**IoT Security and Privacy**

**Lab 2 – ESP32 Flash Hack**

**(70 points)**

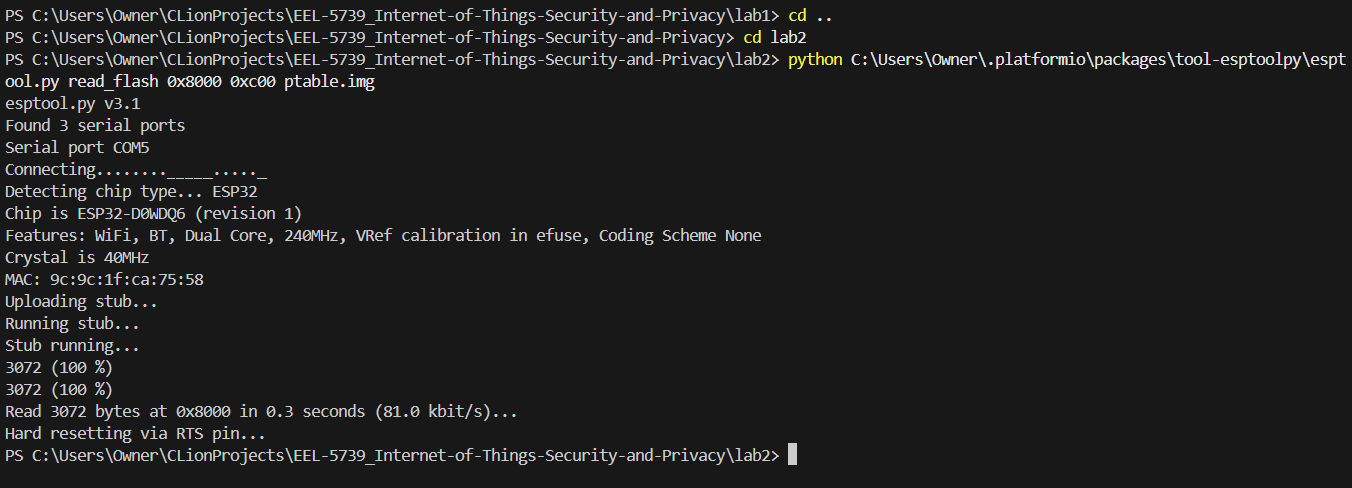
### Instructions

1. This is a group assignment. Each group can have at most two students. There should be a single submission for each group.
2. Answer each question following the original question. Do NOT delete the original question.
3. Refer to [Print screen](http://en.wikipedia.org/wiki/Print_screen) on how to take a screenshot.

**Questions**

1. Please refer to [1] for the use of *esptool*. The following command will retrieve the [partition table](https://docs.espressif.com/projects/esp-idf/en/v3.0-rc1/api-guides/partition-tables.html) of the IoT kit flash in the binary format:
   1. *python C:\Users\$USER\.platformio\packages\tool-esptoolpy\esptool.py read\_flash 0x8000 0xc00 ptable.img*

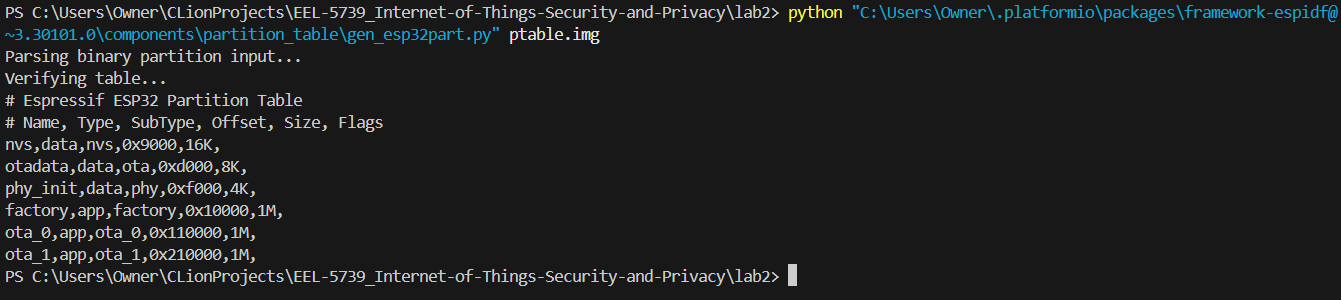
where 0x8000 is the start address of the partition table and 0xc00 is the length of the partition table. $USER is the username of the user that installed platformio. The binary partition table is saved in *ptable.img*. Please provide a screenshot of the command running. [10 point]



* 1. Please refer to [2] for the use of [*gen\_esp32part.py*](https://docs.espressif.com/projects/esp-idf/en/v3.0-rc1/api-guides/partition-tables.html). The following command will print out the partition table of our IoT kit in the CSV (comma-separated values) format. The partition table shows how the flash is partitioned.

*python C:\Users\$USER\.platformio\packages\framework-espidf\components\partition\_table\gen\_esp32part.py ptable.img*

Please provide a screenshot of the command running. [10 point]



* 1. Refer to [2] and explain the partition table that is printed out. [10 point]

1. The following command retrieves the whole flash content although it is also possible to refer to the partition table and select only the occupied part of the flash.
   1. *python C:\Users\$USER\.platformio\packages\tool-esptoolpy\esptool.py read\_flash 0 0x400000 flash\_contents.bin*

where *0* is the starting address and *0x400000* is the length of the flash of the [ESP32-WROOM-32 surface-mount module board](https://en.wikipedia.org/wiki/ESP32) that our IoT kit uses. The whole flash in the binary format is saved in *flash\_contents.bin*. Please provide a screenshot of the command. [10 point]

* 1. Students can use a [hex editor](https://www.tecmint.com/best-hex-editors-for-linux/) (e.g. wxhexeditor, imhex) to search the WiFi credentials in the flash dump. Please provide a screenshot of found WiFi credentials (e.g. password or key) using a hex editor. [10 point]
  2. In an extreme case, an adversary may attempt to write another firmware to the IoT kit with esptool.py. Please demonstrate writing back to the IoT kit with esptool.py. Students should modify the downloaded flash to change the password to be some combination of all group members’ names. Take care that the modified password is the same length as the original. Students must then try to write the changed flash dump to the IoT kit. Please provide a screenshot of the command running. Describe the results (e.g. observe the board, try communicating with it, repeat Lab 1, etc.) and discuss potential reasons for what was observed. [20 points]  
     Hint: Here is a command writing the flash\_contents\_modified.bin to the IoT kit

*python C:\Users\$USER\.platformio\packages\tool-esptoolpy\esptool.py write\_flash 0 flash\_contents\_modified.bin*

**References**

1. [esptool.py](https://github.com/espressif/esptool), Accessed on Mar. 6, 2021

1. [Partition Tables](https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/partition-tables.html?highlight=partition), Accessed on Feb 9, 2023