Reversing Internet of Things from Mobile applications

Axelle Apvrille - FortiGuard Labs, Fortinet

Insomni’hack, Geneva, March 18, 2016
Reversing Internet of Things (IoT) is difficult

Different hardware

Different OS
Linux, Windows Mobile, Android, Contiki, RIOT, TinyOS, Brillo...

Different formats
ELF, BFLT...

Why reverse IoT?
- To understand how (in)secure they are
- To detect and protect against viruses and exploits
So, how do we get started?

Focus first on the mobile app

Apktool, dex2jar, IDA Pro...

It’s faster

First step
Real examples
Real examples
Real examples
A shell on the glasses

- Enable USB debugging on the glasses
- Add udev rule
- Add vendor in
  
  /\.android/adb_usb.ini

$ adb devices
List of devices attached
291052171 device
$ adb -s 291052171 shell
shell@android:/$
System properties

```
shell@android:/ $ getprop ro.boot.bootloader
U-Boot_1.1.4-4.4-SUN^0-dirty
shell@android:/ $ getprop ro.build.description
lean_jet_sun-user 4.1.2 JZ054K 11 release-keys
```

The glasses are using Android 4.1.2 - Jelly Bean
Hey, what hardware is it using?

/system/board properties/soc/revision: OMAP4430

/system/lib/hw/sensors.conf:
- STM LSM9DS0 accelerometer/gyroscope/compass
- STM LPS25 pressure
- TI TMP103 temperature
- Recon Free Fall
- Avago Tech APDS9900 ambient light
System applications

```
shell@android:/system/app $ ls
...
ReconCamera.apk
ReconCompass.apk
ReconItemHost.apk
...
```

Pull them, analyze them
Apktool, dex2jar, JEB, baksmali...

```java
zipcreated:
    ArrayList list = new ArrayList();
    File logfile = new File(this.mContext.getFilesDir() + "//logcatout.txt");
    try {
        Runtime.getRuntime().exec("logcat -d -v threadtime -f " + logfile.getAbsolutePath());
        if (!logfile.exists()) {
            goto label_82;
        }
    } catch (IOException e) {
        ....
    }
    list.add(logfile);
```
Data leak

Recon Jet glasses

Events :
  Start/
  Pause ...
  Battery level
  +
  Logcat, dmesg

ZIP

Hard-coded password

HTTPS

Sent during synchronization

Recon Instruments servers

Insomni'Hack 2016 - A. Apvrille
Example of data

```
{
    "component": "battery_monitor",
    "data1": "99%; 4172mV",
    "data2": "Charging USB",
    "data3": "29",
    "event_type": "BatteryMeasurement",
    "time_stamp": "1434115258015"
}
```

```
{
    "component": "ActivityManager",
    "data1": "com.reconinstruments.jetconnectdevice/.ReconnectSmartphoneActivity",
    "data2": "",
    "data3": "",
    "event_type": "PauseActivity",
    "time_stamp": "1434115211239"
}
```
Vulnerability status

- Vulnerability found
- Vendor contacted
- Issue fixed in Recon OS 4.4 (February 2016)
Why are we investigating toothbrushes?!

Attackers don't care about your teeth, but ...

**TARGETED BUSINESS**
Profile user & family
Sell health plans, hi-tech

**UNDESERVED REWARDS**
Free toothpaste not attractive to attackers
Insurance fraud might become an incentive

**RANSOMWARE**
“I’ll tell your mom you don’t brush your teeth!”
Ransom kids pocket money

**INFECTION VECTOR**
Your toothbrush infects other devices

PRIVACY ISSUES?
WATCH THIS IN THE FUTURE?
LOW REVENUE
WATCH THIS IN THE FUTURE?
Classes and fields: we work out the mappings
So, what?

- Firmware
- Hardware
- Flash
- Battery: level
- Motor: control speed
- Gyroscope: 3 axis
- Accelerometer: 3 axis
- Auto off timer
- Bluetooth: MAC addr
- Serial no

*beamtoothbrush.com/
Dentist map
Database
Discount
Database
User
Database
SYNC HTTPS
BLE

User name, Birthdate, Email, Zip Stars...
Now you’re ready for wiser investigations...

Now, it’s going to be easier & faster to continue the reverse engineering.

Talk to your toothbrush?

Send BLE ATT packets to service/characteristics

- **Firmware OTA** service C05FC343-C076-...
- Beam service 04234F8E-75...
- **Battery level** 6DAC0185-E...
- **Accelerometer** 0227F1B0-FF...
- Auto off and quadrant buzz 19DC94FA-7B...
- ...

Insomni’Hack 2016 - A. Apvrille
There’s an Android app for the alarm

- Protect your house against burglars
- Controllable by SMS

But it’s not very user friendly...

Comply to a strict SMS formatting

💡 So, they created an Android app to assist end-users
In the **outbox**, the SMS contains the **password** and **phone number** of the alarm.

*You get it? You control the alarm!*

Fake data, of course :D

Let's suppose you are a **wise person** and erase the **SMS**

*You are wise, aren't you?*
With the Android app, it’s worse!

Weak protection for password: we can recover alarm’s phone number, password, delay, emergency phone...

Your credentials are at risk even if you erased the SMS!

Without the app, 1 security issue.
With the app, 2 security issues !!!
How to reverse Internet of Things

1. Get the **mobile application**, reverse it
2. Then, use what you have learned to go deeper down and e.g. inspect hardware, protocols etc.
Recap’ (2/2)

- One vulnerability found and fixed
- We know what hardware is used

- We know how to communicate with the toothbrush!
- We know where stars and challenges are handled

- One vulnerability found, advisory published
- Don’t use the app!
Thanks for your attention!

@cryptax or aapvrille (at) fortinet (dot) com
http://www.fortiguard.com
http://blog.fortinet.com

Awesome slides? Thanks! That’s LATEX