Hacking your Smart Coffee Machine

Axelle Apvrille

BSides Munich, August 2020

Background story

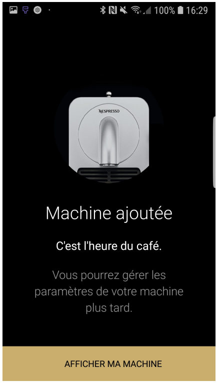
"We need a visual challenge which involves a real/ known/ off-the-shelf smart device"

Smart Coffee Machine

$\longleftarrow$

Bluetooth
Low Energy

No WiFi, no Ethernet...

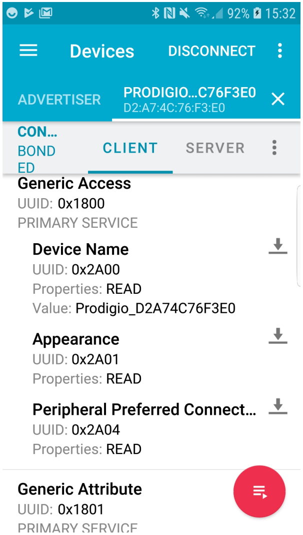


Machine ajoutée

C'est l'heure du café.

Vous pourrez gérer les paramètres de votre machine plus tard.

BLE enumeration



Generic Access
UUID: 0x1800
PRIMARY SERVICE

Device Name
UUID: 0x2A00
Properties: READ
Value: Prodigio_D2A74C76F3E0

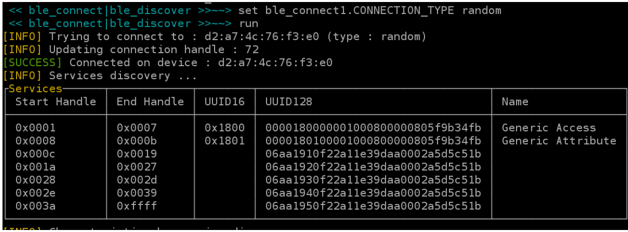
Appearance
UUID: 0x2A01
Properties: READ

Peripheral Preferred Connect...
UUID: 0x2A04
Properties: READ

Generic Attribute
UUID: 0x1801
PRIMARY SERVICE

nRF Connect

Mirage



```
 << ble_connect|ble_discover >>~~> set ble_connect1.CONNECTION_TYPE random
 << ble_connect|ble_discover >>~~> run
[INFO] Trying to connect to : d2:a7:4c:76:f3:e0 (type : random)
[INFO] Updating connection handle : 72
[SUCCESS] Connected on device : d2:a7:4c:76:f3:e0
[INFO] Services discovery ...
┌Services──────────────────────────────────────────────────────────────────────────────────────────────────┐
│ Start Handle │ End Handle │ UUID16 │ UUID128                          │ Name             │
├──────────────┼────────────┼────────┼──────────────────────────────────┼──────────────────┤
│ 0x0001       │ 0x0007     │ 0x1800 │ 0000180000001000800000805f9b34fb │ Generic Access    │
│ 0x0008       │ 0x000b     │ 0x1801 │ 0000180100001000800000805f9b34fb │ Generic Attribute │
│ 0x000c       │ 0x0019     │        │ 06aa1910f22a11e39daa0002a5d5c51b │                   │
│ 0x001a       │ 0x0027     │        │ 06aa1920f22a11e39daa0002a5d5c51b │                   │
│ 0x0028       │ 0x002d     │        │ 06aa1930f22a11e39daa0002a5d5c51b │                   │
│ 0x002e       │ 0x0039     │        │ 06aa1940f22a11e39daa0002a5d5c51b │                   │
│ 0x003a       │ 0xffff     │        │ 06aa1950f22a11e39daa0002a5d5c51b │                   │
└──────────────┴────────────┴────────┴──────────────────────────────────┴──────────────────┘
```

We need authentication

```
sudo gatttool -b d2:a7:4c:76:f3:e0 -I -t random
[d2:a7:4c:76:f3:e0][LE]> connect
Attempting to connect to d2:a7:4c:76:f3:e0
Connection successful
[d2:a7:4c:76:f3:e0][LE]> char-read-hnd 0x000e
Error: Characteristic value/descriptor read failed:
↪   Attribute requires authentication before
↪   read/write
```

How to request Security Mode 1, Level 3

- Level 1. No encryption.
- Level 2. Unauthenticated encryption.
- Level 3. Authenticated encryption

see https://www.oreilly.com/library/view/getting-started-with/9781491900550/ch04.html

gatttool (deprecated)

```
gatttool -b d2:a7:4c:76:f3:e0 -I -t random
                   --sec-level=high
```

bluetoothctl

```
[bluetooth]# connect D2:A7:4C:76:F3:E0
[Prodigio_D2A74C76F3E0]# pair D2:A7:4C:76:F3:E0
Attempting to pair with D2:A7:4C:76:F3:E0
[CHG] Device D2:A7:4C:76:F3:E0 Paired: yes
Pairing successful
```

Mirage (July 2019)

In `.mirage/mirage.cfg`:

```
[ble_connect]
TARGET=d2:a7:4c:76:f3:e0
CONNECTION_TYPE=random

[ble_master]
TARGET=d2:a7:4c:76:f3:e0
CONNECTION_TYPE=random

[ble_pair]
IRK=112233445566778899aabbccddeeff
BONDING=yes
LTK=112233445566778899aabbccddeeff
CSRK=112233445566778899aabbccddeeff
DISPLAY=yes
KEYBOARD=yes
YESNO=no
SECURE_CONNECTIONS=no
CT2=no
MITM=yes
```

Unable to pair though packets are okay...

```
load ble_connect|ble_pair
...
[SUCCESS] Connected on device :
↪  d2:a7:4c:76:f3:e0
...
[SUCCESS] Pairing Method selected :
↪  JustWorks
...
[INFO] Sending CSRK...
[SUCCESS] Sent !
[FAIL] Pairing Failed received : <<
↪  BLE - Pairing Failed Packet |
↪  reason=8 >>
[FAIL] Reason : Unspecified reason
[FAIL] Execution of module ble_pair2
↪  failed !
```

We need authorization

```
[d2:a7:4c:76:f3:e0][LE]> char-read-hnd 0x001c
Error: Characteristic value/descriptor read failed: Attribute requires
↪  authorization before read/write
```

Authentication is not Authorization

Authentication

- *Act of proving an assertion, e.g. identity of a user/computer*
- Done during pairing

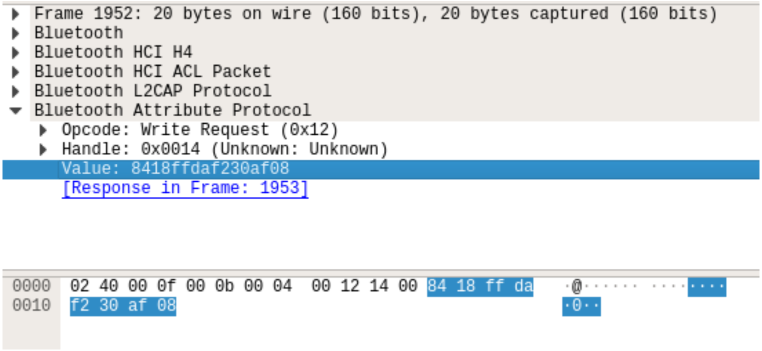
Authorization

- *"Is device X allowed to access/use service Y?"*
- Rare

How do I get authorization?

Eavesdropping for authorization between smartphone and coffee maker

Enable Bluetooth HCI snoop log, then reboot, then adb pull /sdcard/btsnoop_hci.log, then inspect network capture, look for **ATT** protocol and `Write Request` on handle `0x14`:



```
▶ Frame 1952: 20 bytes on wire (160 bits), 20 bytes captured (160 bits)
▶ Bluetooth
▶ Bluetooth HCI H4
▶ Bluetooth HCI ACL Packet
▶ Bluetooth L2CAP Protocol
▼ Bluetooth Attribute Protocol
   ▶ Opcode: Write Request (0x12)
   ▶ Handle: 0x0014 (Unknown: Unknown)
     Value: 8418ffdaf230af08
     [Response in Frame: 1953]
```

```
0000  02 40 00 0f 00 0b 00 04  00 12 14 00 84 18 ff da   ·@······ ········
0010  f2 30 af 08                                        ·0··
```

BrewOperations

```
public Completable writeBrewNow(int coffeeTypeId) {
  return
  ↪   ((Authentication)this.deviceDriver.getBleDevice().getFeature(Authenticat
  .authenticate(this.deviceDriver.write(new OperationKey("Brew"),
  ↪   BrewOperations.COMMAND_KEY_CHARACTERISTIC_DESCRIPTION,
  ↪   BrewByteConversion.getProgrammedBrewPayload(0L,
  ↪   coffeeTypeId))).compose(BrewOperations..Lambda.0.$instance);
}
```

- `authenticate()`
- `COMMAND_KEY_CHARACTERISTIC_DESCRIPTION`
- `getProgrammedBrewPayload(0, coffeeTypeId)`

COMMAND_KEY_CHARACTERISTIC_DESCRIPTION

The characteristic belongs to SERVICE_UUID. The characteristic UUID is created from UUID_TEMPLATE and modifying some bytes:

```
BrewOperations.COMMAND_KEY_CHARACTERISTIC_DESCRIPTION = new
↪  CharacteristicDescription(BrewOperations.SERVICE_UUID,
↪  BrewOperations.UUID_TEMPLATE.evaluate(0x3A42L));
```

We unwrap calls:

- SERVICE_UUID is 06aa**1920**-f22a-11e3-9daa-0002a5d5c51b.
- COMMAND_KEY_CHARACTERISTIC_DESCRIPTION is 06aa**3a42**-f22a-11e3-9daa-0002a5d5c51b

getProgrammedBrewPayload()

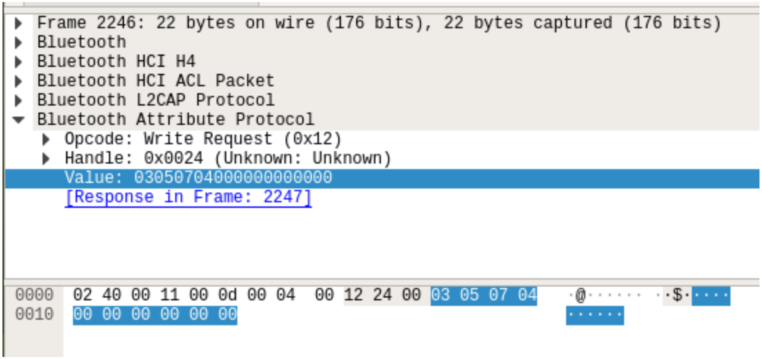
```
public static ByteBuffer getProgrammedBrewPayload(long time, int
↪  coffeeType) {
        ByteBuffer buffer = ByteBuffer.allocate(10);
        buffer.put(BrewByteConversion.PROGRAMMED_BREW_PREFIX);
        buffer.put(ByteConversion.toByteBuffer(((int)time)));
        buffer.put(ByteConversion.toByteBuffer(((short)coffeeType)));
        return buffer;
}
```

03 05 07 04	00 00 00 00	TT TT
Fixed Prefix	Reserved	Coffee Type

Coffee Type

```java
public int getCoffeeTypeIdFromProdigioBrewParameters(...) {
        int v0;
        switch(prodigioBrewParameters) {
            case RISTRETTO: {
                v0 = 0;
                break;
            }
            case ESPRESSO: {
                v0 = 1;
                break;
            }
            case LUNGO: {
                v0 = 2;
                break;
            }
            default: {
                throw new BrewException("Illegal coffee type");
            }
        }
        return v0;
    }
```

You can also get the command from packet capture

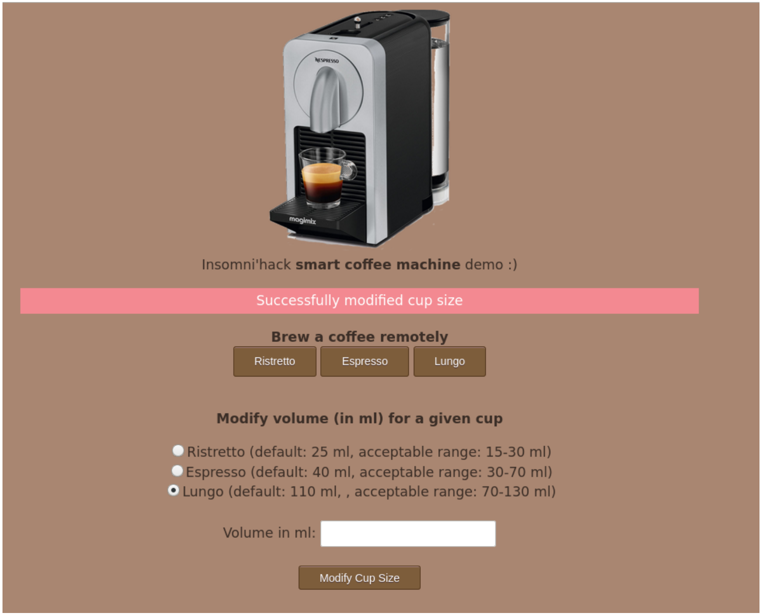


```
▶ Frame 2246: 22 bytes on wire (176 bits), 22 bytes captured (176 bits)
▶ Bluetooth
▶ Bluetooth HCI H4
▶ Bluetooth HCI ACL Packet
▶ Bluetooth L2CAP Protocol
▼ Bluetooth Attribute Protocol
   ▶ Opcode: Write Request (0x12)
   ▶ Handle: 0x0024 (Unknown: Unknown)
     Value: 03050704000000000000
     [Response in Frame: 2247]

0000  02 40 00 11 00 0d 00 04  00 12 24 00 03 05 07 04   ·@······ ··$·····
0010  00 00 00 00 00 00                                  ······
```

Request to brew a Ristretto

Webpresso



Insomni'hack **smart coffee machine** demo :)

Successfully modified cup size

Brew a coffee remotely

Modify volume (in ml) for a given cup

- ○ Ristretto (default: 25 ml, acceptable range: 15-30 ml)
- ○ Espresso (default: 40 ml, acceptable range: 30-70 ml)
- ● Lungo (default: 110 ml, , acceptable range: 70-130 ml)

Volume in ml:

Waow, this looks interesting!

```
public class CupSizeOperations implements CupSize {
  ...
  private Completable writeCupSizeTarget(CupSizeType cupSizeType) {
    return this.deviceDriver.write(new
    ↪  OperationKey("writeCupSizeTarget"),
    ↪  CupSizeOperations.WRITE_CUPE_SIZE_TARGET_CHARACTERISTIC_DESCRIPTION,
    ↪  this.getCupSizeKindByteBuffer(cupSizeType));
  }

  private Completable writeCupSizeVolume(int volume) {
    return this.deviceDriver.write(new
    ↪  OperationKey("writeCupSizeVolume"),
    ↪  CupSizeOperations.VOLUME_CHARACTERISTIC_DESCRIPTION,
    ↪  this.getCupSizeVolumeData(volume));
  }
}
```

Can we customize cup size?

The app does not export this feature!

How to customize volume

1. Connect, pair, get authorization
2. Specify cup (ristretto, espresso, lungo)
3. Customize volume in ml
4. Disconnect

You can't specify *any* volume, there are ranges:

```
CupSizeVolume.RISTRETTO_VOLUME_RANGE = Range.open(15, 30);
CupSizeVolume.ESPRESSO_VOLUME_RANGE = Range.open(30, 70);
CupSizeVolume.LUNGO_VOLUME_RANGE = Range.open(70, 130);
```

Ph0wn 2019 CTF

- **17 teams** found the authorization code
- **1 team** prepared coffee without the app (nearly 2)
- 0 team managed to customize volume (nearly 1)

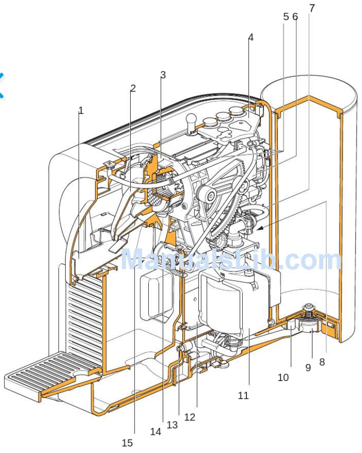
Ph0wn 2019 CTF

- **17 teams** found the authorization code
- **1 team** prepared coffee without the app (nearly 2)
- 0 team managed to customize volume (nearly 1)

Working with BLE

- Encountered multiple crazy bugs: **Use Bluez 5.50+**
- **bluetoothctl** can be scripted using **expect**
- Code at `https://github.com/cryptax/webpresso/`
- There is a combination of buttons to unpair, or factory reset the coffee machine. **Useful**

Hardware



1) Drop stop
2) Steam cover
3) Brewing unit
4) MMI board (Men-Machine Interface) with reinforced silicone keypad
5) Light guides
6) Thermoblock
7) High pressure connector
8) Motor
9) Water tank connector
10) Magnet fixing for water tank (3 permanent magnets)
11) Pump
12) Electronic module with flowmeter
13) Magnet fixing for drip tray
14) Light barrier
15) Position switch for used capsule container

BL600-SA-06

- Bluetooth Low Energy Module
- Based on a ARM Cortex M0 with BLE radio
- Integrated ceramic RF antenna
- Manufactured by Laird

- Spares: PCB with buttons, Flow Meter, Control Module...
- C70 Service Manual
- CitiZ teardown (similar - a few differences)

Thanks for your attention!

@cryptax
@ph0wn - https://ph0wn.org
https://github.com/cryptax/webpresso/
Kudos to ph0wn organizers!

If you have a cool idea for an IoT challenge, please talk to me!