



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

Departament d'Enginyeria Minera, Industrial
i TIC

SC PROJECT

CONFIGURABLE WI-FI AP

Sistemes de Comunicació

Zhou Pan
Enginyeria de Sistemes TIC
Quadrimestre 7
Curs 2023-2024

Índex

1	Introduction	2
2	Design	2
3	Implementation	2
4	Results	4
5	Conclusió	7

1 Introduction

In this project we have to develop a Wi-Fi AP that the access control is based on "Time of day":

Based on a list of MAC addresses and time periods when those MAC addresses are allowed to use services. The other times they are not allowed to use service. And when the end time arrives, only the corresponding users are disconnected.

Furthermore, we have to provide a way of monitoring. Notify the admin when the devices of interest connect. And it should allow a way to provide the list of devices of interest. Device is identified by its MAC address.

2 Design

First of all, a laptop is used to create a Wi-Fi access point, software access point tool 'hostapd'. To provide IP address to STA clients, it uses 'dnsmasq'.

I have created a python program to achieve "Time of day" access control and monitoring.

The strategy is: create a wifi AP with a white list (only those MAC addresses are allowed to connect to the AP). So any other MAC address that is not in the list won't be able to connect it. When a MAC of this list connects, it checks if it is a correct time period. If not, it kicks out. If fine, it allows the connection and it notifies the admin. When the end time arrives, it disconnects only the corresponding user.

To check if the end time has arrived or not, every 'x' time it gets all the current users of the AP, and it checks if each user has arrived its end time or not.

3 Implementation

To achieve all this goals, i have create a python program to manage it.

First of all, define the interface that you want to use as AP, SSID and password for the AP, and a free port for dns. And then we can start the access point configuration:

```
1 interface = "wlo1"
2 SSID_AP = "MyAP"
3 Password_AP = "123123123"
4 dns_port=5353
5 try:
6     start_access_point(interface, SSID_AP, Password_AP, dns_port)
7     monitor = Monitor()
8     ...
```

In the function of start access point, we do a clean up (kill any existing hostapd or dnsmasq process in case it's running). Configure the wireless interface. Extract the allowed MAC list. Write the configuration of the hostapd. And then run hostapd with this configuration. Also, run the dnsmasq service.

```

1     ...
2     hostapd_conf = f"""interface={interface}
3 driver=nl80211
4 ssid={ssid}
5 hw_mode=g
6 channel=6
7 wpa=2
8 wpa_passphrase={password}
9 wpa_key_mgmt=WPA-PSK
10 wpa_pairwise=TKIP
11 rsn_pairwise=CCMP
12 ctrl_interface=/var/run/hostapd
13 ctrl_interface_group=0
14 macaddr_acl=1
15 accept_mac_file=/tmp/hostapd.accept"""
16
17     with open("/tmp/hostapd.conf", "w") as conf_file:
18         conf_file.write(hostapd_conf)
19
20     subprocess.Popen(["sudo", "hostapd", "/tmp/hostapd.conf"]) #run
    hostapd
21     ...

```

This only creates the access point. To manage the "Time of day" access control and the monitoring, i have created a tkinter. After creating the access point, it runs a tkinter interface.

I have implemented a function that gets all the current users of the AP. If a new user gets in, and it's an interested device, it notifies the admin by popping up a message in the window. Also, it checks if they are in the correct time period. If not, they are disconnected.

```

1     ...
2     def update_connected_devices():
3         # show list connected devices
4         connected_devices_listbox.delete(0, 'end') # Clear the
    existing list
5         current_users = get_connected_users(interface)
6         for i in self.connected_interest_users:
7             if i not in current_users:
8                 self.connected_interest_users.remove(i) #the device of
    interest has disconnect by itself
9         for user in current_users:

```

```

10         start_time = list_allowed_mac[user][0]
11         end_time = list_allowed_mac[user][1]
12         if not check_time(start_time, end_time):
13             disconnect_mac(user)
14         else:
15             connected_devices_listbox.insert('end', user)
16             if user in list_interest_device and user not in self.
connected_interest_users:
17                 self.connected_interest_users.append(user)
18                 self.notify_admin(f"new interested device
connecting: {user}")
19             # Schedule the function to run again after a delay
20             self.root.after(1000, update_connected_devices)
21 ...

```

To show the interested devices, there is a button in the bottom of the window when clicking it, it pop up a new window with a list of device of interest.

```

1     def show_interested_devices(self):
2         top = Toplevel()
3         top.title("Interested Devices")
4
5         title_label = Label(top, text="List of Interested Devices")
6         title_label.pack(padx=80, pady=10)
7
8         listbox = Listbox(top, width=25)
9         for mac_address, device_name in list_interest_device.items():
10             listbox.insert('end', f"{device_name}: {mac_address}")
11
12         listbox.pack(padx=80, pady=10)

```

4 Results

Starting the program it starts the hostapd service (creates the access point) and it pops up a window like this:

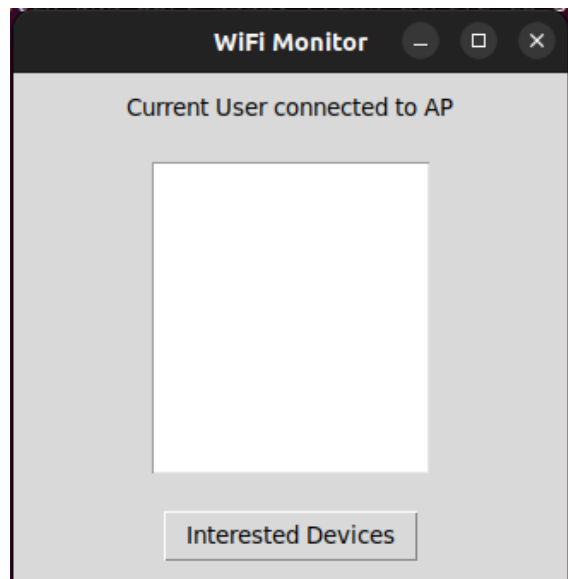


Figura 1: AP monitor

It shows the current user connected to the AP
After a new client connects it, if it's an interested device, it pops up a message to notify the admin:

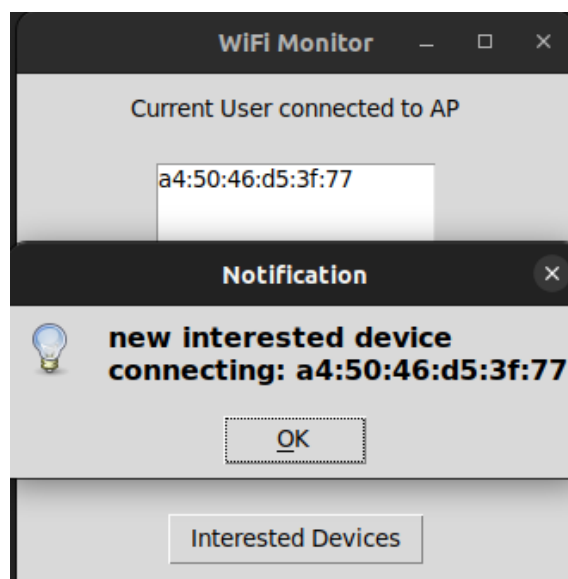


Figura 2: Notify admin

And if it's not an interested device, it does not pop up this message. Only it updates the list:

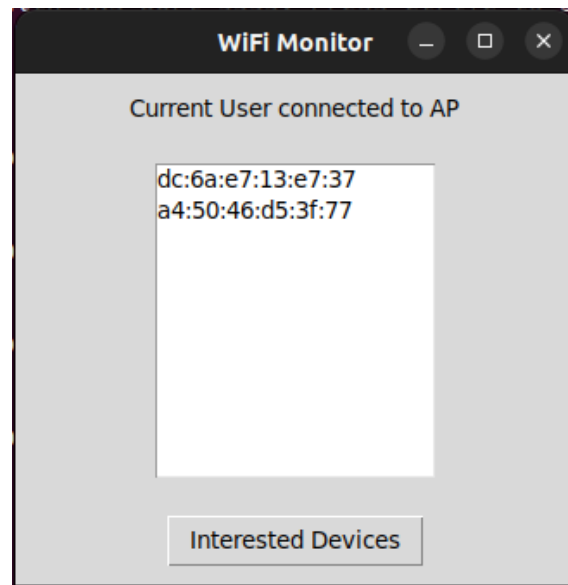


Figura 3: Current clients connected to the AP

When the end time arrives, it kicks out the corresponding user. And the other still maintain connected.

By clicking the button 'Interested Devices', it pops up a new window that shows the list of interested devices:

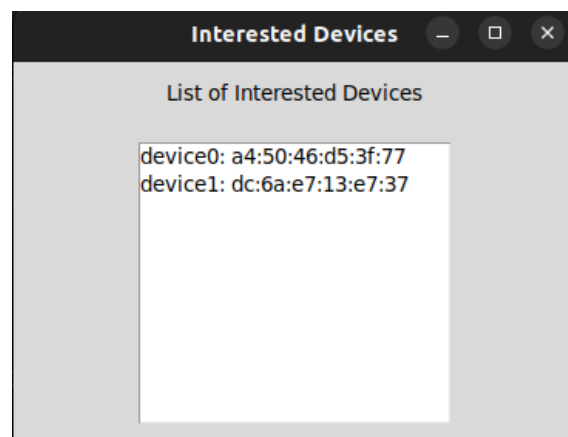


Figura 4: List of interested device

Video link: <https://youtu.be/UbctQKdAOA8>

As we can see in the video, when the end time arrives, only the corresponding user disconnects.

5 Conclusió

In conclusion, i have learnt about how to create a wifi access point using a software tool 'hostapd'.

At first it was difficult. Cause of all the configurations and drivers, etc. which causes errors. And i had to search a lot. Once i have achieved to create a simple wifi AP, the other is to think and implement, most of the part. That wasn't easy at all.

I wanted to create a blacklist and a whitelist to manage the "Time of day"based access control. When a MAC address's end time arrives, the AP kicks out and it moves to a blacklist, which cannot connect again to the AP. When its start times arrives, it moves to the whitelist, which can connect again.

The way I implemented, if the user is in the whitelist but it is in the incorrect time and tries to connect, the AP allows the connection but it kicks out immediately. The way i wanted to implement, it won't allow the connection at the first place.

I have tried to update the whitelist and the blacklist while AP is running without closing it, but i couldn't achieve it. I have found some ways which can reconfigure the lists. However, according to the tests i have done, it does not work. After updating a MAC to the blacklist and removing to the whitelist, this MAC still can connect.

So, the other alternative was closing the AP and reconfigure again the lists and start. But this way, it will disconnect all the user. Hence, not a good option.