FIRMWARE Hardcoded Password

Goals of this challenge

- Flashing the firmware
- Solving the challenge
  - Dumping the firmware
  - Analyzing the firmware source code

Challenge Description: A confidential message is stored on the firmware but protected by a password. You should be able to read the confidential message in the source code

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

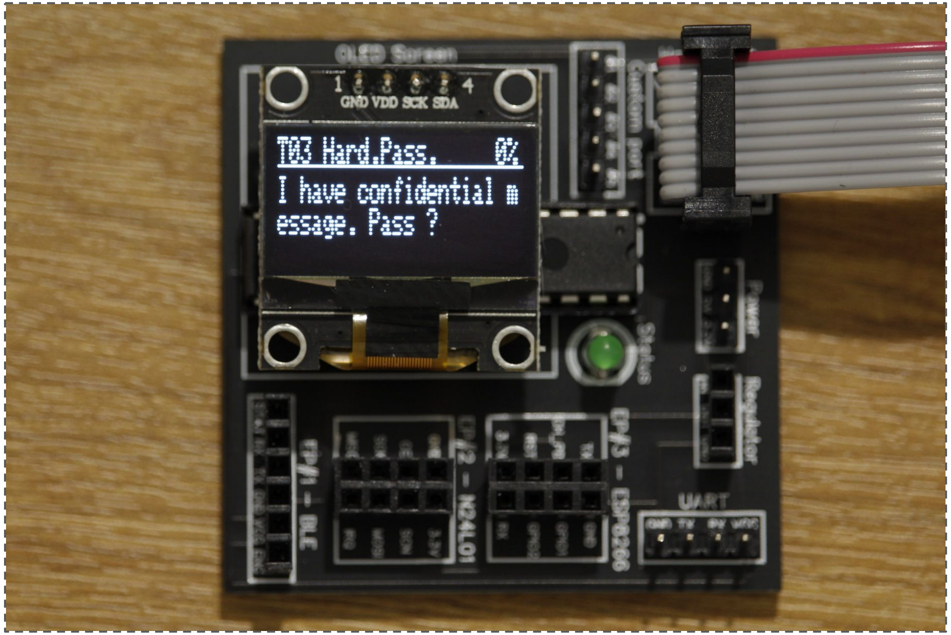
```
#!/bin/bash

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

pushd ./DVID/trainings/firmware/hardcodedPassword/
avrdude -F -v -p atmega328p -P /dev/ttyUSB0 -c usbasp -u -U flash:w:hardcodedPassword.ino.with_bootloader.arduino_standard.hex
popd
```

Note:

Avrdude for windows
https://github.com/mariusgreuel/avrdude/releases/download/v7.0-windows/avrdude-v7.0-windows-windows-arm64.zip
Flash.sh
https://hastebin.skyra.pw/raw/likojafomo

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



```
T03 Hard.Pass.     0%
I have confidential m
essage. Pass ?
```

Note:

Avrdude for windows
https://github.com/mariusgreuel/avrdude/releases/download/v7.0-windows/avrdude-v7.0-windows-windows-arm64.zip
Flash.sh
https://hastebin.skyra.pw/raw/likojafomo

Solving the challenge [dumping the firmware]

- the **avrdude** utility can do multiple things ! For **extracting the firmware**, the command differs only a little from the command to flash the firmware. We will only change the *Memory operation* mode in the **-U** option
- We used **-U flash:w:findTheDatasheet.ino.arduino_standard.hex** to **install** the firmware on the board
- We will use **-U flash:r:firmware.bin** to **dump** the firmware from the board. (We could have also dumped it in **hex format** with **-U flash:r:firmware.hex**)
- The command becomes :

```
avrdude -F -v -p atmega328p -P /dev/ttyUSB0 -c usbasp -u -U flash:r:firmware.bin
```

- The strings command can be used to extract ASCII strings from the raw binary file, which is a pretty useful first step in reversing

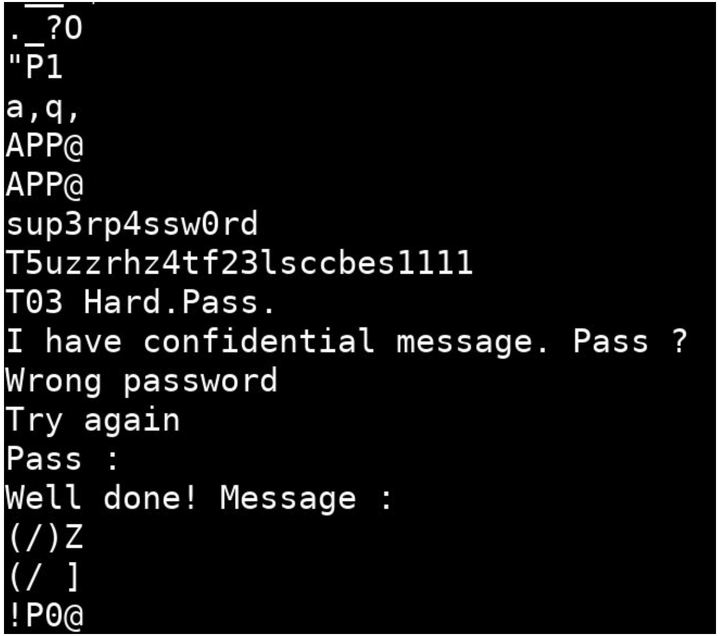
```
strings -a firmware.bin
```

Solving the challenge [dumping the firmware]

- Sup3rp4ssw0rd seems like the password, also we can see other strings like the flag and messages
- Use this script to send the password to DVID board and solve the challenge

```
#!/usr/bin/env python3

import serial

s = serial.Serial("/dev/ttyUSB0", 9600)
s.write(b'sup3rp4ssw0rd\r\n')
```

- This challenge could also be solved by analysing the firmware source code



```
._?O
"P1
a,q,
APP@
APP@
sup3rp4ssw0rd
T5uzzrhz4tf23lsccbes1111
T03 Hard.Pass.
I have confidential message. Pass ?
Wrong password
Try again
Pass :
Well done! Message :
(/)Z
(/ ]
!P0@
```

Solving the challenge [analyzing the firmware source code]

- We also could have solved the challenge another way if we consider we can access the firmware source code. In this case, we can look for hard coded variables in the code of the file hardcodedPassword.ino
- We easily find the hardcoded password on line 18 of the file hardcodedPassword.ino. Of course, the idea here is to learn about realistic attacks, you might not always have the source code of the firmware !

```
18 String password = "sup3rp4ssw0rd";
19 String message = "T5uzzrhz4tf23lsccbes1111";
20 String currentPassword = "";
21 bool finished = false;
22 String rx = "";
23 bool first = true;
```

Note:

Rapid7 How to extract firmware from onboard flash memory part 1
https://www.rapid7.com/blog/post/2019/04/16/extracting-firmware-from-microcontrollers-onboard-flash-memory-part-1-atmel-microcontrollers/