Evil Twin Tutorial

Tags

Evil Twin Attack

Report Title:

Demonstration of Evil Twin Attack using Airgeddon for Cybersecurity Awareness

Name: Faez Nazari

Course: Computer Science (Computer Security)

Institution: Universiti Teknikal Malaysia Melaka (UTeM)

Date: [30 June 2025]

Cover Description:

> This report presents a practical demonstration of an Evil Twin wireless attack utilizing the Airgeddon framework. The objective is to raise cybersecurity awareness by simulating a rogue Wi-Fi scenario designed to capture sensitive credentials through a deceptive captive portal. The findings aim to highlight the social engineering risks and technical vulnerabilities users face when connecting to unsecured wireless networks.

1. Introduction

🎯 Purpose of the Experiment

The objective of this lab exercise is to demonstrate the implementation of an Evil Twin attack using Airgeddon in a controlled environment. By emulating a fake wireless access point, this test simulates how attackers can deceive users into disclosing sensitive information such as login credentials. The goal is to increase awareness of Wi-Fi-based social engineering threats and encourage best practices in wireless security hygiene.

🚨 Relevance to Wi-Fi Security Awareness

Public and corporate Wi-Fi networks are frequently targeted by attackers using rogue access points. Evil Twin attacks are particularly dangerous because they exploit user trust in familiar SSIDs, combined with psychological manipulation via look-alike captive portals. Demonstrating this technique helps highlight how easily unsuspecting users can fall victim to credential theft, and reinforces the importance of avoiding unsecured or spoofed networks.

🕵️ What is an Evil Twin Attack?

An Evil Twin attack involves setting up a malicious wireless access point that mimics the SSID and configuration of a legitimate Wi-Fi network. When victims connect to this spoofed AP, they are redirected to a fake login page or captive portal. Any credentials or data submitted are harvested by the attacker. This form of attack is particularly effective in public areas like universities, airports, or cafes where users tend to connect without verifying authenticity.

🧰 Tools & Technologies Used

The following tools and platforms were used to perform the Evil Twin simulation:

Tool	Description
Kali Linux	A Linux distribution used for penetration testing and digital forensics
Airgeddon	A multi-use bash script for automated wireless attacks
mdk4	Tool used to perform high-speed deauthentication and DoS attacks
hostapd	Utility to create a software-based access point
lighttpd	Lightweight web server used to host the captive portal
Phone (Android)	Used as the victim device to simulate real-world interaction

Lab Setup & Environment

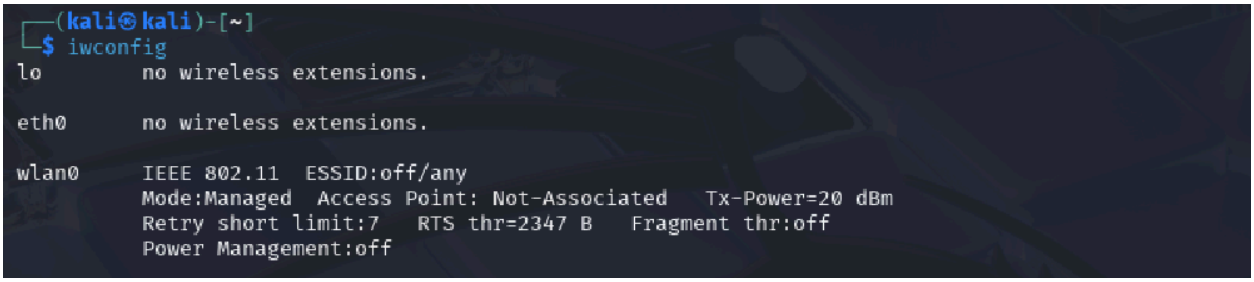
- Devices used (laptop, external Wi-Fi adapter if used)
- Software: Kali version, Airgeddon version
- Network setup (e.g., SSID `Mr.Whitehat`, real AP and client device like Harith's Note20)

Attack Walkthrough

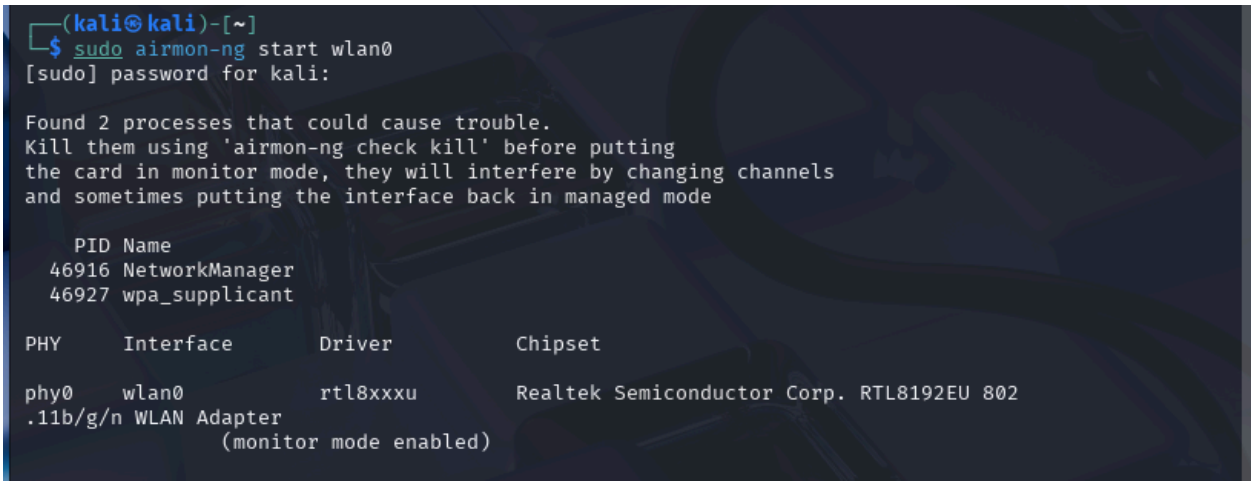
Selecting the Wireless Interface

1 - make your wireless interface in monitor mode

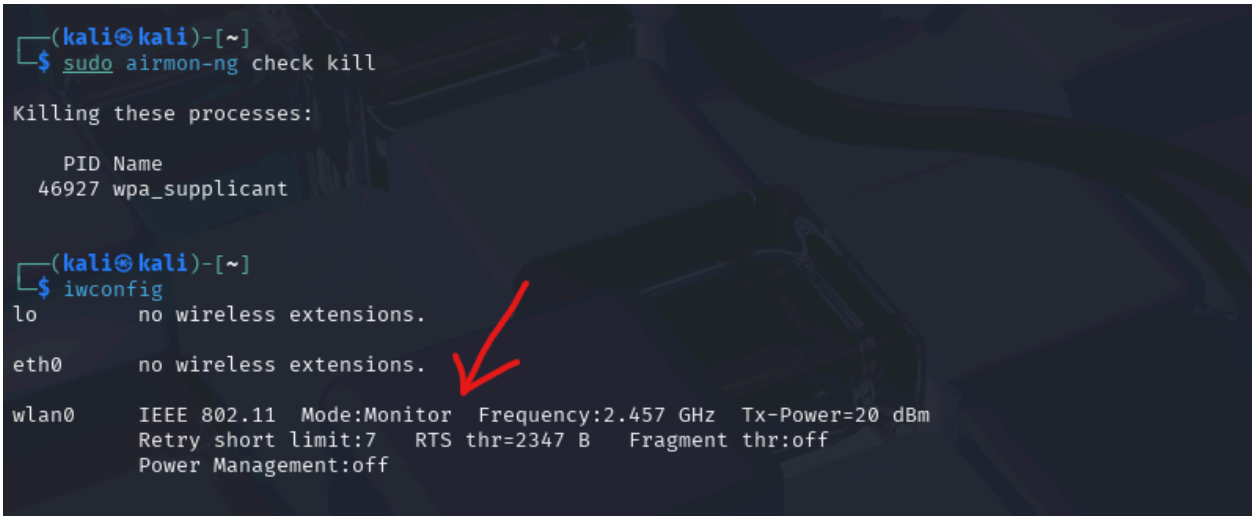
```
iwconfig
```




```
sudo airmon-ng start wlan0
```

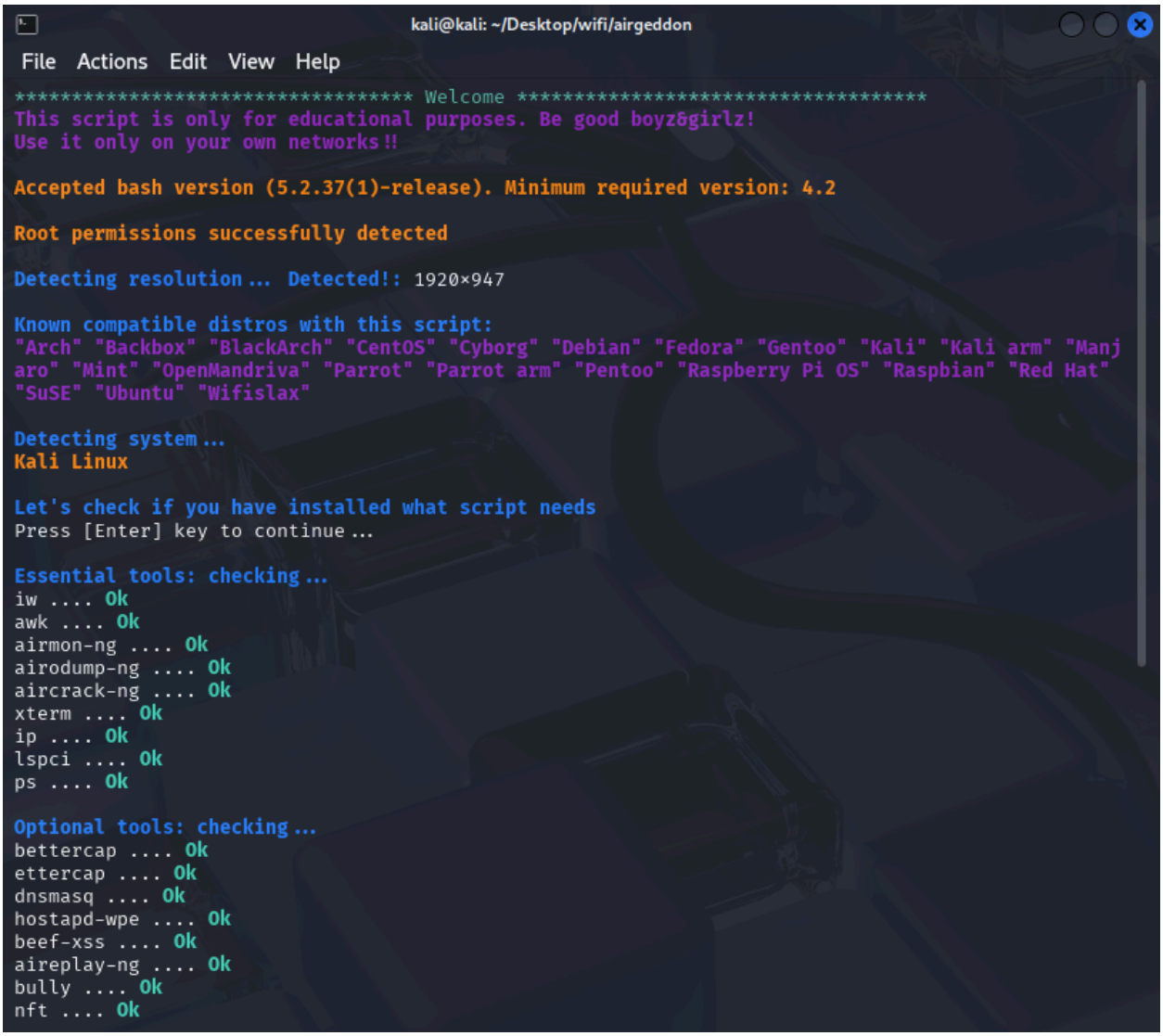


sudo airmon-ng check kill



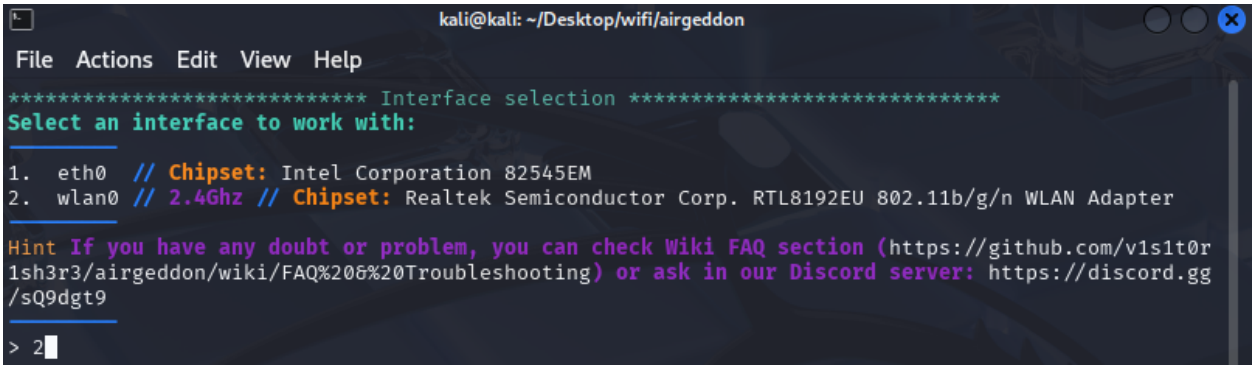
2. Step 0: Downloading Airgeddon from GitHub

git clone https://github.com/v1s1t0r1sh3r3/airgeddon.git
cd airgeddon
sudo bash airgeddon.sh

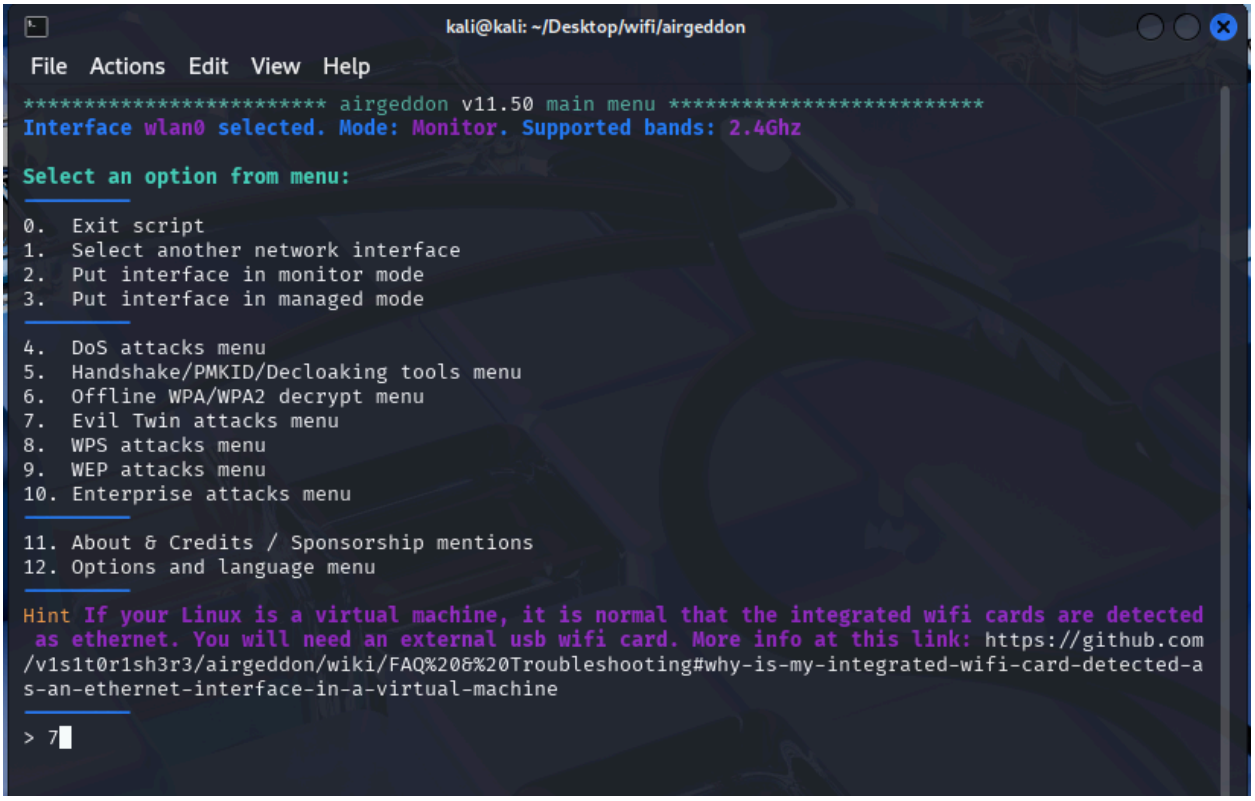


just click **Enter**

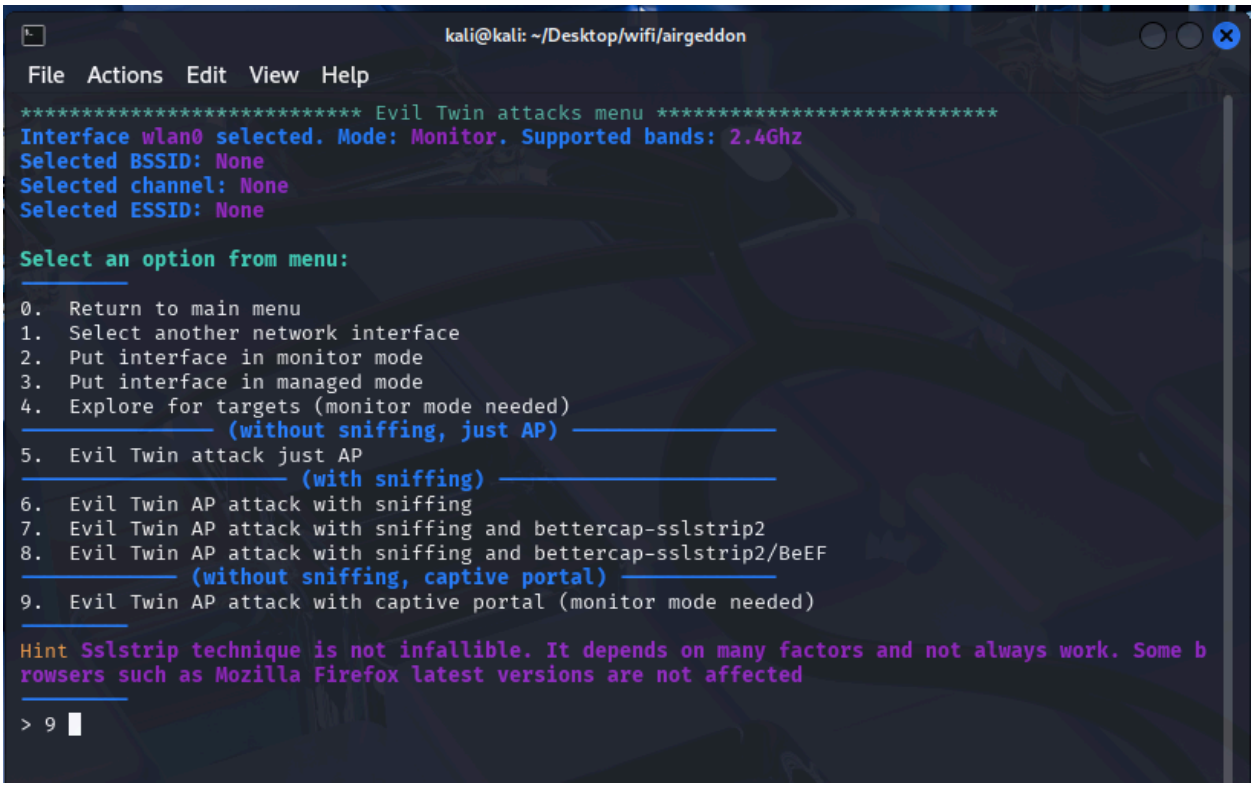
Selecting the Wireless Interface



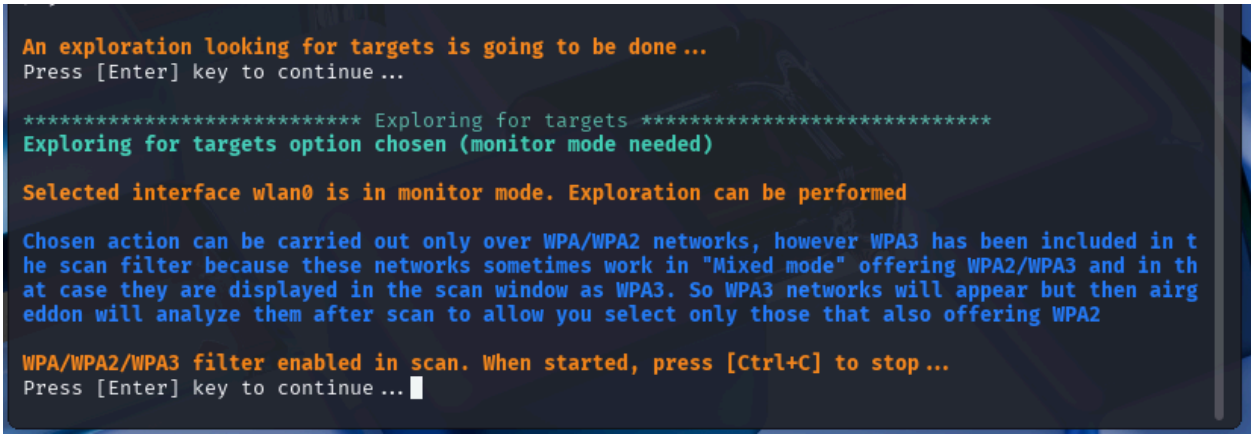
choose **2** and enter



choose no 7 and enter



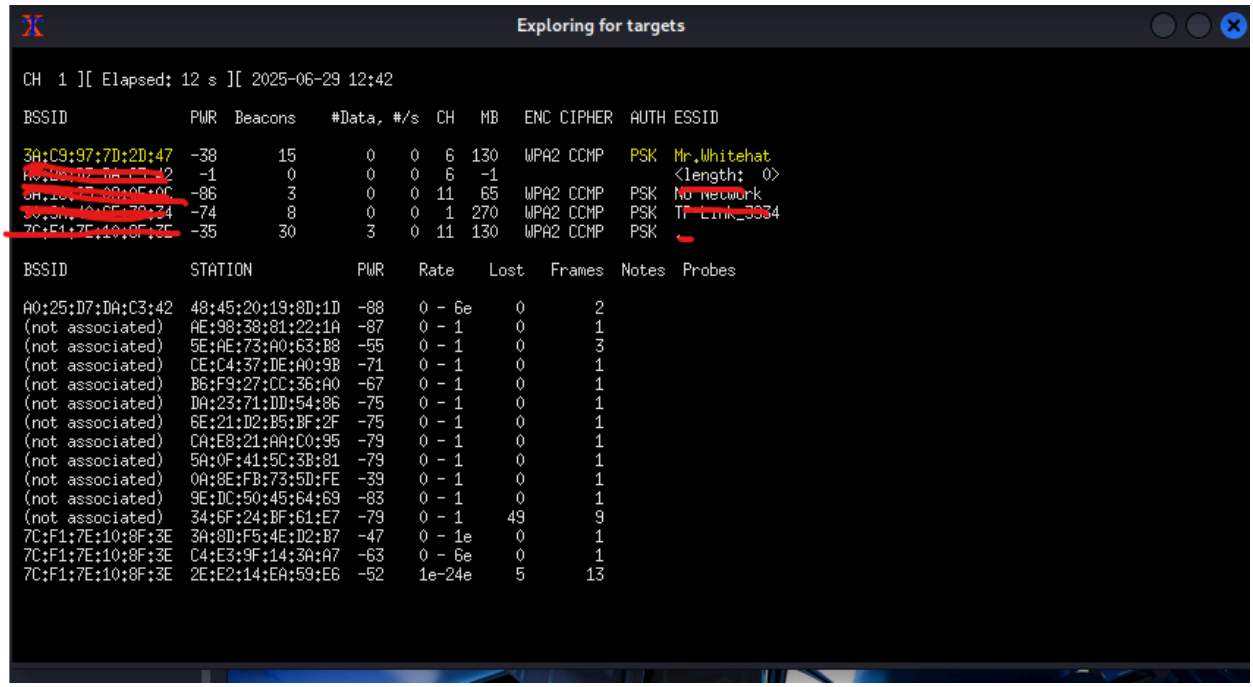
choose no 9



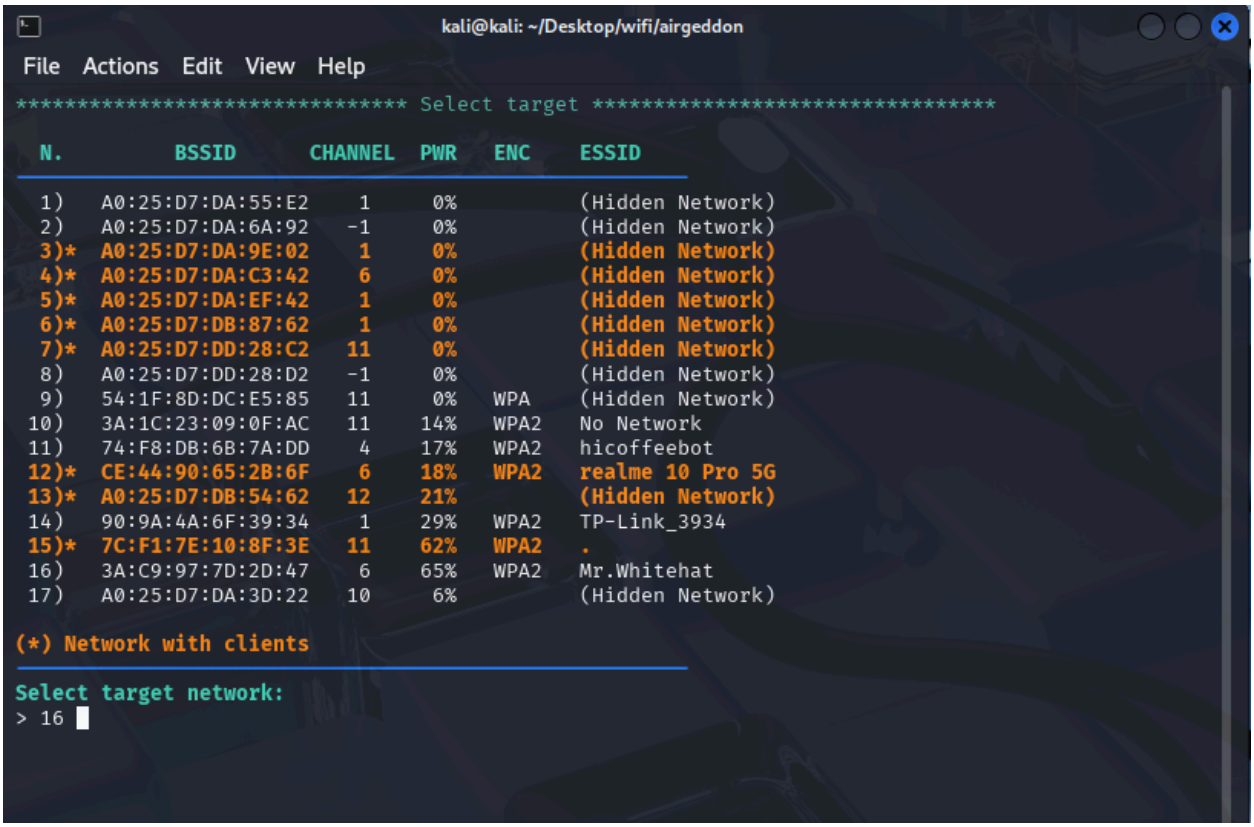
just click enter

4. Scanning for Targets

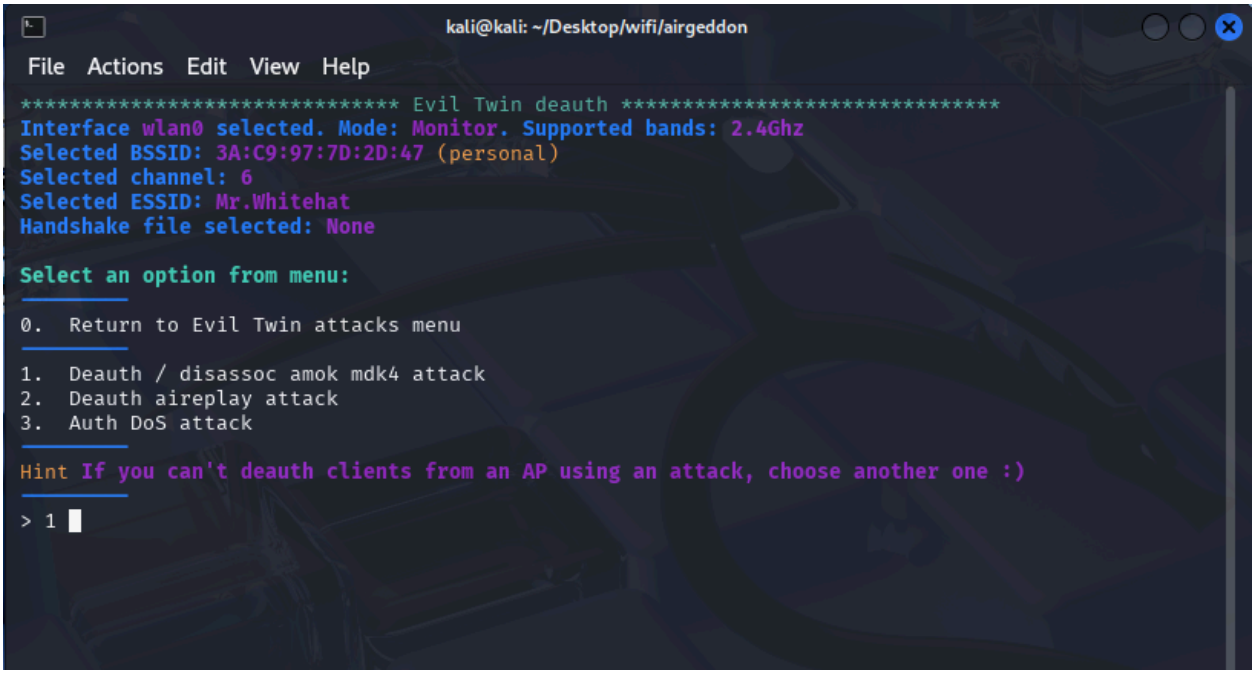
- Airgeddon launches `airodump-ng` scan
- Look for the target SSID (`Mr.Whitehat`)
- Press `Ctrl+C` when it appears
- Select the SSID from the lis



after found your AP



choose your ssid



choose no **1**

💣 Option 1: **mdk4 attack**

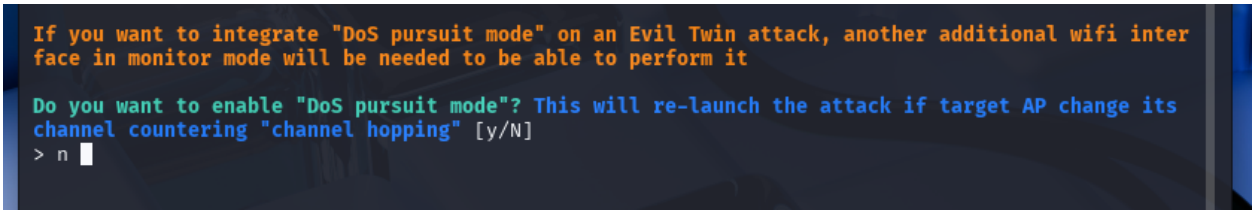
- Sends a *flood* of deauth/disassoc packets
- Works super fast and aggressive—but **only if** `mdk4` **is installed properly**
  ✅ Use this **if you've already got mdk4 installed successfully**

🪂 Option 2: **aireplay attack**

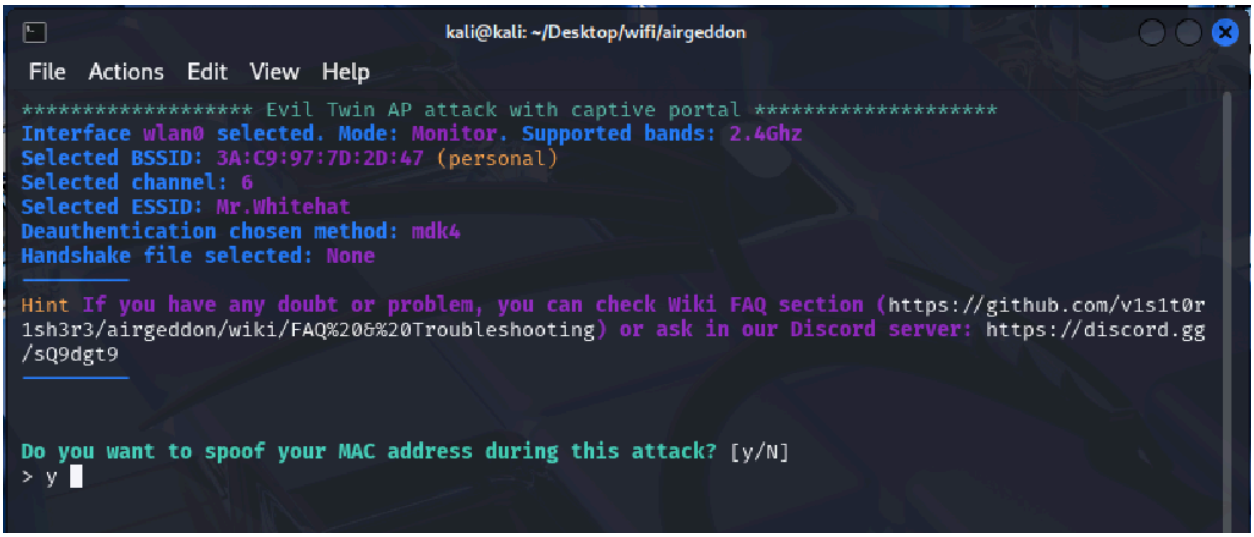
- Classic and reliable
- Sends targeted deauth packets
  ✅ Use this **if mdk4 didn't install** or isn't behaving
  ✅ *You already tried this earlier—so if it seemed weak, try mdk4 this time instead*

🚫 Option 3: **Auth DoS attack**

- Spams the AP's association table
- Less effective in this Evil Twin context
  🔻 Not ideal for handshake capture or client disconnection in your case



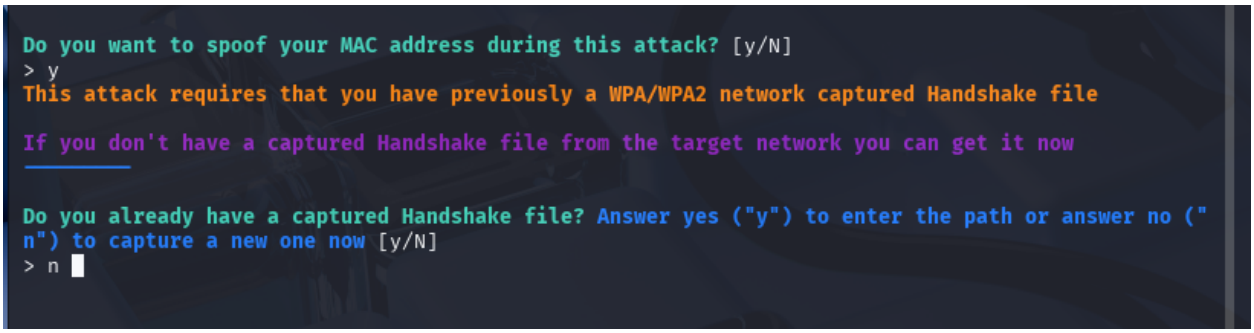
just choose **n**



choose **y**

🥷 `y` — Yes, spoof MAC

- Airgeddon will randomize your MAC address.
- Useful for stealth, evasion, or simulating an attacker.
- Highly recommended for realistic Evil Twin simulations.
  ✅ **Best choice for lab realism** — go with `y`

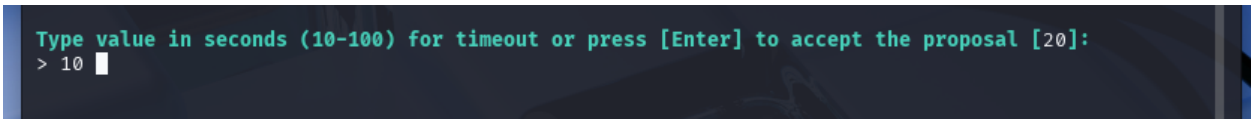


type **n**

and press **Enter**

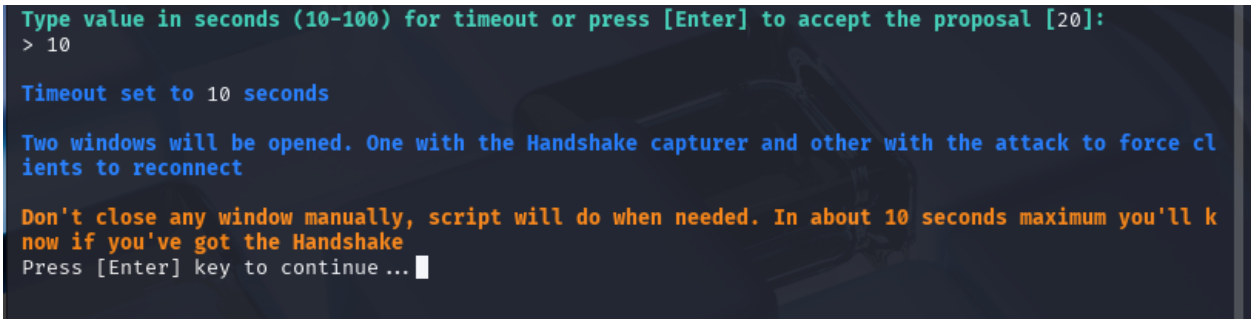
Since you haven't captured a handshake yet, this option tells Airgeddon to:

- Start **deauthing** your phone (so it disconnects from the real `Mr.Whitehat`)
- Simultaneously **listen** for the WPA2 handshake when the phone tries to reconnect



Yep—typing `10` will shorten the handshake capture timeout to 10 seconds. That means Airgeddon will:

- Start listening for WPA/WPA2 handshakes from your **Mr.Whitehat** access point
- Try to deauth your phone and wait **10 seconds** for a reconnection

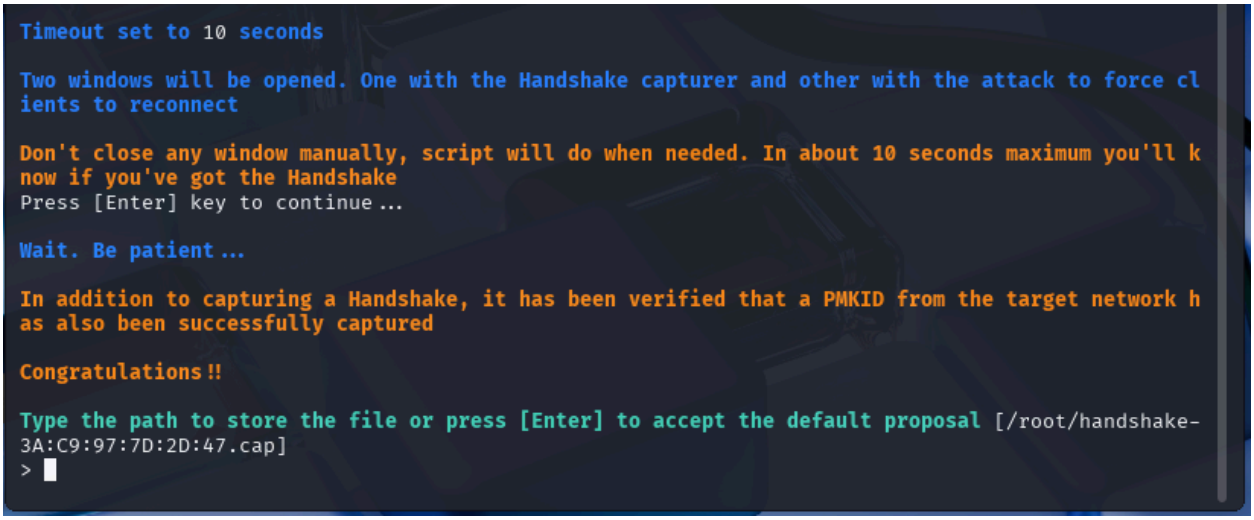


just click **Enter**

Then Airgeddon will launch into:

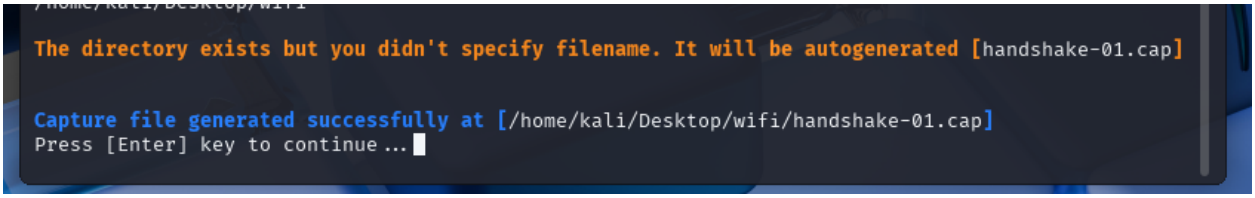
- Deauthing real clients
- Firing up `hostapd` for the fake AP

- Launching the captive portal to capture credentials



Boom! You did it — handshake **and** PMKID captured? That's like catching the fish *and* the secret sauce in one cast 🐟🧪

Now that you've got the credentials capture stored (default path: `/root/handshake-EA:BD:E8:25:13:75.cap`)

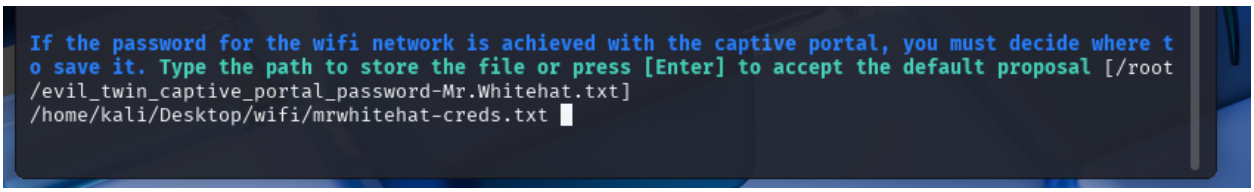


i will save it to

`/home/kali/Desktop/wifi`

my wifi folder

Once you choose your path, Airgeddon will move forward with the Evil Twin setup and spin up the fake AP and phishing portal.



Right now, Airgeddon is asking:

> “If the password is captured via the fake login page, where should it be saved?”

🗂️ Here's What You Can Do:

✅ Option 1: Use Default Path

Just press:

`Enter`

This will save any captured credentials (from your Evil Twin captive portal) to:

`/root/evil_twin_captive_portal_password-Mr.Whitehat.txt`

You can move or rename that file later if needed.

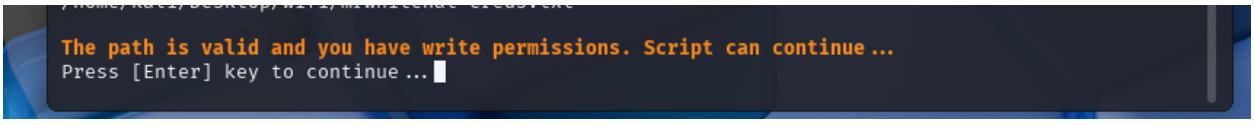
💾 Option 2: Save it to Your Desktop (Organized Style)

If you'd prefer to keep everything neatly in one place, type:

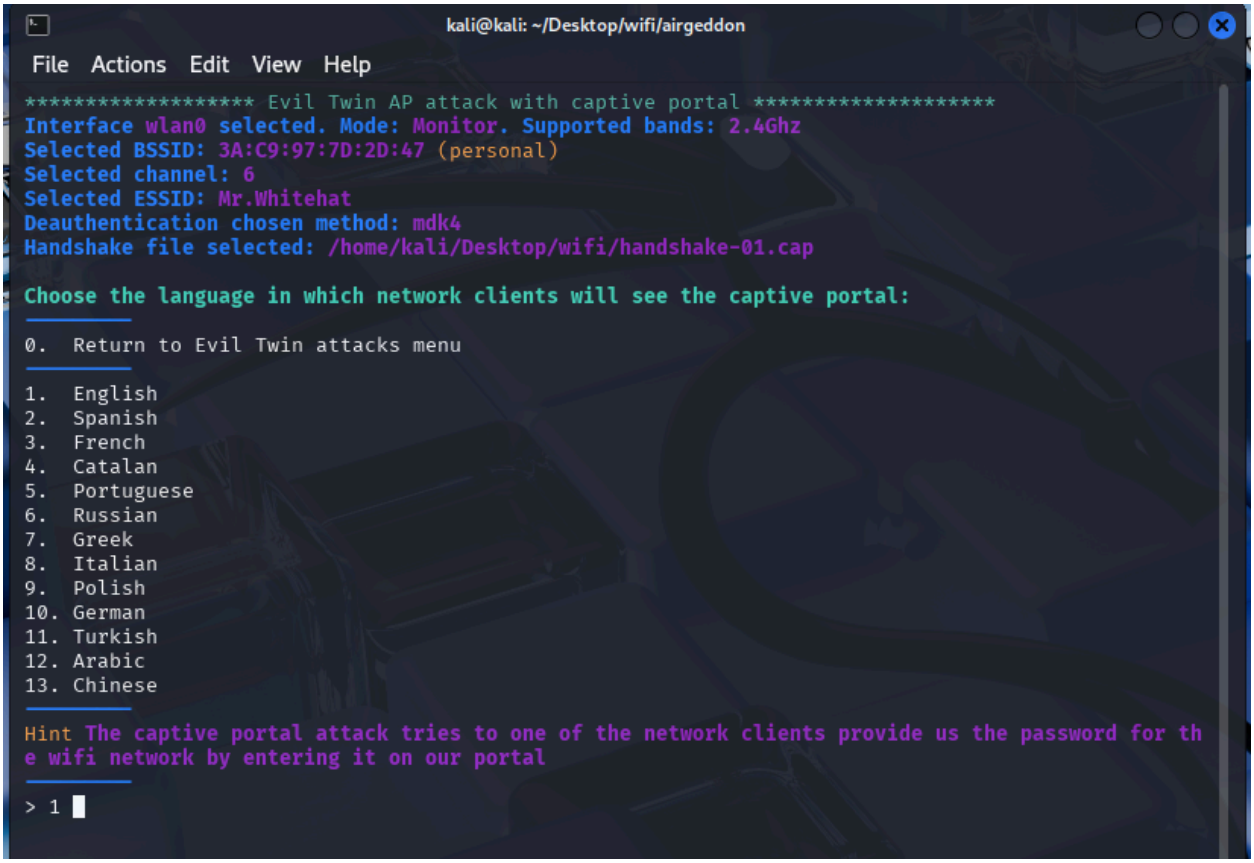
bash

`/home/kali/Desktop/wifi/mrwhitehat-creds.txt`

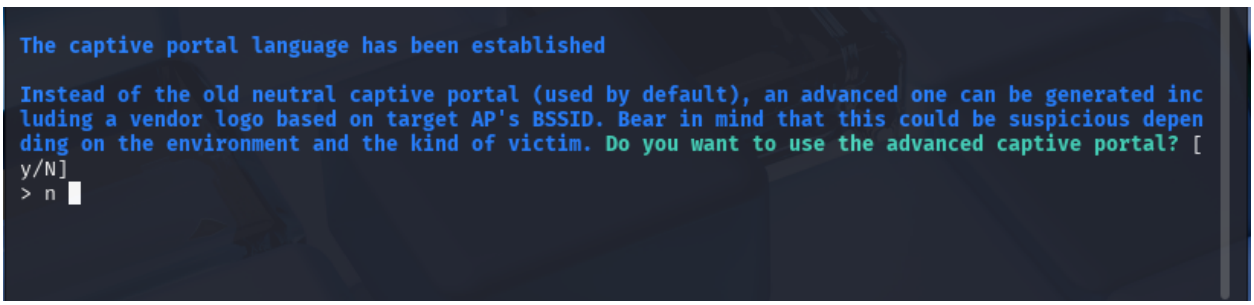
And hit Enter — now both your `.cap` file and any future credentials will live in the same `wifi` folder.



click E**nter**



then click **1**

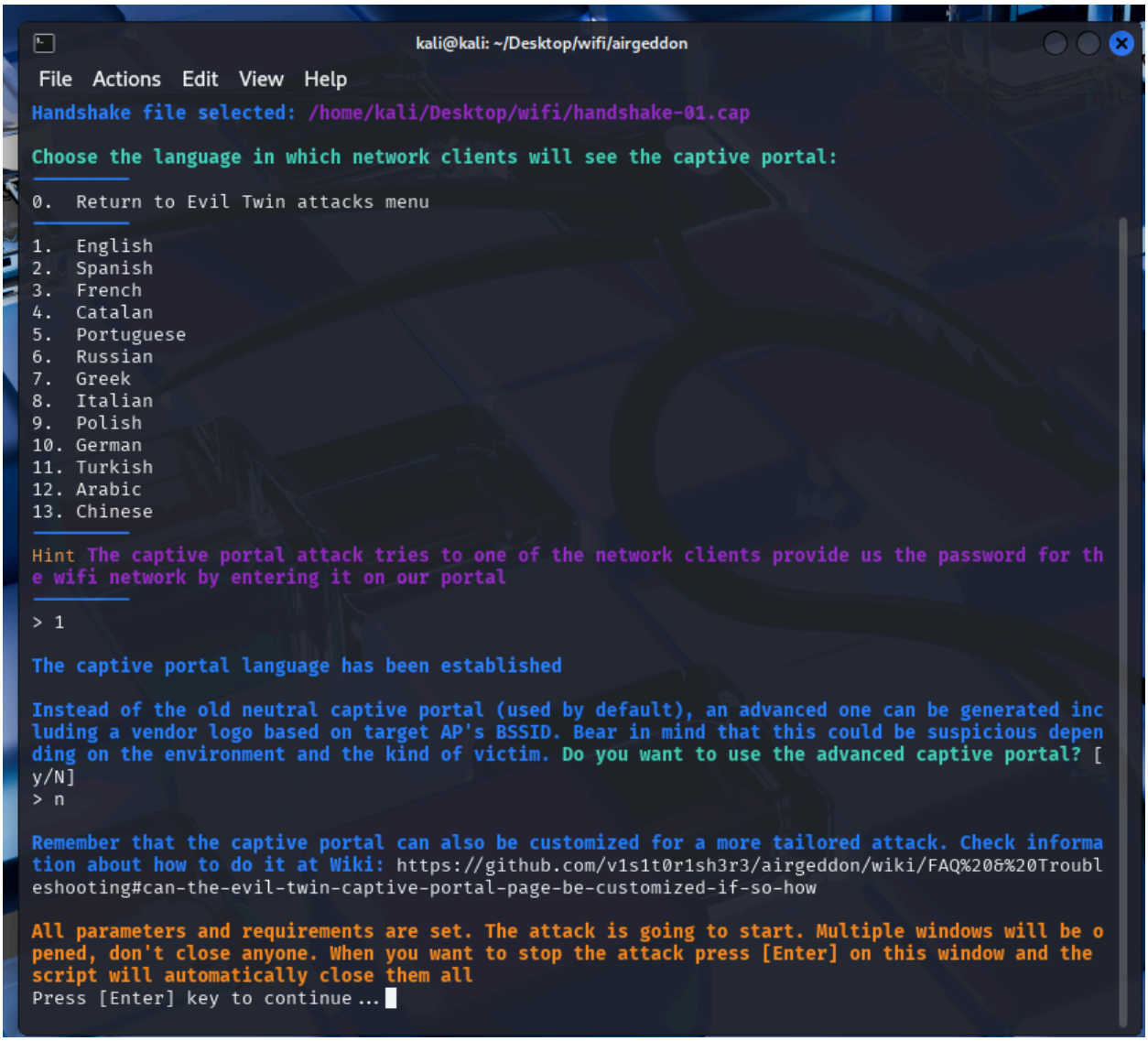


just choose **n**

option `n` or Press Enter — Stick with the **Default Neutral Page**

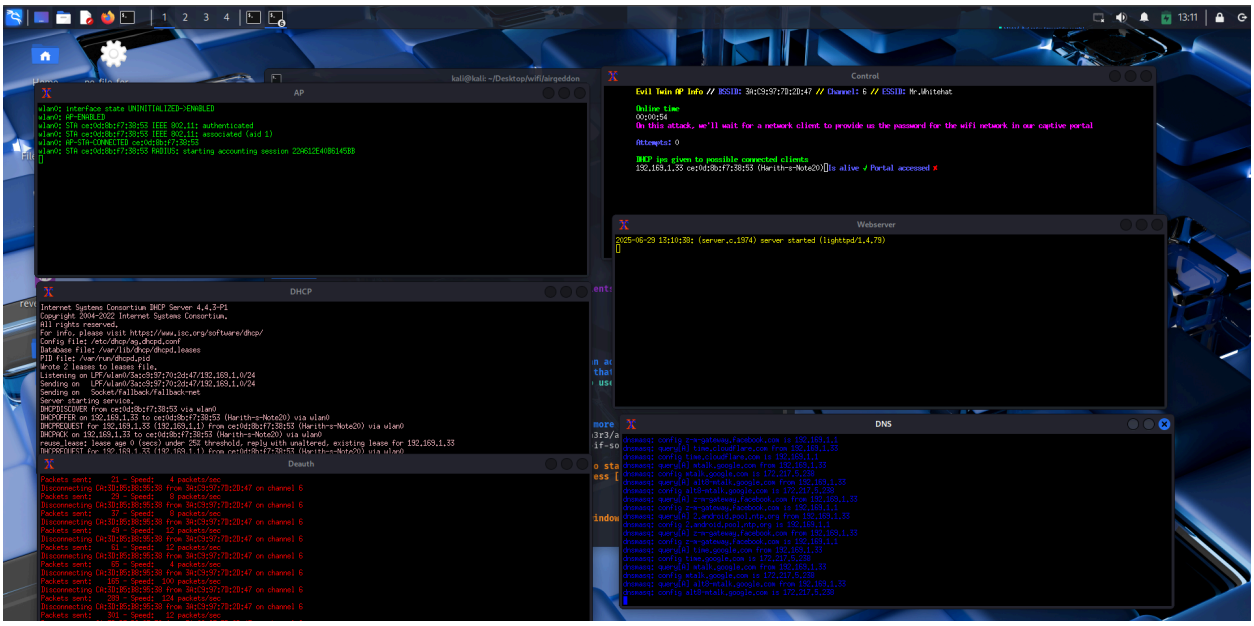
- Clean, generic HTML login form
- No branding, no suspicion
- Perfect for lab testing and credential capture

✅ Ideal if you're just experimenting locally, especially since you've already turned off the real `Mr.Whitehat` AP



you may click **Enter**

then your screen may look like this



What's Running Now

- `hostapd`: broadcasting the **fake AP** `Mr.Whitehat`
- `lighttpd`: serving the **captive portal** (generic HTML login page)
- `mdk4`: hammering the real AP to **deauth clients**
- Logging: ready to **save any creds** entered by a device

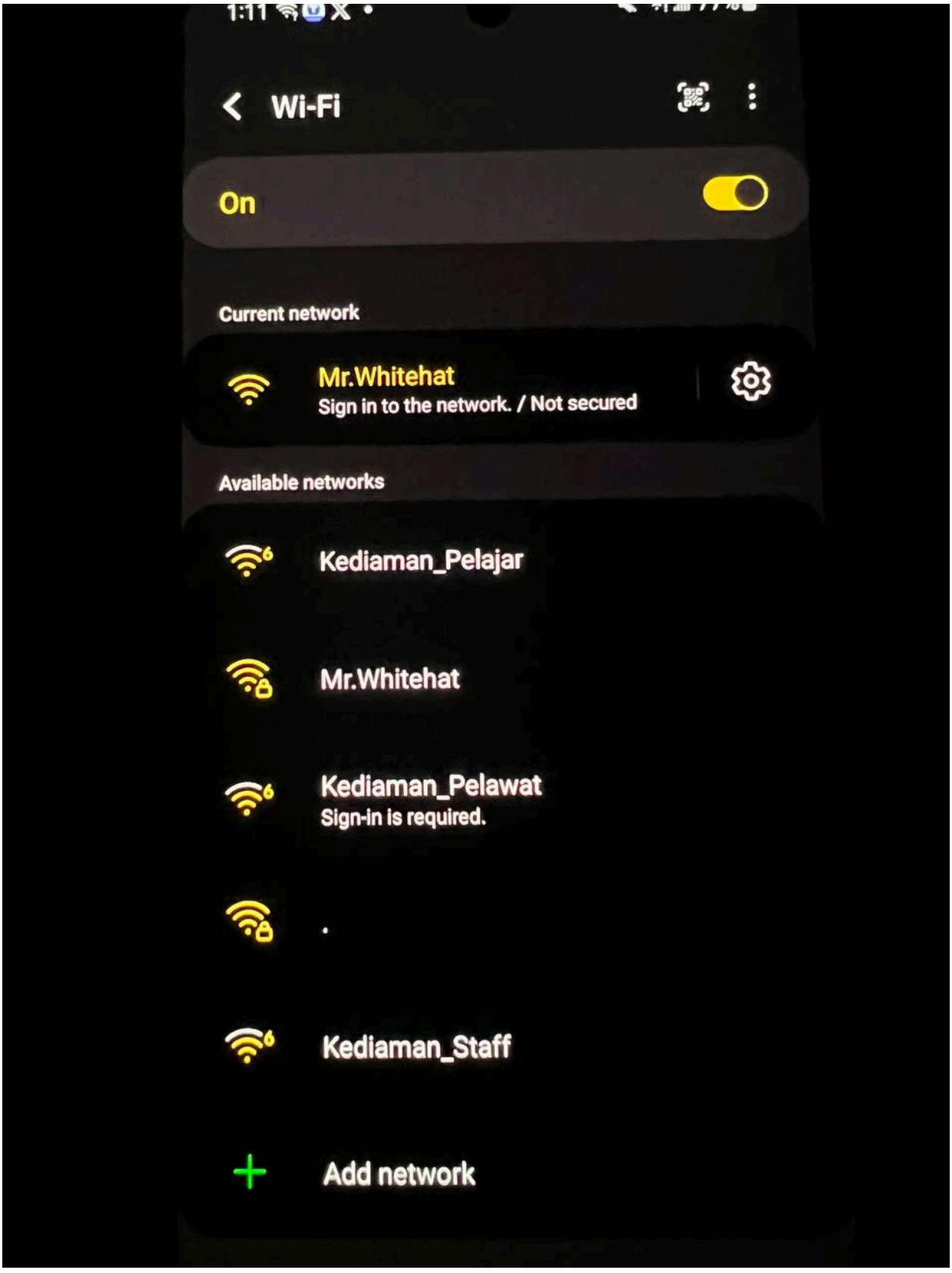
📲 What You Should Do

1. **On your phone**:
    - Go to Wi-Fi settings
    - Look for `Mr.Whitehat` — it should appear **open (no password)**

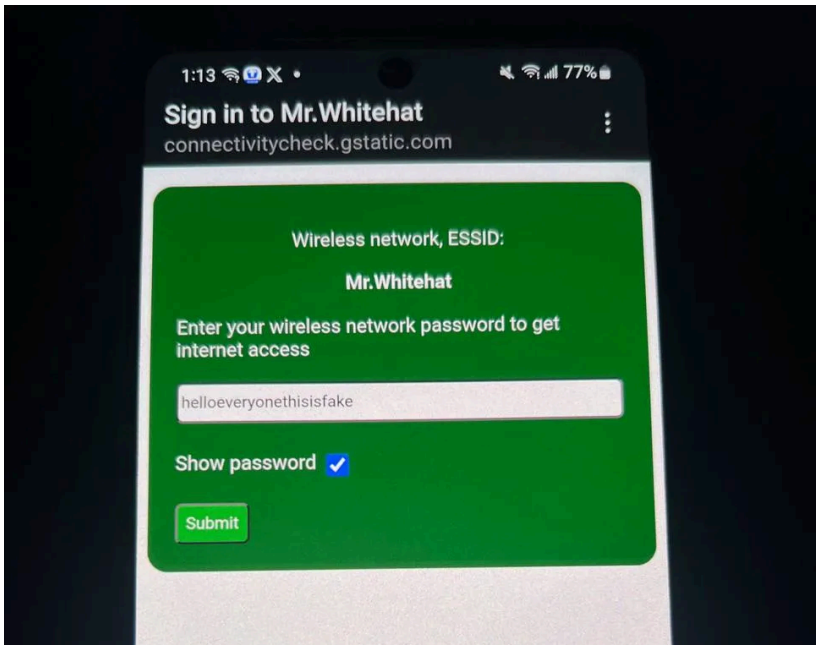
    Connect your phone to `Mr.Whitehat`

    - No password required (that's your rogue AP doing its magic)

in Victim’s Side


Wi-Fi
On
Current network
Mr.Whitehat
Sign in to the network. / Not secured
Available networks
Kediaman_Pelajar
Mr.Whitehat
Kediaman_Pelawat
Sign-in is required.
.
Kediaman_Staff
Add network

choose the fake AP

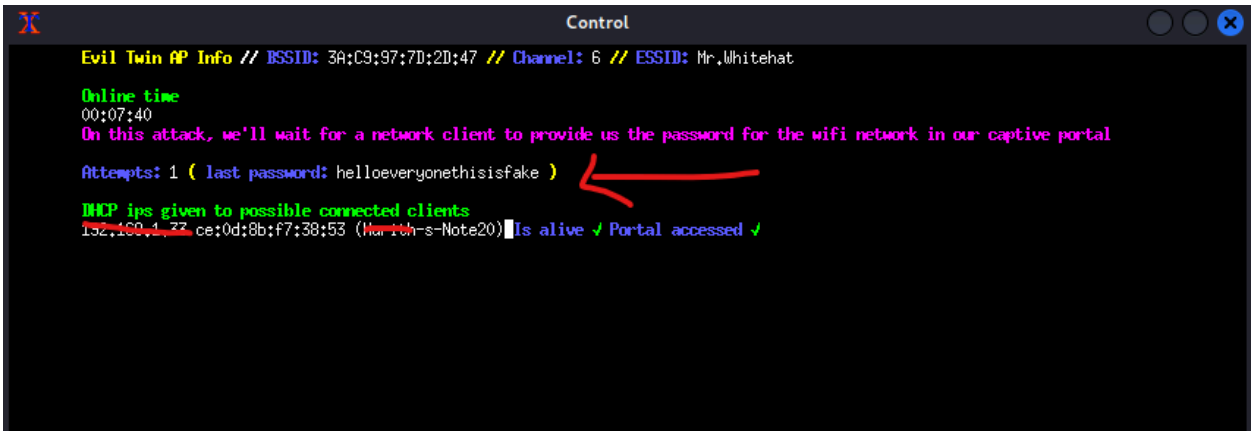


fill the password form

then click **Submit**

---

back to host side



we can see the password we just submitted

Mitigation Tips

- Avoid connecting to public Wi-Fi without verifying the source
- Disable auto-connect to known networks
- Use VPNs to encrypt traffic
- Monitor for rogue APs using tools like `Kismet`, `Wifite`, or enterprise-grade WIDS
- Prefer WPA3 and enterprise authentication when possible