National College of Ireland

Configuration Manual

MSc Research Project
Cyber Security

David Collins
Student ID: X09106081

School of Computing
National College of Ireland

Supervisor: Dr Paul Stynes, Dr Vanessa Ayala-Riverra

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	David Collins
Student ID:	X09106081
Programme:	Cyber Security
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr Paul Stynes, Dr Vanessa Ayala-Riverra
Submission Due Date:	23/09/2021
Project Title:	Configuration Manual
Word Count:	1136
Page Count:	13

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	*David Collins*
Date:	23rd September 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☑
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☑
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☑

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

David Collins
X09106081

1 Introduction

In my paper, Pen-Testing Framework for IoT Devices. A suitable framework for performing a penetration test on an ESP32 Microcontroller is proposed. This configuration manual describes the steps taken to set up the lab environment which would allow the test to be conducted. The tests are derived from the evolving legal regulations and demonstrate how these can be carried out practically.

2 Lab Set-Up

The high-level view of the Lab setup is shown in the Network Diagram. The subsequent parts of Section 2 comprise the instruction required to configure each of the components of this Network.

- OpenWrt router will act as a Firewall.
- ESP32 Nat Router will provide an isolated WiFi network to carry out the tests.
- Kali Linux will have the appropriate tools to carry out the test.
- The laptop will have any additional tools required.
- Raspberry pi zero -w set up to allow testing of the MQTT protocol.
- IoT Device under test.

The ESP32 Nat Router is configured using the WiFi Manager On the Laptop connect to this network the Initial settings has this open network. Once connected navigate to `192.168.4.1`. New values for SSID, Password can be Entered. Add the credentials for the OpenWrt Server allowing Internet connection, firewall rules on the OpenWrt router permitting. Static IP settings can be entered if required. Connect the Kali Linux, Raspberry Pi zero, and Laptop to the ESP32 Nat Router WiFi.

2.1 Network Diagram

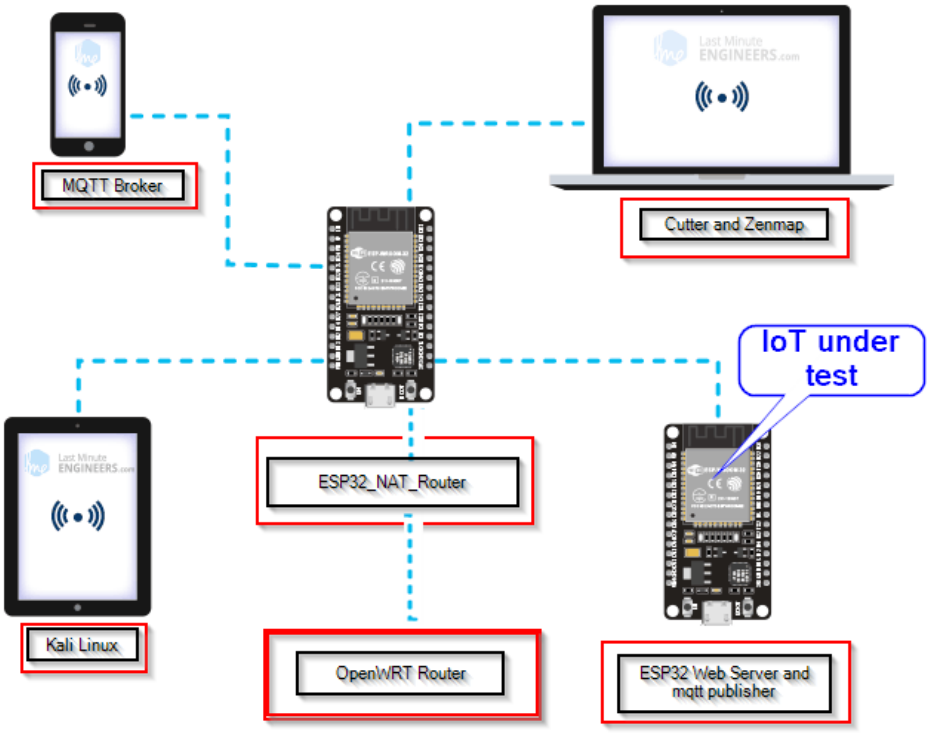


Figure 1: Network Lab Diagram

2.2 OpenWrt Server

- ✓ connect to Web Gui `192.168.1.1`
- ✓ go to Networks - Wireless
- ✓ create a WiFi Network by clicking add.
- ✓ go to Device Configuration
- ✓ open Advance Settings.
- ✓ fill in country code to meet legal regulations.
- ✓ in General, Security enters SSID for the name of the network.
- ✓ in Wireless Security select encryption method WPA2-PSK
- ✓ in Wireless Security select a cypher auto.
- ✓ in Wireless Security enter a wireless password.
- ✓ click save and apply

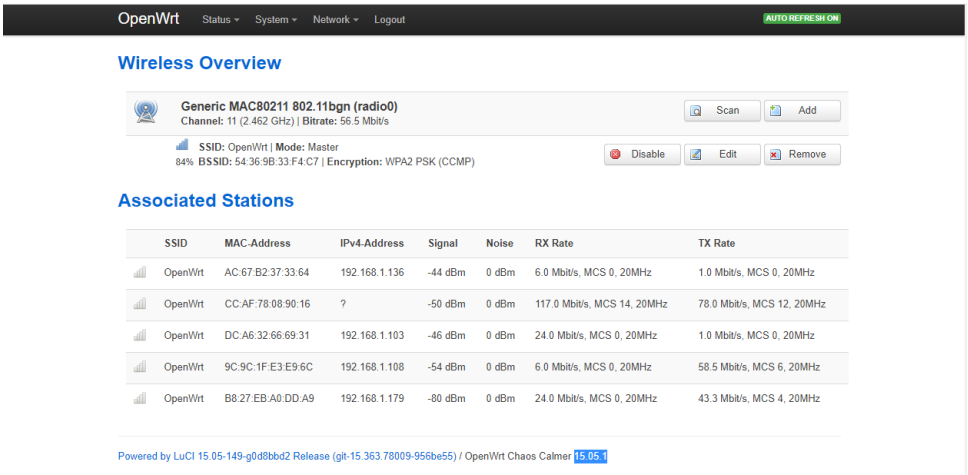


Figure 2: OpenWrt Showing Connected devices

2.3 ESP32 Nat Router

Carry out the following steps to create the ESP32 Nat router.

Step 1: Download the espressif Download tool. from `https://www.espressif.com/en/support/download/other-tools`

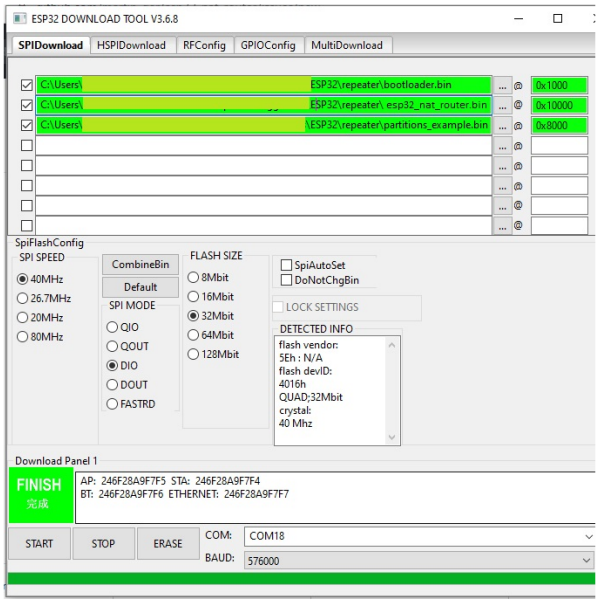


Figure 3: Download tool settings

Step 2: Download the binaries from Github `https://github.com/martin-ger/esp32_nat_router/tree/master/build`.

Step 3: Install the binaries at the following locations:

- bootloader.bin @0x1000.
- esp32_nat_router.bin @ 0x10000

- partitions_example.bin @8000

✓ set SPI Speed to 40MHz.

✓ SPI Mode to DIO

✓ Flash size 32Mbit.

✓ Click start and wait for the download to complete.

2.4 MQTT Broker

Download the Raspbian Jessie image and install it to SD Card. Attach SD card to Raspberry Pi zero w. This device comes with added Wireless and Bluetooth Connectivity. Connect Monitor and Bluetooth Mouse and Keyboard to the PI. Open a command prompt and update the software with the commands. sudo apt update sudo apt dist-upgrade. To install an MQTT Broker and MQTT Client on the Raspberry PI zero the following steps are used.

✓ wget `http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key`

✓ sudo apt-key add mosquitto-repo.gpg.key

✓ cd /etc/apt/sources.list.d/

✓ sudo wget `http://repo.mosquitto.org/debian/mosquitto-wheezy.list`

✓ apt-get update

✓ apt-get install mosquitto

✓ apt-get install mosquitto-clients

To install NODE-RED on the PI Zero run the following installation script

$bash>(curl - sL \quad https : //raw.githubusercontent.com/node - red/raspbian - deb - package/master/resources/update - nodejs - and - nodered)$

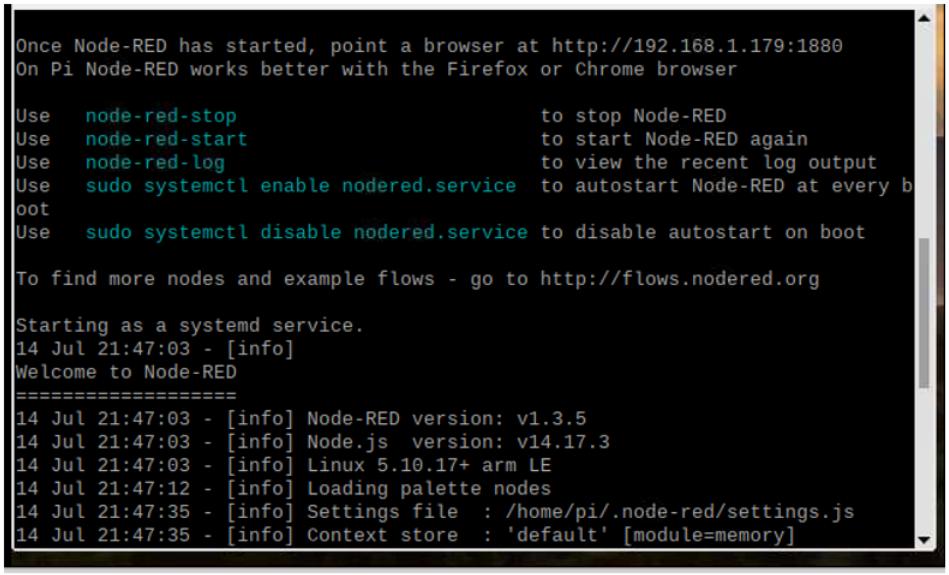


Figure 4: NODE RED Start Stop Commands

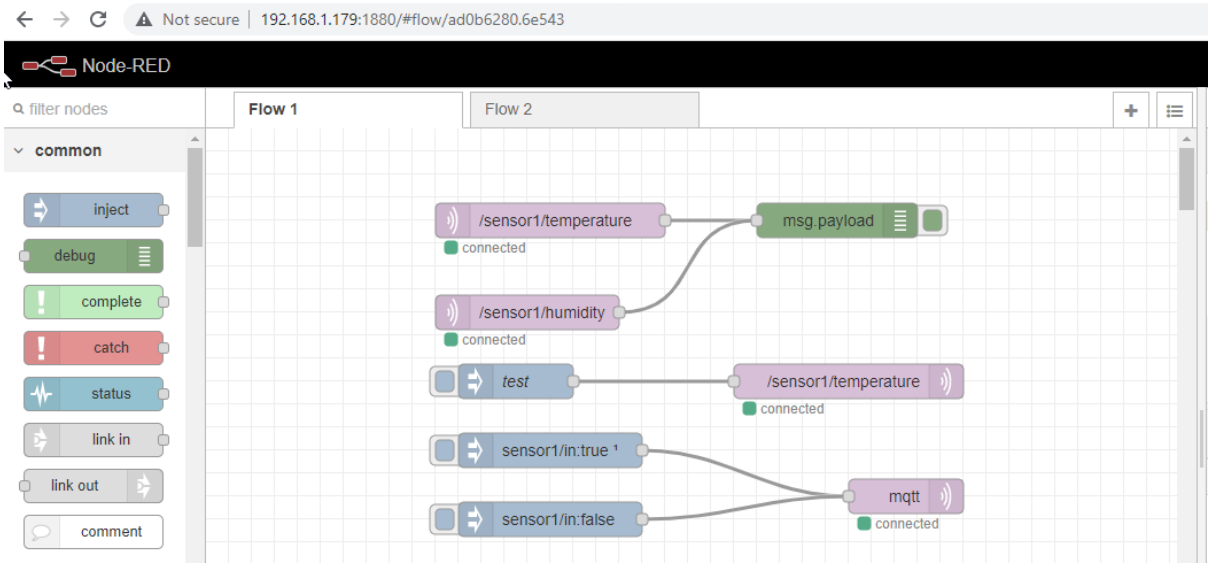
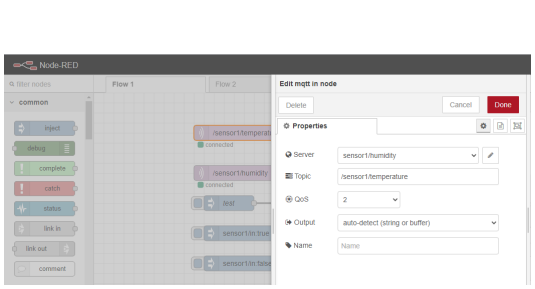
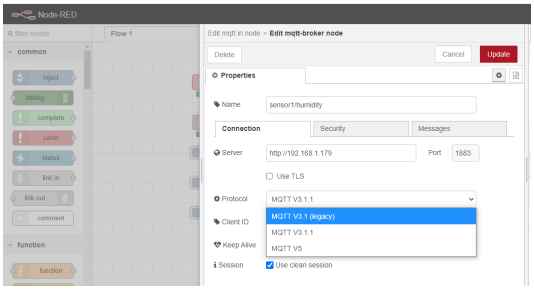


Figure 5: NODE RED MQTT Sketch

Start Node-Red. Create the MQTT sketch and configure the nodes with topic and configuration parameters.

Table 1: Configure MQTT topic and security.





2.5 Kali Linux

To conduct the penetration test extra tools are installed on Kali Linux. For this Lab environment, a Raspberry pi 4 Model B 8GB Ram was used. The kali image was downloaded from `https://www.kali.org/get-kali/#kali-arm` and flashed to a 32 GB sd card using Balana Etcher. On start-up, Kali was updated with the commands.

sudo apt update

sudo apt upgrade

2.5.1 Kali Tools

esptool git clone `https://github.com/espressif/esptool.git`

cd esptool

pip install –user -e.

to test installation use command $python3 \quad esptool -h$

(Fredrik; 2021)

RouterSploit sudo apt-get install python3-pip

git clone `https://www.github.com/threat9/routersploit`.

cd routersploit

$python3 \quad -m \quad pip \quad install -r \quad requirements.txt$

(kleo; 2018)

Wifite git clone `https://github.com/derv82/wifite2.git`.

cd Wifite

$sudo \quad python3 \quad setup.py\, install$

To test installation $sudo \quad wifite -h$ will provide a list of all available commands

(derv; 2018)

nmap nmap should come preloaded on Kali Linux if not it can be installed with the following command.

$sudo \quad apt-get \quad install -y\, nmap$

to test $nmap \quad -version$

(Lyon; 2012)

WireShark $sudo \quad apt-get \quad install -y\, wireshark$

to test $wireshark \quad -h \quad tshark \quad -h$

(*Wireshark · Go Deep.*; 2021)

Visual Studio Code Open a web browser and navigate to `https://code.visualstudio.com/download`

and download the latest armhf.deb file.

to install run the command sudo apt install ./code-xxx-armhf.deb

PlatformIO

- Open VSCode Package Manager.
- Search for the official platformio ide extension.
- Install PlatformIO IDE check that toolbar is present.

Figure 6: PlatformIO toolbar

(Kravets; 2020)

3 ESP32 IoT Test Subject

The IoT device used to test the penetration test framework uses an ESP32 Microcontroller to act as a Web Server and MQTT client. The following steps were used to program the project. Open Visual Studio Code and create a PlatformIO application. Add the following details to the config file. The libraries can be added manually or by using the Libraries tab of PIO Home. The full code can be found on my GitHub Repository `https://github.com/2davecollins/esp`. (Collins; 2021) For this test, sensitive data was extracted to separate credentials.h file for evaluation. Other options for storing credentials would use the Preferences library. (Santos; 2021). The initial plan for this paper was to test an insure IoT devise add security features secure MQTT, HTTPS and test again.

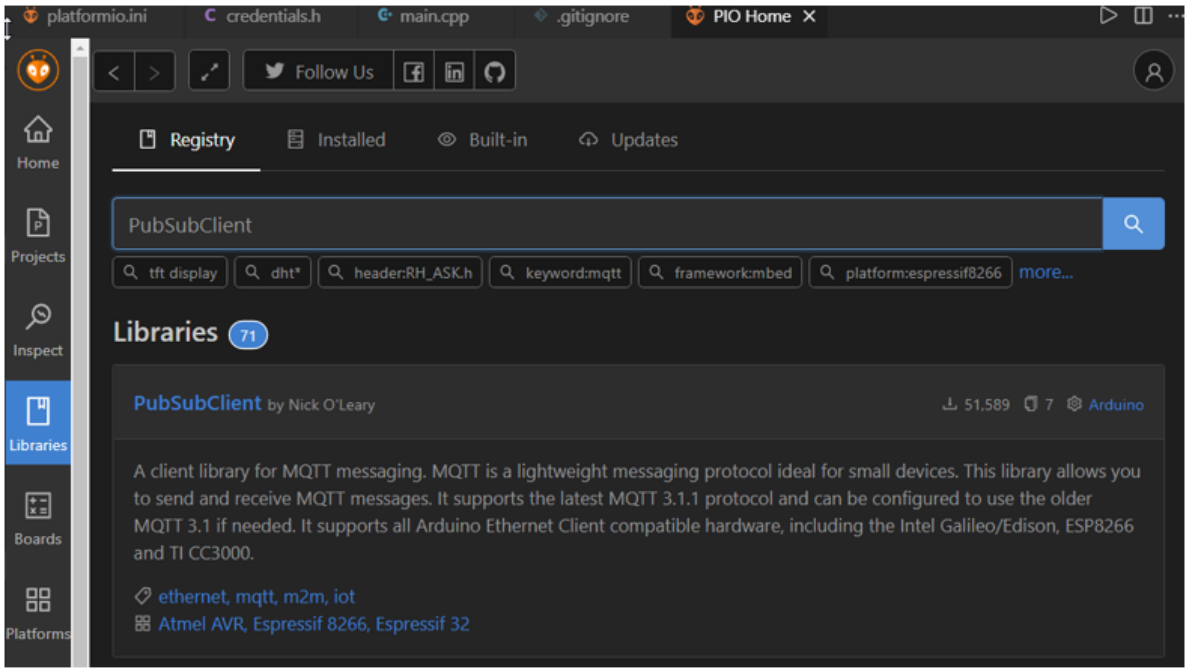


Figure 7: PIO Home add Libraries

Table 2: Configure Credentials platformIO

```
[env:esp32dev]
platform = espressif32
board = esp32dev
framework = arduino
lib_deps =
    ottowinter/ESPAsyncWebServer-esphome@^1.2.7
    ottowinter/AsyncTCP-esphome@^1.2.1
    adafruit/Adafruit Unified Sensor @ ^1.1.4
    adafruit/DHT sensor library@^1.4.2
    ayushsharma82/AsyncElegantOTA @ ^2.2.5
    knolleary/PubSubClient@^2.8
monitor_speed = 115200
check_tool = clangtidy
```

credentials.h > ...

```
#define WIFI_SSID "OpenWrt"
#define WIFI_PASSWD "password"
#define U_NAME "admin"
#define U_PASS "myPassword"
#define MQTT_BROKER "192.168.1.179"
#define MQTT_USER_NAME "cdavid"
#define MQTT_USER_PASSWORD "cdavid"
#define MQTT_INTERVAL 10000;
```

4 Penetration Test

The following checks are carried out as part of the penetration test.

- Identification of IoT Device.
- Ensure Firmware has not been tampered with.
- Perform Static code analysis on firmware.
- Check router for vulnerabilities (default credentials).
- Network Scan for open ports.
- Check for data exposure using Wireshark.

4.1 ESP32 Identification

cd /home/kali/esptool

use esptool chip id function to report on esp32 chip identification.

$$python3\ esptool.py\ chip\_id$$

additional information can be found using esptool flash id function.

$$python3\ esptool.py\ flash\_id$$

4.2 ESP32 Check firmware for tampering

cd /home/kali/esptool

check binary at location 0x1000 against local file bootloader.bin

$$python3\ -m\ esptool\ verify\_flash\ --dif\ yes\ 0x1000\ bootloader.bin$$

check binary at location 0x10000 against local file esp32_nat_router.bin

$$python3\ -m\ esptool\ verify\_flash\ --dif\ yes\ 0x10000\ esp32\_nat\_router.bin$$

check binary at location 0x8000 against local file partitions_example.bin

$$python3\ -m\ esptool\ verify\_flash\ --dif\ yes\ 0x10000\ partitions\_example.bin$$

alternately the hash of the binary can be found by loading it into the Cutter tool and comparing the hash against a previously recorded value.

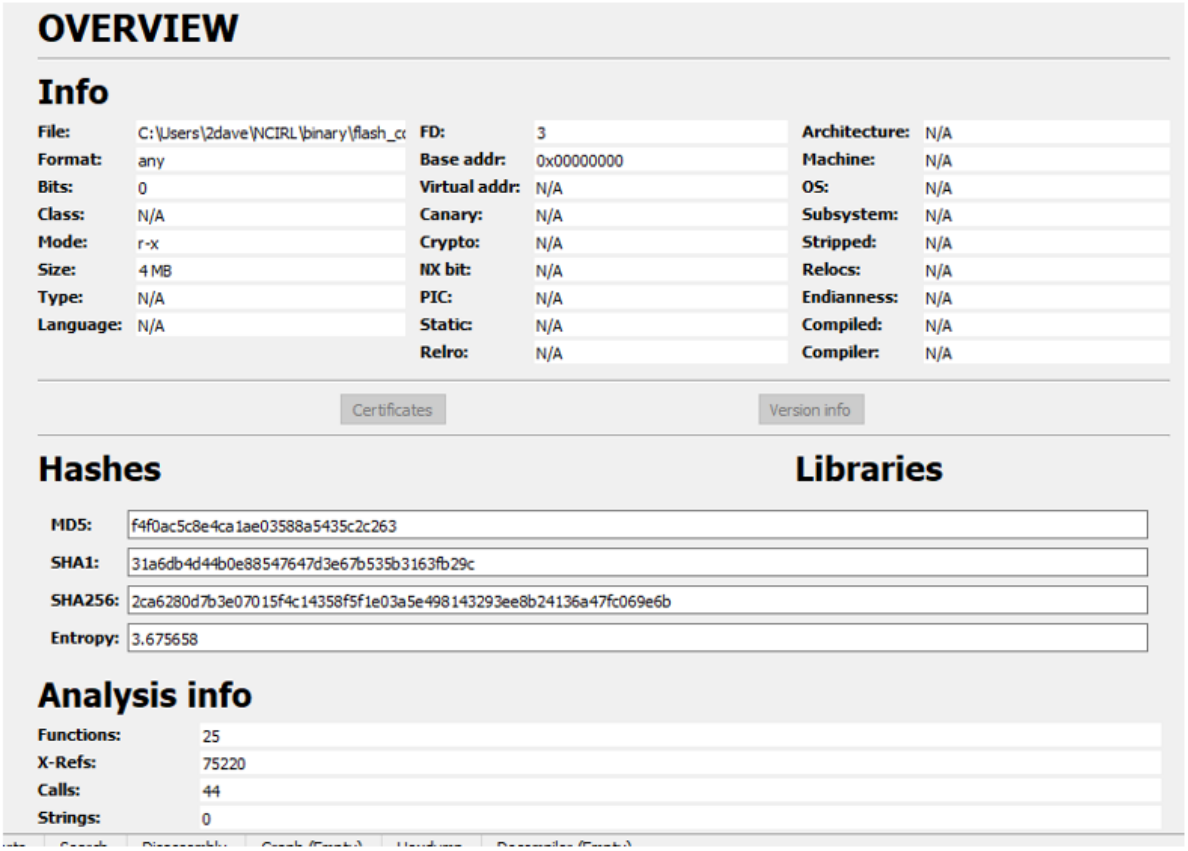


Figure 8: Reading binary hash using cutter tool

4.3 Perform Static code analysis on firmware

Open up the firmware using Visual Studio Code with platformIO extension. Open PIO Home and navigate to the Inspect Tab and Toggle Inspect Memory and Check Code and initiate test with Inspect Button.

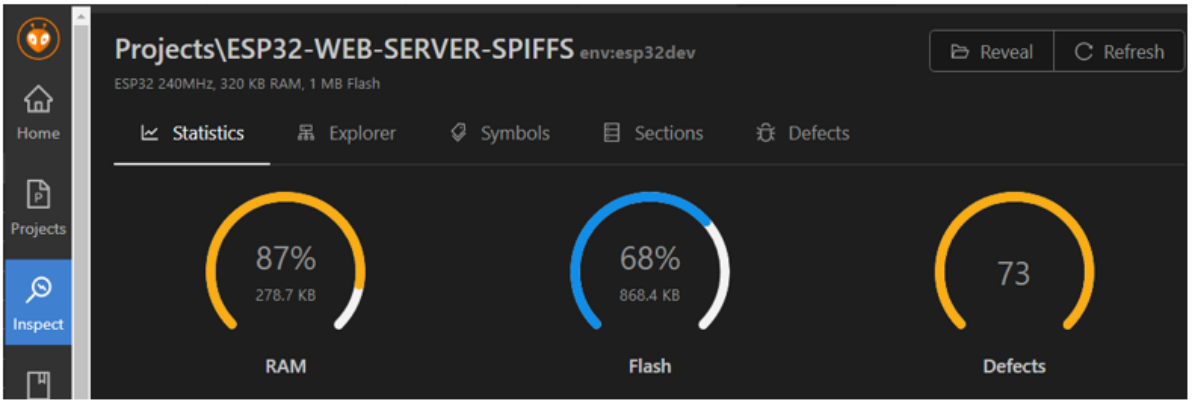


Figure 9: Result Statistics

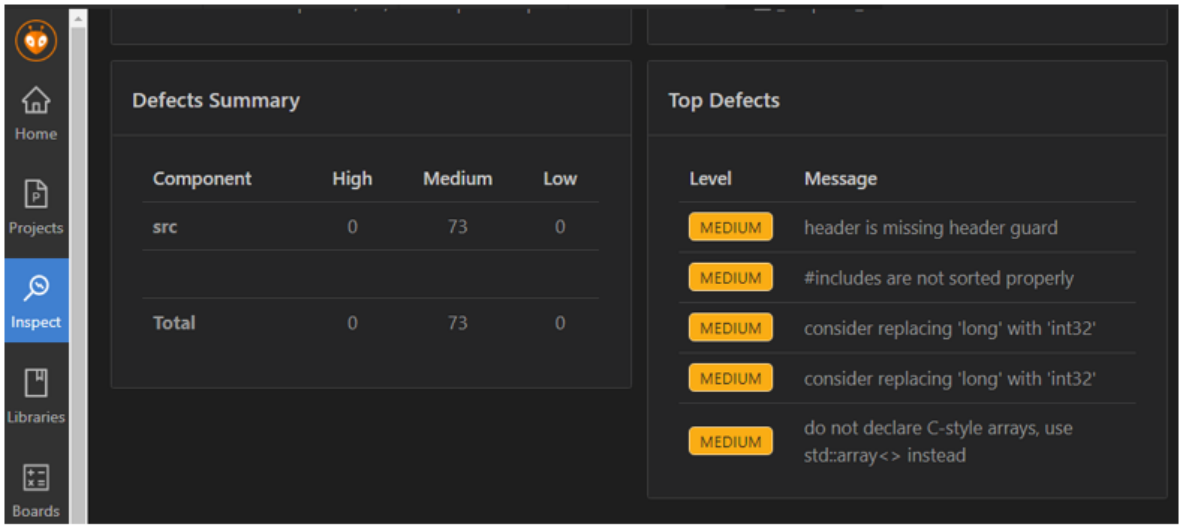


Figure 10: Top Defects Summary

4.4 Check router for vulnerabilities (default credentials) using RouterSploit

cd /home/kali/routersploit

$\#\ python3 \quad rsf.py$

$rsf > use \quad scanners\,autopwn$

$rsf > (AutoPwn) \quad show \quad options$

$rsf > (AutoPwn) \quad set\,target < ipaddressofrouter >$

$rsf > (AutoPwn) \quad run$

4.5 Network Scans for open ports

Use nmap to scan a network for open ports. Use a basic scan to determine the IP addresses on the network.

$$nmap\ 192.168.4.0/24$$

using zenmap for a graphical view of a network using a shorter range and searching for MQTT protocol.

$$nmap\ -p\ 1833\ T4\ -A\ -v\ --open\ 192.168.4.1-7$$

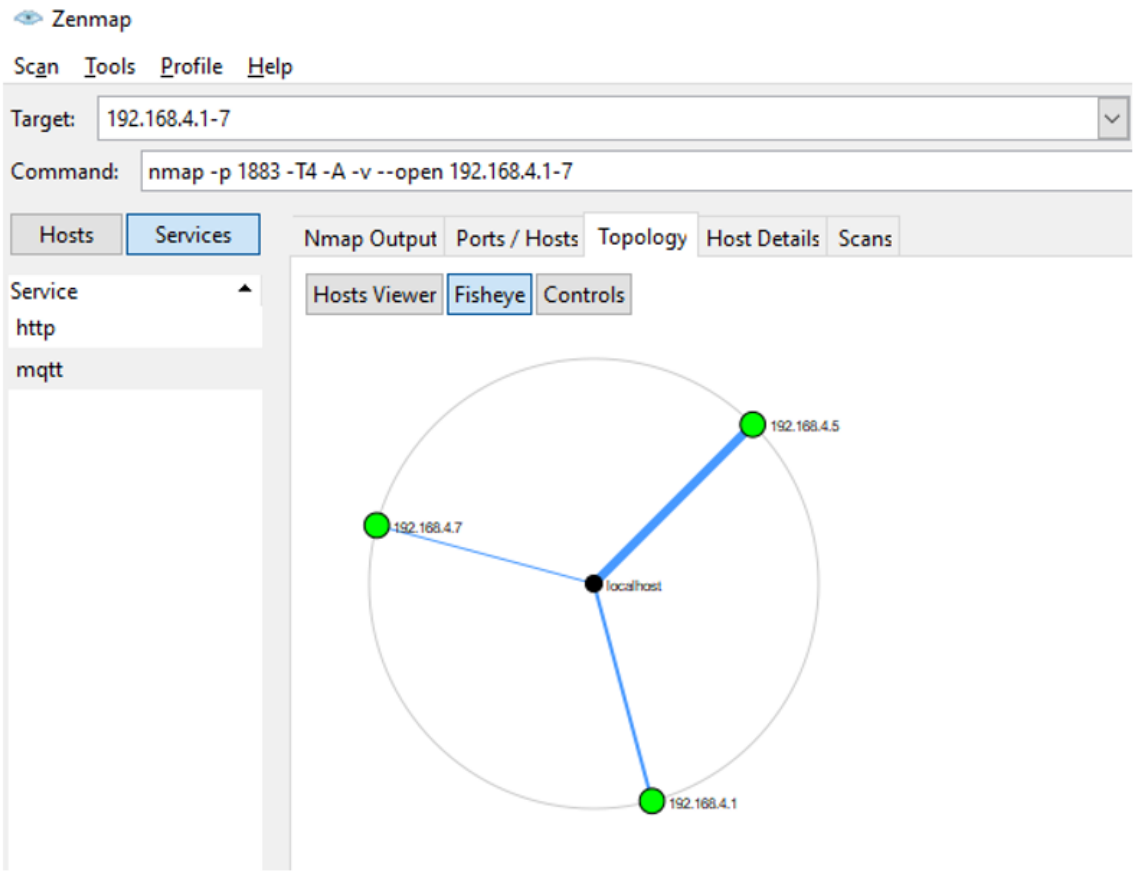


Figure 11: nmap scan for MQTT

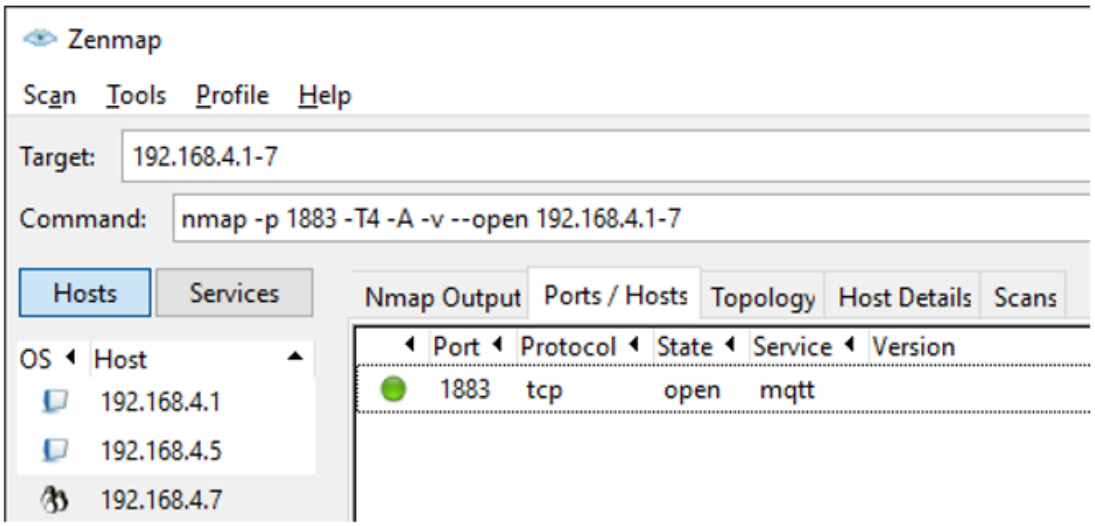


Figure 12: nmap scan results

4.6 Check for data exposure using WireShark

Open Wireshark on Kali Linux. Start NODE-Red and wait for traffic from the MQTT client on IoT under test to be observed at debug node on Node-Red. Open a web browser and observe temp and humidity sensor data from the Web Server. Stop Wireshark and examine the pcap file for exposed data.

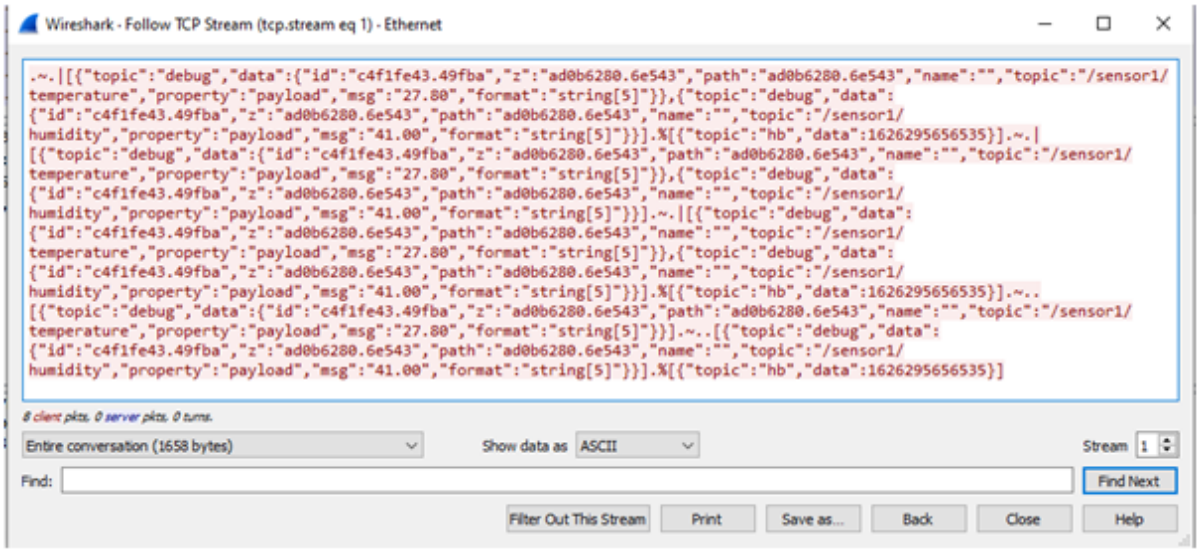


Figure 13: Wireshark exposed data

References

Collins, D. (2021). 2davecollins esp, [Online] Available: `https://github.com/2davecollins/esp`. [Accessed on: August, 10, 2021].

derv (2018). derv82 wifite, [Online] Available: `https://github.com/derv82/wifite`. [Accessed on: August, 13, 2021].

Fredrik, A. (2021). espressif esptool, [Online] Available : https://github.com/espressif/esptool. [Accessed on: July, 12, 2021].

kleo (2018). threat9 routersploit, [Online] Available: https://github.com/threat9/routersploit. [Accessed on: August, 14, 2021].

Kravets, I. (2020). Professional collaborative platform for embedded development, [Online] Available: https://docs.platformio.org/en/latest/.

Lyon, G. (2012). Nmap reference guide, [Online] Available : https://nmap.org/book/man.html. [Accessed on: August, 10, 2021].

Santos, R. (2021). Esp32 save data permantly using preferences library, [Online] Available : https://randomnerdtutorials.com/esp32-save-data-permanently-preferences/. [Accessed on: August, 10, 2021].

Wireshark · Go Deep. (2021). [Online] Available: https://www.wireshark.org/. [Accessed on: August, 13, 2021].