COMMUNICATION SYSTEMS

Configurable Wi-Fi AP

IMAD SALMI , ROGER MIQUEL & ROBERT MIRALLES

CREATING A WI-FI ACCESS POINT

1. Install hostapd and dnsmasq software in Ubuntu:

```
sudo apt-get install hostapd
sudo apt-get install dnsmasq
```

2. Create a configuration file: /etc/hostapd/hostapd.conf

```
sudo emacs /etc/hostapd/hostapd.conf
```

The content of this file is explained in the following slide

interface=wlp1s0 → wlp1s0 is the name of the wireless interface of our laptop.

driver=nl80211 → Specifies the wireless driver to be used by hostapd. The nl80211 driver is commonly used on Linux systems.

ssid=Sisom_AP → Is the name of our Wi-Fi network. This is what users will see when they scan for available networks.

hw_mode=g → Specifies the Wi-Fi mode, such as "**g**” for 2.4 GHz band or “**a**” for 5 GHz band.

channel=6 → Specifies the Wi-Fi channel to be used. We have to choose a channel that is not heavily used in our environment.

macaddr_acl=0 → Controls MAC address filtering. **“0”** means no filtering, and **“1”** means allow only MAC addresses specified in the **“accept_mac_file”**.

auth_algs=1 → Specifies the authentication algorithms. 1 enables WPA (Wi-Fi Protected Access).

ignore_broadcast_ssid=0 → Controls whether the SSID should be hidden (1) or broadcasted (0).

wpa=2 → Specifies the version of WPA to use. 2 indicates WPA2.

wpa_passphrase=Socrative → Specifies the pre-shared key (password) for WPA-PSK authentication.

wpa_key_mgmt=WPA-PSK → Specifies the key management protocols used for authentication.

wpa_pairwise=TKIP → Specifies the pairwise (unicast) cipher suites for WPA.

rsn_pairwise=CCMP → Specifies the pairwise (unicast) cipher suites for RSN (Robust Security Network), used in WPA2.

ctrl_interface=/var/run/hostapd → Configurates a parameter that specifies the directory path for the control interface socket.

Procedure for the creation of AP

- Deactivate the NetworkManager

```
sudo service NetworkManager stop
```

- Assign an IP to our interface

```
sudo ip address add 192.168.1.1/24 dev wlp1s0
```

- Activate our wireless interface

```
sudo ip link set wlp1s0
```

Configuring dnsmasq

In order to dynamically assign IP addresses to the different users that connect to our AP we use the same dnsmasq.

To do this you need to add this in the configuration file /etc/dnsmasq.conf:

```
interface = wlp1s0

dhcp-range=192.168.1.2, 192.168.1.10, 12h
```

And we reset the tool to be able to apply the changes made.

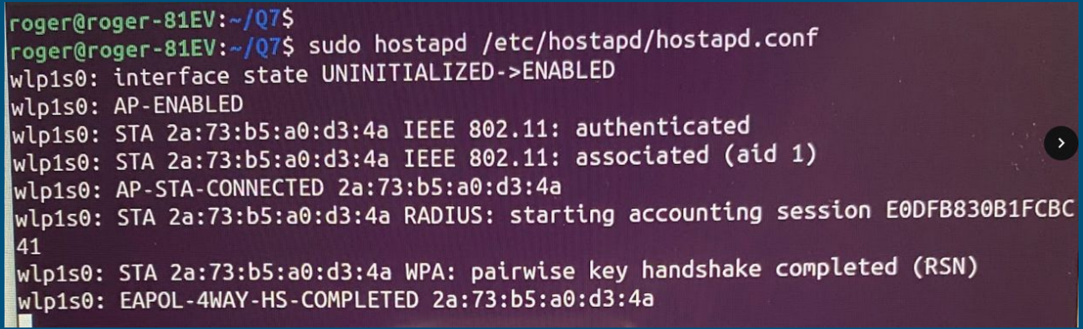
```
sudo service dnsmasq restart
```

We can check for possible errors by doing:

```
systemctl status dnsmasq.service
```

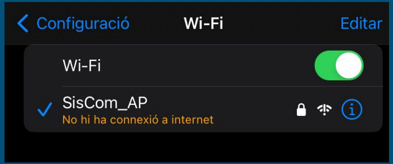
- Activate the AP

sudo hostapd /etc/hostapd/hostapd.conf



```
roger@roger-81EV:~/Q7$
roger@roger-81EV:~/Q7$ sudo hostapd /etc/hostapd/hostapd.conf
wlp1s0: interface state UNINITIALIZED->ENABLED
wlp1s0: AP-ENABLED
wlp1s0: STA 2a:73:b5:a0:d3:4a IEEE 802.11: authenticated
wlp1s0: STA 2a:73:b5:a0:d3:4a IEEE 802.11: associated (aid 1)
wlp1s0: AP-STA-CONNECTED 2a:73:b5:a0:d3:4a
wlp1s0: STA 2a:73:b5:a0:d3:4a RADIUS: starting accounting session E0DFB830B1FCBC
41
wlp1s0: STA 2a:73:b5:a0:d3:4a WPA: pairwise key handshake completed (RSN)
wlp1s0: EAPOL-4WAY-HS-COMPLETED 2a:73:b5:a0:d3:4a
```

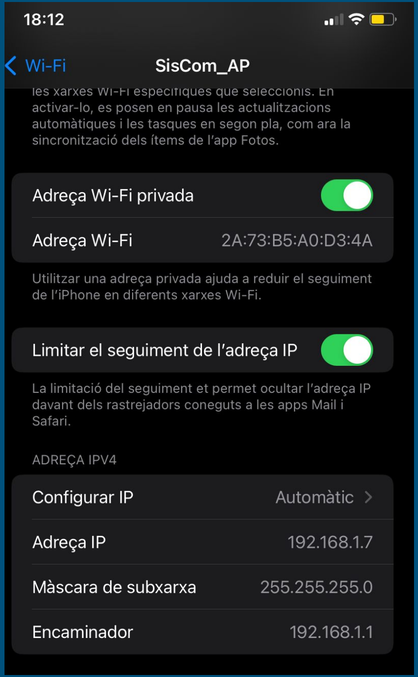
And we connect with our mobile:



Wi-Fi ●

✓ SisCom_AP
No hi ha connexió a internet

We can see the addresses that have been correctly assigned to us:



les xarxes Wi-Fi específiques que seleccionis. En activar-lo, es posen en pausa les actualitzacions automàtiques i les tasques en segon pla, com ara la sincronització dels ítems de l'app Fotos.

Adreça Wi-Fi privada	●
Adreça Wi-Fi	2A:73:B5:A0:D3:4A

Utilitzar una adreça privada ajuda a reduir el seguiment de l'iPhone en diferents xarxes Wi-Fi.

Limitar el seguiment de l'adreça IP ●

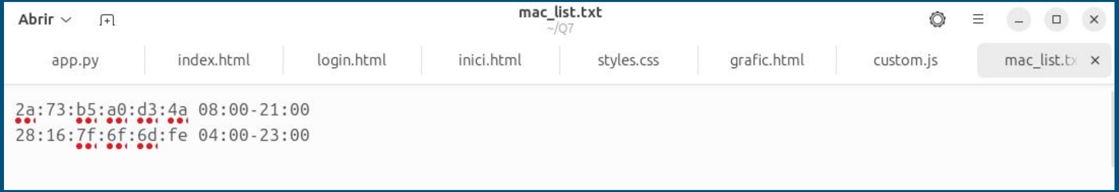
La limitació del seguiment et permet ocultar l'adreça IP davant dels rastrejadors coneguts a les apps Mail i Safari.

ADREÇA IPV4

Configurar IP	Automàtic >
Adreça IP	192.168.1.7
Màscara de subxarxa	255.255.255.0
Encaminador	192.168.1.1

“Time of day” based control

We have created a file named “mac_list.txt” which contains the MAC’S allowed and their time permission. (You can save in the same folder)



```
2a:73:b5:a0:d3:4a 08:00-21:00
28:16:7f:6f:6d:fe 04:00-23:00
```

Then we look with the following command if there is an user connected with us:

sudo arp -a

We obtain their MAC and IP and we look in that file if he can connected or not

“Time of day” based control

In order to remove a user whose MAC address is not in the list or does not meet the time range we use these two commands:

```
sudo arp -d <IP>
sudo hostapd_cli -i wlp1s0 deauthenticate <MAC>
```

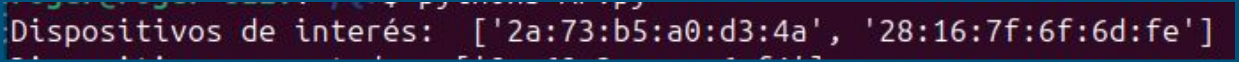
We repeat the control process every 5 seconds, this value can be determined very easily and conveniently

Monitoring

In order to have a list of devices of interest, we have chosen to define the list in the program itself. It could have been done in a file like we did with the list of MACs and their times.

This list is printed on the screen once the program starts.

We also notify you on screen when one of these devices is connected. We could not opt for any other way, since the computer we used to make the AP only has the wireless interface and therefore does not have the Internet when the AP is activated.



```
Dispositivos de interés:  ['2a:73:b5:a0:d3:4a', '28:16:7f:6f:6d:fe']
```

Link of the video demonstration

https://youtu.be/ZqLch0Js73Q

In the video we can see how we have the computer that acts as an AP and we have two different mobile phones to try to connect to the AP.

We can see that when the first mobile is connected, it is disconnected by the python control program.

As for the second mobile, it connects, it shows us on the screen that it has connected, since it is a device of interest and is disconnected by the monitoring program. When the time range in the "mac_list" file is changed, the mobile is no longer disconnected.