BLUETOOTH Characteristics 2

Goals of this challenge

- Flashing the firmware
- Solving the challenge

Challenge Description: A confidential message is stored on the firmware but protected by a password. The goal is to provide a screenshot of the confidential message.

Flashing the firmware

- Connect the **DVID board** to the **computer** using the **USB AVR Programmer**, and start the script
- When the AVR programming has completed, the **board should restart** and you should the following
- Flash.sh

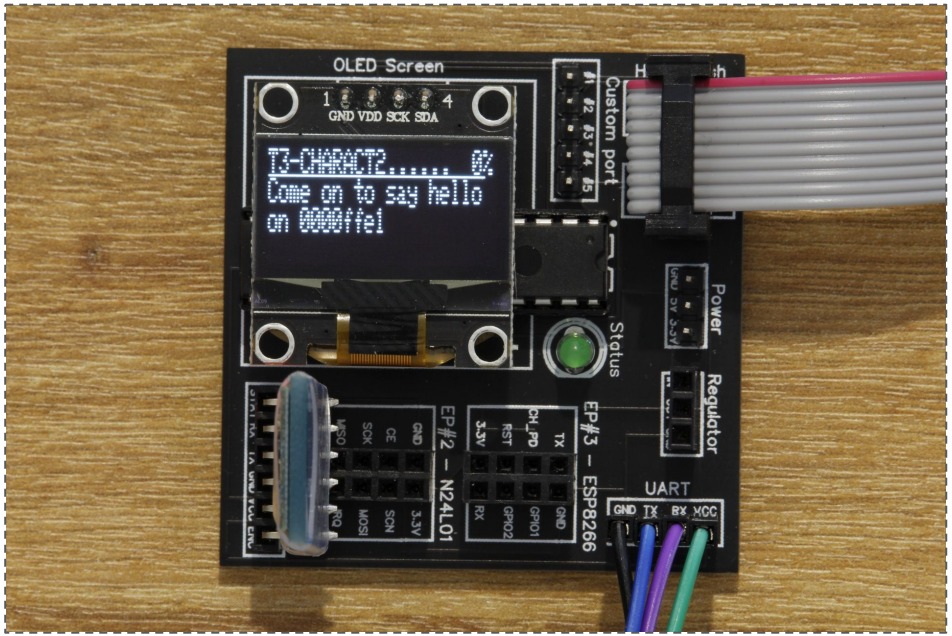
```
#!/bin/bash

if [[ ! -d "./DVID/" ]]; then
    git clone https://github.com/vulcainreo/DVID
fi

pushd ./DVID/trainings/bluetooth/characteristics2/
avrdude -F -v -p atmega328p -P /dev/ttyUSB0 -c
usbasp -u -U
flash:w:characteristics2.ino.with_bootloader.ardui
no_standard.hex
popd
```

Note:

Avrdude for windows
https://github.com/mariusgreuel/avrdude/releases/download/v7.0-windows/avrdude-v7.0-windows-windows-arm64.zip

Flashing the firmware

- Now, we will connect the DVID board to our computer using the USB AVR programmer, and start the script. When the AVR programming has completed, the board should restart and you should see the following

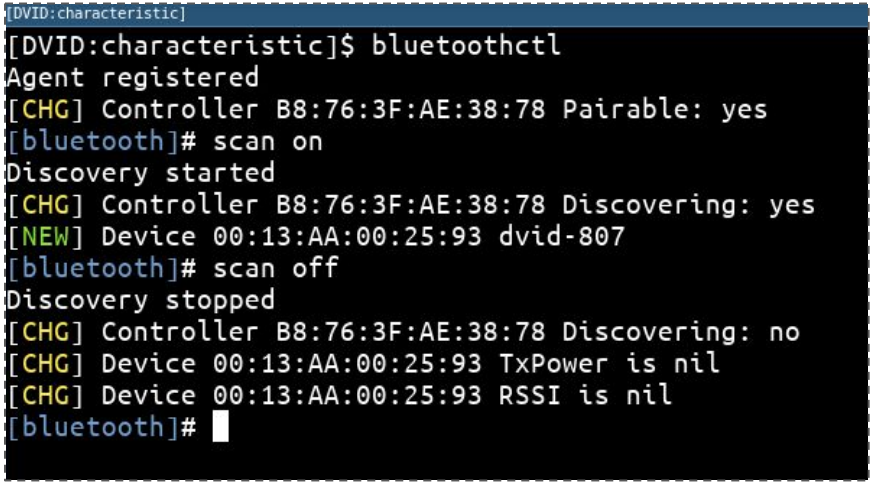


Note:

Avrdude for windows
https://github.com/mariusgreuel/avrdude/releases/download/v7.0-windows/avrdude-v7.0-windows-windows-arm64.zip

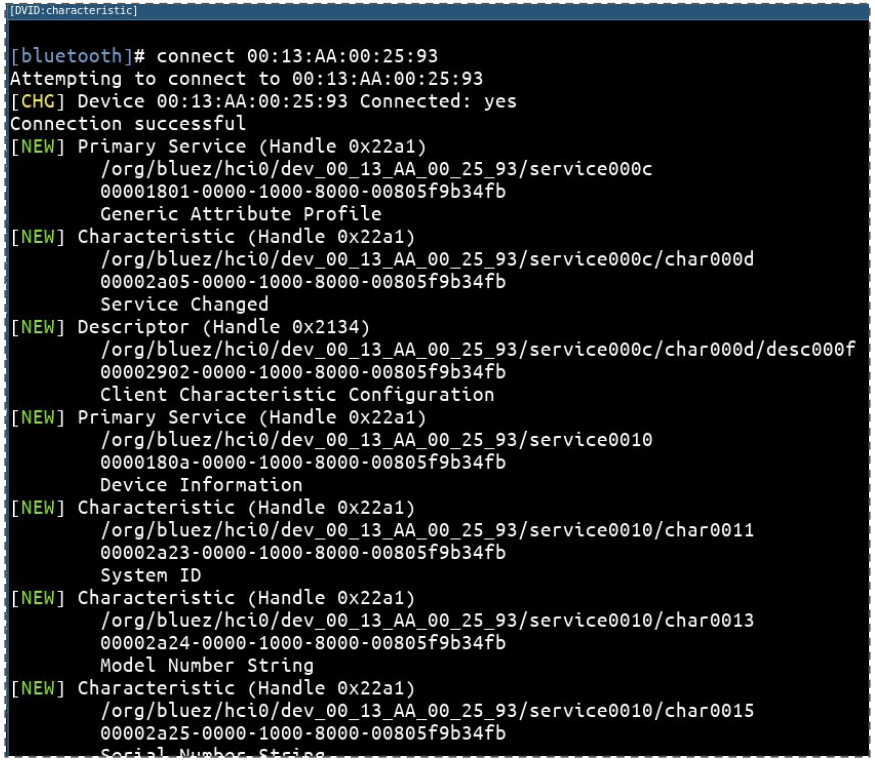
Solving the challenge

- In this challenge, the IoT device sends informations using bluetooth characteristics on `0000ffe1`
- Firstly, we need to do a **bluetooth scan** to find the device
- We can do this inside **bluetoothctl** using **scan on** command. (and **scan off** to stop)



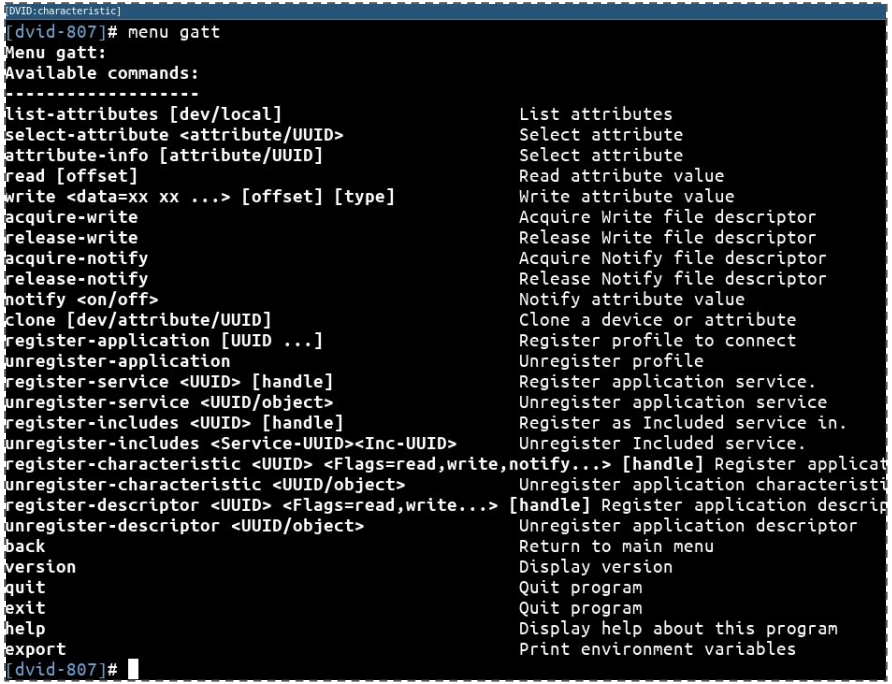
Solving the challenge

- Now that we found the IoT device, we can connect to it by typing **connect** followed by the device's bluetooth address
- To use **bluetooth characteristics**, we need to use a specific menu, called **gatt**



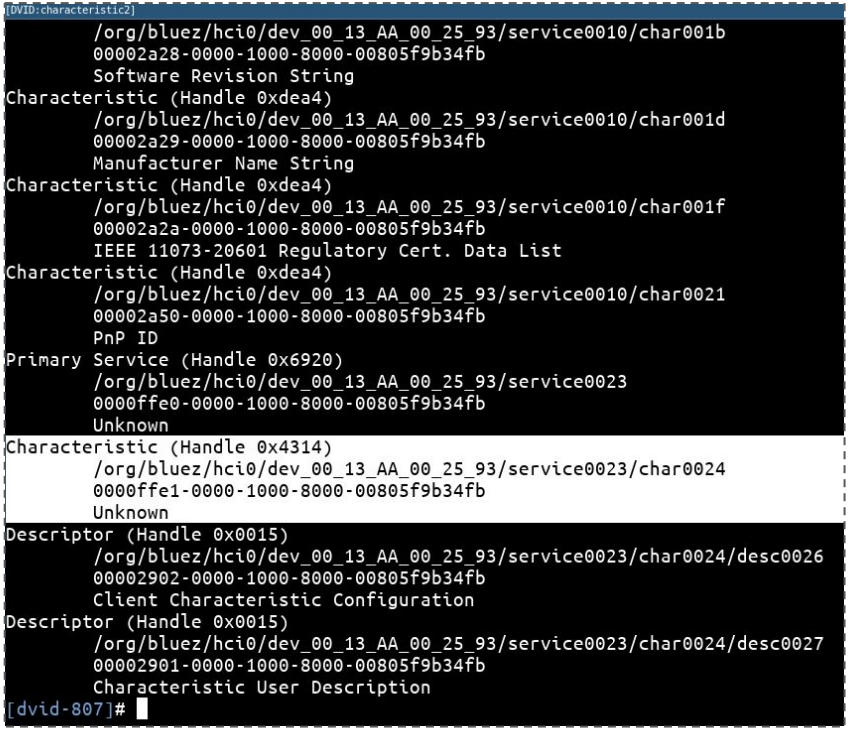
Solving the challenge

- To access it in **bluetoothctl**, type **menu gatt** to enter the gatt menu and **back** to exit the menu
- In the gatt menu we have access to many advanced features, such as listing attributes of a device, reading and writing data to bluetooth services



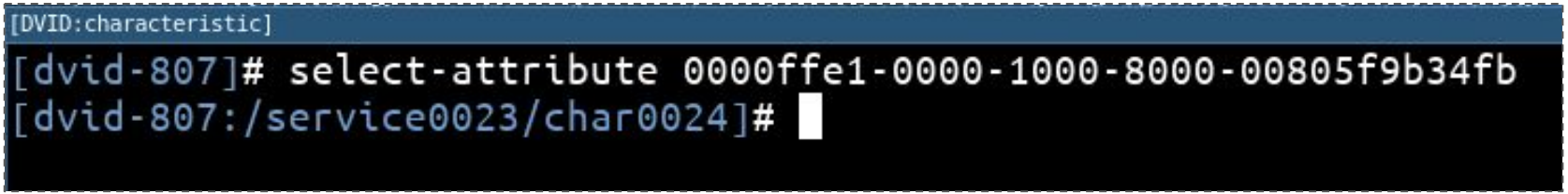
Solving the challenge

- We will now list attributes offered by the bluetooth device using **list-attributes** command



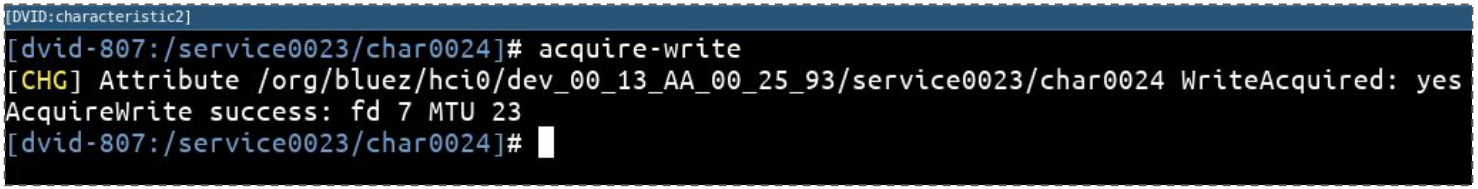
Solving the challenge

- Looking at the screen of the **DVID device**, we can see a message **"Something is leaking on 0000ffe1"**, therefore we will try to interact with the characteristic **0000ffe1-0000-1000-8000-00805f9b34fb**. To do this, we need to select the attribute using **select-attribute <uuid>** command

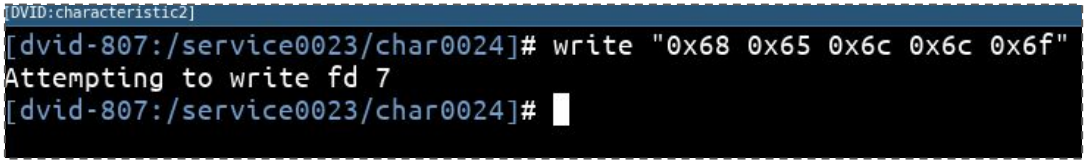


- Then we need to **acquire-write** to connect to the stream and be able to read the values from the **service/characteristic.** We can then type read to read the stream of values coming from the **service/characteristic.**

Solving the challenge



- Now we can use the **write** command to **send data** to the **service/characteristic**. The syntax is **write "0xaa 0xbb 0xcc ..."** with the **hex values of the data you want to send.** For example **hello** is **"0x68 0x65 0x6c 0x6c 0x6f"**. Therefore, the final command is write **"0x68 0x65 0x6c 0x6c 0x6f"**



Flashing the firmware

- We can now see the flag on the DVID screen