Sideload this!

About me slides always feel so narcisisstic

- I’ve given talks at a few of these
- Hilariously, rarely at the same employer twice
- I like math
- And computers
- But mostly, fun problems!

Let’s get started

I’m not GreatScott though

This is a "Come with me on my journey through fighting with this for twelve hours, so you only have to fight with this for 10 hours if you want to try it too."

Spoiler: We don’t sideload anything

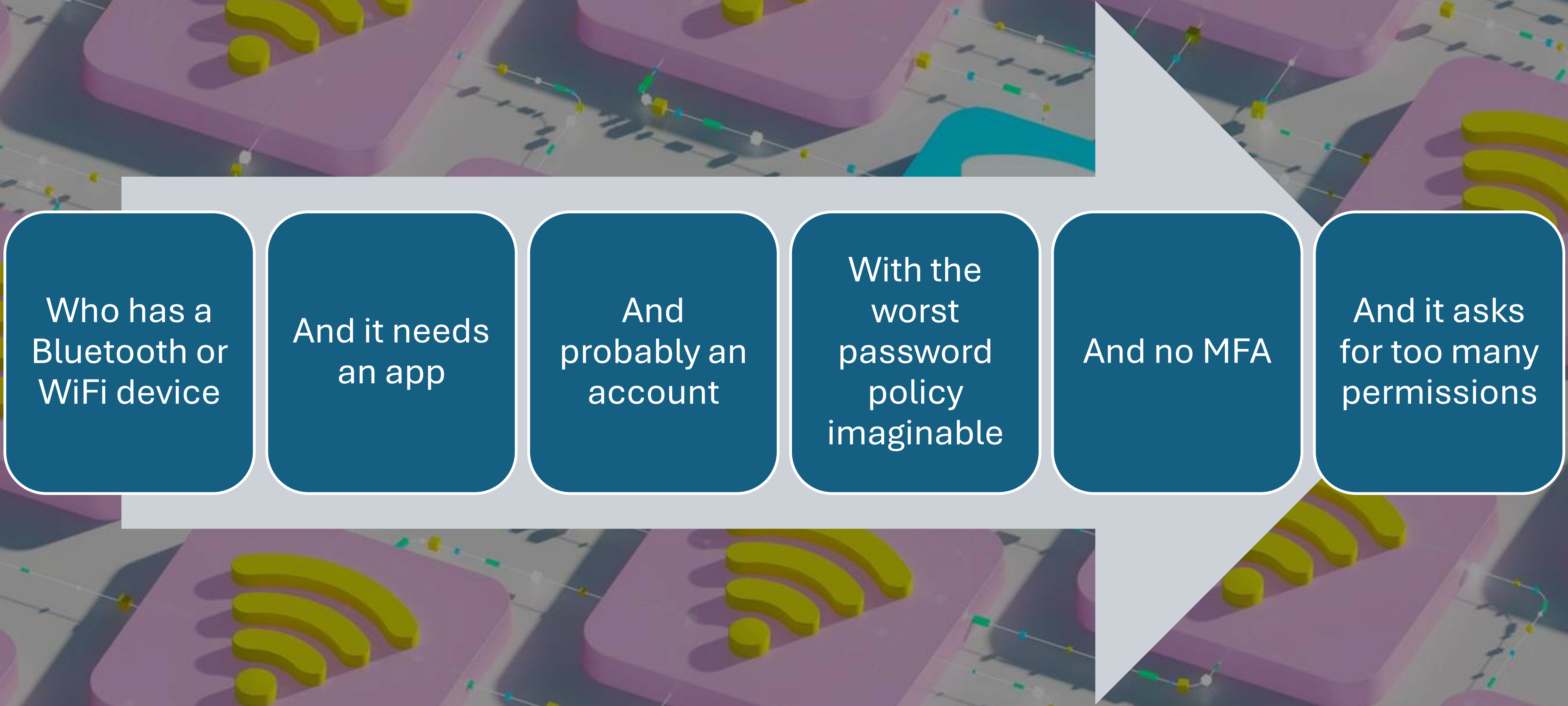
We do

But

Like

Not really

So what are we doing?



Who has a Bluetooth or WiFi device

And it needs an app

And probably an account

With the worst password policy imaginable

And no MFA

And it asks for too many permissions

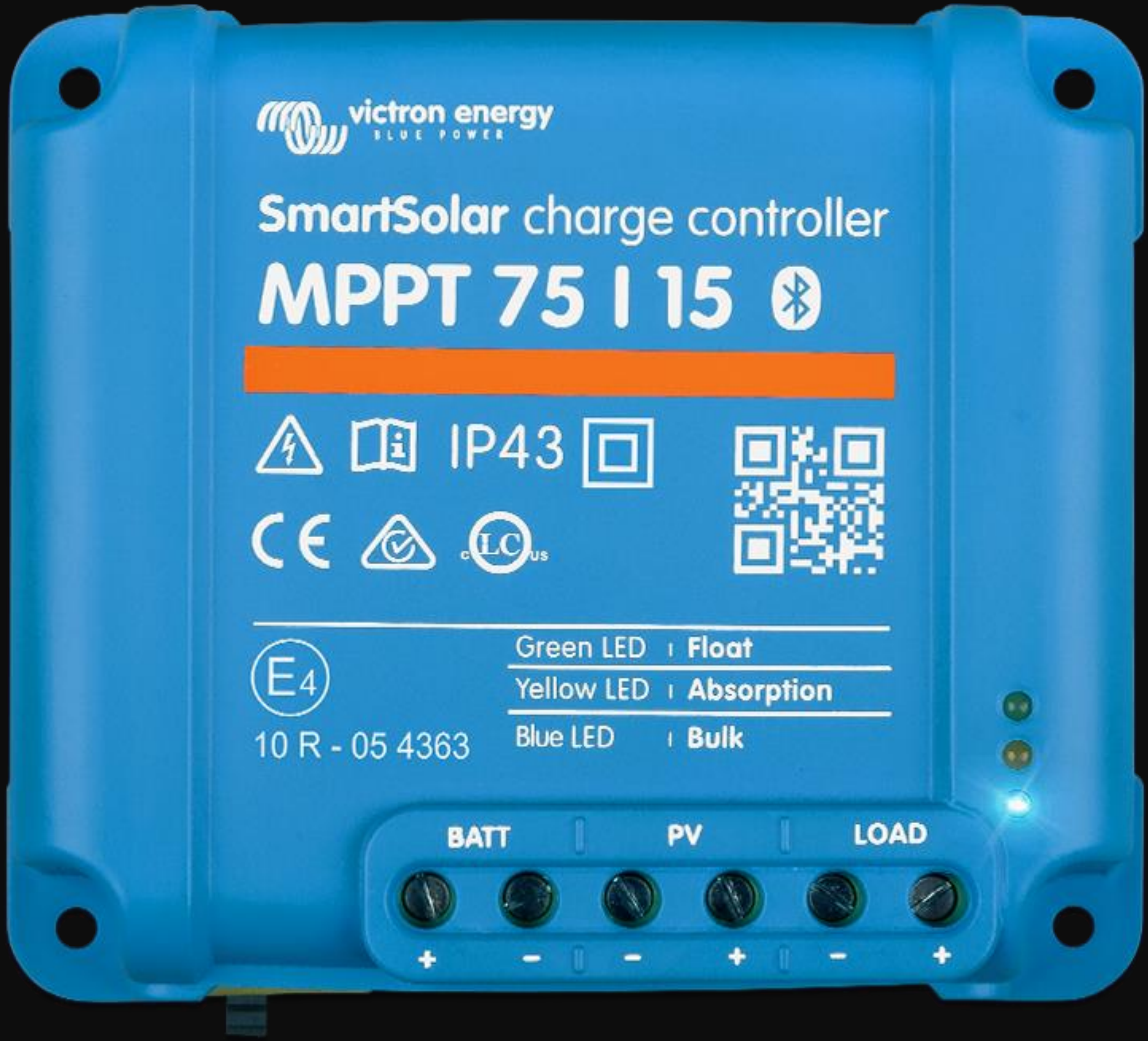
UD24

USB current/voltage/energy meter

The app comes from Mediafire

Victron solar charger

- Play store
- No account
- Minimal permissions
- But this is the only real company on the list



PUPVWMHB
LiFePO4
4.06V VolHigh
4.05V VolLow
4.05V AveVol
0.01V PressureDiff
0 Cycle-Index

Anova

- App from Play store
- Needs an account
- Wireless functions are paid for new subscribers
- Older models, despite still working are getting removed

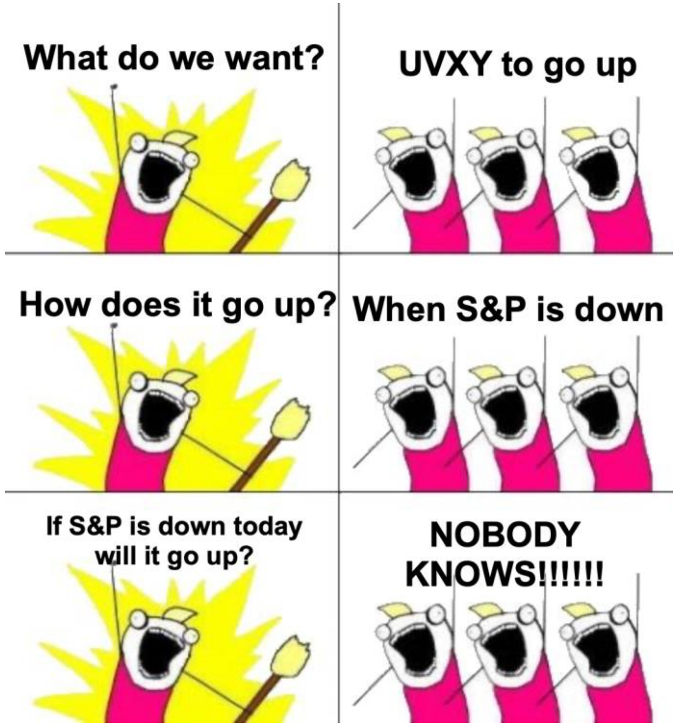
ANOVA
130.0
130.0
MAX
MIN
Anova
9:41 AM
100%
My Recipes
Your favorite time & temp at the tap of a button
Guides
Get the basics with our sous vide guides, by J. Kenji López-Alt
Recipes
Discover 1,000+ from award-winning chefs and writers
CURRENT
73.4°F
TIMER
00:00
SET
66.5°F
Start Cooking

“Personal Massager”

So what is a poor orphan boy to do

Reverse engineering!

- <That meme>



I was going to make this an actual slide, but honestly, this is better than anything I could make.

We’ve got two options

Decompile Android APKs

Sniff traffic from the apps from an Android device

And I suck at Java

“But Mike, how do you capture Bluetooth traffic off of an Android phone?!”

I’m glad you asked

Imagine if you could pipe packets from your phone to wireshark through stdout

Introducing extcap!

WIRESHARK/TSHARK SUPPORT EXTCAP

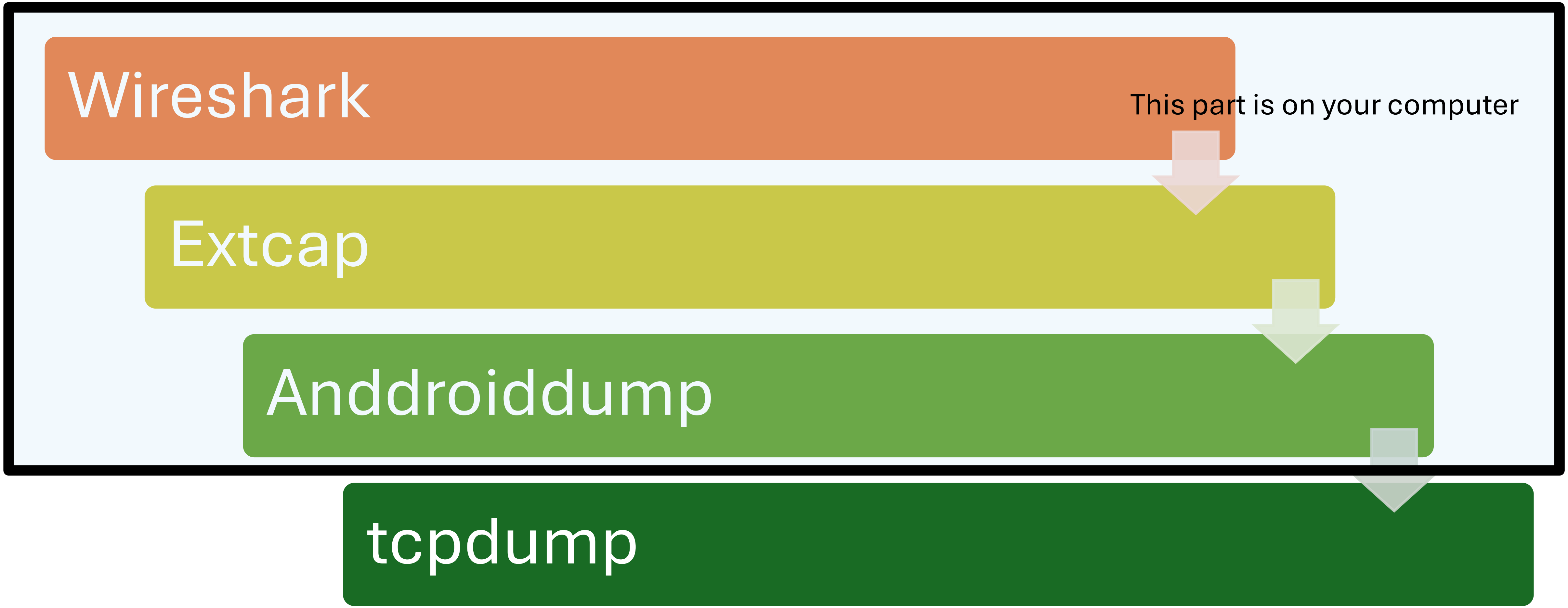
IT’S A WAY OF ADDING EXTERNAL CAPTURE AND LOG SOURCES

LIKE REMOTE-SSHDUMP

OR CISCODUMP

HTTPS://WWW.WIRESHARK.ORG/DOCS/WSDG_HTML_CHUNKED/CHCAPTUREEXTCAP.HTML

The simple example, capturing wifi



Wireshark

This part is on your computer

Extcap

Anddroiddump

tcpdump

Androiddump supported sources

Logcat Main (binary [<=Jelly Bean] or text)

Logcat System (binary [<=Jelly Bean] or text)

Logcat Events (binary [<=Jelly Bean] or text)

Logcat Radio (binary [<=Jelly Bean] or text)

Logcat Crash (text; from Lollipop)

Bluetooth Hcidump [<=Jelly Bean]

Bluetooth Bluedroid External Parser [Kitkat]

Bluetooth BtsnoopNet [>=Lollipop]

WiFi/Ethernet tcpdump [needs tcpdump on phone]

We only care about BtSnoopNet

- It’s the modern Bluetooth HCI capture mechanism
- Enabling it is a delicate ballet
- With snakes
- With arms

Capturing Bluetooth HCI on Android

- Disable Bluetooth
- Go to system settings
- Enable developer options
- Find Bluetooth HCI snooping toggle
- Turn it on
- Turn it off
- Turn Bluetooth on
- Turn Bluetooth off
- Turn HCI snooping on
- Turn Bluetooth on
- Turn Bluetooth off
- Turn Bluetooth on

Here’s how it works

TURNING SNOOPING OFF RESETS THE CAPTURE STATE

TURNING SNOOPING ON STARTS CAPTURE

TURNING BLUETOOTH ON MAKES THE SETTING TAKE EFFECT

So how does Wireshark + BTSnoopNet work?

BTSnoopNet writes packets to a file in the system tree

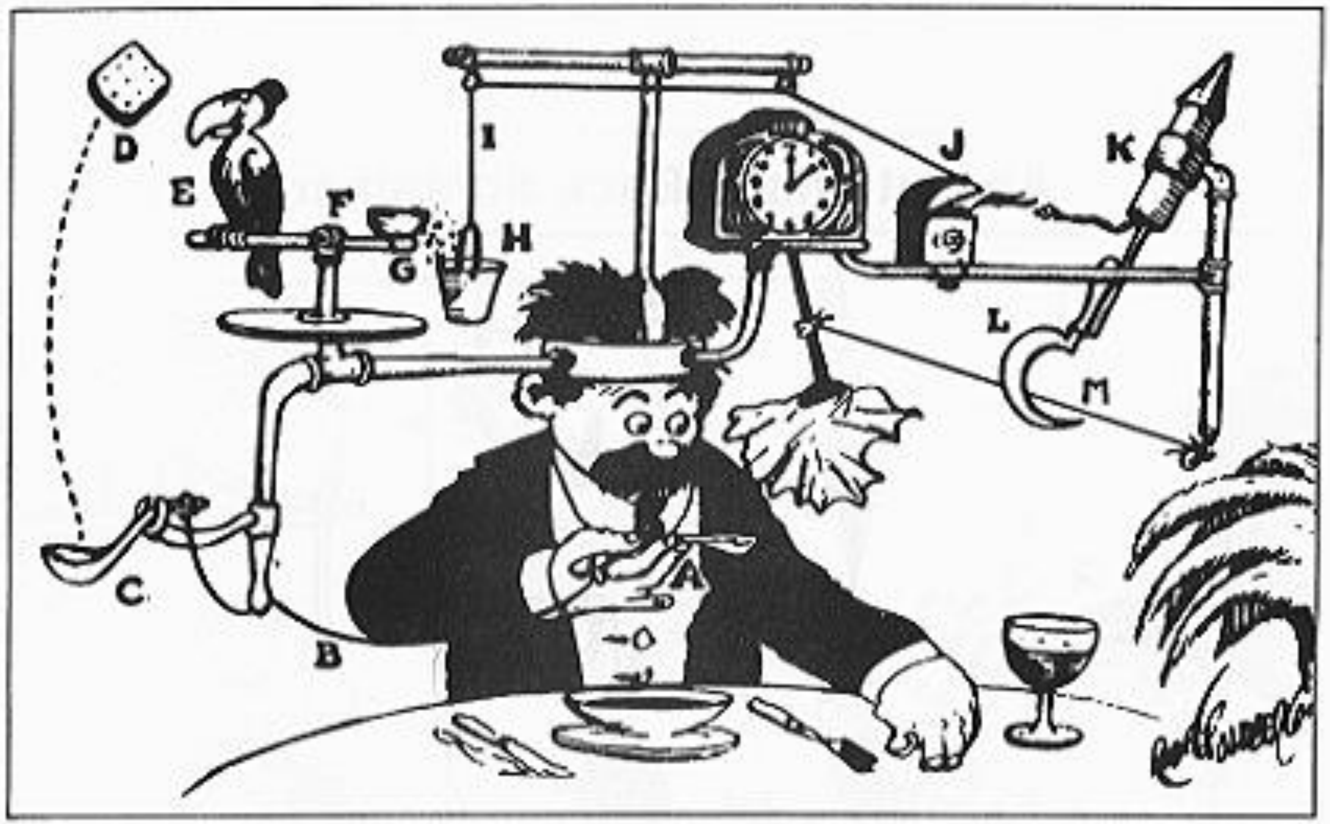
And
uses
t

Wireshark invokes extcap with a host and port

Extca
to th

realtime

Self-Operating Napkin



What does BtSnoopNet do?

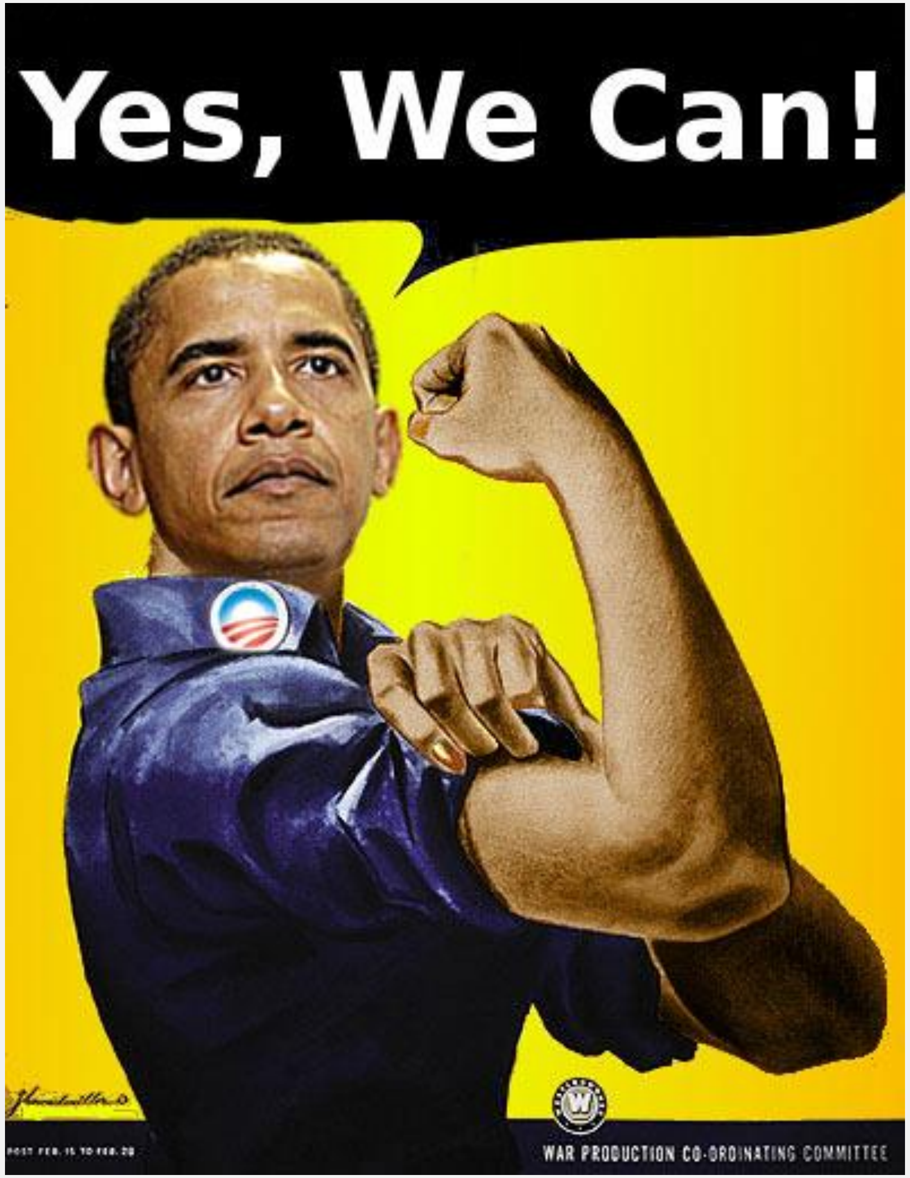
IT WRITES TO A FILE IN A ROOT-ACCESSIBLE LOCATION.

SO THAT MEANS…

TO USE IT YOU NEED ROOT!

Let’s pretend we don’t have a rooted device

- Can we still do this?



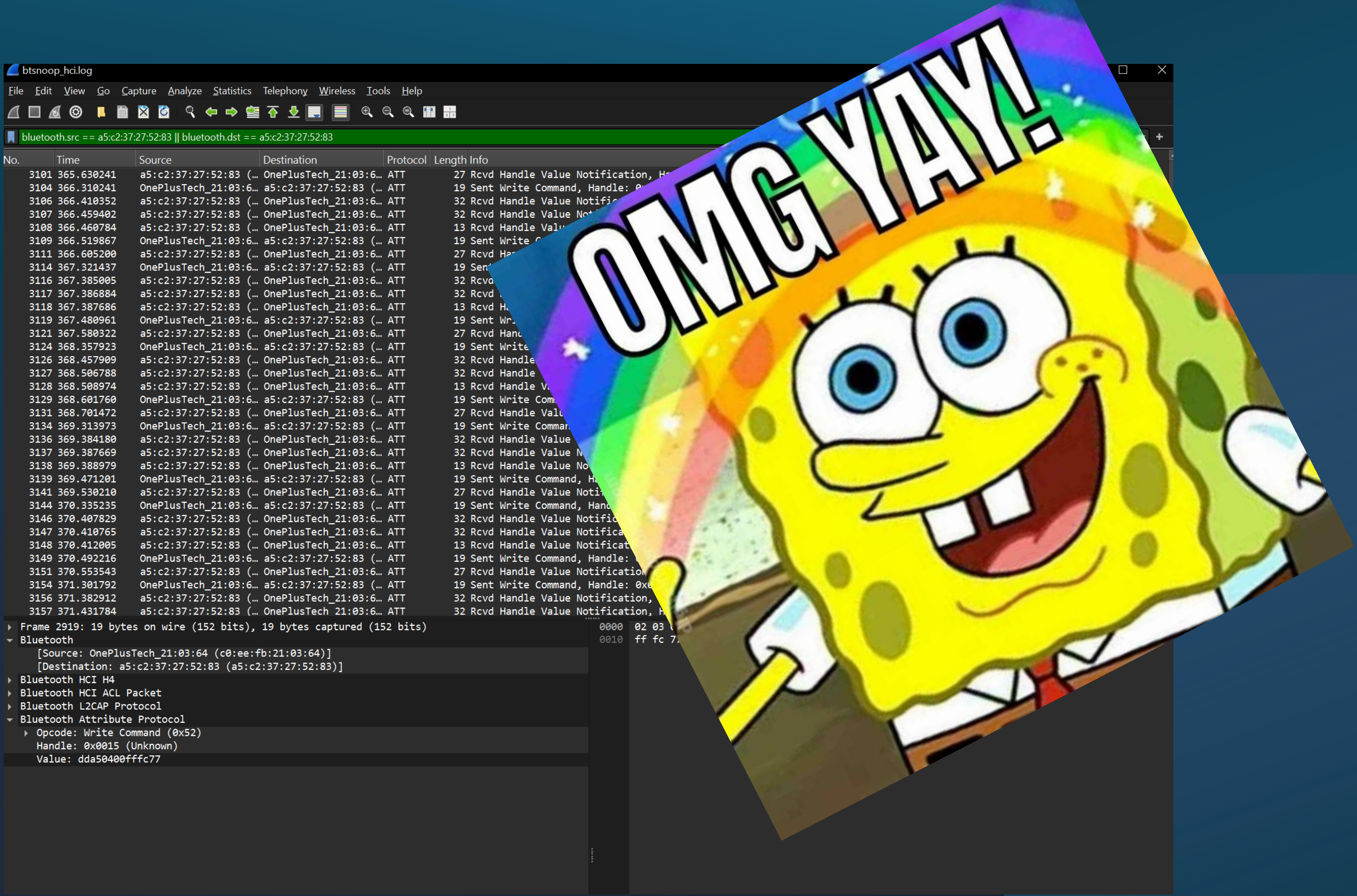
How do we get the logs off?

- Declare a bug report!
- Bug reports are non-privileged actions
- But will contain HCI snooping logs if enabled
- But it isn’t realtime
- So correlating events is a lot harder
- Timestamps are your friend? 🤷‍♂️

But we’ve got them!

UNPACK THEM FROM THE BUGREPORT ZIP

DRAG AND DROP INTO WIRESHARK!



btsnoop_hci.log

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

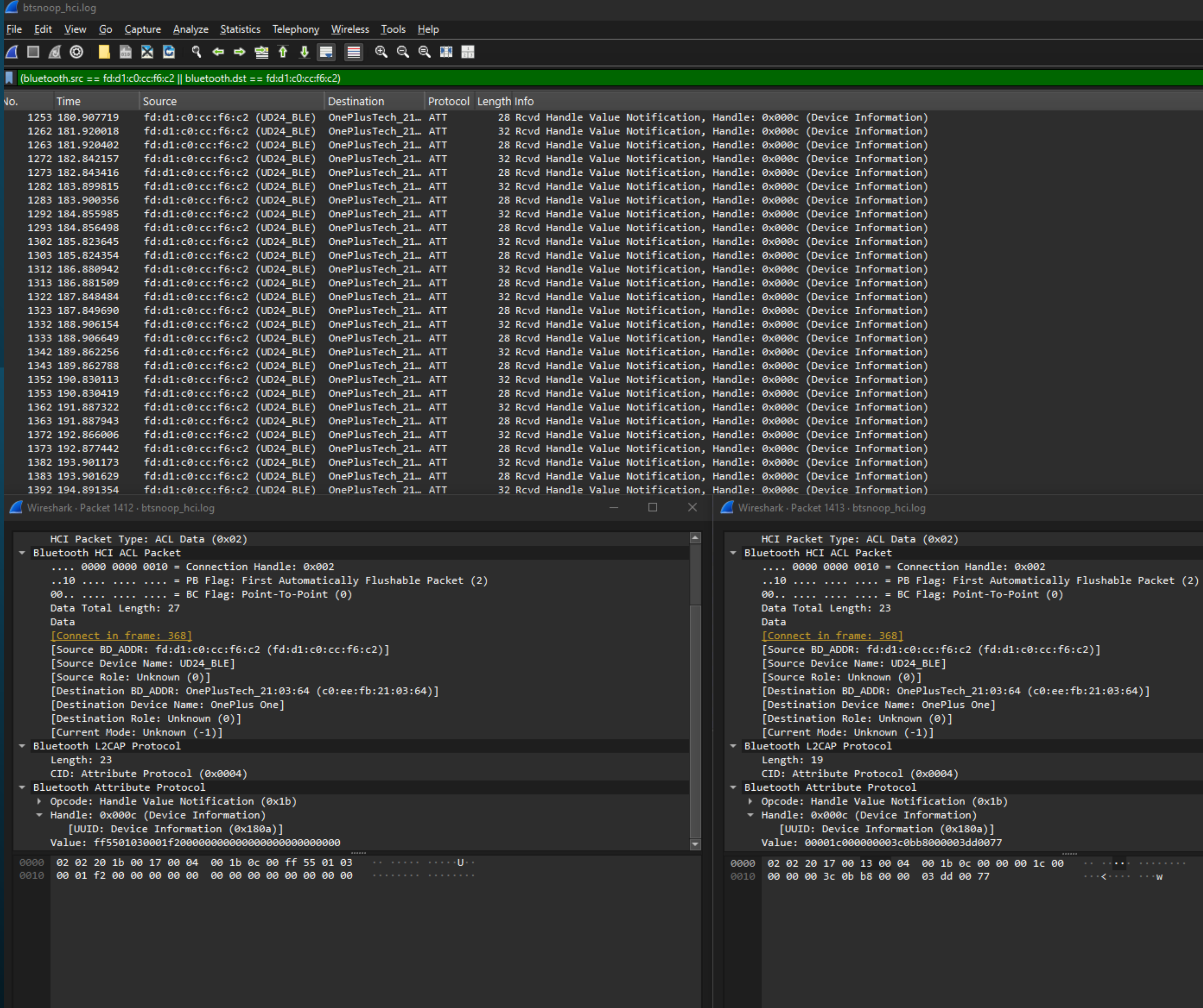
bluetooth.src == a5:c2:37:27:52:83 || bluetooth.dst == a5:c2:37:27:52:83

```
No.   Time        Source                Destination           Protocol Length Info
 3101 365.630241  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      27 Rcvd Handle Value Notification, Ha
 3104 366.310241  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write Command, Handle: 0
 3106 366.410352  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle Value Notific
 3107 366.459402  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle Value Not
 3108 366.460784  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      13 Rcvd Handle Valu
 3109 366.519867  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write C
 3111 366.605200  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      27 Rcvd Ha
 3114 367.321437  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sen
 3116 367.385005  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd
 3117 367.386884  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd
 3118 367.387686  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      13 Rcvd H
 3119 367.480961  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Wr
 3121 367.580322  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      27 Rcvd Hand
 3124 368.357923  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write
 3126 368.457909  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle
 3127 368.506788  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle
 3128 368.508974  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      13 Rcvd Handle V
 3129 368.601760  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write Comm
 3131 368.701472  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      27 Rcvd Handle Valu
 3134 369.313973  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write Comman
 3136 369.384180  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle Value
 3137 369.387669  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle Value N
 3138 369.388979  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      13 Rcvd Handle Value No
 3139 369.471201  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write Command, H
 3141 369.530210  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      27 Rcvd Handle Value Notif
 3144 370.335235  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write Command, Hand
 3146 370.407829  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle Value Notific
 3147 370.410765  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle Value Notifica
 3148 370.412005  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      13 Rcvd Handle Value Notificat
 3149 370.492216  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write Command, Handle:
 3151 370.553543  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      27 Rcvd Handle Value Notificatio
 3154 371.301792  OnePlusTech_21:03:6… a5:c2:37:27:52:83 (… ATT      19 Sent Write Command, Handle: 0xe
 3156 371.382912  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle Value Notification,
 3157 371.431784  a5:c2:37:27:52:83 (… OnePlusTech_21:03:6… ATT      32 Rcvd Handle Value Notification, H
```

```
> Frame 2919: 19 bytes on wire (152 bits), 19 bytes captured (152 bits)
v Bluetooth
    [Source: OnePlusTech_21:03:64 (c0:ee:fb:21:03:64)]
    [Destination: a5:c2:37:27:52:83 (a5:c2:37:27:52:83)]
> Bluetooth HCI H4
> Bluetooth HCI ACL Packet
> Bluetooth L2CAP Protocol
v Bluetooth Attribute Protocol
  > Opcode: Write Command (0x52)
    Handle: 0x0015 (Unknown)
    Value: dda50400fffc77
```

```
0000  02 03
0010  ff fc 7
```

Let’s start with a USB current meter



```
(bluetooth.src == fd:d1:c0:cc:f6:c2 || bluetooth.dst == fd:d1:c0:cc:f6:c2)
```

```
No.  Time       Source                        Destination       Protocol Length Info
1253 180.907719 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1262 181.920018 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1263 181.920402 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1272 182.842157 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1273 182.843416 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1282 183.899815 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1283 183.900356 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1292 184.855985 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1293 184.856498 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1302 185.823645 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1303 185.824354 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1312 186.880942 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1313 186.881509 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1322 187.848484 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1323 187.849690 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1332 188.906154 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1333 188.906649 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1342 189.862256 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1343 189.862788 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1352 190.830113 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1353 190.830419 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1362 191.887322 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1363 191.887943 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1372 192.866006 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1373 192.877442 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1382 193.901173 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1383 193.901629 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          28 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
1392 194.891354 fd:d1:c0:cc:f6:c2 (UD24_BLE)  OnePlusTech_21… ATT          32 Rcvd Handle Value Notification, Handle: 0x000c (Device Information)
```

Wireshark · Packet 1412 · btsnoop_hci.log

```
  HCI Packet Type: ACL Data (0x02)
▾ Bluetooth HCI ACL Packet
  .... 0000 0000 0010 = Connection Handle: 0x002
  ..10 .... .... .... = PB Flag: First Automatically Flushable Packet (2)
  00.. .... .... .... = BC Flag: Point-To-Point (0)
  Data Total Length: 27
  Data
  [Connect in frame: 368]
  [Source BD_ADDR: fd:d1:c0:cc:f6:c2 (fd:d1:c0:cc:f6:c2)]
  [Source Device Name: UD24_BLE]
  [Source Role: Unknown (0)]
  [Destination BD_ADDR: OnePlusTech_21:03:64 (c0:ee:fb:21:03:64)]
  [Destination Device Name: OnePlus One]
  [Destination Role: Unknown (0)]
  [Current Mode: Unknown (-1)]
▾ Bluetooth L2CAP Protocol
  Length: 23
  CID: Attribute Protocol (0x0004)
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Handle Value Notification (0x1b)
  ▾ Handle: 0x000c (Device Information)
      [UUID: Device Information (0x180a)]
    Value: ff5501030001f20000000000000000000000000000

0000  02 02 20 1b 00 17 00 04  00 1b 0c 00 ff 55 01 03   ·· ······ ·····U··
0010  00 01 f2 00 00 00 00 00  00 00 00 00 00 00 00 00   ········ ········
```

Wireshark · Packet 1413 · btsnoop_hci.log

```
  HCI Packet Type: ACL Data (0x02)
▾ Bluetooth HCI ACL Packet
  .... 0000 0000 0010 = Connection Handle: 0x002
  ..10 .... .... .... = PB Flag: First Automatically Flushable Packet (2)
  00.. .... .... .... = BC Flag: Point-To-Point (0)
  Data Total Length: 23
  Data
  [Connect in frame: 368]
  [Source BD_ADDR: fd:d1:c0:cc:f6:c2 (fd:d1:c0:cc:f6:c2)]
  [Source Device Name: UD24_BLE]
  [Source Role: Unknown (0)]
  [Destination BD_ADDR: OnePlusTech_21:03:64 (c0:ee:fb:21:03:64)]
  [Destination Device Name: OnePlus One]
  [Destination Role: Unknown (0)]
  [Current Mode: Unknown (-1)]
▾ Bluetooth L2CAP Protocol
  Length: 19
  CID: Attribute Protocol (0x0004)
▾ Bluetooth Attribute Protocol
  ▸ Opcode: Handle Value Notification (0x1b)
  ▾ Handle: 0x000c (Device Information)
      [UUID: Device Information (0x180a)]
    Value: 00001c000000003c0bb8000003dd0077

0000  02 02 20 17 00 13 00 04  00 1b 0c 00 00 00 1c 00   ·· ····· ·········
0010  00 00 00 3c 0b b8 00 00  03 dd 00 77               ···<···· ···w
```

What do we get

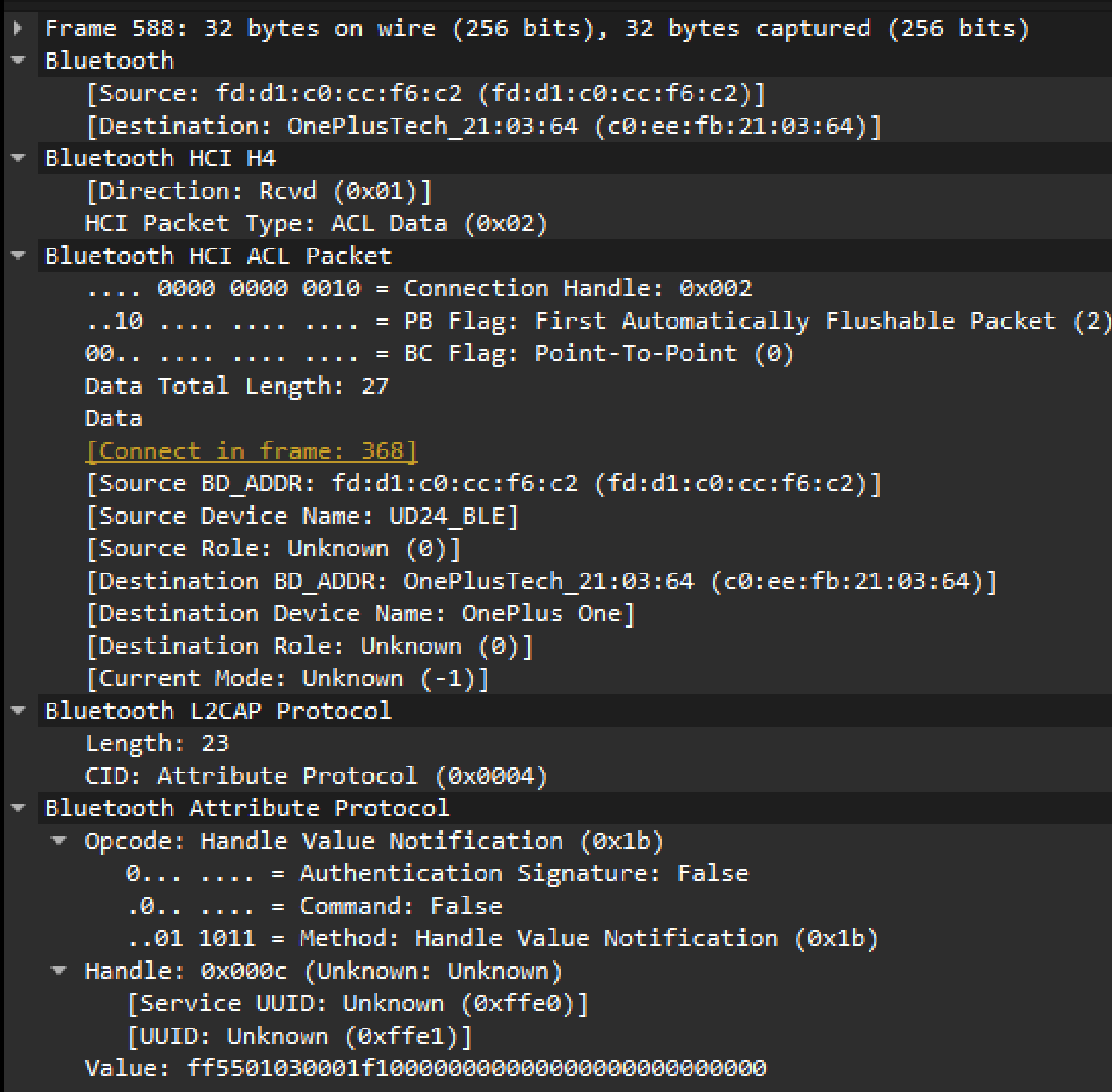
- We get full Bluetooth LE stack
- Full GATT protocol details
- MACs and direction
- Payloads
- Timestamps

Ok, now for a quick quide on EBNF

BLE

So what are we seeing

- Attribute handle new-value notifications
- Two PDUs, one with 20 bytes, one with 16
- ff5501030001f2000000000000000000000000000000001c000000003c0bb8000003dd001f



```
▸ Frame 588: 32 bytes on wire (256 bits), 32 bytes captured (256 bits)
▾ Bluetooth
      [Source: fd:d1:c0:cc:f6:c2 (fd:d1:c0:cc:f6:c2)]
      [Destination: OnePlusTech_21:03:64 (c0:ee:fb:21:03:64)]
▾ Bluetooth HCI H4
      [Direction: Rcvd (0x01)]
      HCI Packet Type: ACL Data (0x02)
▾ Bluetooth HCI ACL Packet
      .... 0000 0000 0010 = Connection Handle: 0x002
      ..10 .... .... .... = PB Flag: First Automatically Flushable Packet (2)
      00.. .... .... .... = BC Flag: Point-To-Point (0)
      Data Total Length: 27
      Data
      [Connect in frame: 368]
      [Source BD_ADDR: fd:d1:c0:cc:f6:c2 (fd:d1:c0:cc:f6:c2)]
      [Source Device Name: UD24_BLE]
      [Source Role: Unknown (0)]
      [Destination BD_ADDR: OnePlusTech_21:03:64 (c0:ee:fb:21:03:64)]
      [Destination Device Name: OnePlus One]
      [Destination Role: Unknown (0)]
      [Current Mode: Unknown (-1)]
▾ Bluetooth L2CAP Protocol
      Length: 23
      CID: Attribute Protocol (0x0004)
▾ Bluetooth Attribute Protocol
    ▾ Opcode: Handle Value Notification (0x1b)
          0... .... = Authentication Signature: False
          .0.. .... = Command: False
          ..01 1011 = Method: Handle Value Notification (0x1b)
    ▾ Handle: 0x000c (Unknown: Unknown)
          [Service UUID: Unknown (0xffe0)]
          [UUID: Unknown (0xffe1)]
      Value: ff5501030001f100000000000000000000000000
```

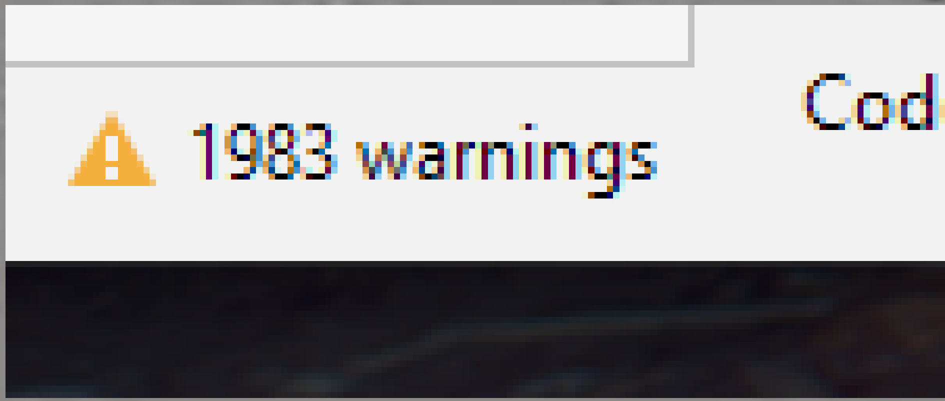
That’s a pile of garbage

So let’s crack open the APK

I don't know what I'm doing, so

JADX was the tool I grabbed

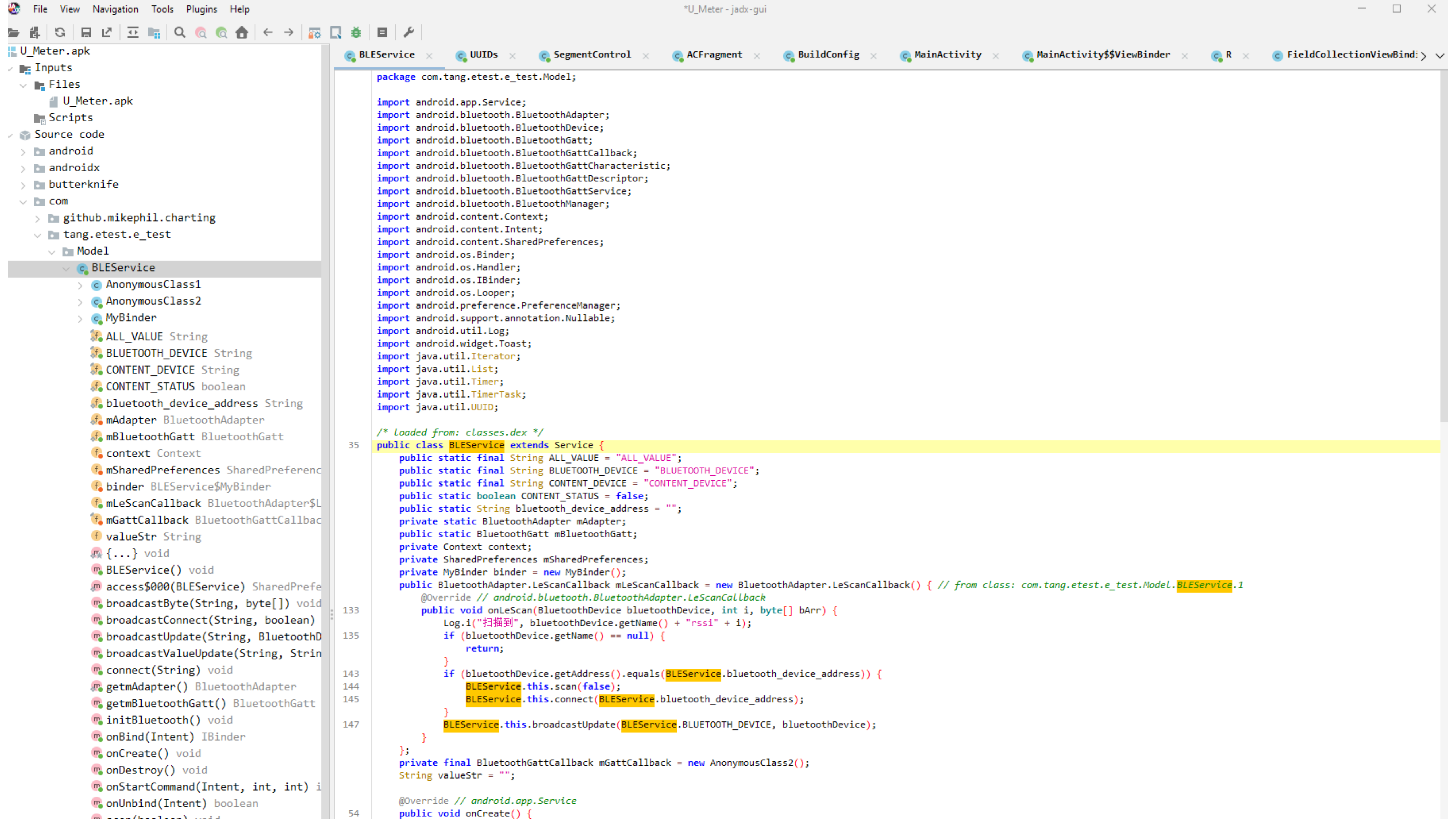
Opened up the APK



⚠ 1983 warnings

Cod

We could be off to a worse start

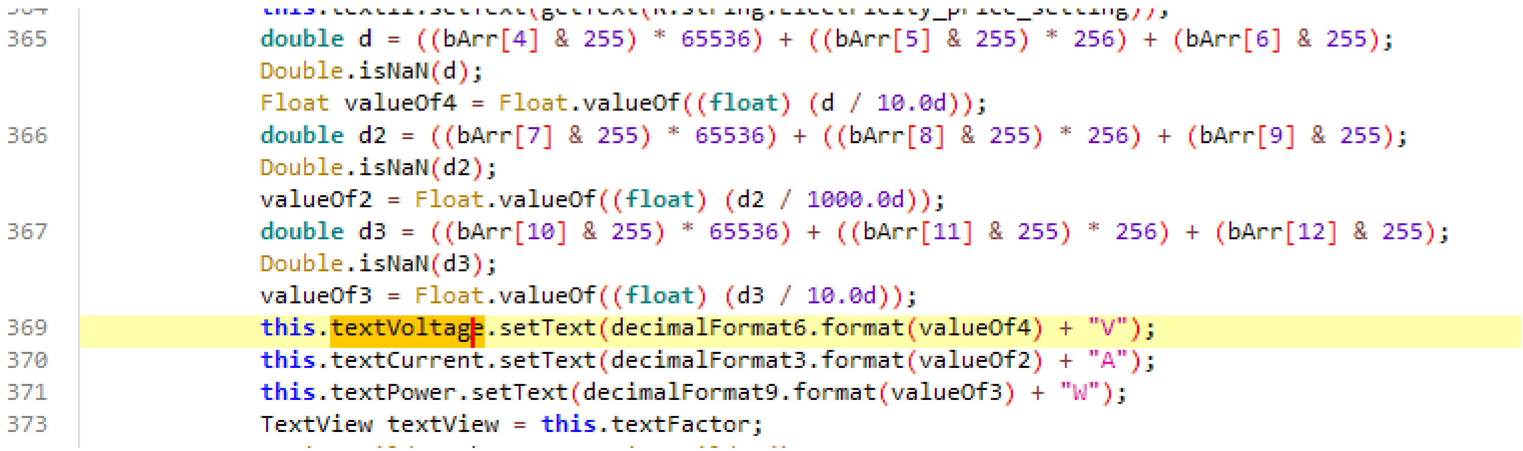


```
U_Meter.apk
  Inputs
    Files
      U_Meter.apk
    Scripts
  Source code
    android
    androidx
    butterknife
    com
      github.mikephil.charting
      tang.etest.e_test
        Model
          BLEService
            AnonymousClass1
            AnonymousClass2
            MyBinder
            ALL_VALUE String
            BLUETOOTH_DEVICE String
            CONTENT_DEVICE String
            CONTENT_STATUS boolean
            bluetooth_device_address String
            mAdapter BluetoothAdapter
            mBluetoothGatt BluetoothGatt
            context Context
            mSharedPreferences SharedPreferenc
            binder BLEService$MyBinder
            mLeScanCallback BluetoothAdapter$L
            mGattCallback BluetoothGattCallbac
            valueStr String
            {...} void
            BLEService() void
            access$000(BLEService) SharedPrefe
            broadcastByte(String, byte[]) void
            broadcastConnect(String, boolean)
            broadcastUpdate(String, BluetoothD
            broadcastValueUpdate(String, Strin
            connect(String) void
            getmAdapter() BluetoothAdapter
            getmBluetoothGatt() BluetoothGatt
            initBluetooth() void
            onBind(Intent) IBinder
            onCreate() void
            onDestroy() void
            onStartCommand(Intent, int, int) i
            onUnbind(Intent) boolean
```

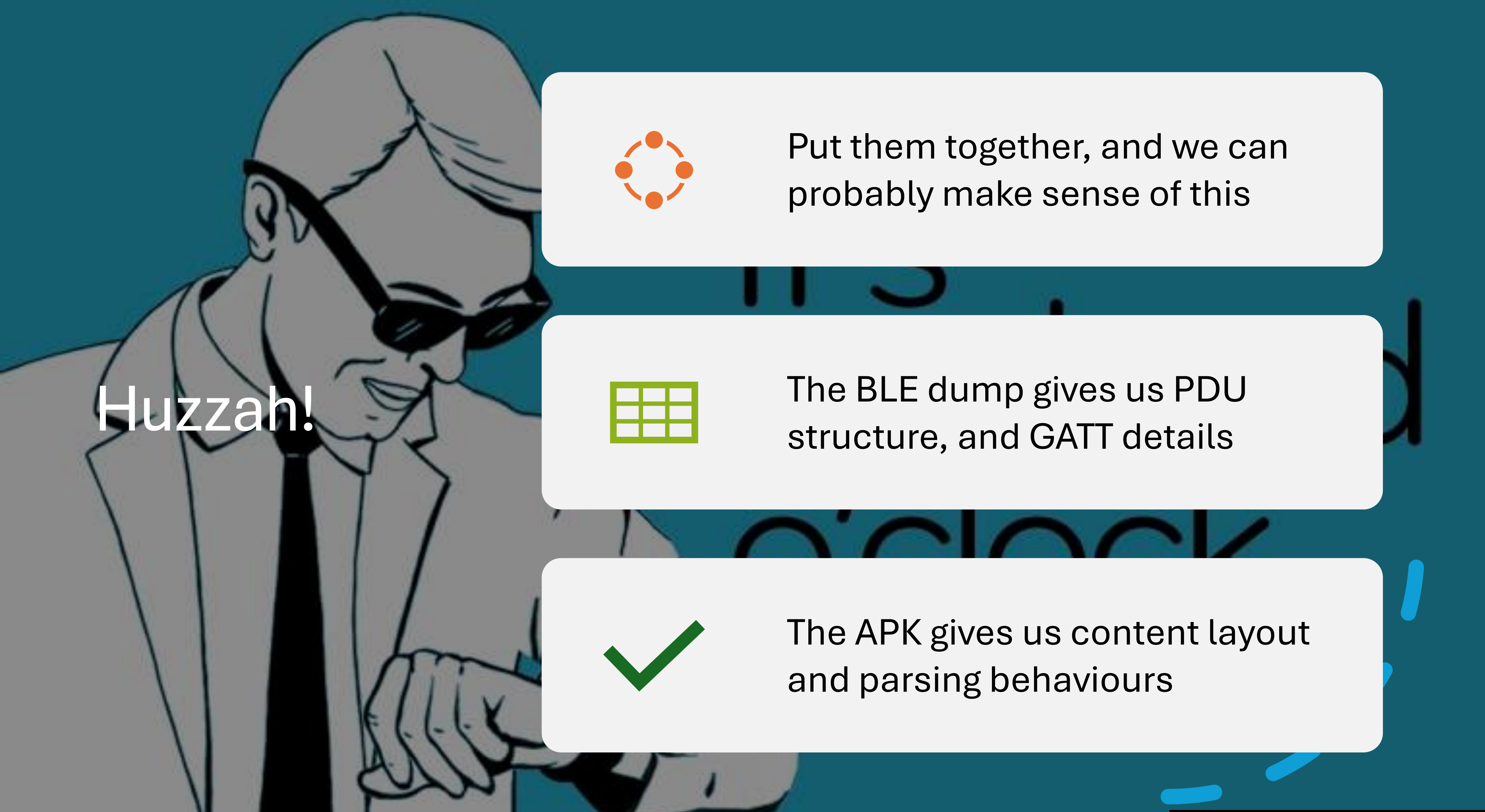
```java
package com.tang.etest.e_test.Model;

import android.app.Service;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.bluetooth.BluetoothGatt;
import android.bluetooth.BluetoothGattCallback;
import android.bluetooth.BluetoothGattCharacteristic;
import android.bluetooth.BluetoothGattDescriptor;
import android.bluetooth.BluetoothGattService;
import android.bluetooth.BluetoothManager;
import android.content.Context;
import android.content.Intent;
import android.content.SharedPreferences;
import android.os.Binder;
import android.os.Handler;
import android.os.IBinder;
import android.os.Looper;
import android.preference.PreferenceManager;
import android.support.annotation.Nullable;
import android.util.Log;
import android.widget.Toast;
import java.util.Iterator;
import java.util.List;
import java.util.Timer;
import java.util.TimerTask;
import java.util.UUID;

/* loaded from: classes.dex */
public class BLEService extends Service {
    public static final String ALL_VALUE = "ALL_VALUE";
    public static final String BLUETOOTH_DEVICE = "BLUETOOTH_DEVICE";
    public static final String CONTENT_DEVICE = "CONTENT_DEVICE";
    public static boolean CONTENT_STATUS = false;
    public static String bluetooth_device_address = "";
    private static BluetoothAdapter mAdapter;
    public static BluetoothGatt mBluetoothGatt;
    private Context context;
    private SharedPreferences mSharedPreferences;
    private MyBinder binder = new MyBinder();
    public BluetoothAdapter.LeScanCallback mLeScanCallback = new BluetoothAdapter.LeScanCallback() { // from class: com.tang.etest.e_test.Model.BLEService.1
        @Override // android.bluetooth.BluetoothAdapter.LeScanCallback
        public void onLeScan(BluetoothDevice bluetoothDevice, int i, byte[] bArr) {
            Log.i("扫描到", bluetoothDevice.getName() + "rssi" + i);
            if (bluetoothDevice.getName() == null) {
                return;
            }
            if (bluetoothDevice.getAddress().equals(BLEService.bluetooth_device_address)) {
                BLEService.this.scan(false);
                BLEService.this.connect(BLEService.bluetooth_device_address);
            }
            BLEService.this.broadcastUpdate(BLEService.BLUETOOTH_DEVICE, bluetoothDevice);
        }
    };
    private final BluetoothGattCallback mGattCallback = new AnonymousClass2();
    String valueStr = "";

    @Override // android.app.Service
    public void onCreate() {
```

This seems like a good start!

- 2 hours of clicking around like a dumdum later…

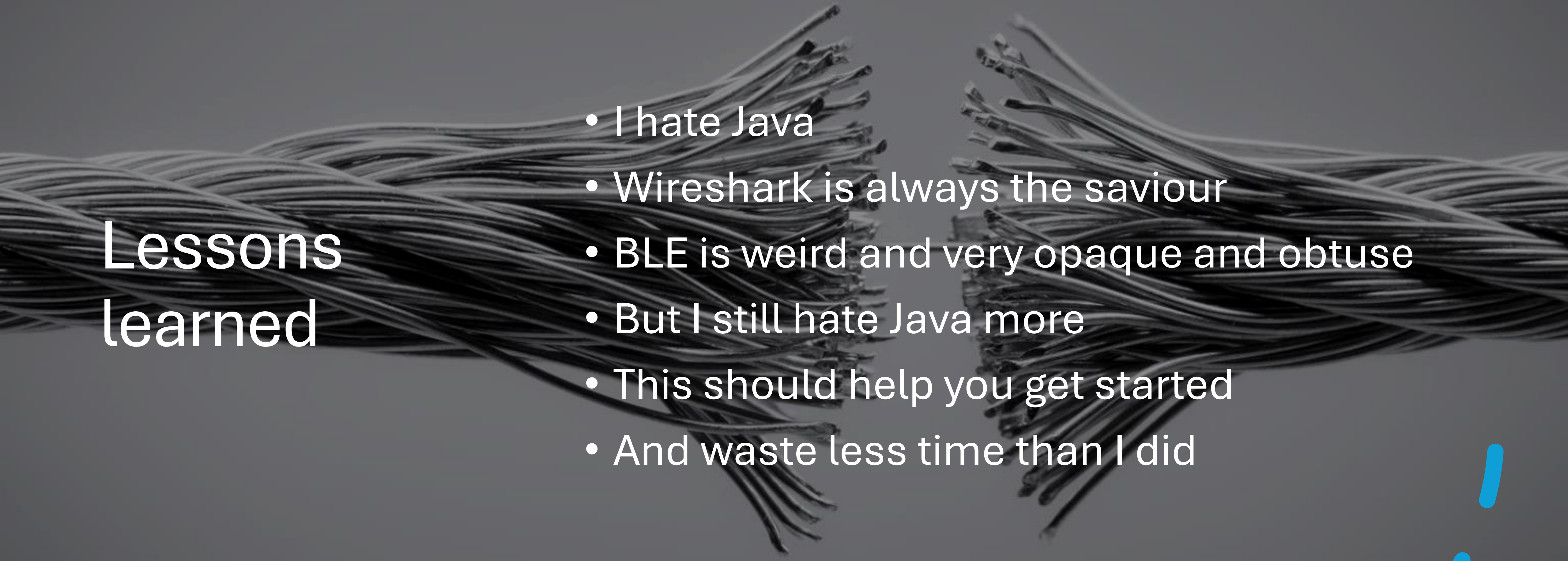


```
this.textTI.setText(getText(R.string.electricity_price_setting));
double d = ((bArr[4] & 255) * 65536) + ((bArr[5] & 255) * 256) + (bArr[6] & 255);
Double.isNaN(d);
Float valueOf4 = Float.valueOf((float) (d / 10.0d));
double d2 = ((bArr[7] & 255) * 65536) + ((bArr[8] & 255) * 256) + (bArr[9] & 255);
Double.isNaN(d2);
valueOf2 = Float.valueOf((float) (d2 / 1000.0d));
double d3 = ((bArr[10] & 255) * 65536) + ((bArr[11] & 255) * 256) + (bArr[12] & 255);
Double.isNaN(d3);
valueOf3 = Float.valueOf((float) (d3 / 10.0d));
this.textVoltage.setText(decimalFormat6.format(valueOf4) + "V");
this.textCurrent.setText(decimalFormat3.format(valueOf2) + "A");
this.textPower.setText(decimalFormat9.format(valueOf3) + "W");
TextView textView = this.textFactor;
```



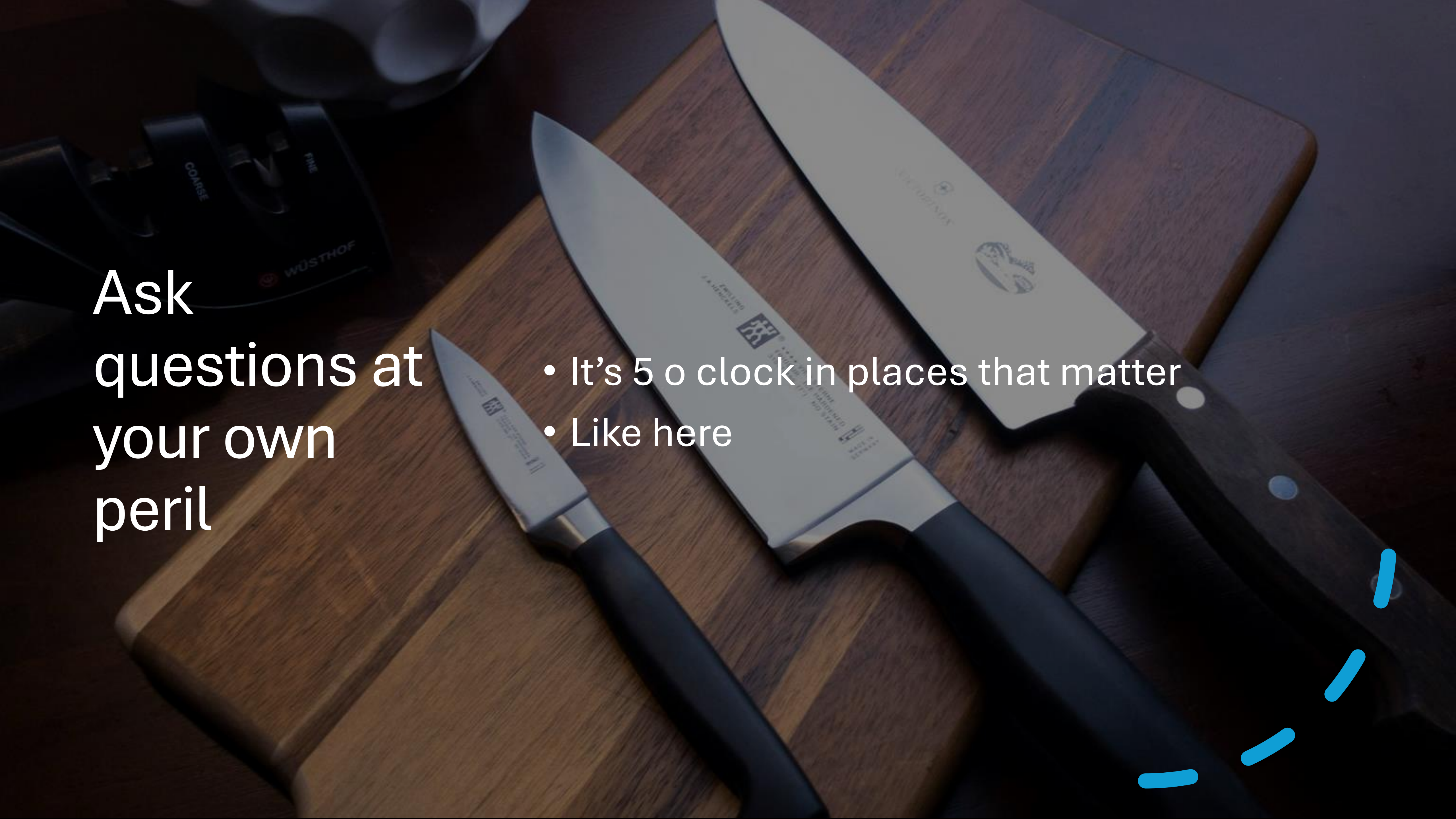
Huzzah!

- Put them together, and we can probably make sense of this
- The BLE dump gives us PDU structure, and GATT details
- The APK gives us content layout and parsing behaviours



Lessons learned

- I hate Java
- Wireshark is always the saviour
- BLE is weird and very opaque and obtuse
- But I still hate Java more
- This should help you get started
- And waste less time than I did



Ask questions at your own peril

- It’s 5 o clock in places that matter
- Like here