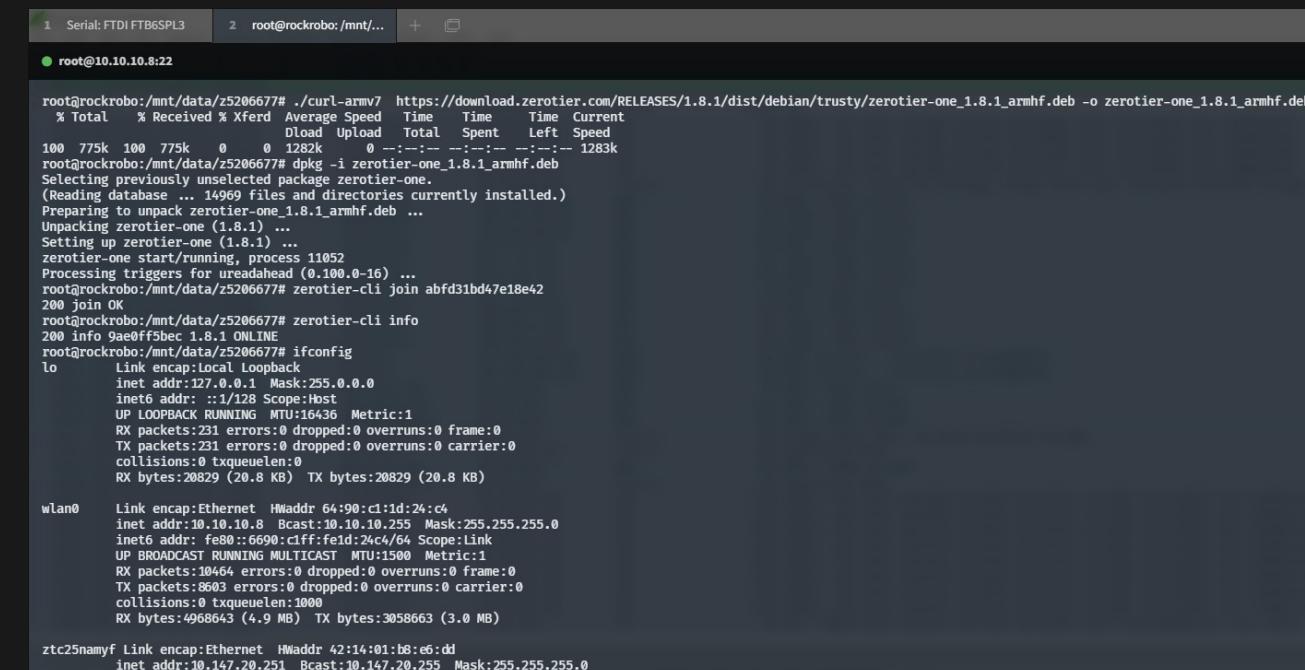
- ID: abfd
- Name: abit_suspicious_dont_you_think
- Type: PRIVATE
- Status: OK
- Ethernet MAC: 42:53:... (redacted)
- Virtual NIC Device: ethernet_32784
- Virtual NIC MTU: 2800
- Ethernet Broadcast: enabled
- Ethernet Bridging: prohibited
- DNS Domain: (not configured)
- DNS Servers: (none)
- Allow Managed IPs: checked
- Allow Global Internet IPs: unchecked
- Allow Default Route Override: unchecked

Managed IP: 10.147.20.87/24

Managed Routes: 10.147.20.0/24 via (lan)

Ethernet Multicast Subscriptions: 01:00:5e:00:00:01, 01:00:5e:00:00:fb, 01:00:5e:00:00:fc

- 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



The terminal session shows the following commands and output:

```
1 Serial:FTDI FTB65PL3 2 root@rockrobo:/mnt/...
root@10.10.10.8:22
root@rockrobo:/mnt/data/z5206677# curl-armv7 https://download.zerotier.com/RELEASES/1.8.1/dist/debian/trusty/zerotier-one_1.8.1_armhf.deb -o zerotier-one_1.8.1_armhf.deb
% Total % Received % Xferd Average Speed Time Time Current
100 775k 100 775k 0 0 1282k 0 --:-- --:-- --:-- 1283k
root@rockrobo:/mnt/data/z5206677# dpkg -i zerotier-one_1.8.1_armhf.deb
Selecting previously unselected package zerotier-one.
(Reading database ... 14969 files and directories currently installed.)
Preparing to unpack zerotier-one_1.8.1_armhf.deb ...
Unpacking zerotier-one (1.8.1) ...
Setting up zerotier-one (1.8.1) ...
zerotier-one start/running, process 11052
Processing triggers for ureadahead (0.100.0-16) ...
root@rockrobo:/mnt/data/z5206677# zerotier-cli join abfd31bd47e18e42
200 join OK
root@rockrobo:/mnt/data/z5206677# zerotier-cli info
200 info Qae0ff5bec 1.8.1 ONLINE
root@rockrobo:/mnt/data/z5206677# ifconfig
lo Link encap:Local Loopback
    inet addr:127.0.0.1 Mask:255.0.0.0
    inet6 addr: ::/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:231 errors:0 dropped:0 overruns:0 frame:0
        TX packets:231 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:20829 (20.8 kB) TX bytes:20829 (20.8 kB)
wlan0 Link encap:Ethernet HWaddr 64:90:c1:1d:24:c6
    inet addr:10.10.10.8 Bcast:10.10.10.255 Mask:255.255.255.0
    inet6 addr: fe80::6690:c1ff:fe1d:24c6/64 Scope:Link
        UP BROADCAST RUNNING MTU:1500 Metric:1
        RX packets:10464 errors:0 dropped:0 overruns:0 frame:0
        TX packets:8603 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:4968643 (4.9 MB) TX bytes:3058663 (3.0 MB)
ztc25namyf Link encap:Ethernet HWaddr 42:14:01:01:08:e6:0d
    inet addr:10.147.20.251 Bcast:10.147.20.255 Mask:255.255.255.0
```

Thesis B | Observations

Thesis B - Investigating tcpdump

(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?

Thesis B - Investigating rrlogd

(some) Interesting Files

mmcb1k0p8/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?

The image shows two side-by-side debugger windows from Immunity Debugger, both titled "rrlogd (ELF Graph)".

Left Window: Shows the main function entry point. The assembly code includes instructions like movw r0, #0x8c8c, movt r0, #2 {data_28c8c, "/dev/shm/rrlogd.pid"}, and b1 r0, #0x1533e. Below the assembly is a detailed control flow graph (CFG) showing the flow between various functions and memory locations.

Right Window: Shows a specific function: sub_1b2d4(int32_t arg1, int32_t arg2, int32_t arg3). The assembly code includes fopen(arg1, "r"), fgets(var_41c, 0x400, r0), and if (r0 == 0) {model}. Below the assembly is another CFG showing the execution flow through various branches and conditions.

Both windows provide a comprehensive view of the program's internal logic and data structures, highlighting the complexity of the logging and configuration handling.

(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

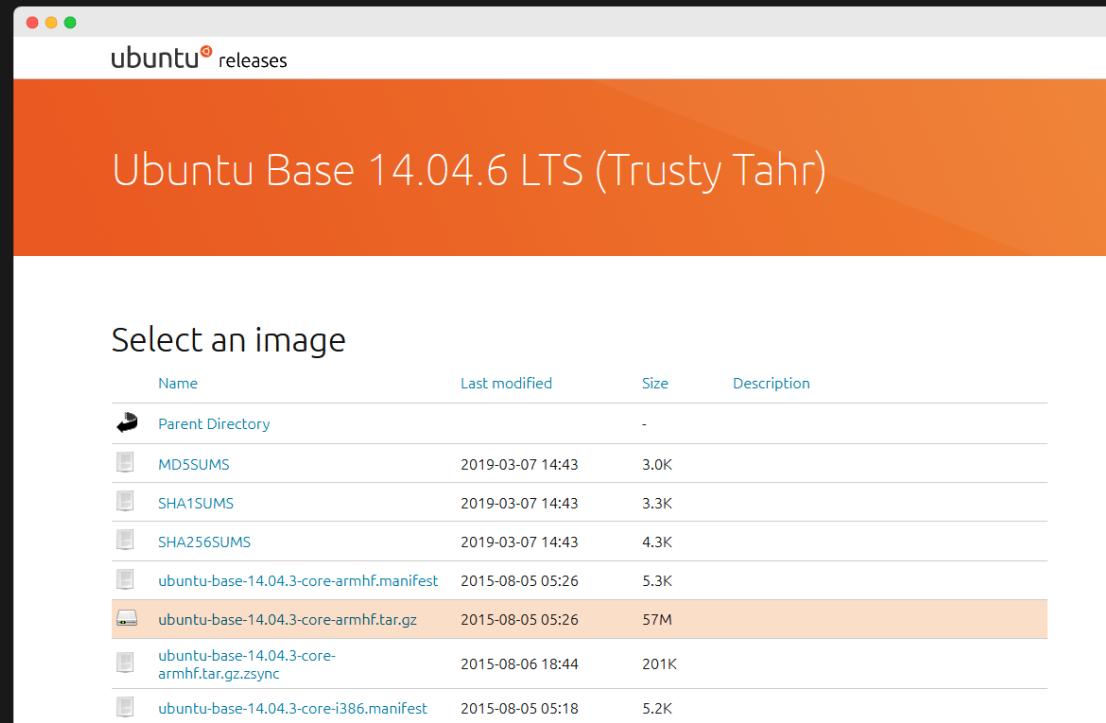
- mmcblk0p6/vinda
- mmcblk0p6/adb.conf
- mmcblk0p8/var/log/upstart/adbd.log

Thesis B | Future Challenges

Thesis B - Comparing files

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

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

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

- WAN - security, PII
- LAN - traffic? (tcpdump)
- Firmware upgrade
- Update to a newer firmware version and look at changes
- Check what files gets cleared during a format
- Binary assessment
- Verify IPv6 SSH access

The IPv6 World

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

Let's Talk Threats

Talk about threat models

What files?

- compare against stock
- lots of hash matches

threat models

work done in thesis c

hooking into code..

Network analysis

Address the statement

Thesis in a Year

The screenshot shows a digital dashboard with a teal header bar containing the title "CSE Thesis Devlog TL;DR". Below the header, there are three columns of tasks:

- Research** (left column):
 - Wednesday 29/09/2021
 - Tuesday 5/10/2021
 - Tuesday 12/10/2021
 - Monday 18/10/2021
 - Monday 1/11/2021
 - Wednesday
- Hardware Hacking** (middle column):
 - Monday 25/10/2021
 - Tuesday 26/10/2021
 - Friday 29/10/2021
 - Sunday 01/05/2022
 - Friday 24/06/2022
- Software Hacking** (right column):
 - Monday 25/10/2021
 - Friday 29/10/2021
 - Saturday 30/10/2021
 - Wednesday 02/03/2022
 - Filesystem inspection 19/03/2022
 - Install software

Thank You

Andrew Wong

w: featherbear.cc/UNSW-CSE-Thesis

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