SBA Research

COMET

Competence Centers for Excellent Technologies

All Your Fitness Data Belongs to You

Reverse Engineering the Huawei Health Android App

Christian Kudera

Bundesministerium Verkehr, Innovation und Technologie
Bundesministerium Digitalisierung und Wirtschaftsstandort
FFG Forschung wirkt.
wirtschaftsagentur wien Ein Fonds der Stadt Wien
FWF Der Wissenschaftsfonds.
netidee OPEN INNOVATIONS

Who Am I

- Security analyst and researcher at SBA Research
- PhD candidate (TU Wien)
- Mainly interested in IoT, embedded and hardware security

Overview

Motivation

Bluetooth Low Energy (BLE)

Static Program Analysis

Dynamic Program Analysis

The Huawei Link Protocol v2

Conclusion

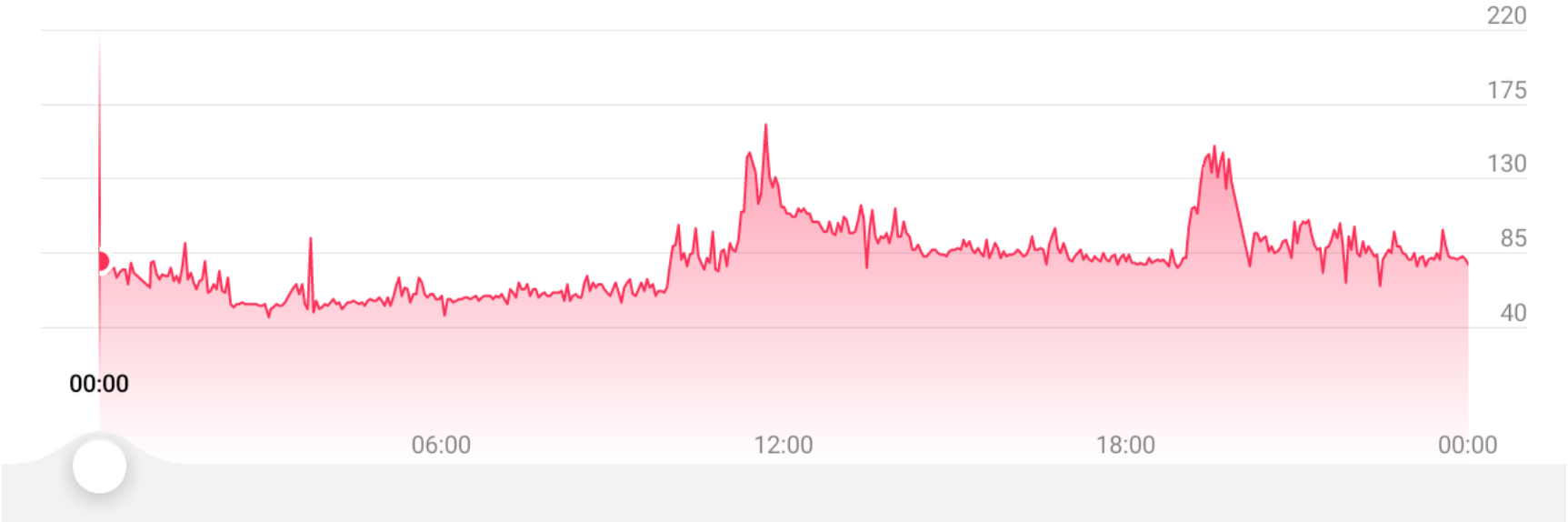
Motivation

Motivation for the Huawei Watch GT

- My reasons for a fitness wearable:
  - Pulse monitoring
  - Tracking of my activities
  - Motivation to do more sport ☺

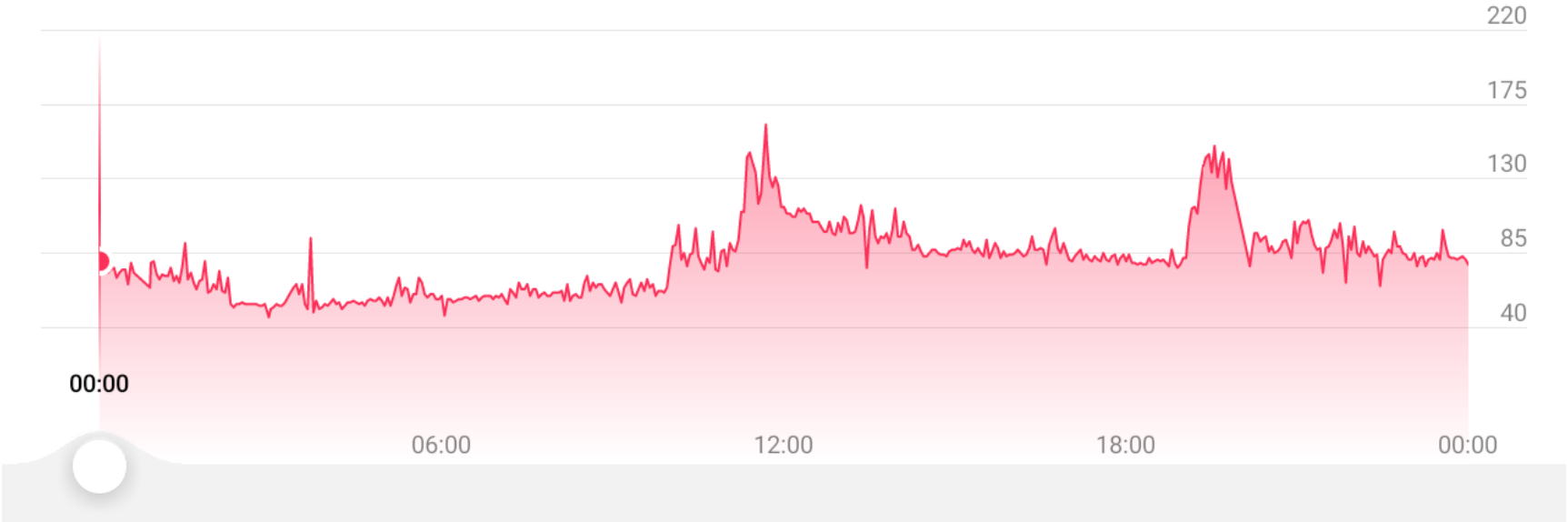
- My reasons for the Huawei Watch GT:
  - Incredible battery life (~14 days)
  - Affordable price (~180€)
  - Reliable hardware (according to reviews)

My Heart Rate Over One Day

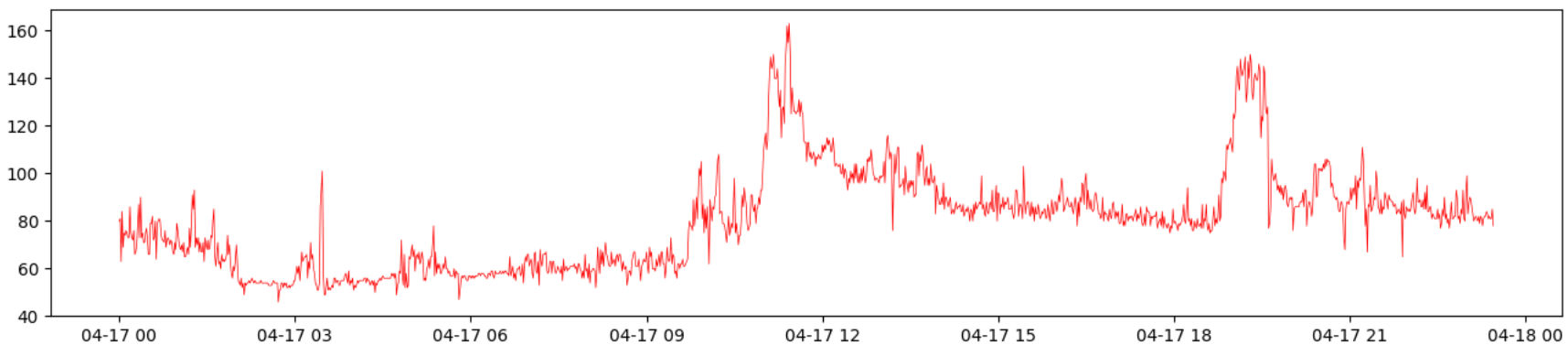


Screenshot Huawei Health App

My Heart Rate Over One Day



Screenshot Huawei Health App



Screenshot Python Matplotlib Plot

Motivation for RE the Huawei Health Android App[3]

- No dependency on the Huawei ecosystem
  - I don’t want to upload my personal data
  - Lack of export possibilities[1]
  - End of product life cycle (cf. discontinuation of Suunto Movescount[2])
- Fun ☺

[1] https://uk.community.huawei.com/watch-gt-50/watch-gt-with-no-strava-integration-1285

[2] https://www.suunto.com/en-gb/Content-pages/digital-service-transition/

[3] https://play.google.com/store/apps/details?id=com.huawei.health

My Objectives …

- … for reversing of the Huawei Health app were:
  - To understand the communication between the smartphone and the watch
  - Extract my personal local stored data from the Huawei Health app (encrypted SQLite database)

- … of this talk are:
  - Introducing generally applicable methods for reverse engineering Android applications
  - Sharing my results regarding the Huawei Health app and the communication protocol

Bluetooth Low Energy (BLE)

BLE: Generic Attributes (GATT)

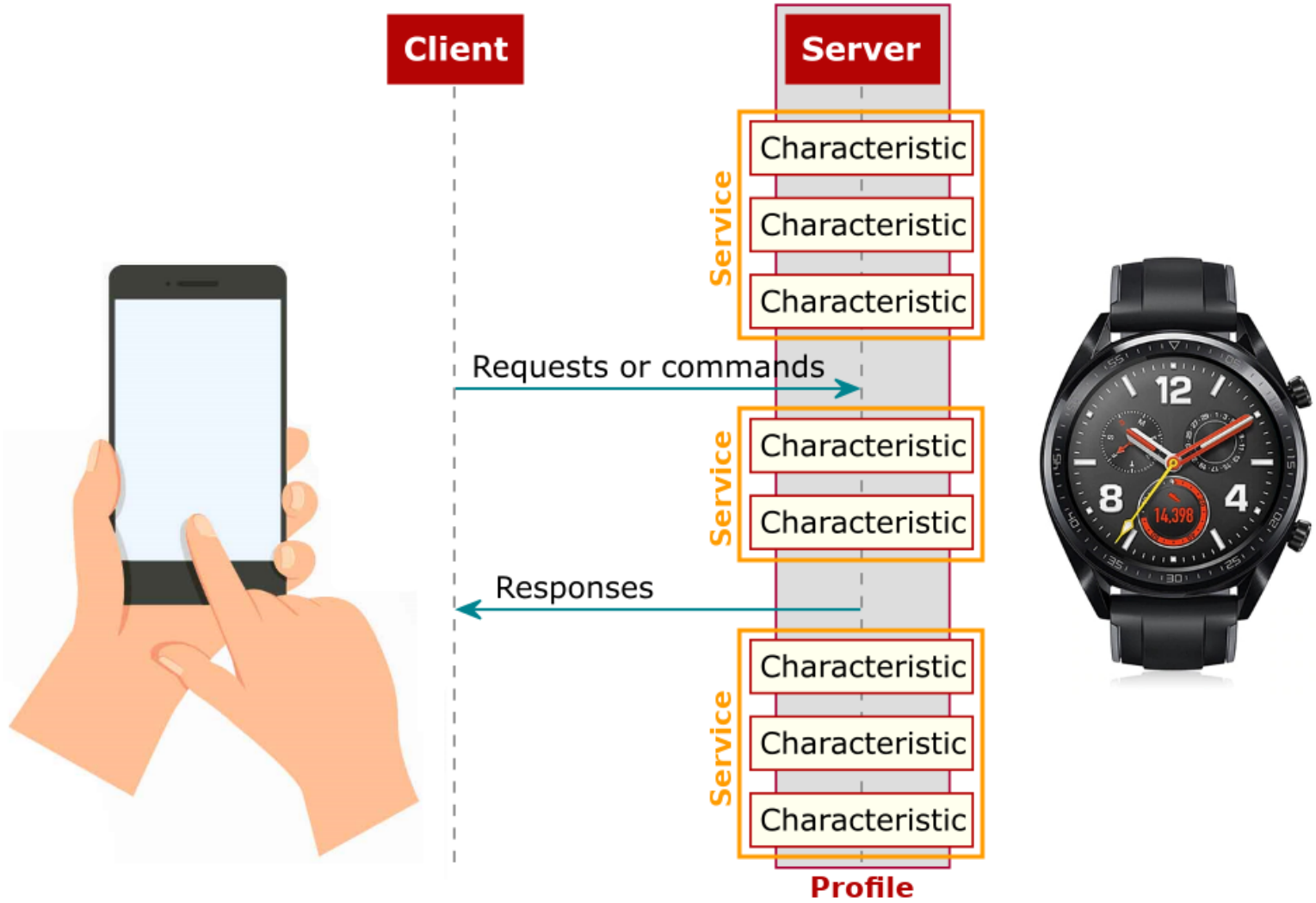
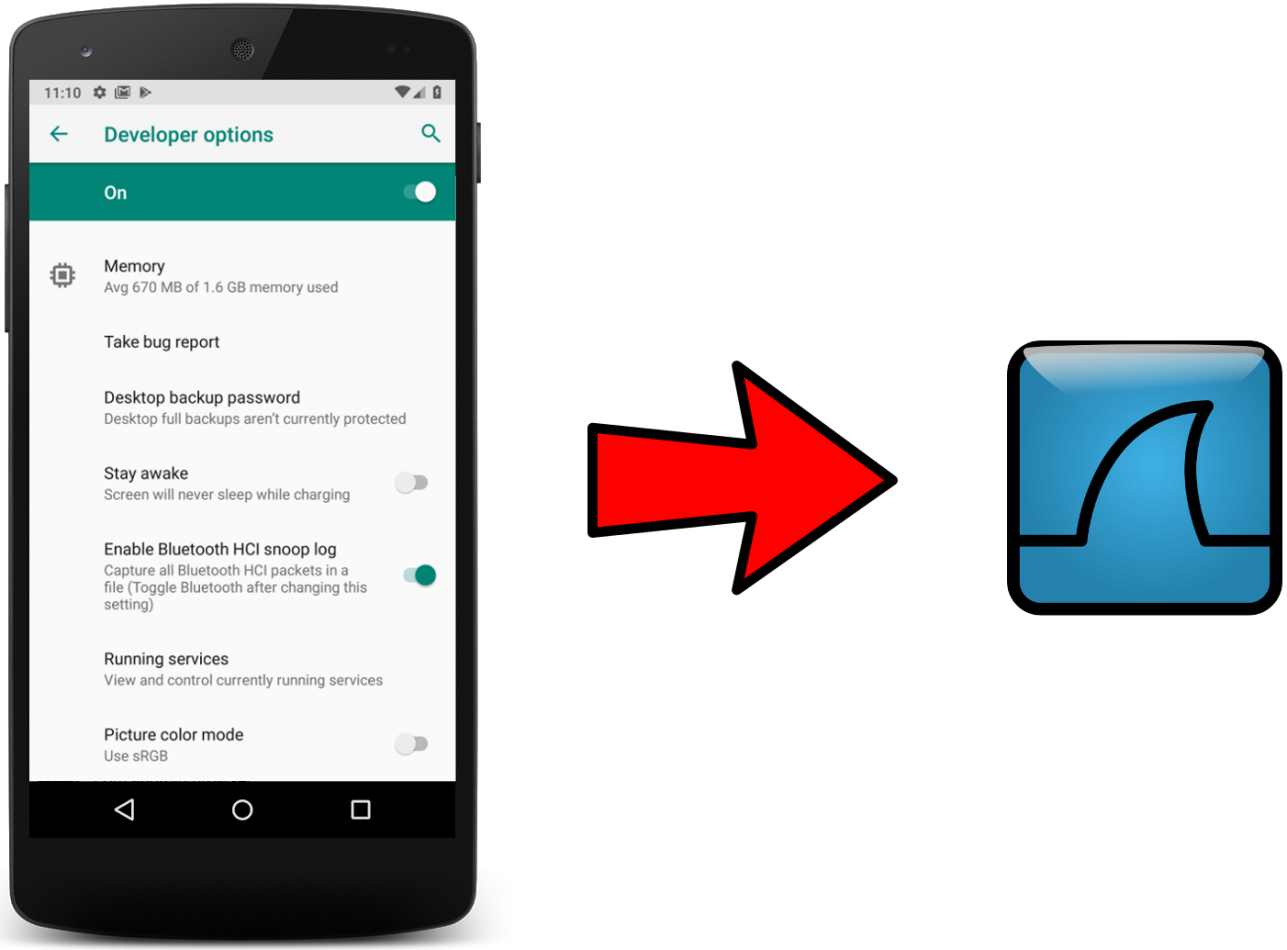
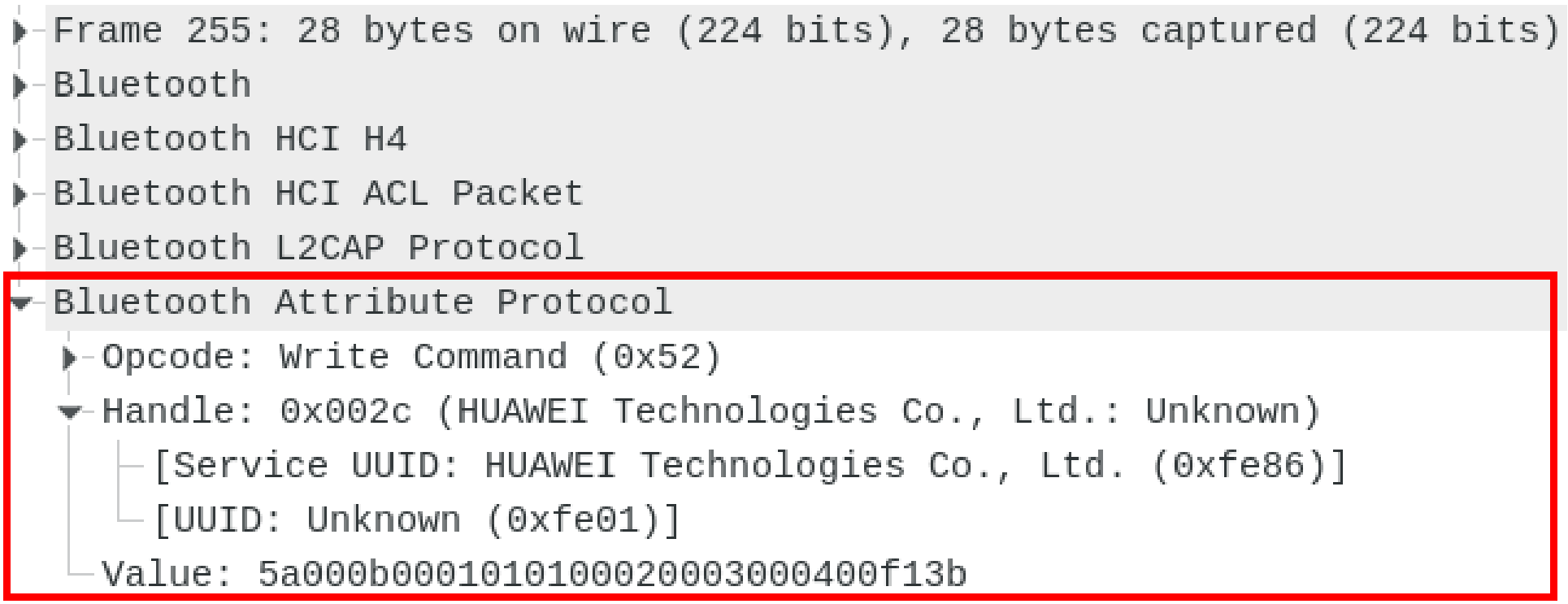


Figure source: `http://dev.ti.com/tirex/content/simplelink_cc2640r2_sdk_1_35_00_33/docs/ble5stack/ble_user_guide/html/ble-stack/gatt.html`

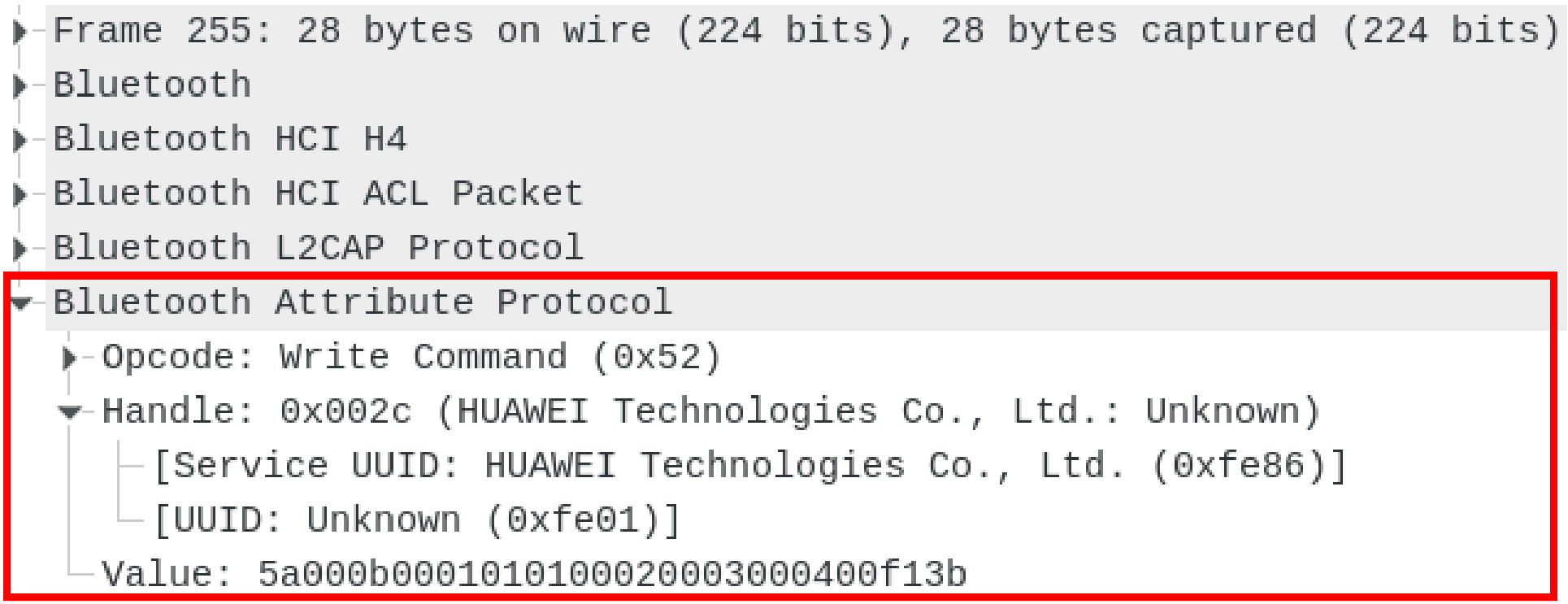
Bluetooth Sniffing on Android



Bluetooth Sniffing on Android – First Write Command



Bluetooth Sniffing on Android – First Write Command



What does this write command do?

```
5A 00 0B 00 01 01 01 00 02 00 03 00 04 00 F1 3B
```

Static Program Analysis

Extraction of Android Apps

APKMirror[4]

- Archive of downloadable Android apps, so called APKs (Android Application Packages)
- Download of different (older) versions
- Only free apps are available (APKMirror has a no-piracy policy)
- Cryptographic signature guarantees origin
- Unfortunately, not all free Android apps are available

[4] `https://www.apkmirror.com`

Android Debug Bridge (adb)[6]

- Command-line tool that lets you communicate with a device
- Part of the Android SDK Platform Tools[5]
- Can be used to extract apps from **non-rooted** or **rooted** devices:

[5] https://developer.android.com/studio/releases/platform-tools

[6] https://developer.android.com/studio/command-line/adb

Android Debug Bridge (adb) – Example

Find the apk name of the Huawei Health app:

```
$ adb shell pm list packages | grep huawei
package:com.huawei.health
```

Find the path of the Huawei Health app:

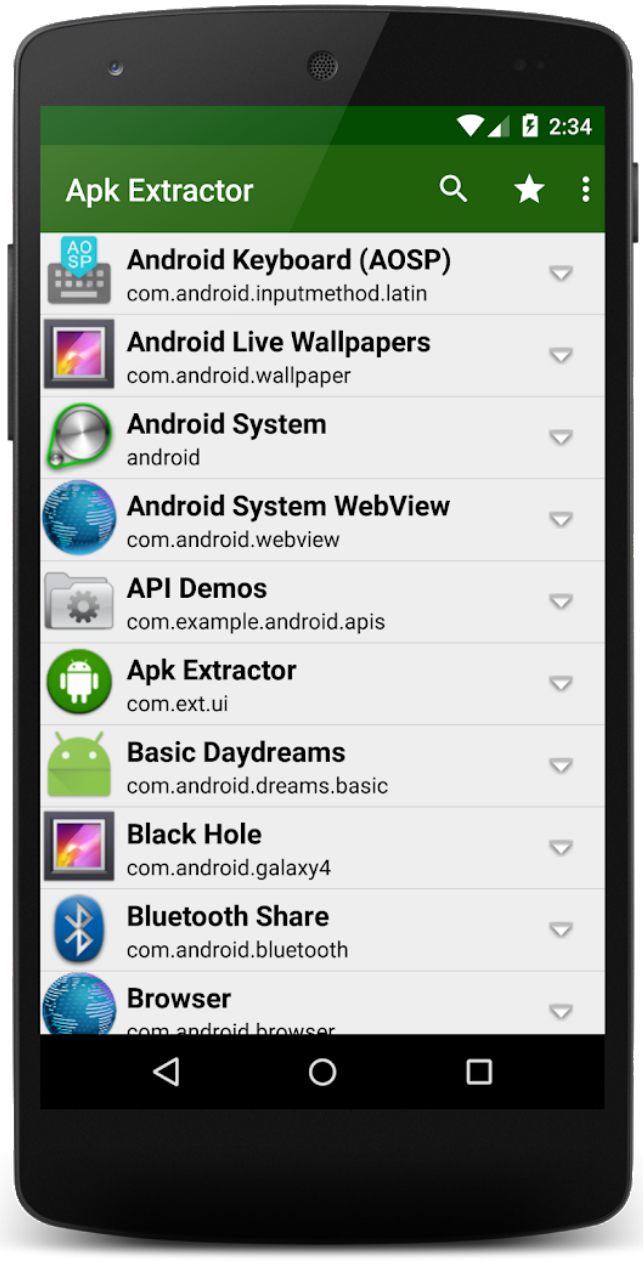
```
$ adb shell pm path com.huawei.health
/data/app/com.huawei.health-1/base.apk
```

Download the Huawei Health app to your system:

```
$ adb pull /data/app/com.huawei.health-1/base.apk
base.apk  6.0 MB/s (78453174 bytes in 12.387s)
```

APK Extractor

- Extracts apps that are installed on an android device
- Copies the extracted apps to SD card
- Does not require root access
- Disadvantages:
  - Advertisement in the app
  - You have to trust the developer (e.g., no modification of apps)



Static Program Analysis

Decoding of Android Apps

Decoding of Android Apps

- APKs are zip files containing resources and assembled java code
- If you simply unzip an APK, you would be left with files such as `classes.dex` and `resources.arsc`
- The files in the archive are compiled sources and not human readable
  - E.g., `AndroidManifest.xml` is in format *Android binary XML*
- With APKTool[7] apps can be decoded

[7] `https://github.com/iBotPeaches/Apktool`

Decoding with APKTool

```
$ java -jar apktool.jar decode health.apk
I: Using Apktool 2.4.0 on health.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
[...]
I: Baksmaling classes11.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

Decoding with APKTool – AndroidManifest.xml

```
$ java -jar apktool.jar decode health.apk
I: Using Apktool 2.4.0 on health.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Baksmaling classes2.dex...
[...]
I: Baksmaling classes11.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
```

AndroidManifest.xml

- Every app must have a `AndroidManifest.xml` file in its root directory
- The manifest provides essential information about the app to the Android system

- Interesting from the analyst's point of view:
  - Is debugging allowed?
  - Is the creation of backups allowed (i.e., is it possible to extract the app's data)?

AndroidManifest.xml – Debugging and Backups

- The debugging of the Huawei Health app is not permitted
- It’s not allowed to backup the app’s data

- Huawei Health `AndroidManifest.xml`:

```
[...]

<application android:allowBackup="false" android:debuggable="false" ...

[...]
```

Static Program Analysis

Repatching of Android Apps

Repatching with APKTool

- With APKTool changes (e.g., `AndroidManifest.xml`, smali code, resources) in the decoded app can be made and afterwards the app can be rebuilt

Repatching with APKTool

- With APKTool changes (e.g., `AndroidManifest.xml`, smali code, resources) in the decoded app can be made and afterwards the app can be rebuilt

Repatching with APKTool

- ~~With APKTool changes (e.g., `AndroidManifest.xml`, smali code, resources) in the decoded app can be made and afterwards the app can be rebuilt~~
- Known issue[8] for apps which use AndResGuard[9] (obfuscation tool)
  - Rebuilding resources leads to an error
  - Repatching `AndroidManifest.xml` requires the decoding of `resources.arsc`
  - Huawei Health app: Only source (e.g., `classes.dex`) can be modified with APKTool

[8] `https://github.com/iBotPeaches/Apktool/issues/1361`

[9] `https://github.com/shwenzhang/AndResGuard`

Repatching with APKTool – In Theory

Rebuild the prior decoded app:

```
$ java -jar apktool.jar build sample/ -o sample_new.apk
```

Download an open source Android signing framework:

```
$ git clone https://github.com/appium/sign
```

Sign the APK with the Android test certificate

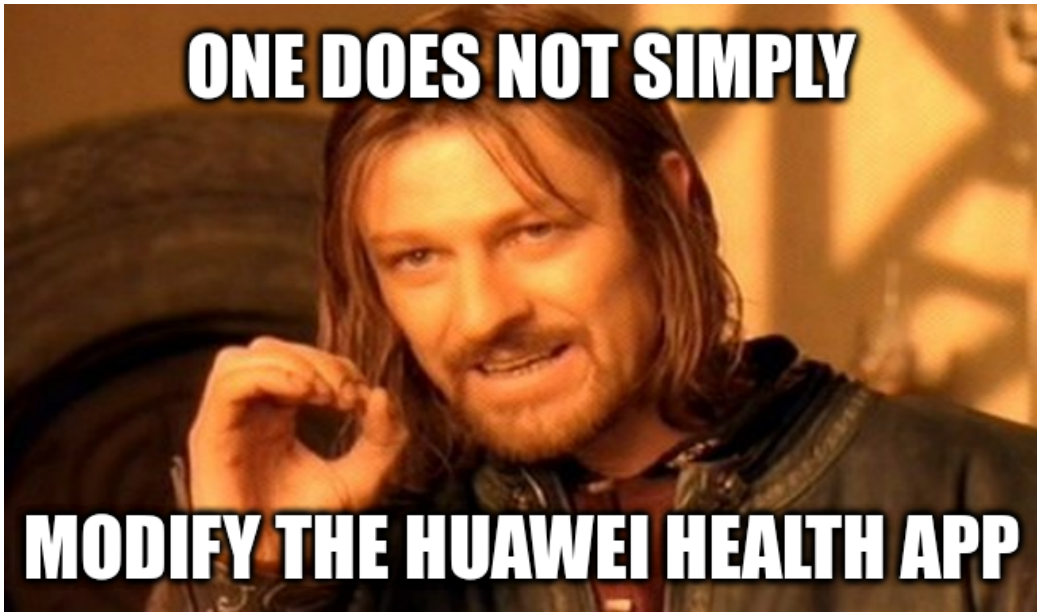
```
$ java -jar sign/dist/signapk.jar sign/testkey.x509.pem \
  sign/testkey.pk8 sample_new.apk sample_new_signed.apk
```

Install the repatched app on your device:

```
$ adb install sample_new_signed.apk
```

Repatching with APKTool – It's not that simple

- The Huawei Health app checks its signature during startup
- If the signature does not match with the stored signature, it exits with an error
- Two possible solutions:
  - Find the signature check and modify it (time-consuming)
  - Modify the app during runtime (easy, we will come to that)



Static Program Analysis

Extracting the App Data

Extracting App Data – Run-As Method

If debugging is permitted (`AndroidManifest.xml`), accessing app data is possible with the `run-as` command

The following command copies the app data to `/sdcard`:

```
$ adb shell 'run-as com.example.app sh -c \
  "cp -r /data/data/com.example.app /sdcard"'
```

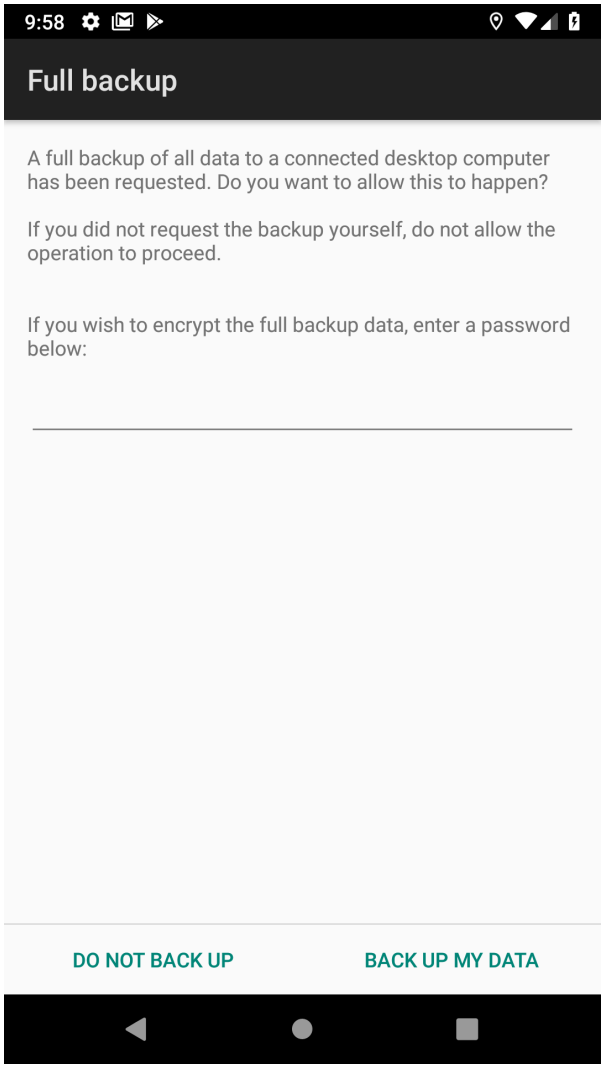
Download the app data from `/sdcard` to your system:

```
$ adb pull /sdcard/com.example.app
```

Extracting App Data – Backup Method

```
$ adb backup -f backup.ab -apk com.example.app
 Now unlock your device and confirm the backup operation...
```

- If the creation of backups is permitted (`AndroidManifest.xml`), a backup can be created with the `backup` command
- The backup (`.ab` file) can then be extracted with *Android backup extractor*[10]



[10] `https://github.com/nelenkov/android-backup-extractor`

Extracting App Data – Rooted Method

Copy Huawei Health data folder to `/sdcard`:

```
$ adb shell su -c 'cp -r /data/data/com.huawei.health/ /sdcard/'
```

Download the app data from `/sdcard` to your system:

```
$ adb pull /sdcard/com.huawei.health
  /sdcard/com.huawei.health/: 325 files pulled
```

Static Program Analysis

Decompilation, Deobfuscation and Code Analysis

Code Execution on Android

- An Android Application Package (APK) contains one or more Dalvik Executables (DEX) (`classes.dex`, `classes2.dex`, `classes<n>.dex`)
  - The Dalvik Executable specification limits the total number of methods that can be referenced within a single DEX file (e.g., `classes.dex`) to 65.536
  - Since the Huawei Health app has 175.369 methods[11], multiple DEX files are necessary
- A Dalvik Executable (DEX) contains bytecode which is executed by the Android runtime (ART)

[11] Counted with `https://github.com/google/android-classyshark`

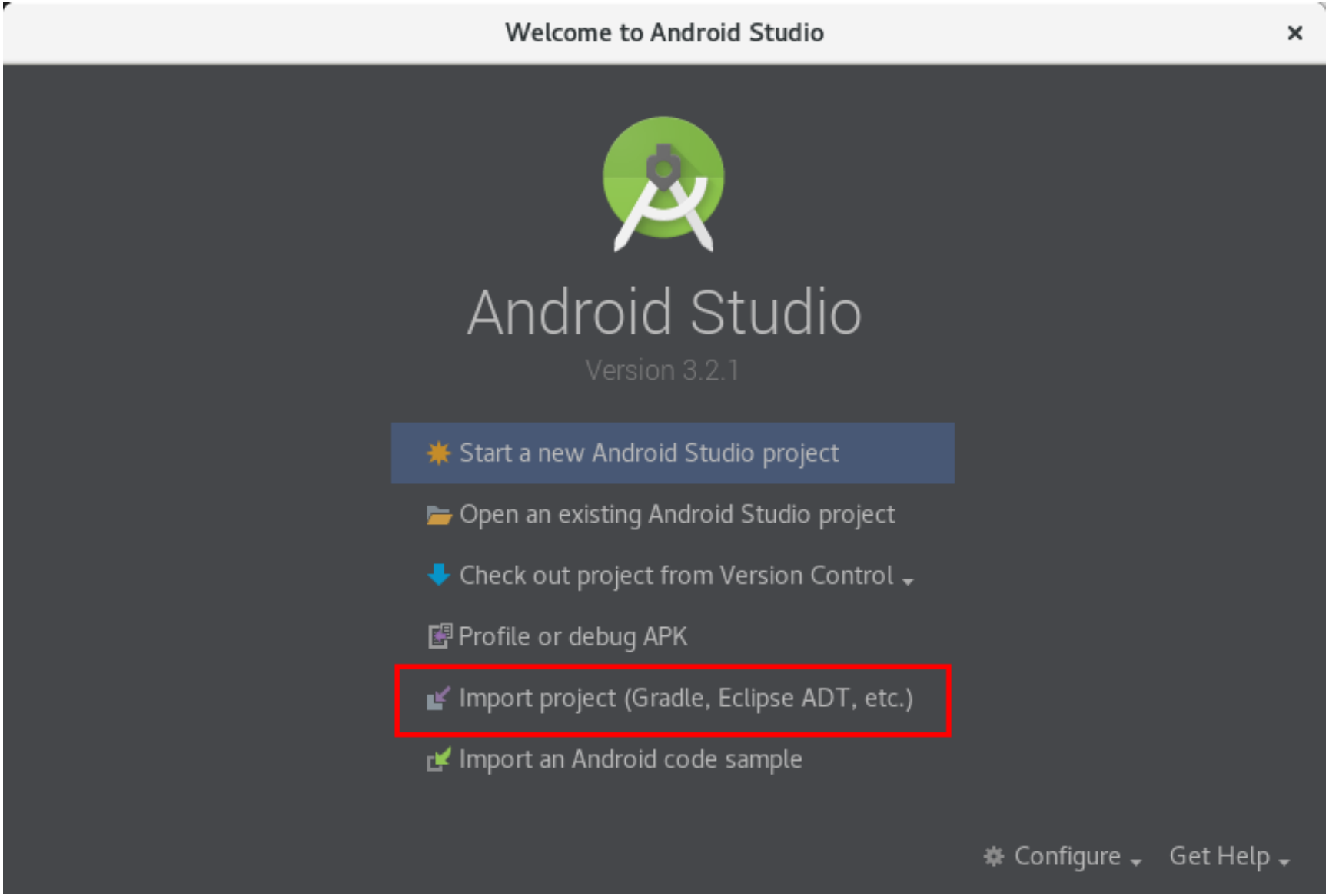
Decompilation with Jadx[15]

- Command line and GUI tool to decompile Android DEX and Apk files to Java source code
  - Static source code analysis can be done with the GUI, but I prefer Android Studio[12]
  - Also has a deobfuscation functionality, but I prefer to do it manually due to false positives
  - Sometimes Jadx is not able to decompile some methods, then you have to try other decompilers (e.g., CFR[13], dex2jar[14])
- Decompilation of the Huawei Health app:

```
$ jadx --no-res -d health/ health.apk
```

[12] https://developer.android.com/studio
[13] http://www.benf.org/other/cfr/
[14] https://github.com/pxb1988/dex2jar
[15] https://github.com/skylot/jadx

Android Studio – Import



Generic Attributes (GATT) with the Android SDK[16]

BluetoothGatt

Added in API level 18

```
public final class BluetoothGatt
extends Object implements BluetoothProfile
```

java.lang.Object
↳ android.bluetooth.BluetoothGatt

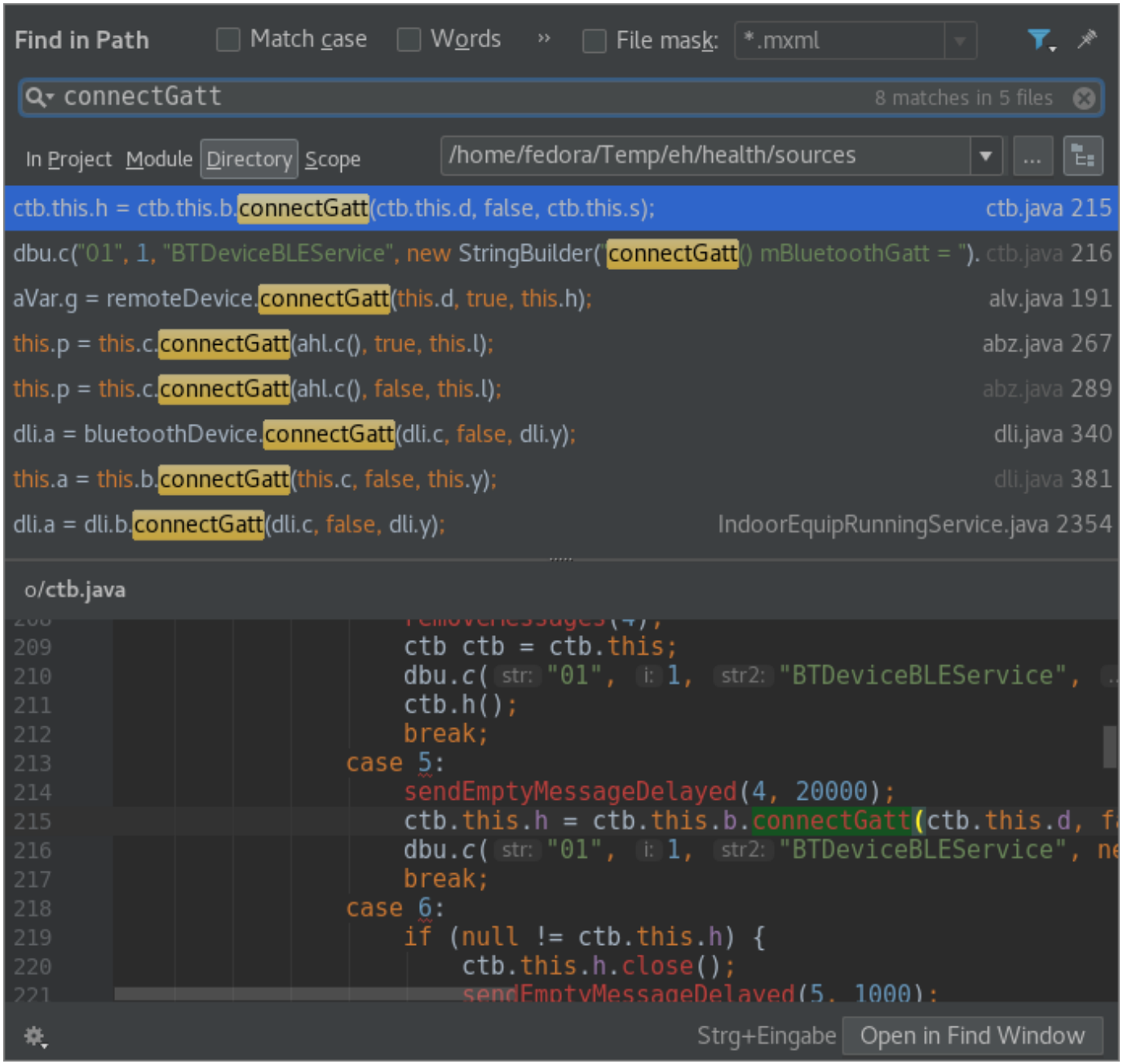
Public API for the Bluetooth GATT Profile.

This class provides Bluetooth GATT functionality to enable communication with Bluetooth Smart or Smart Ready devices.

To connect to a remote peripheral device, create a `BluetoothGattCallback` and call `BluetoothDevice#connectGatt` to get a instance of this class. GATT capable devices can be discovered using the Bluetooth device discovery or BLE scan process.

[16] https://developer.android.com/reference/android/bluetooth/BluetoothGatt

Android Studio – Find in all Files (CTRL + SHIFT + F)



Android Studio – onServicesDiscovered Method

```
public final void onServicesDiscovered(BluetoothGatt bG, int i) {
  [...]
  dbu.c("01", 1, "BTDeviceBLEService", "Service discover success.");
  ctb.f.removeMessages(4);
  BluetoothGattService service = ctb.h.getService(
    UUID.fromString("0000fe86-0000-1000-8000-00805f9b34fb"));
  if (service != null) {
    dbu.c("01", 1, "BTDeviceBLEService", "Service UUID find success");
    synchronized (ctb.q) {
      ctb.g = service.getCharacteristic(
        UUID.fromString("0000fe01-0000-1000-8000-00805f9b34fb"));
    }
    ctb.m = service.getCharacteristic(
      UUID.fromString("0000fe02-0000-1000-8000-00805f9b34fb"));
    ctb.b();
    return;
  }
  [...]
}
```

Android Studio – Deobfuscation through Logging

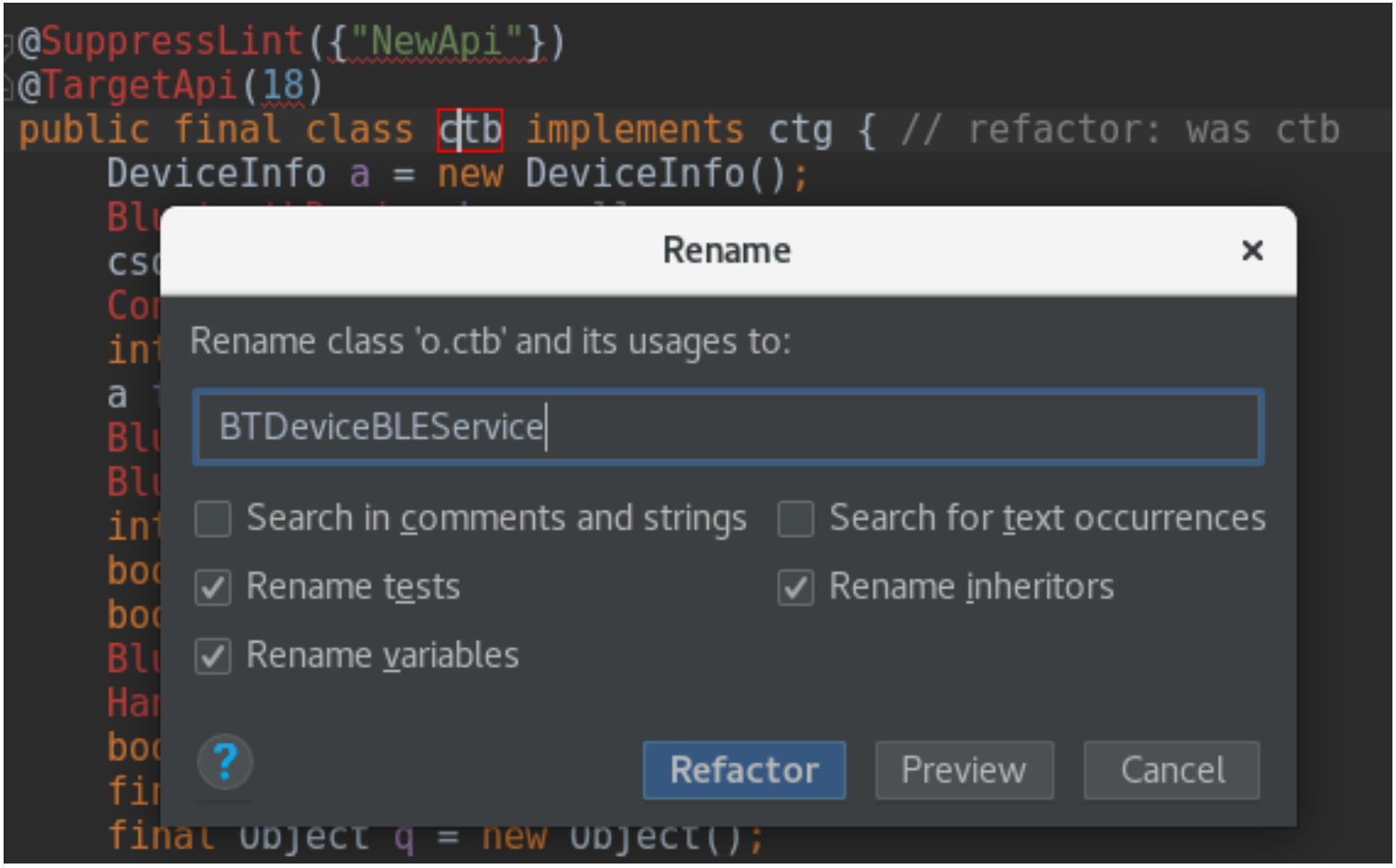
Class and method name:

```
private String c(String str) {
  [...]
  dbu.c("01", 1, "BTDeviceBLEService", "getTokenValue()");
  [...]
}
```

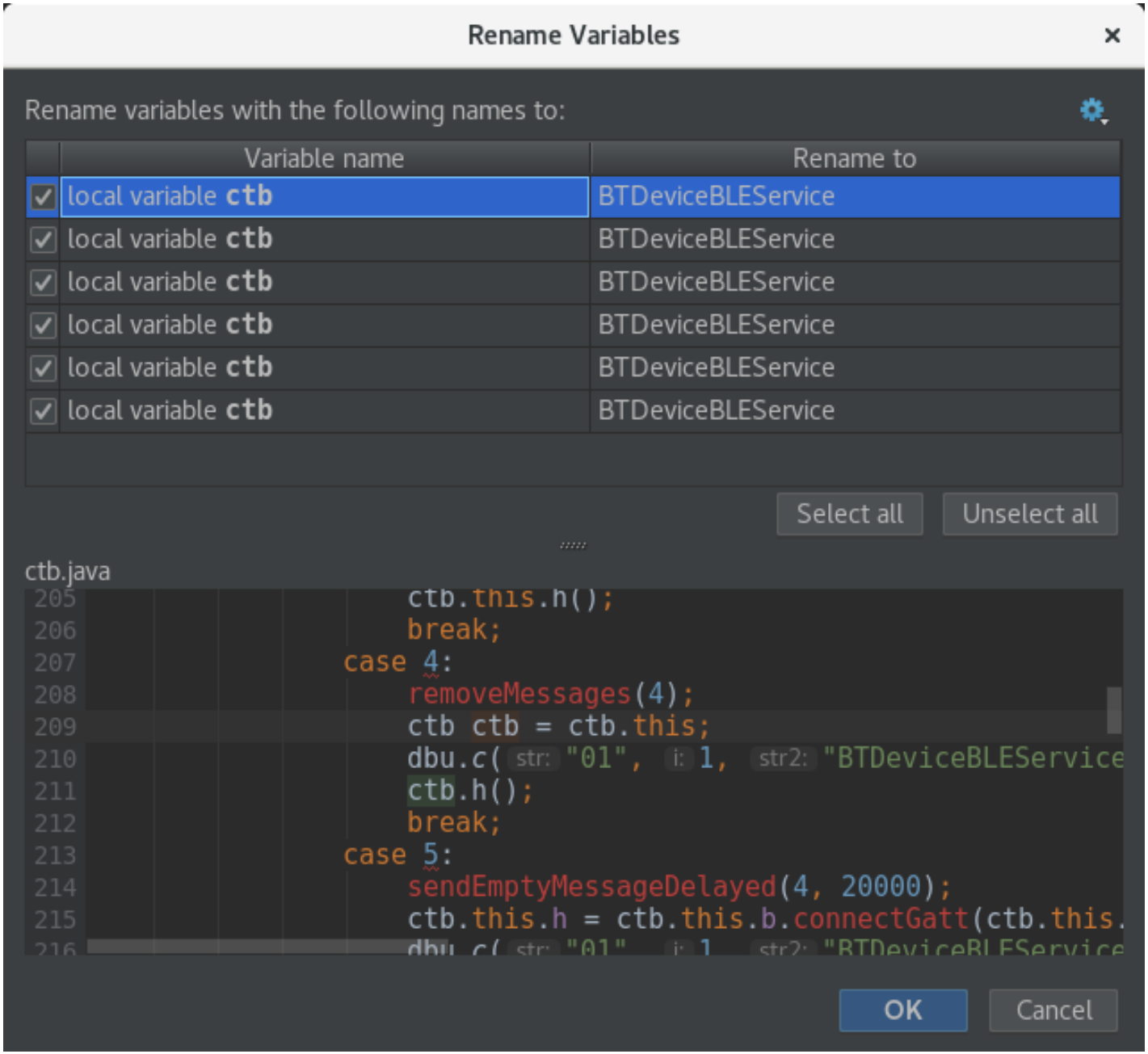
Variable name:

```
dbu.c("01", 1, "", "Device name = " + this.x.toString());
```

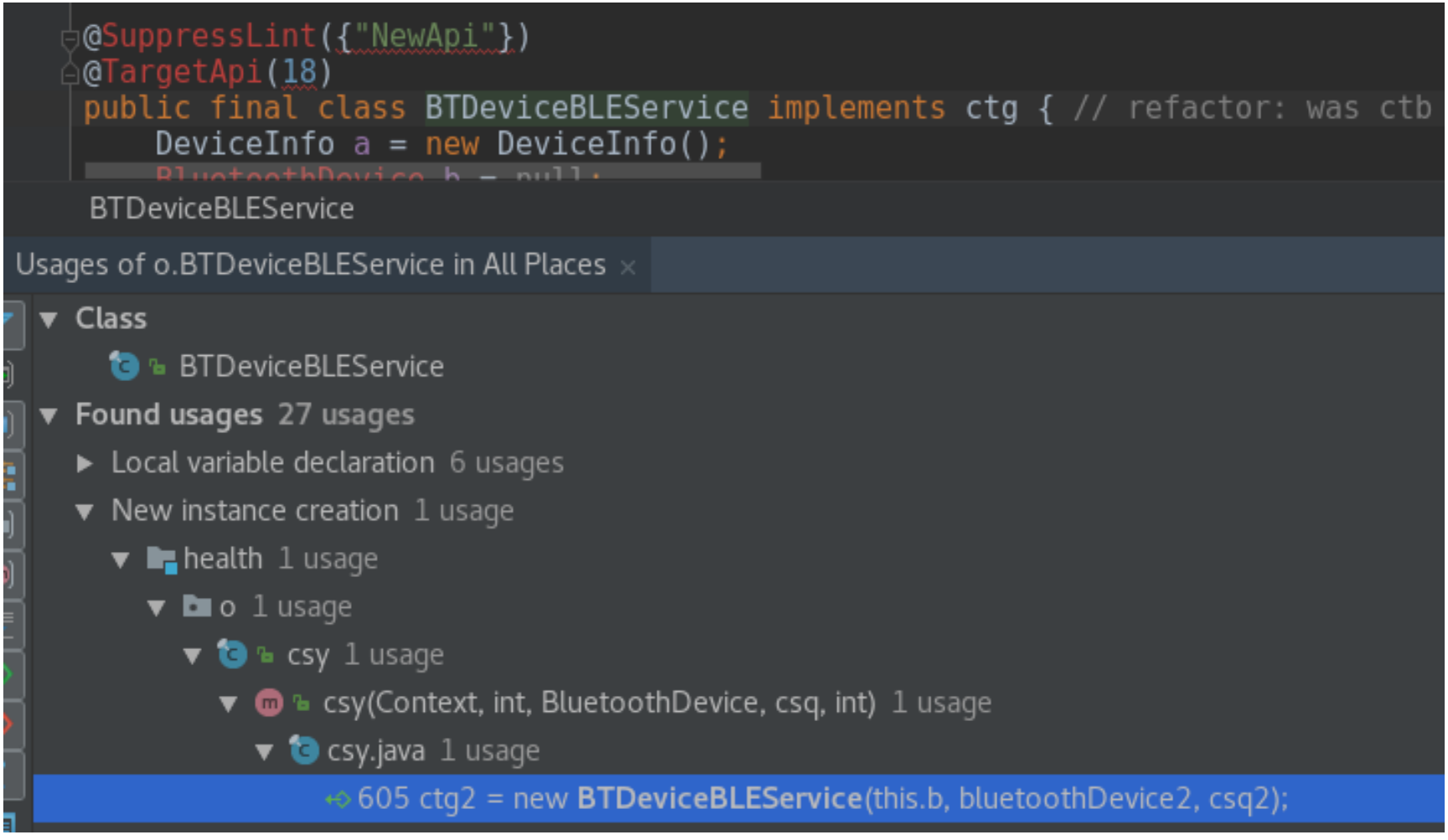
Android Studio – Refactor (SHIFT + F6)



Android Studio – Refactor (SHIFT + F6)



Android Studio – Find Usages (ALT + SHIFT + 7)



Static Code Analysis

What does this write command do?

```
5A 00 0B 00 01 01 01 00 02 00 03 00 04 00 F1 3B
```

```java
// refactor: public static csu e(int i)
public static BTHandshakeManager getBTDeviceLinkParameter(int i) {
  [...]
  if (i == 0) {
    [...]
  } else {
    byte[] bArr2 = {1, 1, 1, 0, 2, 0, 3, 0, 4, 0};
    Integer valueOf2 = Integer.valueOf(10);
    csu.c = (valueOf2 == null ? null : valueOf2).intValue();
    byte[] copyOf2 = Arrays.copyOf(bArr2, 10);
    csu.b = copyOf2 == null ? null : copyOf2;
  }
  [...]
  return csu;
}
```

Static Code Analysis – It Requires Some Effort …

Dynamic Program Analysis

Frida

Frida[17]

- “Dynamic instrumentation toolkit for developers, reverse-engineers, and security researchers”
- Scriptable
  - Inject own scripts
  - Function hooking
  - Trace application code
- Portable
  - Windows, macOS, GNU/Linux, iOS, Android, QNX
- Requires a rooted device
- Requires the installation and execution of `frida-server` on the device:

```
$ adb shell su -c "frida-server"
```

[17] `https://www.frida.re/`

In which database are the fitness data stored?

```
$ adb shell su -c "ls -l /data/data/com.huawei.health/databases"
  -rw-rw---- [...]     20480 HwCPBackupDatas.db
  -rw------- [...]    180224 com_huawei_health101010.db
  -rw------- [...]    106496 com_huawei_health20003.db
  -rw------- [...]      4096 com_huawei_health20004.db
  -rw------- [...]     12288 com_huawei_health20005.db
  -rw------- [...]   2207744 hihealth_003.db
  -rw-rw---- [...]     40960 open_service4.db
  -rw-rw---- [...]      8720 open_service4.db-journal
```

Which processes are executed by the Health App?

```
$ adb shell ps | grep huawei.health
  u0_a175   16691 [...] com.huawei.health
  u0_a175   16720 [...] com.huawei.health:DaemonService
  u0_a175   16830 [...] com.huawei.health:PhoneService
```

Decrypt `hihealth_003.db` – Frida Python Wrapper

```python
import sys, frida

def my_message_handler(message, payload):
    print(message, payload)

if(len(sys.argv) != 2):
    print("Error: Please specify process name!")
    sys.exit(1)

session = frida.get_usb_device().attach(sys.argv[1])

with open("frida.js") as f:
    script = session.create_script(f.read())

script.on("message", my_message_handler)
script.load()

# prevent the python script from terminating
sys.stdin.read()
```

start_frida.py

```
$ python3 attach_frida.py com.huawei.health:DaemonService
```

Decrypt `hihealth_003.db` – Frida Script

```
Java.perform(function () {
  Java.choose("net.sqlcipher.database.SQLiteDatabase", {
    onMatch: function(instance) {
      var path = instance.mPath.value
      if(path.includes("hihealth_003.db")) {
        /* Remove hihealth_003.db.plain if exists */
        var File = Java.use("java.io.File");
        var file = File.$new(path + ".plain");
        file.delete();

        /* create non-encrypted copy of hihealth_003.db */
        instance.rawExecSQL(
          "ATTACH DATABASE '" + path + ".plain' AS plaintext KEY '';"
        );
        instance.rawExecSQL("SELECT sqlcipher_export('plaintext');");
        instance.rawExecSQL("DETACH DATABASE plaintext;");
      }
    },
    onComplete: function() {
      console.log("onComplete");
    }
  });
});
```

frida.js

Structure of `hihealth_003.db`

Name	Typ	Schema
Tabellen (27)		
config_data		CREATE TABLE config_data(_id integer primary key not null,start_time integer not null,end_time in
config_stat_day		CREATE TABLE config_stat_day(_id integer primary key not null,date integer not null,hihealth_type
goal_value		CREATE TABLE goal_value(_id integer primary key not null,goal_type integer,goal_value double,goa
hihealth_account		CREATE TABLE hihealth_account(_id integer primary key not null,huid text not null,app_id integer n
hihealth_app		CREATE TABLE hihealth_app(_id integer primary key not null,package_name text not null,app_name
hihealth_authorization		CREATE TABLE hihealth_authorization(_id integer primary key not null,app_id integer no null,user_i
hihealth_dataclient		CREATE TABLE hihealth_dataclient(_id integer primary key not null,client_uuid text not null,user_id
hihealth_datatype		CREATE TABLE hihealth_datatype(_id integer primary key not null,version text,value_type integer n
hihealth_device		CREATE TABLE hihealth_device(_id integer primary key not null,device_unique_code text not null,d
hihealth_permission		CREATE TABLE hihealth_permission(_id integer primary key not null,cloud_id integer no null,scope.
hihealth_source_order		CREATE TABLE hihealth_source_order(_id integer primary key not null,type_id integer not null,clier
hihealth_stat_day		CREATE TABLE hihealth_stat_day(_id integer primary key not null,date integer not null,hihealth_typ
hihealth_temp		CREATE TABLE hihealth_temp(_id integer primary key not null,tempKey integer not null,tempValue
hihealth_type_inherit		CREATE TABLE hihealth_type_inherit(_id integer primary key not null,type_id integer no null,parent
hihealth_unit		CREATE TABLE hihealth_unit(_id integer primary key not null,unit_string_id text not null,uni_string.
hihealth_user		CREATE TABLE hihealth_user(_id integer primary key not null,huid text not null,nick_name text,hea
real_time_health		CREATE TABLE real_time_health(_id integer primary key not null,start_time integer not null,end_tir
sample_point		CREATE TABLE sample_point(_id integer primary key not null,start_time integer not null,end_time i
sample_point_health		CREATE TABLE sample_point_health(_id integer primary key not null,start_time integer not null,en
sample_point_health_stress		CREATE TABLE sample_point_health_stress(_id integer primary key not null,start_time integer not
sample_sequence		CREATE TABLE sample_sequence(_id integer primary key not null,start_time integer not null,end_t
sample_session		CREATE TABLE sample_session(_id integer primary key not null,start_time integer not null,end_tim
sample_session_core		CREATE TABLE sample_session_core(_id integer primary key not null,start_time integer not null,en
sample_session_health		CREATE TABLE sample_session_health(_id integer primary key not null,start_time integer not null,e
sync_anchor		CREATE TABLE sync_anchor(_id integer primary key not null,cloud_code integer no null,main_user.
sync_cache		CREATE TABLE sync_cache(_id integer primary key not null,userId integer not null,dataType integer
user_preference		CREATE TABLE user_preference(_id integer primary key not null,key text not null,value text not nul


HELL
YEAH

The Huawei Link Protocol v2

The Huawei Link Protocol v2

- This section will give a brief overview of structure of the Huawei Link Protocol v2
- Parts of my research are already published[18] in an issue on GitHub (Gadgetbridge project[19])
  - I'm not the only one you is doing research on the Huawei Link Protocol v2, but perhaps the only one who is doing it for the Huawei Watch GT
  - We compared our results and it seems that Huawei is using the protocol for different wearables:
    - Huawei Band 3 Pro
    - Huawei Watch GT
    - Honor Band 4 (Honor is a sub-brand of Huawei)

[18] https://github.com/Freeyourgadget/Gadgetbridge/issues/1021#issuecomment-450450598

[19] https://github.com/Freeyourgadget/Gadgetbridge

The Huawei Link Protocol v2

Message Structure

Message Structure

What does this write command do?

```
5A 00 0B 00 01 01 01 00 02 00 03 00 04 00 F1 3B
```

Message Structure

What does this write command do?

```
5A 00 0B 00 01 01 01 00 02 00 03 00 04 00 F1 3B
```

- Messages can be transferred in a **single package** or in a **multi package**
- **Single package**: Can be *not sliced* or *sliced*
- **Multi package**: Not further investigated, since I never recorded the transfer of a multi package

Package Structure – Single Package Not Sliced

```
0x5a LLLL 0x00 CONTENT CRC16
```

Field	Length [bytes]	Description
LLLL	2	length of CONTENT + 1
CONTENT	min: 0 max: frame size - 6	content
CRC16	2	CRC-CCITT (XModem)

CONTENT Structure

```
CONTENT: SS CC TLVs
```

Field	Length [bytes]	Description
SS	1	service id
CC	1	command id
TLVs	varies	arbitrary number of TLV

TLV Structure

```
TLV: TAG LENGTH DATA
```

Field	Length [bytes]	Description
TAG	1	tag of the TLV
LENGTH	varint	length of DATA
DATA	defined by LENGTH	data of TLV

Variable-Length Integer (varint): Splits an integer into 7-bit byte array, uses the most-significant-bit (the 8th bit) to tell if it is the last byte

The Huawei Link Protocol v2

Command “Request Link Parameter”

Command “Request Link Parameter”

~~What does this write command do?~~

Request the link parameters from the peripheral:

```
5A 00 0B 00 01 01 01 00 02 00 03 00 04 00 F1 3B
```

constants, message length, service id, command id, TLVs, CRC16

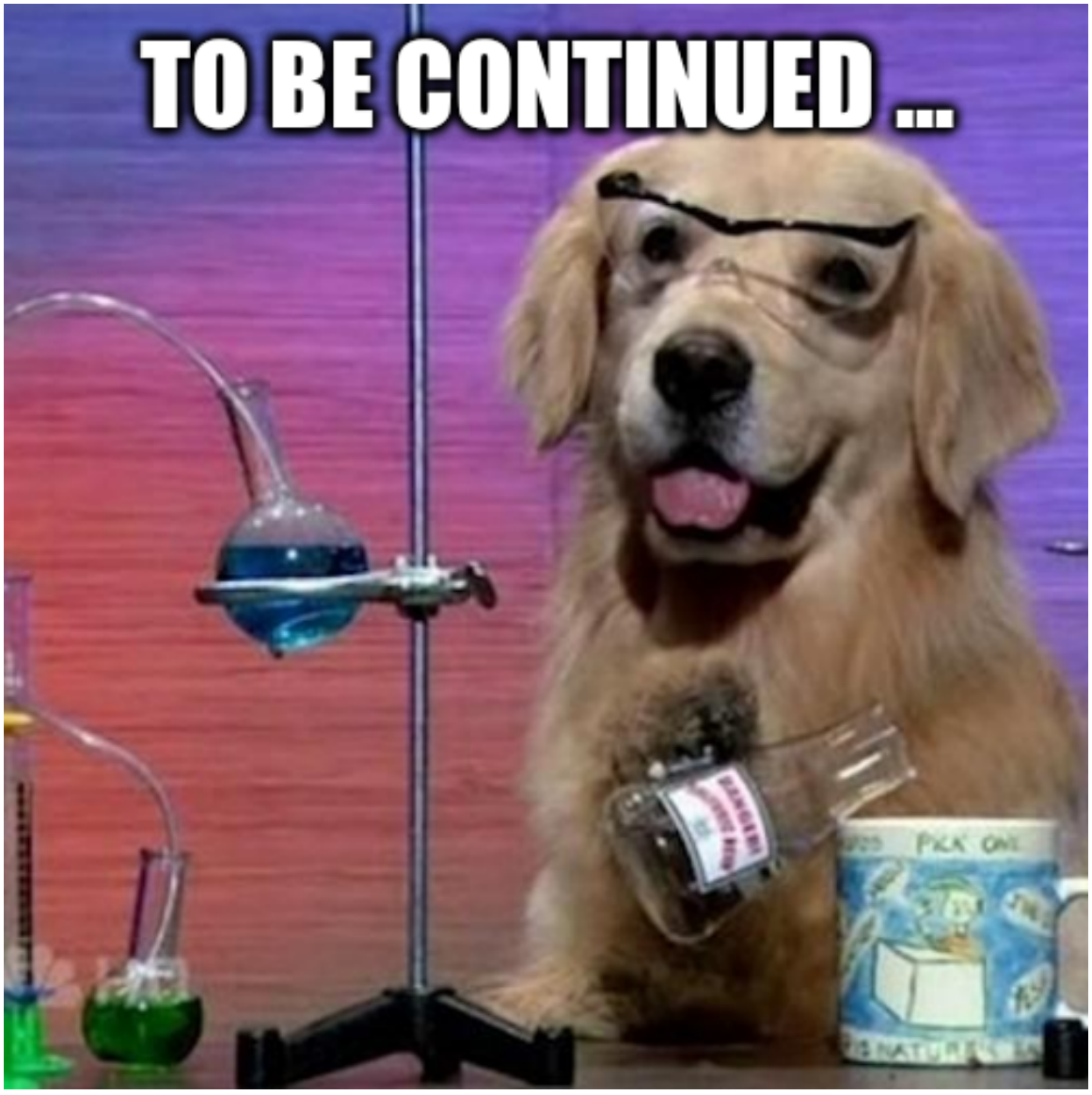
Tag	Length [bytes]	Description
0x01	0x00	request peripheral protocol version
0x02	0x00	request peripheral max frame size
0x03	0x00	request bt_service_mtu
0x04	0x00	request BLE connection interval [ms]

Response for Command “Request Link Parameter”

```
5A 00 26 00 01 01 01 01 02 02 02 00 FE 03 02 00 14 04 02 00
0A 05 12 00 01 F7 B1 EC 24 3C A5 DD 75 50 AB A7 DB 06 77 B4
38 71 85
```

constants, message length, service id, command id, TLVs, CRC16

Tag	Length [bytes]	Description
0x01	0x01	peripheral protocol version
0x02	0x02	peripheral max frame size
0x03	0x02	bt_service_mtu
0x04	0x02	BLE connection interval [ms]
0x05	0x12	XX Y*16 XX auth version Y*16 randA


TO BE CONTINUED ...

Conclusion

Conclusion

- Reversing Android applications isn't that hard, but requires some effort
- If you are interested in a communication protocol, normally it's easier to RE the corresponding app instead of to the embedded device
- If you want to protect your Android application, you should remove the logging strings in the release version
- It's possible to use Huawei wearables without the related Huawei ecosystem
  - Side note: For Fitbit it seems that this isn't possible without firmware modification[20]

[20] Easterhegg 2018: Hacking your Fitbit, `https://media.ccc.de/v/TNYPFB`

Help the SBA Researchers

- Why do people think that their hardware devices are genuine and trustworthy?
  - E.g., hardware wallets, yubikeys, smartphones, etc.

- Complete the online survey[21] and take part in the **raffle** for 3 x **Amazon vouchers 50€** each and 10 x 3 packages of **Zotter chocolate**

- The survey should only take **15-20 minutes**

- Please enter my mail address[22] on the begin of the survey ☺

[21] https://de.surveymonkey.com/r/FFRFTGV

[22] ckudera@sba-research.org

Theses and Job Offer

- SBA Research is always searching[23] for motivated TU Wien students (Computer Science, Electrical Engineering) for **bachelor's thesis** or **master's thesis**

- SBA Research offers professional services to customers
  - **Job offer:** We are always looking for
    - Information Security Consultants
    - Technical IT Security Consultants

- Contact: ckudera@sba-research.org

[23] https://www.sba-research.org/research/bachelor-master-phd-thesis-supervision/

SBA Research

COMET

Competence Centers for
Excellent Technologies

Christian Kudera

ckudera@sba-research.org

Bundesministerium
Verkehr, Innovation
und Technologie

Bundesministerium
Digitalisierung und
Wirtschaftsstandort

FFG
Forschung wirkt.

wirtschafts
agentur
wien
Ein Fonds der
Stadt Wien

FWF
Der Wissenschaftsfonds.

netidee
OPEN INNOVATIONS