iGNITE Technologies

Wireless Penetration Testing

Fluxion

WWW.HACKINGARTICLES.IN

Contents

Introduction

Fluxion is a tool that can be used to perform Penetration Testing or Security Auditing on Wireless Access Points. It uses Social Engineering to grab the authentication password from the users. It tries to gather the WPA/WPA2 key from the target Access Point by performing a phishing attack. Two attacks can be performed using Fluxion. One is the Handshake Snooper attack and another is Captive Portal.

The Handshake Snooper attack tries to gather the WPA/WPA2 authentication hashes from the 4-way handshake. It uses the de-authenticator to disconnect all users that are connected to the targeted access point and then when the users try to reconnect to the access point, it captures the hashes. These hashes can be used by the Captive Portal attack,

The Captive Portal Attack tries to gather the targeted access point's WPA/WPA2 password by creating a rouge network. In a general sense, it performs an Evil-Twin attack where a network is created with the same SID and all the users are disconnected from the targeted access point. Then with the use of phishing attacks the users are fooled into providing the password for the targeted access point.

Note: To perform attacks using Fluxion, you need an external Wi-Fi card with monitoring mode.

Installation

Now that we are aware of the abilities of the Fluxion tool, it is time to install it on our machine. We will be using Kali Linux for this particular demonstration. Fluxion is not available on Kali Linux by default and there is no method directly. We need to clone its repository from its official **GitHub**. We see that it has been downloaded in the directory named fluxion. Inside it, we found directories such as attacks, bin, and docs and a shell script by the name fluxion.sh. In previous versions, there was a different installation file but now all that is required is to add the parameter -i to perform the installation and dependency checks.

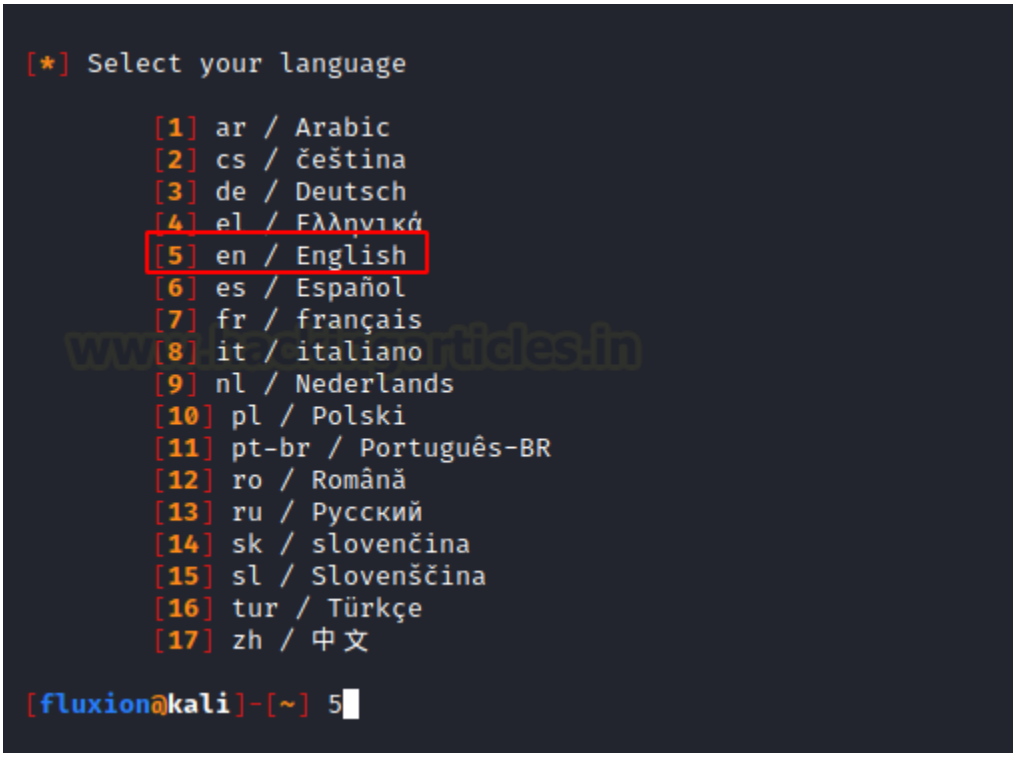
```
git clone https://github.com/FluxionNetwork/fluxion.git
cd fluxion
ls
./fluxion.sh -i
```

```
┌──(root💀kali)-[~]
└─# git clone https://github.com/FluxionNetwork/fluxion.git
Cloning into 'fluxion'...
remote: Enumerating objects: 7940, done.
remote: Counting objects: 100% (39/39), done.
remote: Compressing objects: 100% (26/26), done.
remote: Total 7940 (delta 13), reused 32 (delta 12), pack-reused 7901
Receiving objects: 100% (7940/7940), 32.70 MiB | 9.11 MiB/s, done.
Resolving deltas: 100% (3932/3932), done.

┌──(root💀kali)-[~]
└─# cd fluxion

┌──(root💀kali)-[~/fluxion]
└─# ls
attacks  bin  CODE_OF_CONDUCT.md  _config.yml  CONTRIBUTING.md  docs  fluxion.sh

┌──(root💀kali)-[~/fluxion]
└─# ./fluxion.sh -i
```

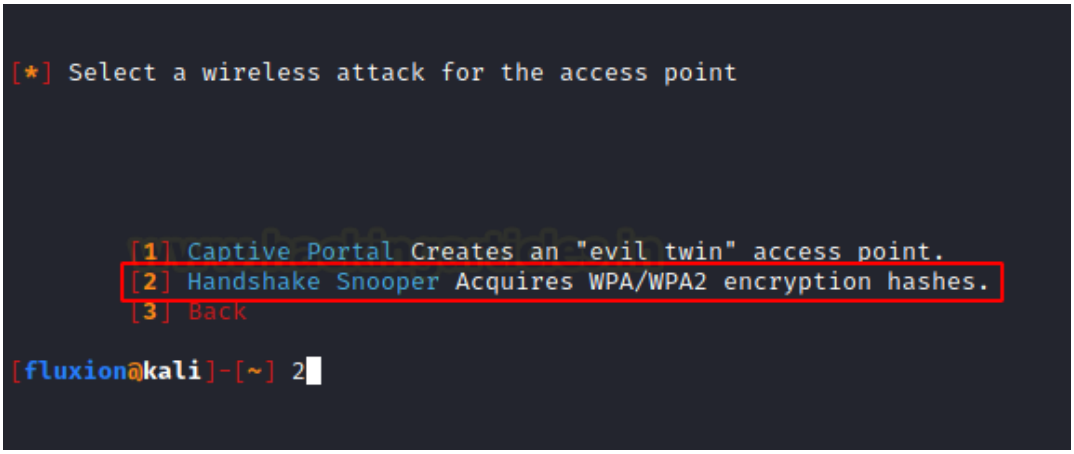
We will be greeted with the logo art of the Fluxion while it checks for the dependencies on its own. If there are any dependencies are tagged as Missing, it is advisable to install them on your own. In this particular demonstration, we have all the dependencies installed so we will ignore some of these and move on.

Capture the SIDs

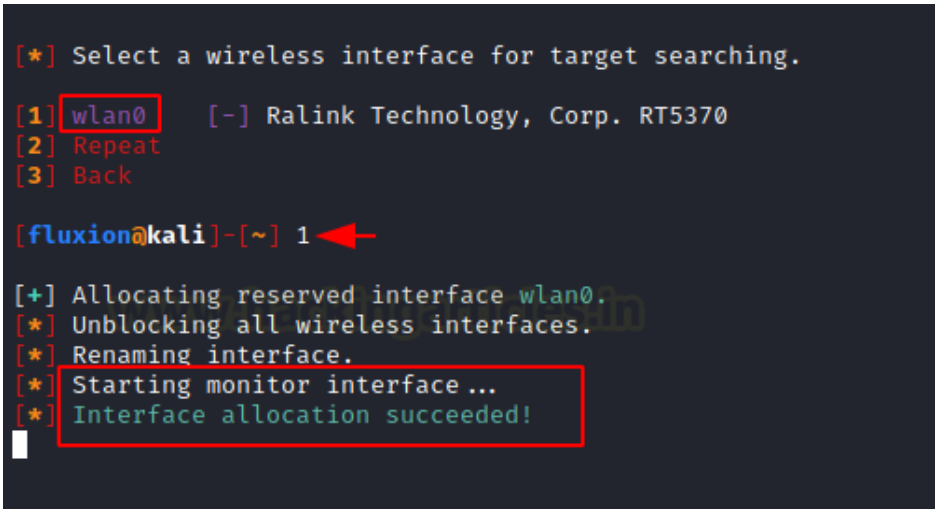
Then we are provided with the Language Selection Menu. We want to select English so we will enter the number from the selection menu and press Enter key.



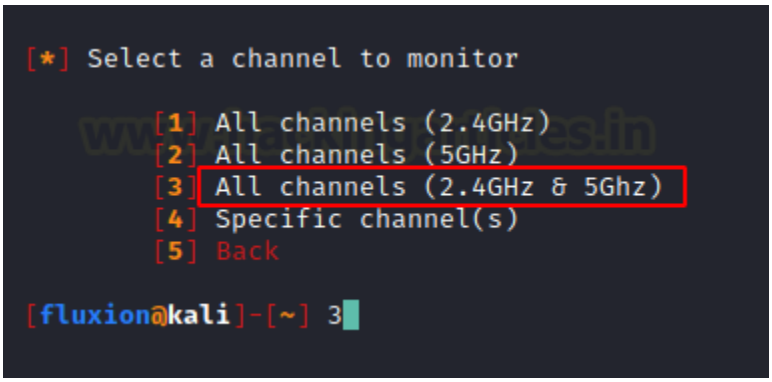
Moving on, we now have to select the Attack that we want to perform on the Access Point. We require to capture the handshake between the network router and the genuine user. We will use that handshake to test and try to get the credential required for gaining access to the Access Point. Hence, we will need to select the Handshake Snooper. The Handshake Snooper attack attempts to retrieve WPA/WPA2 authentication hashes (the 4-way handshake), to be used later by the Captive Portal attack for key verification.



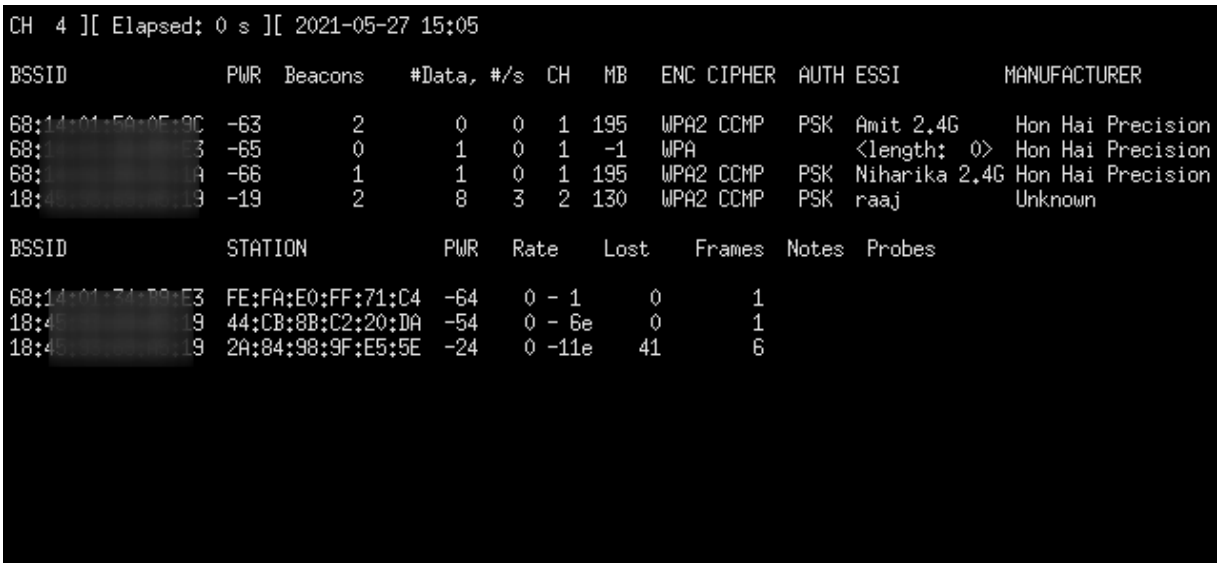
After selecting the Wireless Attack, now we are required to select the Wireless Interface that we will use for searching Target. We have the Wireless Device connected to the wlan0 interface; hence we will select it. After selecting we see that Fluxion starts the monitor interface on the device.



Next, we are required to select the Channel that is supposed to be Monitoring. Since a lot of Wireless Access Points in current days can vary from 2.4GHz to 5GHz so we will choose all the channels in that range.

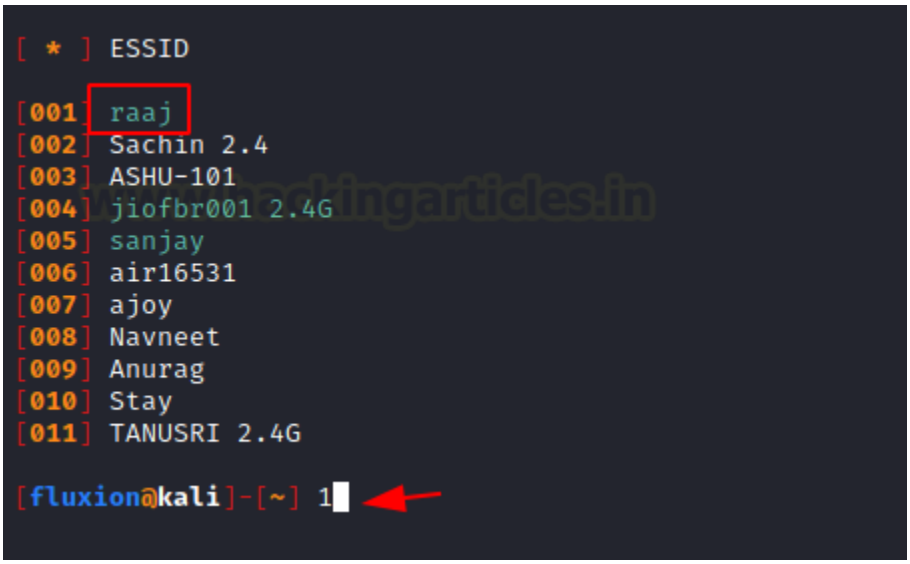


This will open a new window as shown in the image below. This will look for all the possible targets in the network reach and make sure to let the process run for some time and till you have your target visible in the window. **Press Ctrl + c** on the xterm window after locating your target or after a certain time is passed.



Configuration for Handshake Capture

Now back to the original terminal with Fluxion running. Based on the previous process, we will have the Wi-Fi List of potential targets. In our demonstration, we want to target the raaj Wi-Fi. Hence, we enter the number next to it.



Now we are inquiring about the target tracking interface. If in your network environment, you have another wireless interface that you want to use for performing target tracking then you can select it. In our demonstration, we are using the single interface for tracking as well. So, we will select Skip.

```
[*] Select a wireless interface for target tracking.
[*] Choosing a dedicated interface may be required.
[*]

[1] wlan0    [*] Ralink Technology, Corp. RT5370
[2] Skip
[3] Repeat
[4] Back

[fluxion@kali]-[~] 2
```

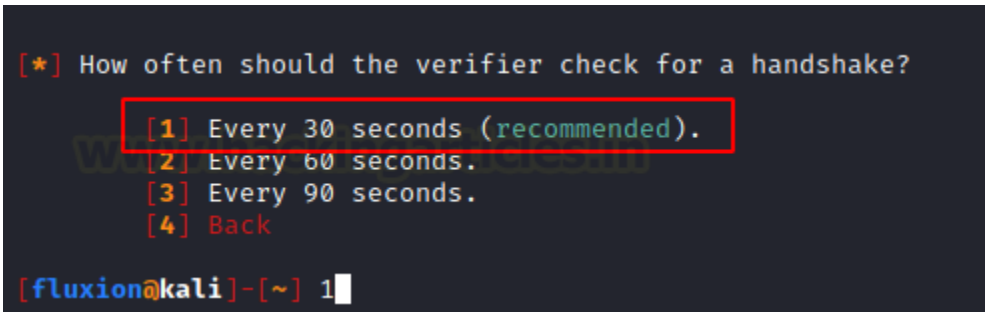
We are now at the stage where we need to select the handshake retrieval method. There are 3 methods provided by Fluxion. The first Method is the Monitor or Passive mode. A passive method of attack forces us to go completely silent, making the attack subtle or undetectable, and allowing for better listening. This method should work best for situations where the target is far away. The downside is the fact the device must keep listening until someone connects to the target access point, which could take a very long time. The other two methods aireplay-ng and mdk4 both are aggressive. These use the deauthenticator. They send de-authentication packets to the users or devices connected to the target access point's clients. We can say this method is aggressive as it jams the connection between the target access point and its users. Once the connection is jammed or disconnected some of the users will try to reconnect with the device which will send the 4-way handshake but this time Fluxion will capture that handshake. You are free to choose any method as both are equally effective. However, we will use the mdk4 method.

```
[*] Select a method of handshake retrieval

     [1] Monitor (passive)
     [2] aireplay-ng deauthentication (aggressive)
     [3] mdk4 deauthentication (aggressive)
     [4] Back

[fluxion@kali]-[~] 3
```

Next, we are required to choose the tool that will be used to verify the hash on the captured valid handshake. Here, we see that the aircrack-ng method is termed unreliable since it is not updated for some time. We will choose the cowpatty verification as it is recommended by Fluxion itself.

```
[*] Select a method of verification for the hash

     [1] aircrack-ng verification (unreliable)
     [2] cowpatty verification (recommended)
     [3] Back

[fluxion@kali]-[~] 2
```

Moving on, we have to choose the duration at which the Fluxion should check for the Handshake. Again, this depends on the environment you are working in and if you want to be discreet. Since we are demonstrating the attack, we will choose to check for Handshake every 30 seconds. This is recommended by Fluxion as well.



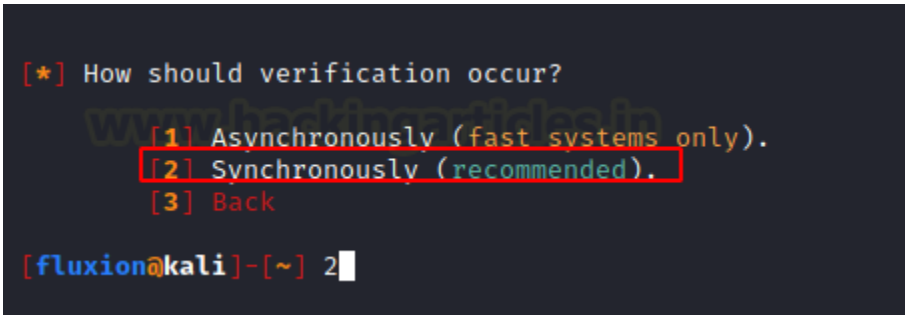
Next, we have to decide on the verifier's synchronicity. It sets the process of verification that occurs with the capturing of data. It asks if we want to capture data simultaneously or back-to-back.

Let's understand the difference between the two.

The Asynchronous option will initiate the verifier while the system is still in the process of capturing data. As this is multitasking at a higher level so requires more thread. If you are running your attacking OS such as Kali Linux directly on the system then you can use it but if you are running Kali Linux as a Virtual Machine like us, then it can cause issues since we have limited threads that are available to Kali.

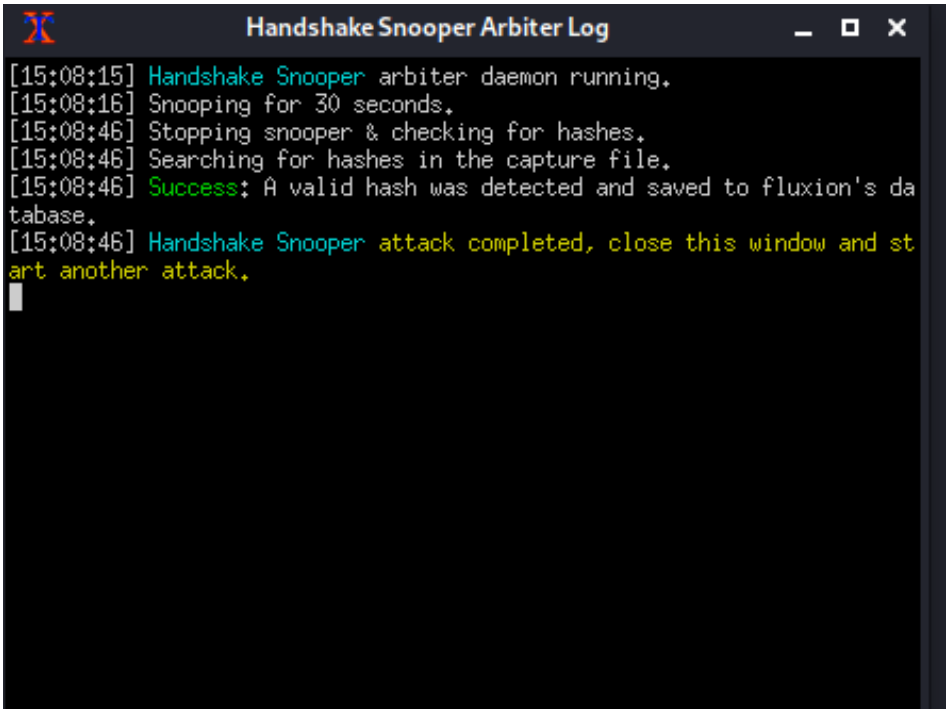
The Synchronous option will stop capturing data before it tries to check for the handshake. As this is not multitasking hence, this option will not cause an issue with low threads. However, there is a downside to choosing these methods as they will stop capturing data so you might lose some handshakes. But since we selected to check the verifier every 30 seconds realistically, we shouldn't miss handshakes.

Again, as we are using Kali on Virtual Machine, we will be choosing the Synchronously Method.

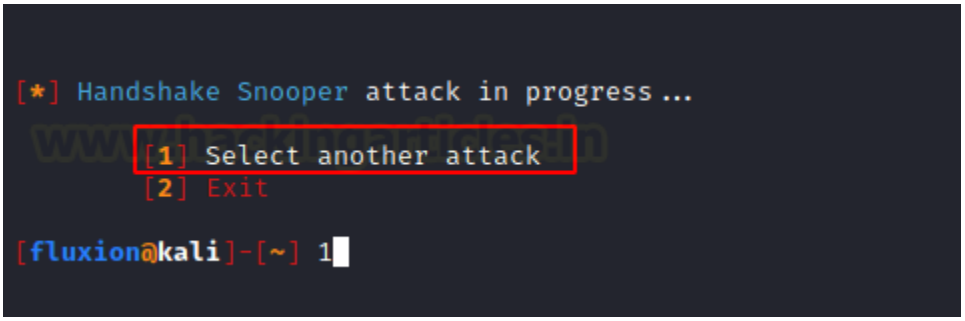


Handshake Snooper Attack

That was the last option that we are required to configure. Now the attack will begin and an xterm window will appear. It is the Log Viewer. It displays the events as they begin. The de-authentication of all the clients will start in a few moments, all the users will be disconnected to the Wi-Fi device. Then when any of those users or devices try to reconnect with the Wi-Fi we will be able to capture the handshake. We can see that the attack was successful and we were able to get a valid hash as demonstrated below. At this moment we can close the log viewer and move on to the next attack.

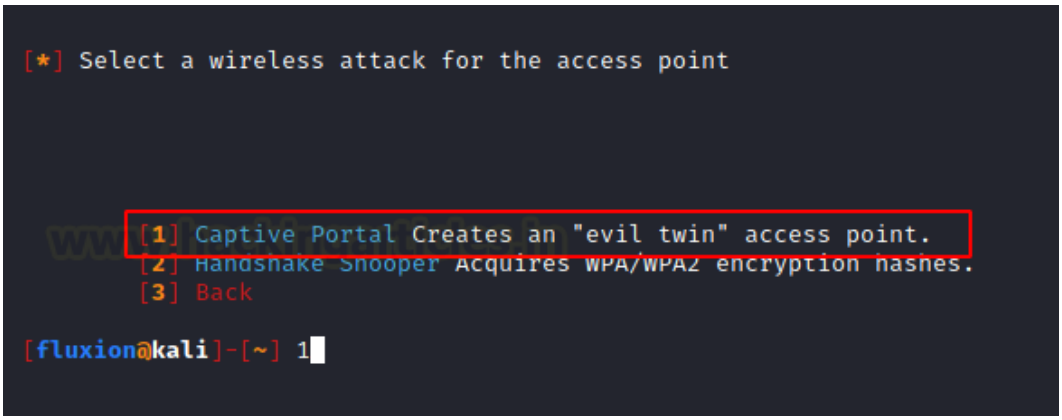


Since we have captured the handshake, we can use this handshake to perform the Captive Portal Attack which is also known as the Evil Twin Attack. As soon as we close the Handshake Sooper Log xterm window, we will be asked we want to select another attack as demonstrated below.

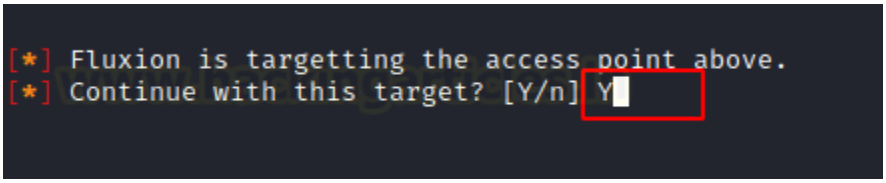


Configuration for Captive Portal Attack

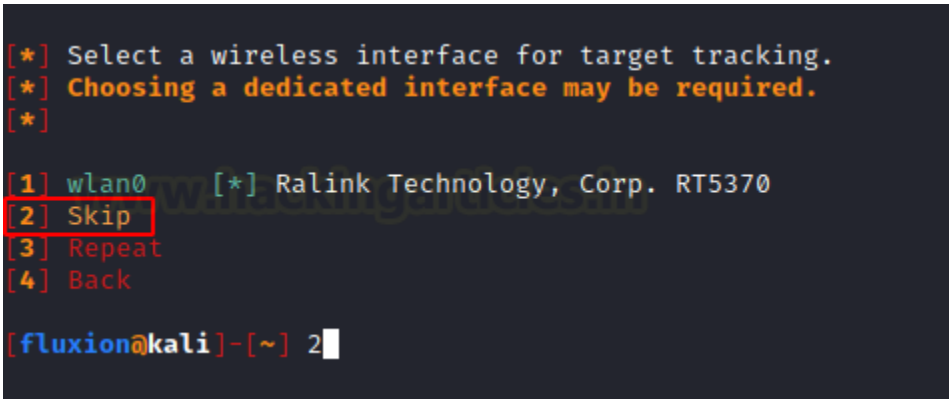
The Captive Portal Attack or as we generally know it as the Evil Twin Attack is the type of attack where we attempt to extract the target access point's WPA/WPA2 key by using a rogue network with an authentication portal that captures the credentials. It is recommended that this type of attack must be performed in close encounters as the attacker machine or Kali Linux serves as the captive portal. This means that the users will need to connect to our machine hence the Wi-Fi signal strength must be strong. After capturing the handshake using Handshake Snooper, we will now again be redirected to the attacker selection menu as before. This time we will select the Captive Portal.



Here we are asked if we want to use the same access point as before. If you were performing the captive attack directly, then you will be required to select the access point name as we did while capturing the handshake earlier. For now, we will continue with this target.



Fluxion asks us for the interface to be used for target tracking. Again, if you are in an environment where you have multiple wireless interfaces which can be used for tacking, choose that interface. Otherwise, skip this step.



We now need to choose the wireless interface that Fluxion will use to send the de-authentication signals. Since we need the wireless interface, we will choose the wlan0.

```
[*] Select an interface for the access point.

[1] docker0  [+] Ralink Technology, Corp. RT5370
[2] eth0     [-] Intel Corporation 82545EM Gigabit Ethernet Cont
[3] wlan0    [*] Ralink Technology, Corp. RT5370
[4] Repeat
[5] Back

[fluxion@kali]-[~] 3
```

Now we are required to choose the methods or tools that can be used for the de-authentication of users from Wi-Fi. Again, this is based on the type of environment and personal preference. All three methods are equally effective. We will be choosing the mdk4 methods as we used it while capturing handshake earlier and it worked swiftly.

```
[*] Select a method of deauthentication

[1] mdk4
[2] aireplay
[3] mdk3

[fluxion@kali]-[~] 1
```

Now, it is time to configure to Rouge Access Point that will capture the credentials from the users. Since we are not using the airbase-ng from the start of the demonstration, we will not use it here as well. Also, it is slow as compared to the hostapd option. So, we choose RogueAP – hostapd.

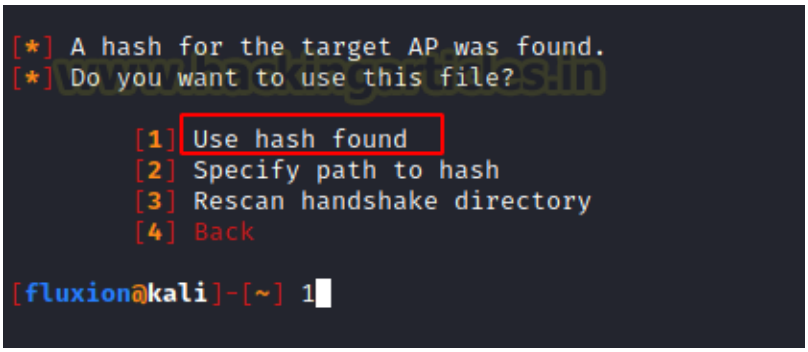
```
[1] Rogue AP - hostapd (recommended)
[2] Rogue AP - airbase-ng (slow)
[3] Back

[fluxion@kali]-[~] 1
```

Next, we will select the hash verifier method as we did while performing the Handshake Snooper Attack. Aircrack-ng is the default hash identifier that is used by Fluxion but it seems to be unreliable and as we used cowpatty before we can use it again. We select hash – cowpatty.

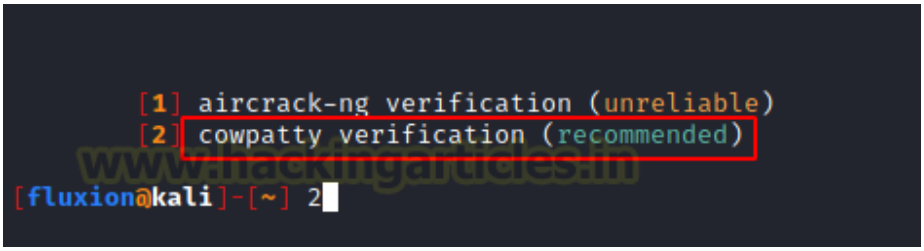
```
[1] hash - cowpatty
[2] hash - aircrack-ng (default, unreliable)
[3] Back

[fluxion@kali]-[~] 1
```

This is the stage where we provide the captured handshake. If you have the handshake in the form of a capture file (.cap) file then you can choose the second option and provide the path to the handshake file.

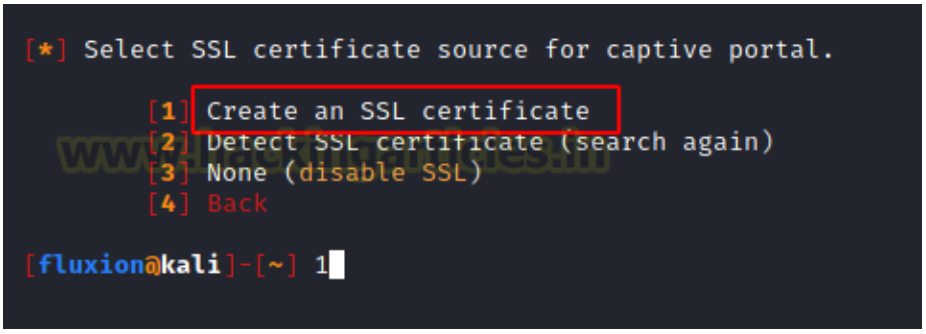
Since we captured the handshake using the Handshake Snooper in the same session, the hash was automatically detected by Fluxion. Hence, we choose the Use hash found option.



Since we choose the cowpatty hash in the previous step, we should use the cowpatty verification method. Even if that is not the case the aircrack-ng verification is unreliable and the cowpatty verification method is recommended by Fluxion. Hence, we choose the cowpatty method.



Next, we are required to select an SSL/TLS certificate source for the captive portal. We can choose the disable SSL option but it will create suspicion as the generally captive portal is SSL supported. If you have a certificate then it will be detected automatically. Since we don't have one, we choose to Create an SSL Certificate option.



Now we need to choose the type of internet connectivity for the rouge network that the users will connect to. We can choose emulated but, in our testing, it was found that it was creating problems with iOS users and Android users. It is useful for the attackers who don't want to make the captive portal more genuine. Since users will connect as it will show that Internet Access is available. But when users will connect to our rouge network, they will be presented with the captive portal. We choose the disconnected method for this demonstration as it has less failure rate.

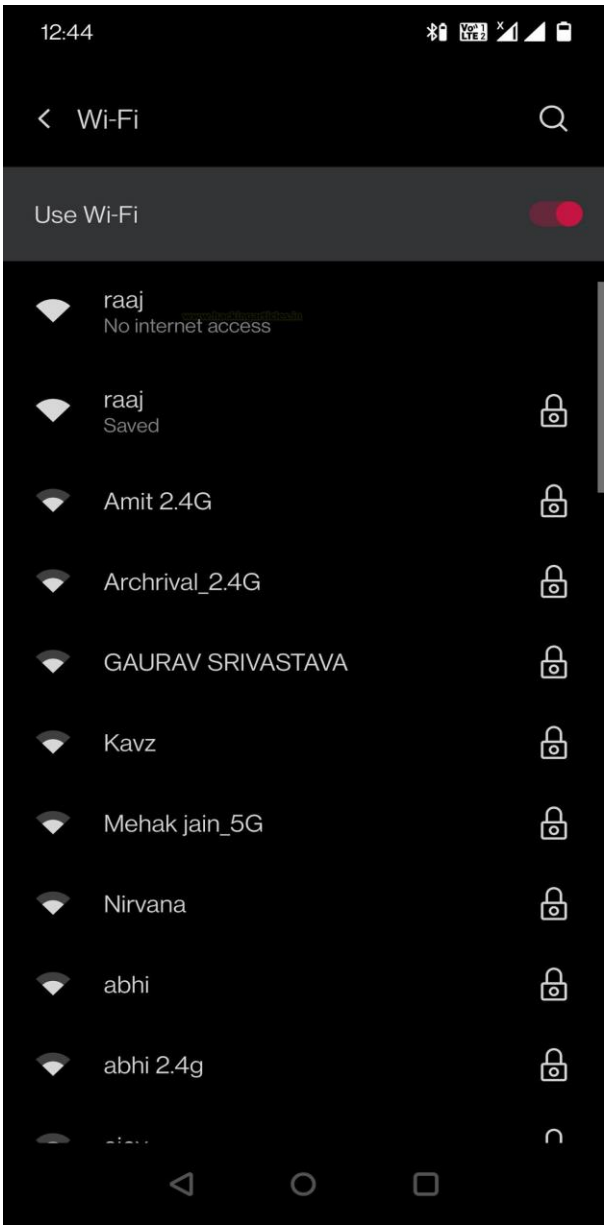
```
[*] Select an internet connectivity type for the rogue network.

        [1] disconnected (recommended)
        [2] emulated
        [3] Back

[fluxion@kali]-[~] 1
```

Now we need to choose the Portal template. By default, Fluxion will have templates that seem very generic. In real-life scenarios, some careful users will not be fooled by these portals. You can scour the internet for the templates that resemble the portal the user has for their network device. For this demonstration, we will be selecting the generic portal with the English language.

```
[01] Generic Portal                    Arabic
[02] Generic Portal                  Bulgarian
[03] Generic Portal                    Chinese
[04] Generic Portal                      Czech
[05] Generic Portal                     Danish
[06] Generic Portal                      Dutch
[07] Generic Portal                    English
[08] Generic Portal                     French
[09] Generic Portal                     German
[10] Generic Portal                      Greek
[11] Generic Portal                     Hebrew
[12] Generic Portal                  Hungarian
[13] Generic Portal                 Indonesian
[14] Generic Portal                    Italian
[15] Generic Portal                  Norweigan
[16] Generic Portal                     Polish
[17] Generic Portal                 Portuguese
[18] Generic Portal                   Romanian
[19] Generic Portal                    Russian
[20] Generic Portal                    Serbian
[21] Generic Portal                      Slovak
[22] Generic Portal                   Slovenian
[23] Generic Portal                    Spanish
[24] Generic Portal                       Thai
```

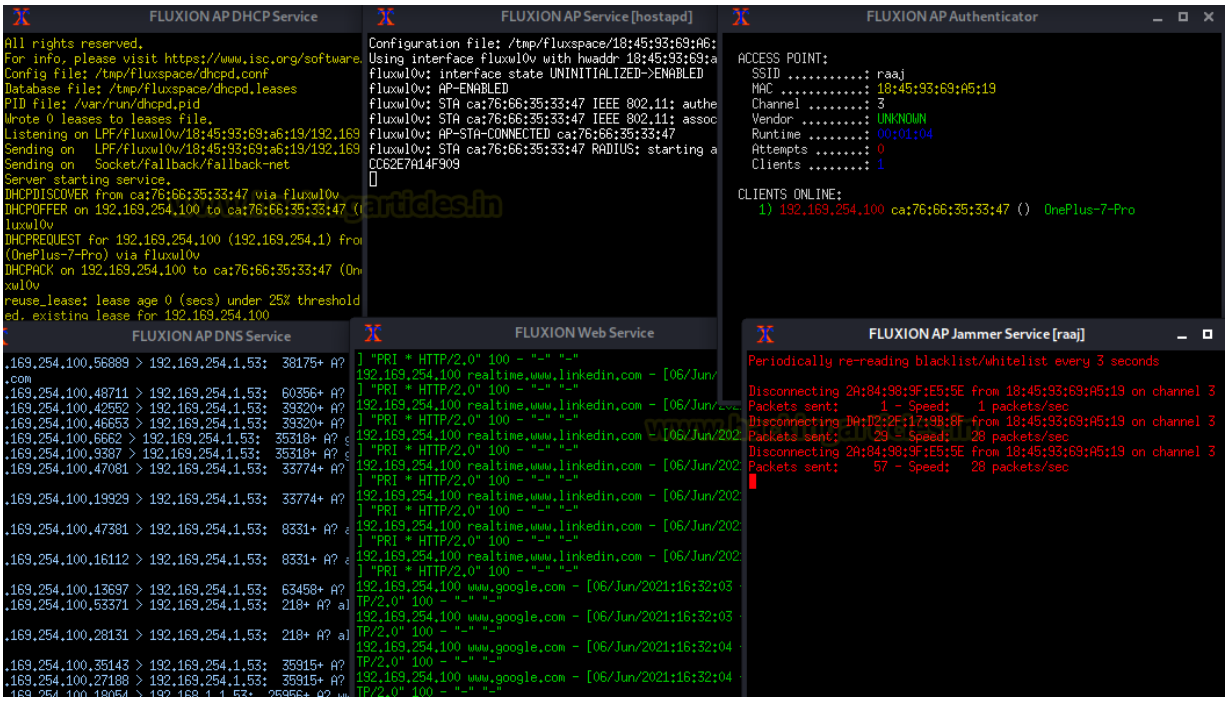
Captive Portal Attack

This sums up the configuration process of the Captive Portal Attack. Now the Fluxion will perform the de-authentication for all the users of the target access point. Any users will be disconnected from their Wi-Fi and will be presented with two networks. One a genuine Network and another a rouge network. The image shown below shows how the attack looks from the victim's perspective. We see that there are two networks by the same name raaj. As we didn't use the emulated option, we see that there is No Internet Access on the rouge network that we created. You can see that the rouge network above the genuine network can convince the user that the rouge network is the real one and they will connect to that network

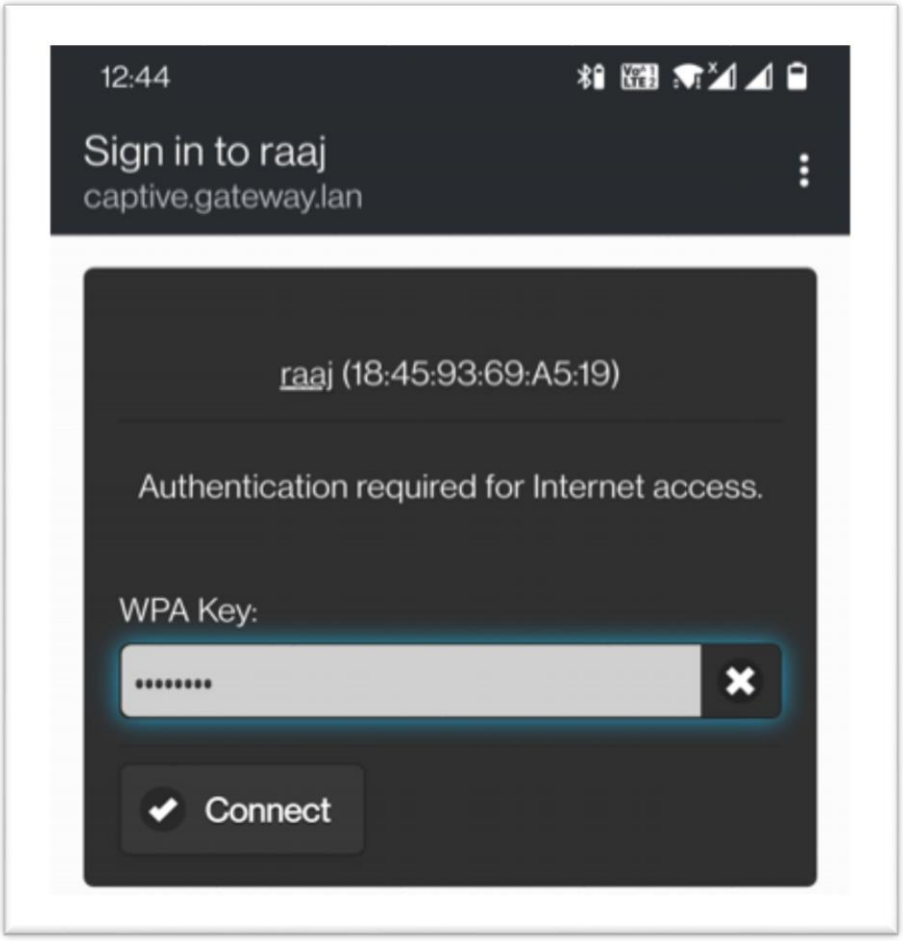


Back to the attacker's perspective. After we choose the template for the rouge network, we will see that there are multiple xterm windows pop up. Let's discuss them. First from the Left-hand side is the DHCP Service. When the victim will connect to our rouge network, it is the responsibility of this DHCP service to emulate the connection and provide an IP address to our victim device. Moving to the right we see the hostapd Window. This is our Captive Portal. This logs the activity of our victims when they browse our Captive Portal. Moving right we see the AP Authenticator window. This logs the SSID, MAC, and other information that is relayed from the victim's device. We will be able to see the attempts of the user if they provide an incorrect password. We can see the listing of Clients that are connected to our rouge network. Moving down from the Left-hand side, we have the DNS Service. As we are not providing internet access all the DNS queries generated from the victim's device will be answered by this service. Moving to the right, we have the log of the Web Service that is hosted for the victim. At last, we have the Jammer Service that is in charge of de-authentication of various devices that are connected to our targeted Access Point.

We saw a similar one while we performed the Handshake Snooper attack. As soon as the victim enters the correct credential for the Access Point, we will be prompted in the Authenticator window.



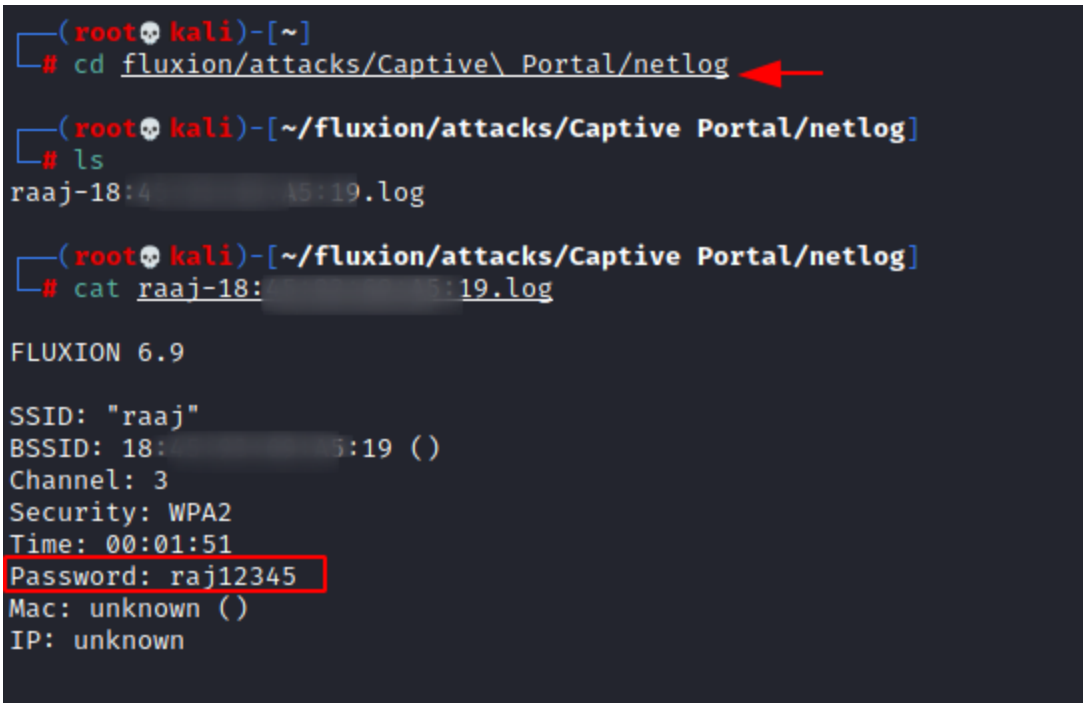
We saw how the attacker will be keeping an eye on the logs to see the activity of the victim. Let's switch back to the Victim perspective. When clicking the rouge network from the Wi-Fi section in the previous step the victim will be greeted with an in-browser portal based on the generic template we provided. We can see that it shows the correct name of the Access Point in the heading. Then provides a generic URL. Next, it asks for the WPA key from the victim and the victim enters the password which starts the process of the 4-way handshake and its hash is matched with the correct hash that we captured earlier. If the user enters an incorrect password, they will be asked to enter the password again until the hash matches.



As soon as the victim enters the correct password connecting the access point, the handshake capture hash is matched with the one we captured earlier and the victim is moved to be connected to the genuine access point and the deauthenticator or jammer is stopped. All the users that were disconnected were allowed to connect with the correct access point. The captured password is saved at /root/fluxion/attacks/Captive Portal/netlog/ as demonstrated below.

Now that we know the location of the password file, we traverse to the location and find the log file with the name of the access point that we targeted. Upon reading the log file we can see the password in cleartext. It was raj12345.

```
cd fluxion/attacks/Captive\ Portal/netlog
ls
cat raaj -18:###########.log
```



This completes the attack; we successfully got the correct password for the raaj Access Point.

Conclusion

Fluxion is one of the best tools when it comes to performing penetration tests or security auditing of Wireless Access Points. In this article, we saw two attacks supported by Fluxion. It was a Handshake Snooper attack and Captive Portal attack. These are not new attacks they have been in the community for quite some time but with updating security protocols and changing environment, Fluxion is still working effectively.

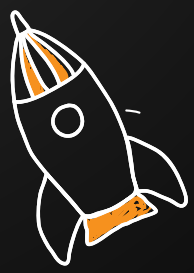
iGNITE Technologies

JOIN OUR TRAINING PROGRAMS

CLICK HERE

BEGINNER

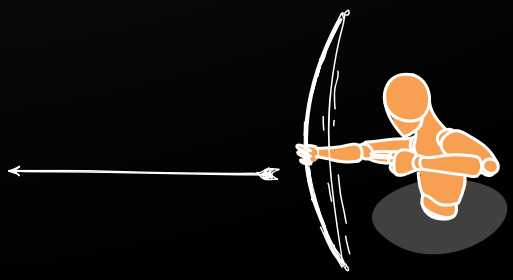
- Ethical Hacking
- Network Pentest
- Bug Bounty
- Wireless Pentest
- Network Security Essentials



ADVANCED

- Burp Suite Pro
- Android Pentest
- Web Services-API
- Advanced Metasploit
- Pro Infrastructure VAPT
- CTF
- Computer Forensics

EXPERT



- Red Team Operation
  - APT's - MITRE Attack Tactics
  - Active Directory Attack
  - MSSQL Security Assessment
- Privilege Escalation
  - Windows
  - Linux