http://linux-news.org/index.php/2011/08/10/cracking-wepwpa2-networks-with-aircrack-ng-linux/

Cracking WEP/WPA/2 networks with Aircrack-ng [Linux]

Now that you have hopefully installed the Aircrack-ng suite and familiarized yourself with some basic Linux commands, we can start cracking WEP and WPA1/2 networks to see the differences in security *Wired Equivalent Privacy* (WEP) and Wi-fi Protected Access (WPA) provide.

Now, lets start. Open up a new terminal and lets begin (all typed commands are underlined; read the notes section for optional commands):

1. Make sure you have a "monitoring" interface, this means that your network interface (the thing that interacts with networks) can scan for open/encrypted networks.
   To check what interfaces you have, type "iwconfig" into your terminal and it will list out which interfaces are currently up, and which mode they are in (look for "mode: managed" or "mode: monitor").
   Check out my blog post about networking in Linux for more on "iwconfig" and the different modes available.

Type:

```
airmon-ng start [interface]
```

if your interface is in "managed" or any other mode (ad-hoc, etc) it needs to be switched into monitor mode. Sometimes it will create a new interface for the monitoring, for example, my wireless is "wlan0" and it creates "wlan0mon" or "mon0" for monitoring.
Once it is in "monitor" mode, you can begin.

2. Make sure you can inject packets into the chosen network (find a network with Kismet (I'll review Kismet later) or your network manager (either Wicd, or network-manager), or with the "airodump-ng [interface]" command in a new terminal. This creates a new .cap file, though).
   Type:

```
aireplay-ng -9 -e [network name] -a [your MAC address] [interface]
```

This makes sure that you can use your network card to input packets (data) into the targeted network. Your NIC (network interface card) must support injection.

3. If you can inject, start dumping captured IVs (Initialization Vectors) into a .cap (capture) file with command:

```
airodump-ng (-c x) –bssid [target network MAC] -w [output prefix] [interface]
```

Note: -c x is channel x, where x is 1-11 and not necessary, although, if you know the channel, I would suggest doing the correct channel.
This will bring up a nice interface with your targeted network, the BSSID (MAC that you entered), the "PWR," or how close you are (lower is better!), the "Beacons," which networks send automatically,

the #Data, which is the data packets that have been sent over the network (which you have just started capturing!), the #/s which is data packets/s (higher is better for capturing faster!), the "CH," or channel (I'll go over this later), the "MB," the "ENC," or encryption (WEP/WPA/OPEN), the CIPHER (related to the ENC), the AUTH (pass-key or other), and finally the ESSID which is the English or ASCII network name that humans understand more easily than a Hex BSSID.

4. Now we have to do a "fake authentication" on the network. This is pretty self explanatory, but it authenticates you with the access point. If you didn't run this, the access point would return "deauthenticated" packets, not allowing you to inject packets back into the system.

Type:

```
aireplay-ng -1 0 -e [network name] -a [target network MAC] -h [your MAC address] [interface]
```

It should respond "Association successful ☺" if not, try again until it works.
This may take a while, so don't fret if it doesn't work right away. I've had to do this three or four times or more with new terminals and locations until I finally got it, it's just luck sometimes.

5. Reinject ARP (Address Resolution Protocol) packets back into the network to create network activity. To review ARP, check out my ARP information post and read it thoroughly, it isn't long and gives a good explaination what ARP is all about. What we're basically doing is sending fake messages to create data packets on the network so we can record and crack their password!

Type:

```
aireplay-ng -3 -b [target network MAC] -h [your MAC address] [interface]
```

It should say "Read xxxx packets (got xxxx ARP requests), sent xxxx packets…" and network activity should increase.

6. Crack the WEP key! Type:

```
aircrack-ng -b [target network MAC] *.cap
```

Note: you can enter the ACTUAL file name instead of "*.cap" if you know it, or whatever "output prefix" you entered, then *.cap (all in a line, since it concatinates -xxxxx_xxxx after the prefix and before .cap).

7. Crack the WPA/WPA2 key (if you're not cracking WEP)! Type:

```
aircrack-ng -w [password list] -b [target network MAC] *.cap
```

Note: You must have captured the WPA handshake, and again, substitute your capture file accordingly.

For WEP cracking, this should run a terminal with "Tested xxxx keys (got xxxx IVs) and a bunch of gibberish HEX underneath. You can run this while you inject packets. It should find the key eventually unless the network admin or creator disconnects the network or you go out of range of it. Sometimes it only takes as little as 5000 keys, and other times 250,000 keys.

My record is about 2-3 minutes while sitting on a toilet in a flea market; it's fun to see how quickly WEP is broken, **so remember ALWAYS use WPA2 with a non-dictionary passkey.** You can review more tips about securing your home network at my post here.

For WPA cracking, it runs through a list of passwords (in Backtrack 5 there is a darkc0de.lst with almost a million, if not more, passwords) and checks every one for a match; thus taking quite a bit longer, and if the password is not in the list, impossible to crack through this method.
The aircrack-ng suite includes the below programs, try playing around with them. If you enter the name then –help or -h, usually (almost always) a help page appears with all the commands you can enter.

Name — What program does

aircrack-ng Cracks WEP and WPA (Dictionary attack) keys.
airdecap-ng Decrypts WEP or WPA encrypted capture files with known key.
airmon-ng Placing different cards in monitor mode.
aireplay-ng Packet injector (Linux, and Windows [with Commview drivers]).
airodump-ng Packet sniffer: Places air traffic into PCAP or IVS files and shows information about networks.
airtun-ng Virtual tunnel interface creator.
airolib-ng Stores and manages ESSID and password lists; Increases the KPS of WPA attacks
packetforge-ng Create encrypted packets for injection.
Tools Tools to merge and convert.
airbase-ng Incorporates techniques for attacking client, as opposed to Access Points
airdecloak-ng removes WEP cloaking from pcap files
airdriver-ng Tools for managing wireless drivers
airolib-ng stores and manages ESSID and password lists and compute Pairwise Master Keys
airserv-ng allows you to access the wireless card from other computers.
buddy-ng the helper server for easside-ng, run on a remote computer
easside-ng a tool for communicating to an access point, without the WEP key
tkiptun-ng WPA/TKIP attack
wesside-ng automatic tool for recovering wep key.