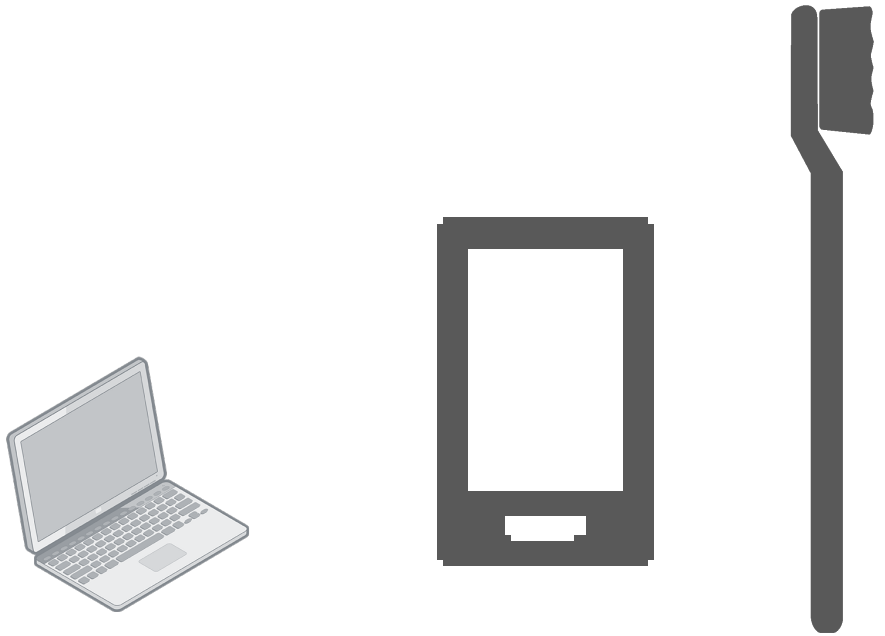
Reversing Internet of Things from Mobile applications

Axelle Apvrille - FortiGuard Labs, Fortinet

AREA 41, Zurich, June 10, 2016

Why reverse IoT?



1.5B 1.8B 2B

Number of units in 2014

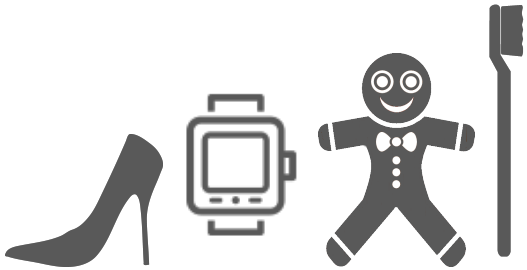
- To understand how (in)secure they are
- To protect ourselves against viruses, exploits, privacy leaks
- More features & interactivity

Hot prediction

Yes, there will be (many) viruses on IoT (one day)

Reversing Internet of Things (IoT) is difficult

Different hardware



Different OS

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

Different formats

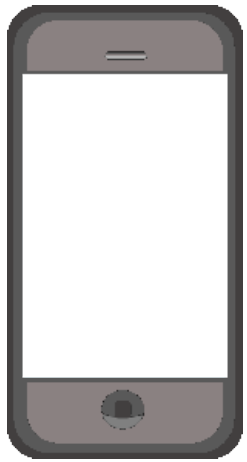
ELF, BFLT...

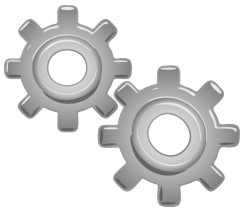
Research

e.g firmware.re

So, how do we get started?

Focus first on the mobile app





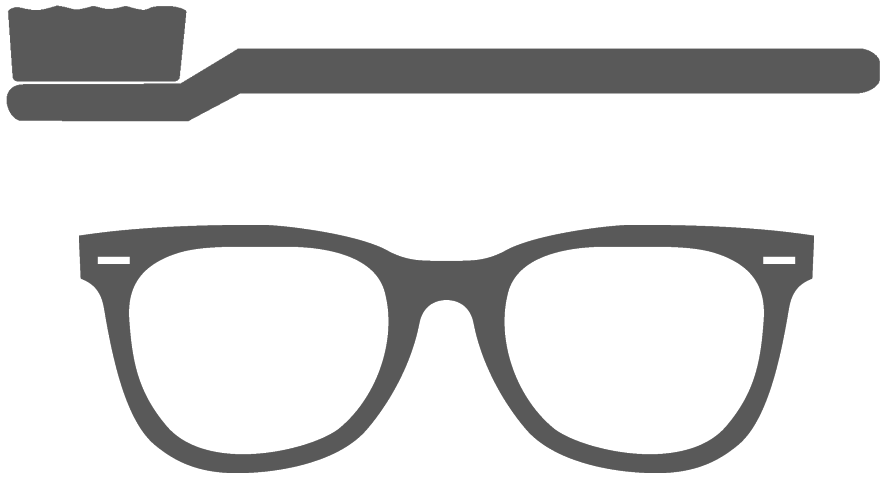
Apktool, dex2jar, IDA Pro...

It's faster

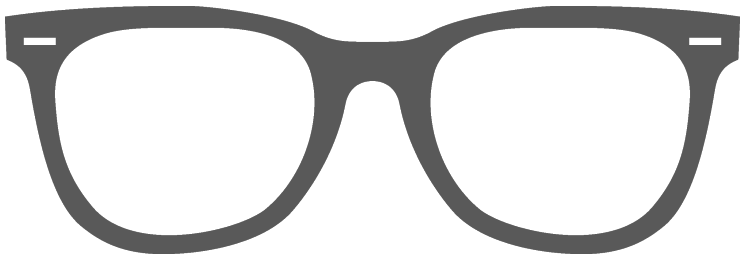
First step

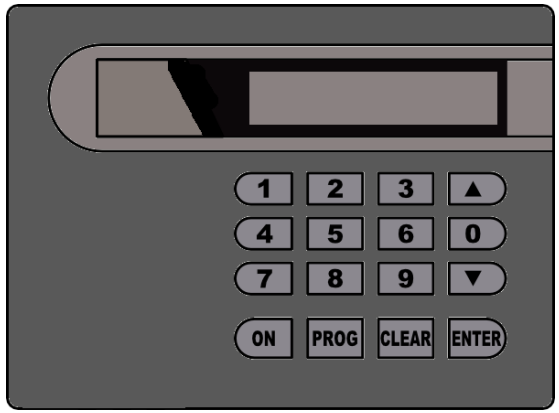
Real examples

Real examples



Real examples



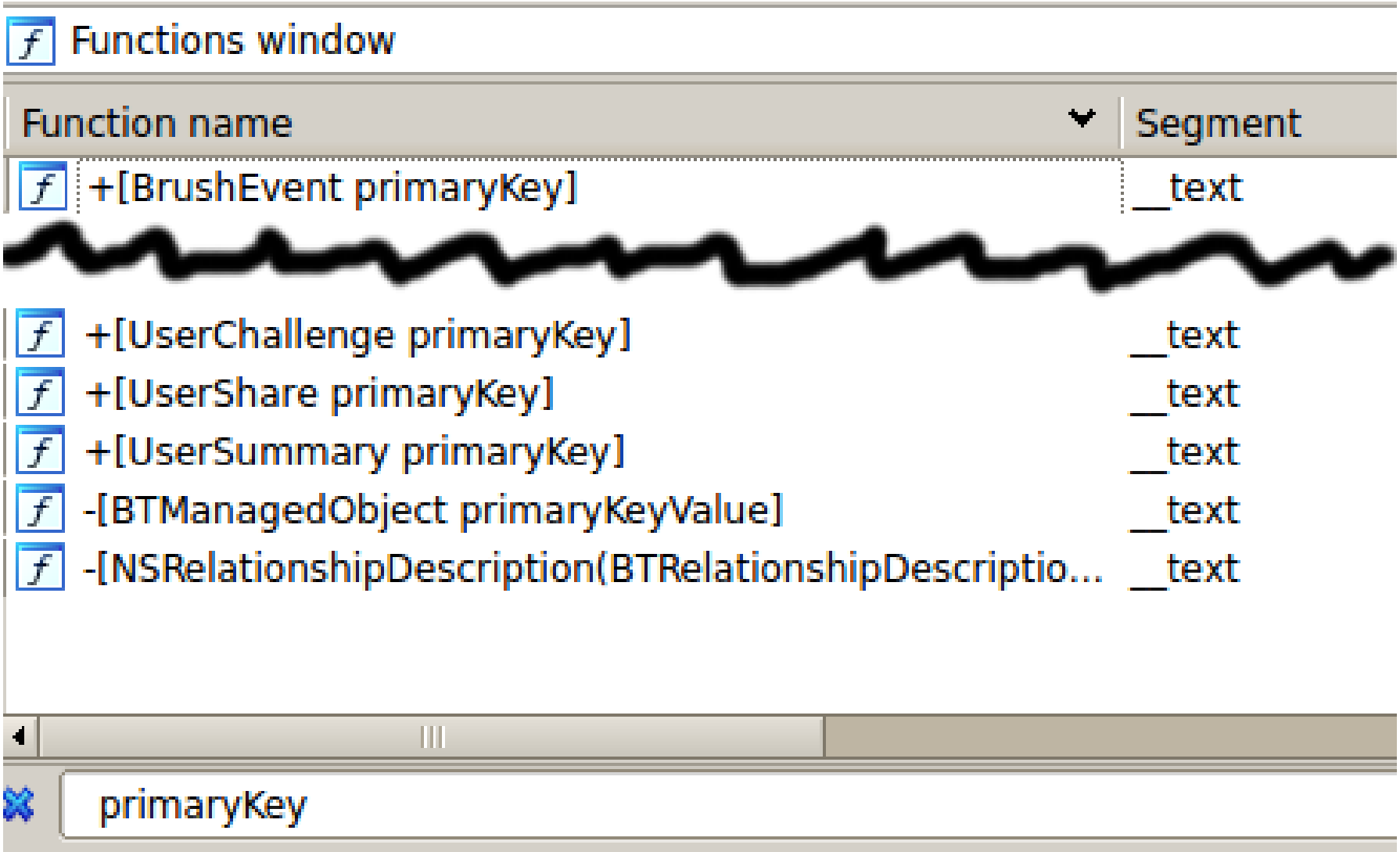


Beam Toothbrush - Smart Glasses - Safety Alarm

Where to start?

Let's reverse the Android and/or iOS applications

List SQL tables



Getting table columns

```
BLX     _objc_retainAutoreleasedReturnValue
MOV     R1, #(cfstr_Firstname - 0x15362) ; "firstName"
ADD     R1, PC  ; "firstName"
MOV     R2, #(cfstr__ - 0x1536C) ; "."
ADD     R2, PC  ; "."
MOV     R3, #(_objc_msgSend_ptr_0 - 0x15376)
ADD     R3, PC ; _objc_msgSend_ptr_0
LDR     R3, [R3] ; __imp__objc_msgSend
MOV     R9, #(selRef_componentsSeparatedByString_ - 0x15382)
ADD     R9, PC ; selRef_componentsSeparatedByString_
MOV     R12, #(cfstr_First_name - 0x1538C) ; "first_name"
ADD     R12, PC ; "first_name"
STR     R0, [SP,#0x130+var_30]
STR.W   R12, [SP,#0x130+var_58]
```

Results: What You Learn

Insured

- insuredID: primary key
- title
- first_name
- middle_initial
- last_name
- post_name
- relation_to_-policy_holder
- gender
- ...

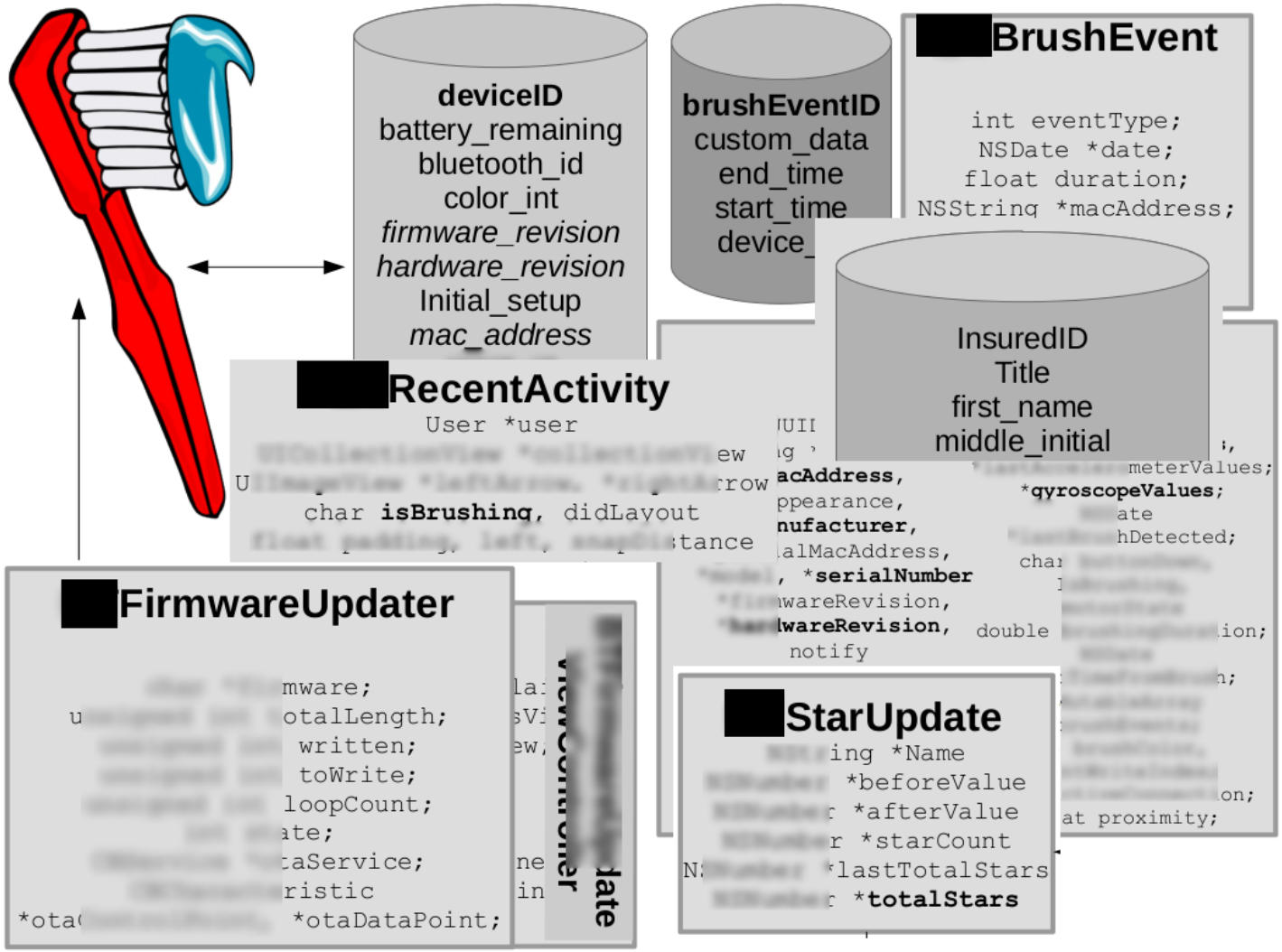
Device

- deviceID: primary key
- battery_remaining
- bluetooth_id
- color_int
- firmware_revision
- hardware_revision
- initial_setup
- mac_address
- ...

Getting methods of classes

- Methods: _OBJC_INSTANCE_METHODS
- Instance variables: _OBJC_INSTANCE_METHODS - contains name and type of variables

Results: What You Learn



BLE characteristics

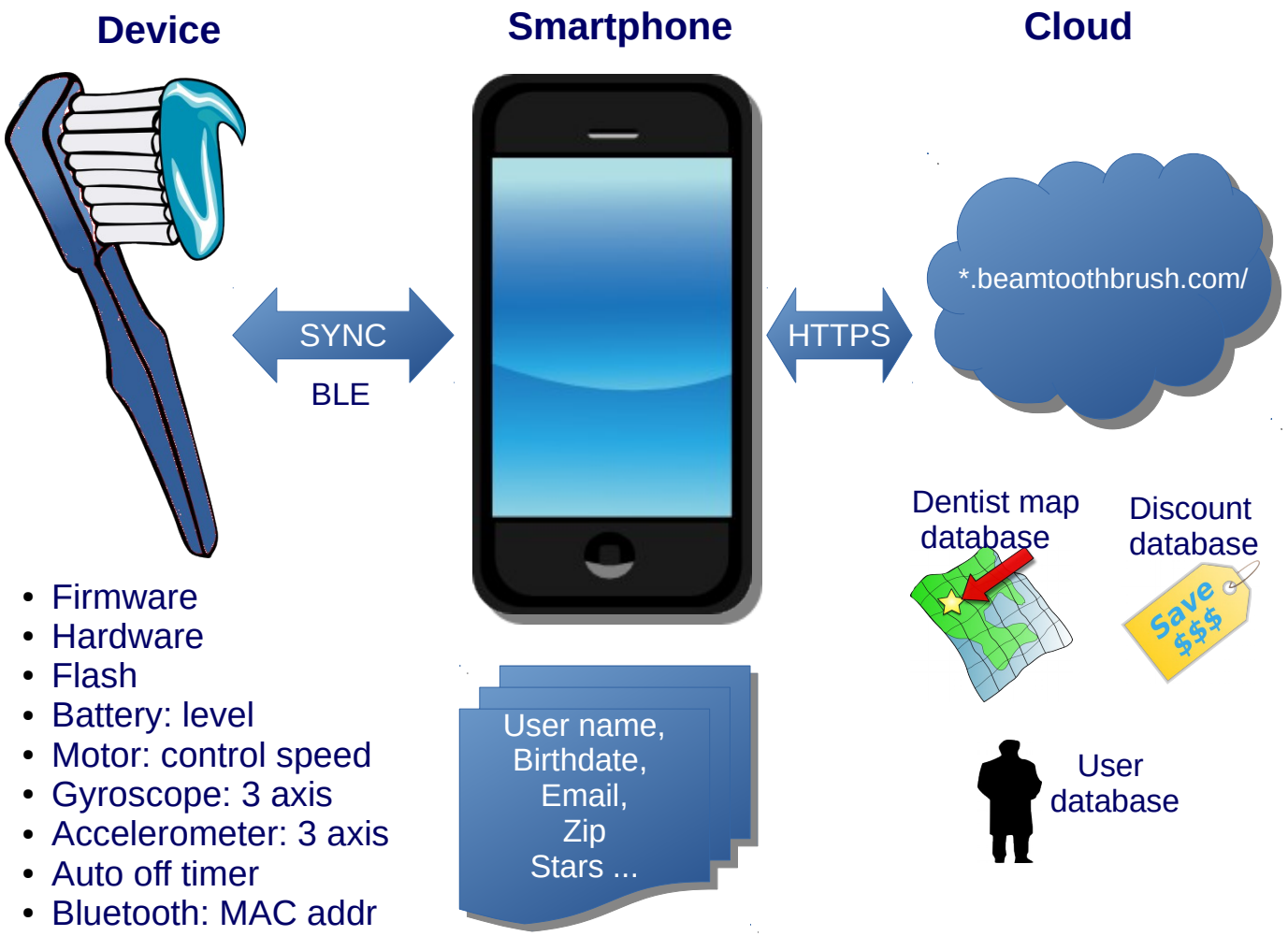
```
public void writeQuadrantBuzz(BLEDevice device, boolean arg6, boolean arg7) {
    BluetoothGatt gatt = this.getBluetoothGatt(device);
    if(gatt != null) {
        this.send2charac(gatt, "04[redacted]", "19DC94FA-7BB3-4248-[redacted]
                ByteSerialize.boolean2byte(arg6, arg7));
    }
}

public void setMotorSpeed(BLEDevice arg5, float arg6) {
    BluetoothGatt v0 = this.getBluetoothGatt(arg5);
    if(v0 != null) {
        this.send2charac(v0, "04[redacted]", "833DA694-51C5-4418[redacted]",
                ByteSerialize.float2byte(arg6));
    }
}
```

Results: What You Learn

UUID	Description
a8902afd-4937-4346-...	Boolean indicator for active brushing
267b09fd-fb8e-4bb9-...	Motor state
3530b2ca-94f8-4a1d-...	Current time
833da694-51c5-4418-...	Motor speed
19dc94fa-7bb3-4248-...	Auto-off and quadrant buzz indicators (2 bits)
6dac0185-e4b7-4afd-...	Battery level (2 bytes)
0971ed14-e929-49f9-...	Brush color (1 byte)
0227f1b0-ff5f-40e3-...	Accelerometer data (6 bytes)
ed1aa0cf-c85f-4262-...	Gyroscope (6 bytes)
cf0848aa-ccdb-41bf-...	Button state

So, what?



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

Why are we investigating toothbrushes?!

Come on!

Attackers don't care about our teeth!!!

Why are we investigating toothbrushes?!

Come on!

Attackers don't care about our teeth!!!

True

Why are we investigating toothbrushes?!

Come on!

Attackers don't care about our teeth!!!

True

but you are missing the point!

Attackers care about **money**

Toothbrush attack scenario 1/4: Targeted business

You own a connected toothbrush?

Attacker knows:

- **Name** and **ages** of member of your **family**
- Likely to be **wealthy high tech user**
- You value your **health**

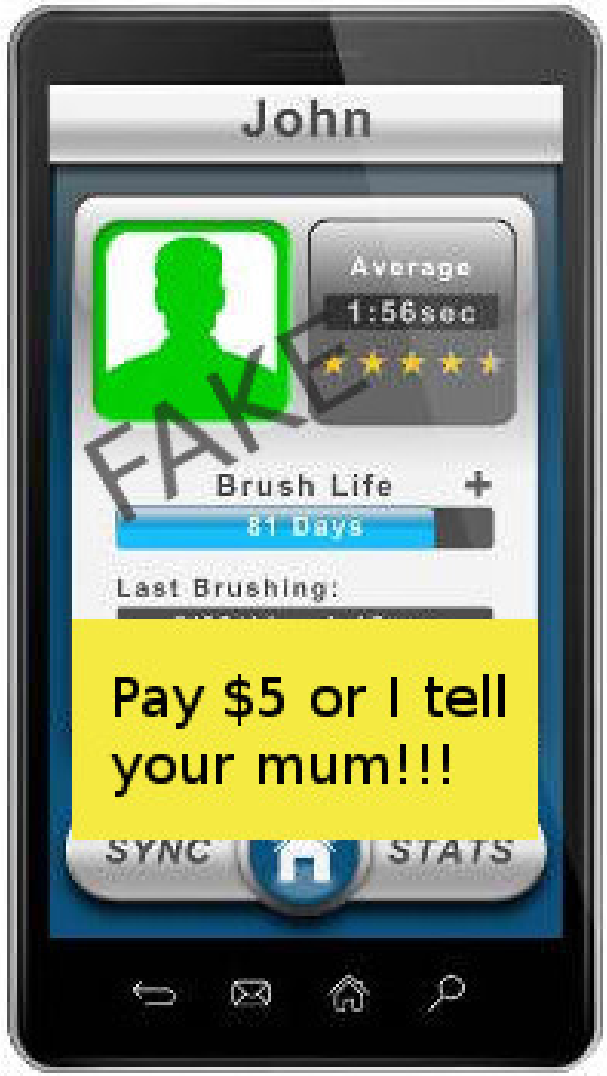
$\rightarrow$ Sell health plans + high-tech ads

Known business, more or less legal, privacy issues

Toothbrush Attack Scenario 2/4: Ransomware

Ransom kids pocket money
See also Candid Wüest, *Is Ransomware coming to IoT devices?*

Efficient but low revenue



Toothbrush Attack Scenario 3/4: Undeserved rewards

What can the attacker get for forged brushing rewards:

- Free toothpaste ;) *not very attractive*
- Insurance fraud... *more scary*

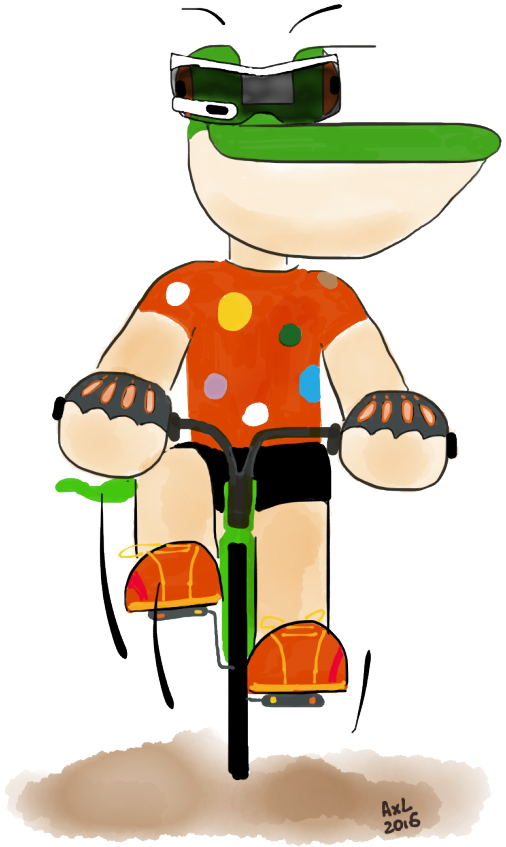
Watch for this in the future

Toothbrush Attack Scenario 4/4: Infection vector

Your toothbrush (or other IoT) infects other devices

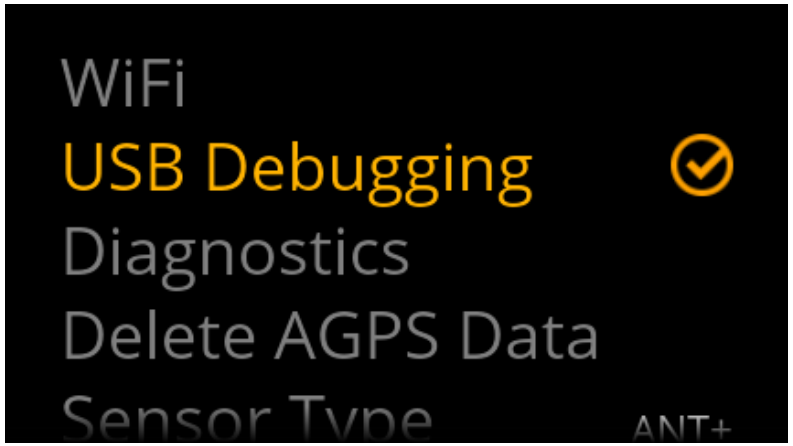
Watch for this in the future

Toothbrush - Recon Jet Smart Glasses - Safety Alarm



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:/ $
```

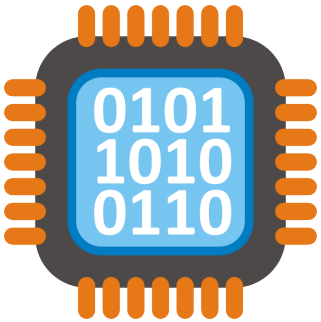
System properties

```
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**

Hardware investigations

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

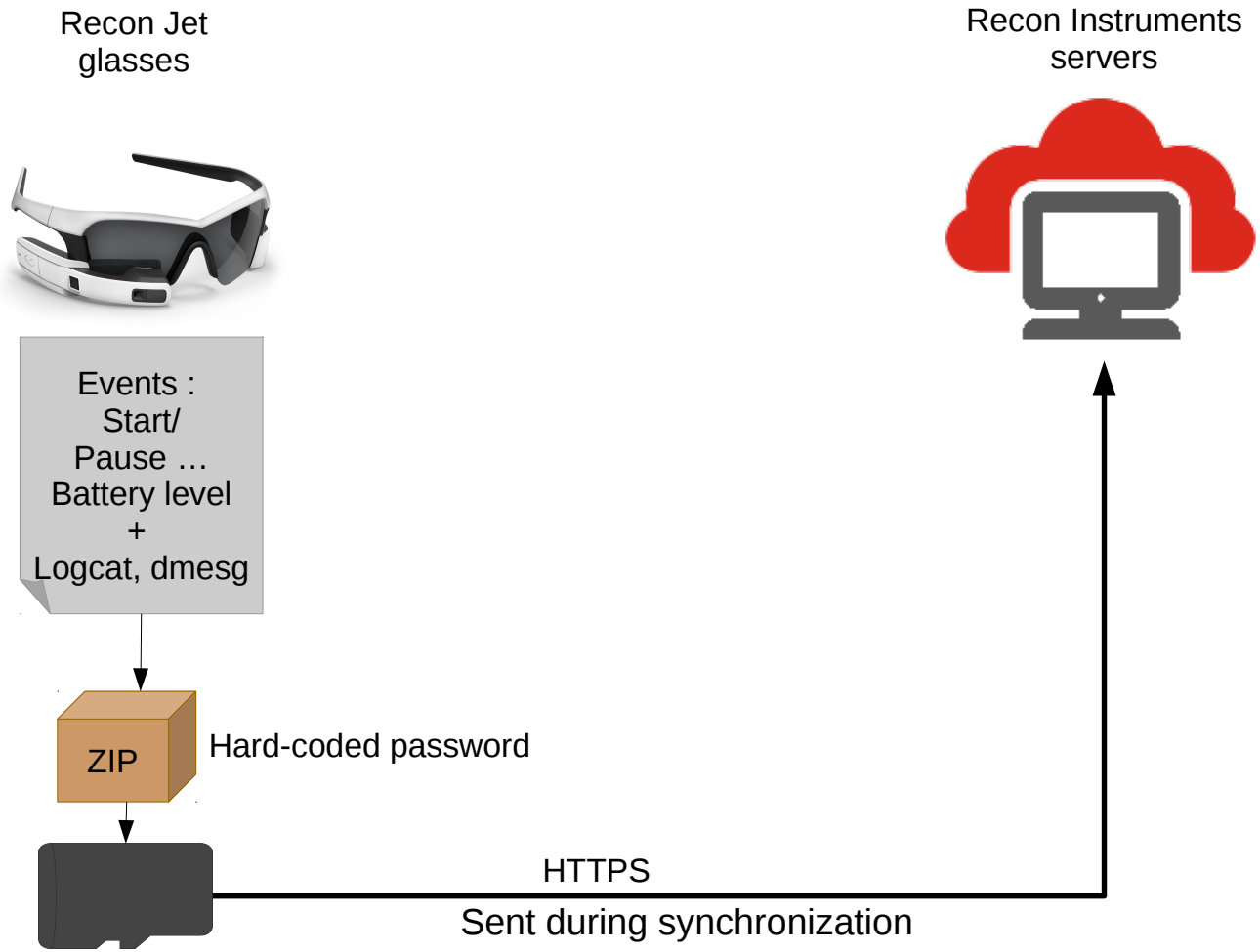
```
shell@android:/system/app $ ls
...
ReconCamera.apk
ReconCompass.apk
ReconItemHost.apk
...
```

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

```
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;
        }

        list.add(logfile);
    }
```

Data leak


Recon Jet glasses
Recon Instruments servers
Events : Start/ Pause … Battery level + Logcat, dmesg
ZIP
Hard-coded password
HTTPS
Sent during synchronization

Example of data

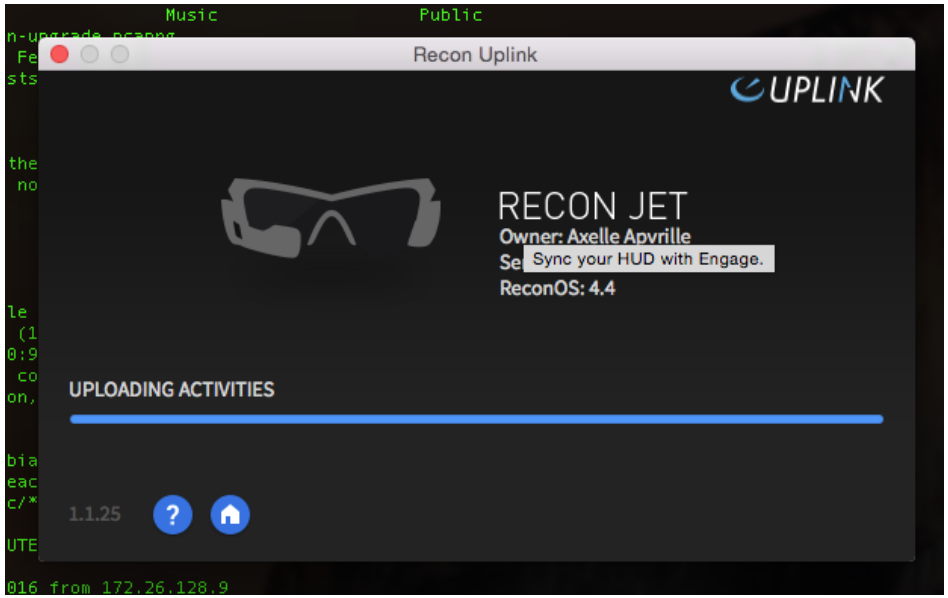

```
 {
            "component": "battery_monitor",
            "data1": "99%; 4172mV",
            "data2": "Charging USB",
            "data3": "29",
            "event_type": "BatteryMeasurement",
            "time_stamp": "1434115258015"
        },
 {
            "component": "ActivityManager",
 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



Toothbrush - Smart Glasses - Meian Home Safety Alarm

There's an Android app for the alarm

- Protect your house against burglars
- Controllable by SMS

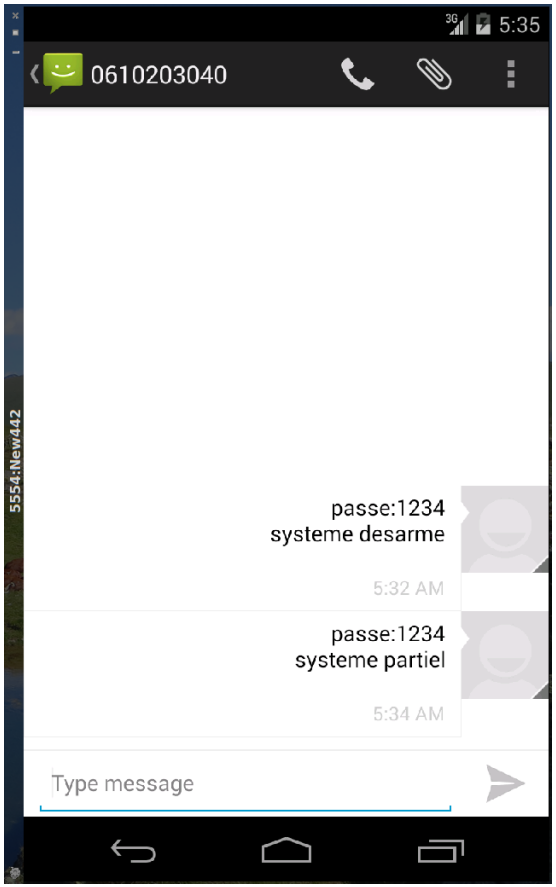
But it's not very user friendly...

Comply to a strict SMS formatting

So, they created an **Android app** to assist end-users

(Known?) Security issue

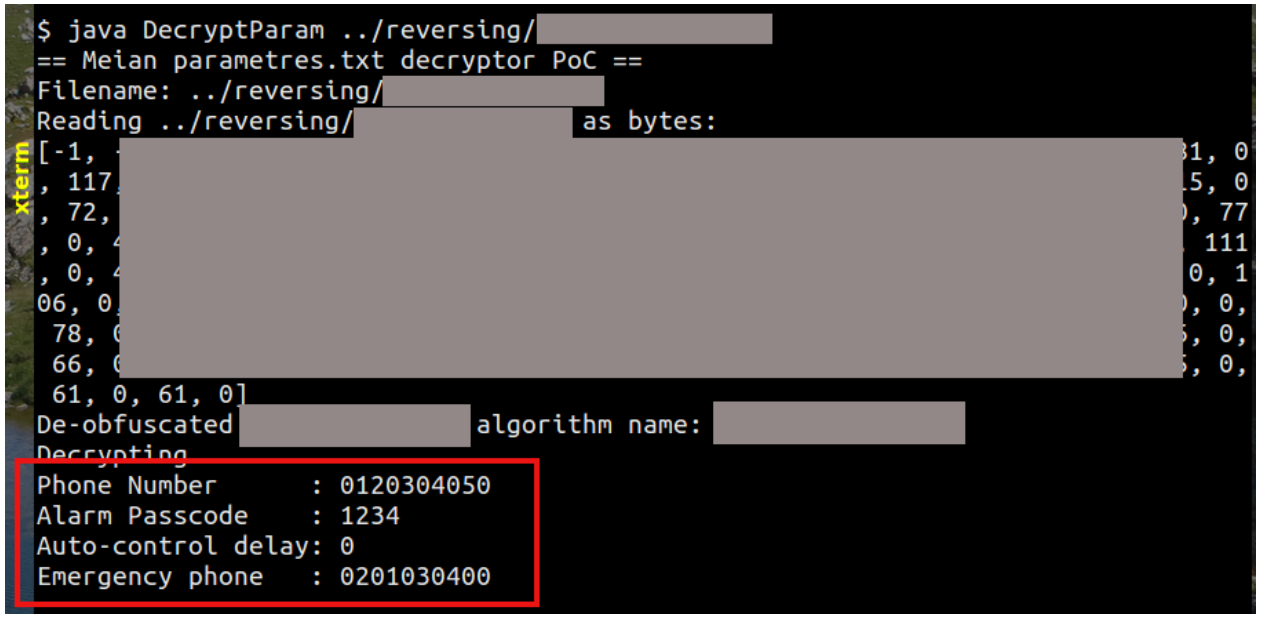
In the **outbox**, the SMS contains the **password** and **phone number** of the alarm.
You get it? You control the alarm!



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** !!!

Recap' (1/2)

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)

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

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

- One vulnerability found, advisory published
- Don't use the app!

I'm an IoT vendor: what can I do?

DO

- Design security of IoT from the beginning
- Review security of mobile apps
- Help security researchers work on your devices
- Consider open sourcing some - or all - of your code

DO NOT

Do not underestimate your device / data

It may not seem interesting to **you** but an attacker has different goals! There are viruses out there

"Okay, now I'll obfuscate my mobile code!"

That's plain **stupid** and inefficient! Security by obscurity has **never** worked

Acknowledgements

Thanks to Beam Technologies for providing a free user account for testing purposes.
Thanks to Recon Instruments for their responsiveness.
That's how security works and improves, folks!

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