Table des matières

1) Attaques hors ligne

Avant de commencer le TP je me rend sur moodle et je récupère le dossier compressé

a) J'ouvre le fichier .ino récupéré sur moodle et je modifie les identifiants et mots de passe dans le programme :

```
#include <WiFi.h>
#include <WebServer.h>
#define CHANNEL 5
const char *ssid="ESP-SAM";
const char *pass="samy12345";
const char *www_realm="Authentification ESP32";
const char *www_username="saminternet";
const char *www_password="12345678";
IPAddress ip(192,168,1,1);
IPAddress gateway(192,168,1,254);
IPAddress subnet(255,255,255,0);
WebServer server(80);
```

Je fais attention à mettre un mot de passe sans doublons pour "www_password" sinon il sera impossible par la suite de le craquer.

Je fais attention aussi à ne pas choisir le même canal que les autres.

Je compile et je téléverse le programme puis je me connecte au wifi de l'ESP et je vérifie que ma page d'authentification fonctionne bien :

192.168.1.1/login

b) Dans arduino je vais dans fichier → préfrences puis je coche afficher les résultats détaillés pendant :

Maintenant je recompile et retéléverse mon programme et je cherche où se situe mon fichier .bin crée pendant la compilation :

```
/tmp/arduino_build_476503/AP_HTTP.ino.bin 0x8000
```

Mon fichier .bin se trouve donc dans /tmp/snap.arduino/tmp/arduino_build_476503

Je peux donc donner mon fichier .bin à mon voisin

c) Maintenant que j'ai le fichier binaire il faut que je retrouve le mot de passe que mon voisin a mis. Je peux utiliser hexdump pour pouvoir lire le fichier binaire. Je peux aussi utiliser la commande cut pour enlever le premier champ du fichier qui risque de me gêner et ensuite utiliser grep pour retrouver le mot de passe à 8 chiffres :

```
root@samuel-MS-7B51:/home/samuel/Téléchargements# hexdump -C
AP_HTTP.ino.bin |cut -f 2- -d " " |egrep [0-9]{8}
 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
 2e 00 37 38 39 34 35 36  31 32 00 67 75 69 67 75  |..78945612.guigu|
 74 75 76 77 78 79 7a 30  31 32 33 34 35 36 37 38  |tuvwxyz012345678|
 00 4e 41 4e 00 30 31 32  33 34 35 36 37 38 39 61  |.NAN.0123456789a|
 62 63 64 65 66 00 30 31  32 33 34 35 36 37 38 39  |bcdef.0123456789|
 78 36 30 30 30 30 30 30  30 20 2b 20 28 75 61 72  |x60000000 + (uar|
```

d) Pour cette question je suis obligé d'utiliser ma propre puce car je suis confiné à la maison

Je vais donc devoir retrouver mes informations (identifiant et mot de passe) mais sans avoir le fichier .bin qui m'a été donné au préalable.

Voici la commande passée par arduino lors du téléversement (Cette commande me permet de savoir où se situe le fichier esptool.py) :

```
python
/home/samuel/snap/arduino/50/.arduino15/packages/esp32/tools/esptool_py/
2.6.1/esptool.py --chip esp32 --port /dev/ttyUSB0 --baud 921600 --before
default_reset --after hard_reset write_flash -z --flash_mode dio --flash_freq 80m
--flash_size detect 0xe000
/home/samuel/snap/arduino/50/.arduino15/packages/esp32/hardware/esp32/1.
0.4/tools/partitions/boot_app0.bin 0x1000
/home/samuel/snap/arduino/50/.arduino15/packages/esp32/hardware/esp32/1.
```

```
0.4/tools/sdk/bin/bootloader_qio_80m.bin 0x10000 /tmp/arduino_build_476503/
AP_HTTP.ino.bin 0x8000 /tmp/arduino_build_476503/AP_HTTP.ino.partitions.bin
```

Maintenant je fais un flash_id pour obtenir des infos sur ma carte et ensuite les mettre dans un fichier .bin à part :

```
root@samuel-MS-7B51:~# python
/home/samuel/snap/arduino/50/.arduino15/packages/esp32/tools/esptool_py/
2.6.1/esptool.py flash_id
esptool.py v2.6
Found 2 serial ports
Serial port /dev/ttyUSB0
Connecting....
Detecting chip type... ESP32
Chip is ESP32D0WDQ5 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding
Scheme None
MAC: 84:0d:8e:e6:7f:7c
Uploading stub...
Running stub...
Stub running...
Manufacturer: 20
Device: 4016
Detected flash size: 4MB
Hard resetting via RTS pin...
```

Maintenant que j'ai les infos je peux envoyer les données dans un fichier .bin que je lirai ensuite :

```
root@samuel-MS-7B51:~# python
/home/samuel/snap/arduino/50/.arduino15/packages/esp32/tools/esptool_py/
2.6.1/esptool.py read_flash 0x10000 0x40000 flash_contents.bin
```

Je peux finir en lisant le contenu du fichier :

```
root@samuel-MS-7B51:~# hexdump -C flash_contents.bin |cut -f 2- -d " " |
egrep [0-9]{8}
 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
 30 30 30 30 30 30 30 30  30 30 30 30 30 30 30 30  |0000000000000000|
 31 32 33 34 35 36 37 38  00 73 61 6d 69 6e 74 65  |12345678.saminte|
 78 79 7a 30 31 32 33 34  35 36 37 38 39 2b 2f 00  |xyz0123456789+/.|
 00 30 31 32 33 34 35 36  37 38 39 61 62 63 64 65  |.0123456789abcde|
 66 00 30 31 32 33 34 35  36 37 38 39 41 42 43 44  |f.0123456789ABCD|
 28 28 28 28 28 28 30 78  36 30 30 30 30 30 30 30  |((((((0x60000000|
00040000
```

Je retrouve bien mon mot de passe

e) Je commence par effacer la mémoire de ma puce :

```
root@samuel-MS-7B51:~# python
/home/samuel/snap/arduino/50/.arduino15/packages/esp32/tools/esptool_py/
2.6.1/esptool.py erase_flash
esptool.py v2.6
Found 2 serial ports
Serial port /dev/ttyUSB0
Connecting....
Detecting chip type... ESP32
Chip is ESP32D0WDQ5 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding
Scheme None
MAC: 84:0d:8e:e6:7f:7c
Uploading stub...
Running stub...
Stub running...
Erasing flash (this may take a while)...
Chip erase completed successfully in 8.8s
Hard resetting via RTS pin...
```

Une fois supprimé je retéléverse le programme dans ma carte

f) Je fais une copie de la mémoire flash :

```
root@samuel-MS-7B51:~# python
/home/samuel/snap/arduino/50/.arduino15/packages/esp32/tools/esptool_py/
2.6.1/esptool.py -p /dev/ttyUSB0 -b 921600 read_flash 0 0x400000
flash_contents.bin
esptool.py v2.6
Serial port /dev/ttyUSB0
Connecting........_
Detecting chip type... ESP32
Chip is ESP32D0WDQ5 (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding
Scheme None
MAC: 84:0d:8e:e6:7f:7c
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 921600
Changed.
4194304 (100 %)
4194304 (100 %)
```

```
Read 4194304 bytes at 0x0 in 52.7 seconds (637.1 kbit/s)...
Hard resetting via RTS pin...
```

Zone mémoire pour le PSK :

```
root@samuel-MS-7B51:~# hexdump -C flash_contents.bin | egrep
'\|.*samy12345'
0000a540  73 61 6d 79 31 32 33 34  35 00 00 00 00 00 00 00  |
samy12345.......|
```

Zone mémoire pour le SSID :

```
root@samuel-MS-7B51:~# hexdump -C flash_contents.bin | egrep '\|.*ESP-SAM'
0000a4c0  07 00 00 00 45 53 50 2d  53 41 4d 00 c7 04 00 00  |....ESP-SAM.....|
00011190  31 32 33 34 35 00 45 53  50 2d 53 41 4d 00 25 30  |12345.ESP-
SAM.%0|
```

g) Je change le SSID et le mot de passe et je retéléverse le programme. Maintenant je regarde quels sont les 3 mémoires tampons :

```
root@samuel-MS-7B51:~# hexdump -C flash_contents.bin | egrep '\|.*ESP-SAM-
TEST'
0000a860  0c 00 00 00 45 53 50 2d  53 41 4d 2d 54 45 53 54  |....ESP-SAM-
TEST|
```

```
root@samuel-MS-7B51:~# hexdump -C flash_contents.bin | egrep
'\|.*samy123456789'
0000a8e0  73 61 6d 79 31 32 33 34  35 36 37 38 39 00 00 00  |
samy123456789...|
```

2) Attaque en fonctionnement à distance

a)

3) Attaque sur un serveur HTTPS