Project: Configurable Wi-Fi AP

Zhou Pan

Communication Systems

Introduction

- Develop a Wi-Fi AP that has “Time of day” based access control
  - 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 connect
  - When the “end time” arrives, only the corresponding users are disconnected
- Monitoring: Notify the “admin” when the “devices of interest” connect
  - System should allow a way to provide the list of “devices of interest”
  - Device is identified by its MAC address

Design

- Tools used: laptop with a wireless interface, ‘hostapd’, ‘dnsmasq’
- Python program to manage time based control and monitoring (with tkinter)
  - 0. Define lists of allowed MAC and device of interest
  - 1. Set up the configuration (interface, hostapd with a whitelist, dnsmasq)
  - 2. Run service hostapd and dnsmasq
  - 3. Run the monitor (tkinter)
  - 4. Check every x time which devices are connected to the AP
  - 5. Then check the period time of each device → if the end time arrives, it kicks out

Implementation

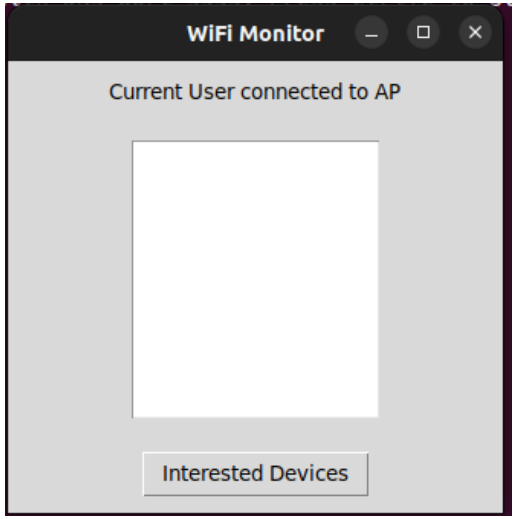
- I have implemented different functions:
  - def start_access_point(interface, ssid, password, dns_port=5353): this function starts the access point and dnsmasq
  - def check_time(start_time, end_time): return True if the current time is inside start_time and end_time
  - def get_connected_users(interface): return a list of the current user connected to the AP
  - def disconnect_mac(client_mac): disconnect the corresponding user mac
- Monitor: class Monitor(threading.Thread):
  - def show_interested_devices(self): pop up a new window that shows the interested devices
  - def notify_admin(self, msg): pop up a message box with a given message to notify admin
  - def update_connected_devices(): check current user connected to the AP and decide if it's allowed or not to the connection

- This specific function is the main to manage the time access control and the notification.

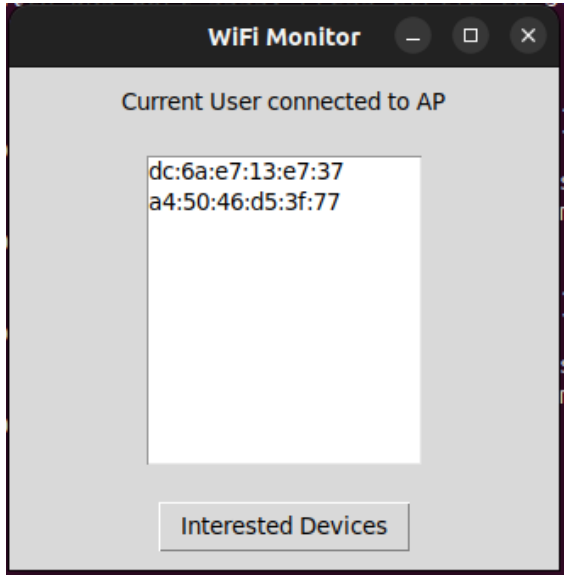
```
def update_connected_devices():
    # show list connected devices
    connected_devices_listbox.delete(0, 'end')  # Clear the existing list
    current_users = get_connected_users(interface)
    for i in self.connected_interest_users:
        if i not in current_users:
            self.connected_interest_users.remove(i) #the device of interest has disconnect by itself
    for user in current_users:
        start_time = list_allowed_mac[user][0]
        end_time = list_allowed_mac[user][1]
        if not check_time(start_time, end_time):
            disconnect_mac(user)
        else:
            connected_devices_listbox.insert('end', user)
            if user in list_interest_device and user not in self.connected_interest_users:
                self.connected_interest_users.append(user)
                self.notify_admin(f"new interested device connecting: {user}")
    # Schedule the function to run again after a delay
    self.root.after(1000, update_connected_devices)
```

Results

- Open the Python program and it pops up this window:



- After a client connects, it shows a list of the current user connected to the AP:

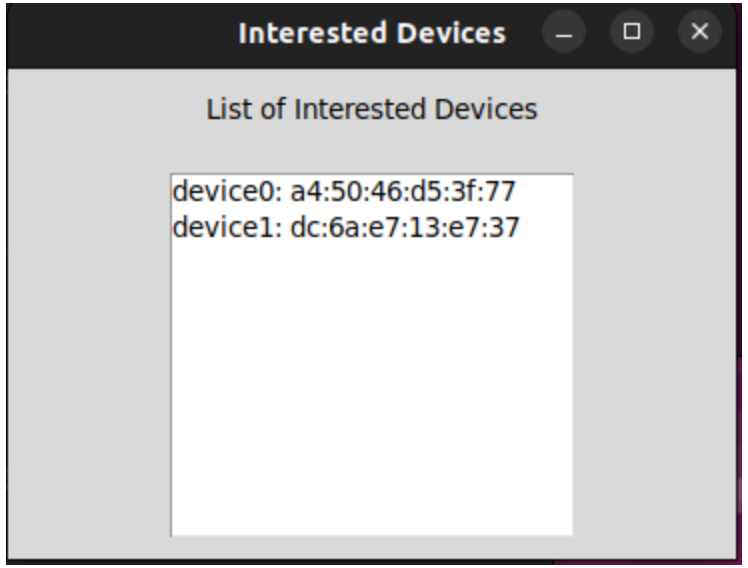


Results

- If a client is in the list of interested devices, it pops up a message box to notify the admin:

Results

- By clicking the button at the bottom ‘Interested Devices’, it shows a new window with a list of device of interest, name and its mac address:



Results

- Also, we can see information in the terminal too:
  - Which device has connected, disconnected etc.

```
Access Point started with DNS port 5353. Press Ctrl+C to stop.
wlo1: interface state UNINITIALIZED->ENABLED
wlo1: AP-ENABLED
wlo1: STA a4:50:46:d5:3f:77 IEEE 802.11: authenticated
wlo1: STA a4:50:46:d5:3f:77 IEEE 802.11: associated (aid 1)
wlo1: AP-STA-CONNECTED a4:50:46:d5:3f:77
wlo1: STA a4:50:46:d5:3f:77 RADIUS: starting accounting session 6052C9837488C620
wlo1: STA a4:50:46:d5:3f:77 WPA: pairwise key handshake completed (RSN)
wlo1: EAPOL-4WAY-HS-COMPLETED a4:50:46:d5:3f:77
wlo1: STA dc:6a:e7:13:e7:37 IEEE 802.11: authenticated
wlo1: STA dc:6a:e7:13:e7:37 IEEE 802.11: associated (aid 2)
wlo1: AP-STA-CONNECTED dc:6a:e7:13:e7:37
wlo1: STA dc:6a:e7:13:e7:37 RADIUS: starting accounting session 3FE89CD4A81D8C29
wlo1: STA dc:6a:e7:13:e7:37 WPA: pairwise key handshake completed (RSN)
wlo1: EAPOL-4WAY-HS-COMPLETED dc:6a:e7:13:e7:37
Selected interface 'wlo1'
                        wlo1: AP-STA-DISCONNECTED a4:50:46:d5:3f:77
OK
```

Conclusion

- I have to study this new tool, hostapd, to create a Wi-Fi access point. I had to search a lot.
- I think the way I implemented is not the best option. 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.
- 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. I have tried and couldn’t achieve it.