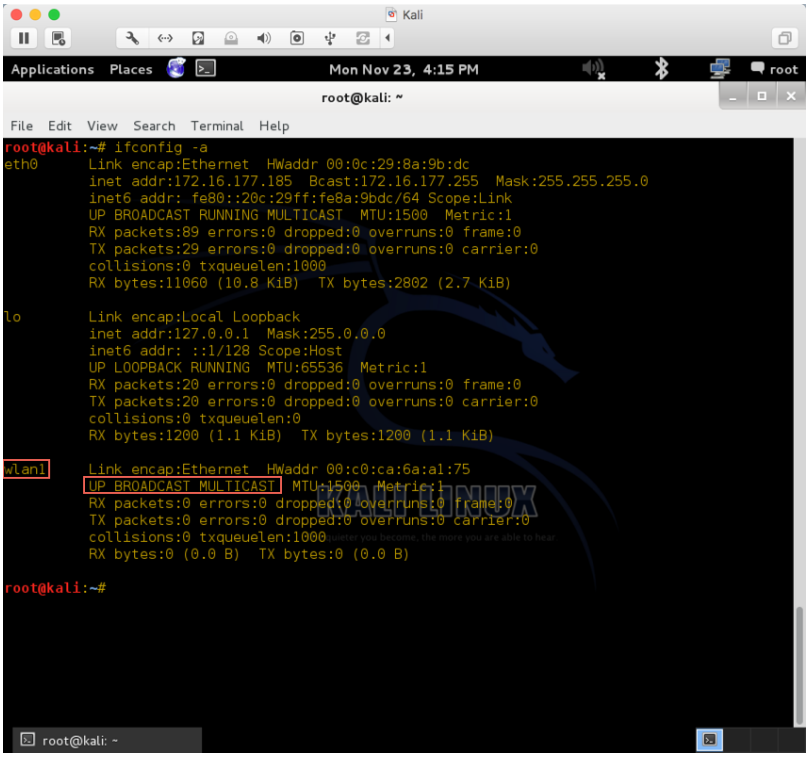
Changing several characteristics of the wireless card

Basic tools

To retrieve a list of interfaces (even the inactive ones)

```
ifconfig -a
```



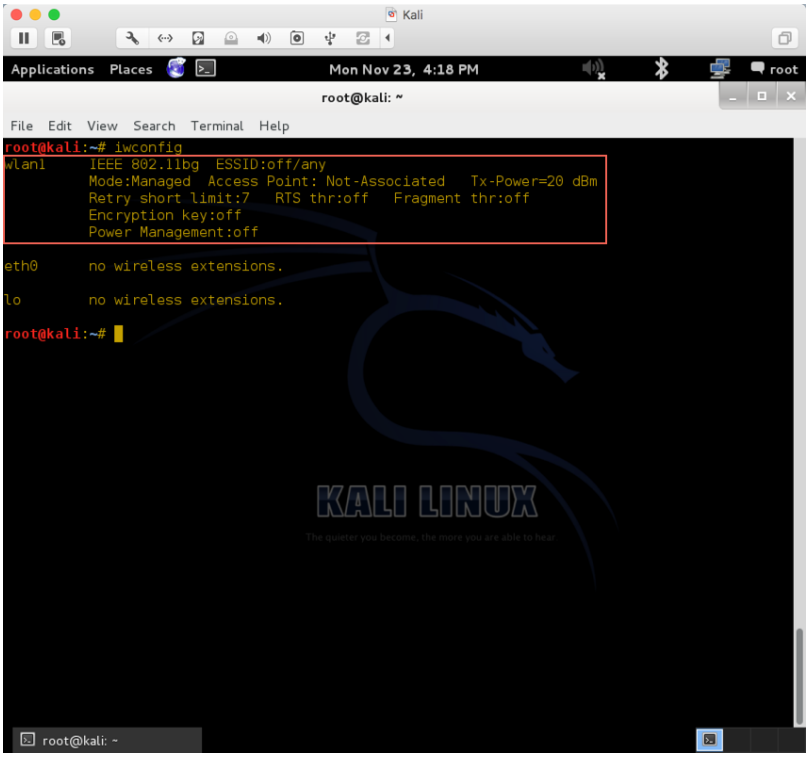
Typically, wireless interfaces are represented as wlanXX

If the wireless interface is on the DOWN state (disabled), then we should enable it before doing anything meaningful with it

```
ifconfig <interface> up
```

To see the characteristics of the wireless extensions of the interfaces on our system

```
iwconfig
```



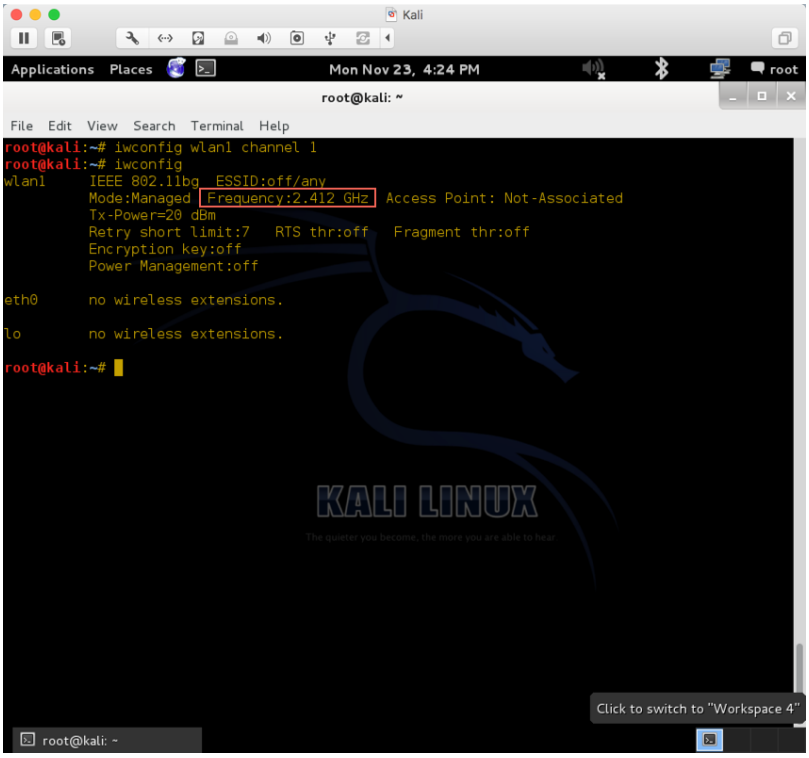
In the case of our example the only wireless interface is the wlan1

Changing the channel

To change the channel of the card

```
iwconfig <interface> channel <channel number>
```

After doing so, if you run the `iwconfig` command again you will notice that the card is set to 2.412 GHz which corresponds to the frequency of the first channel.

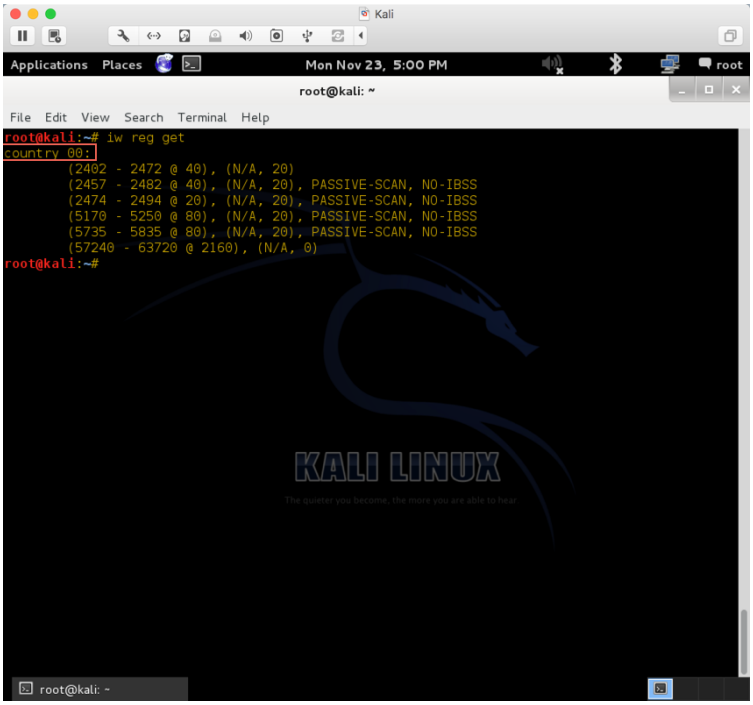


Changing the transmission power

The region of the device is an important setting which indirectly dictates the strength of the signal in which the card transmits. Different countries have different legislations regarding the maximum strength of the signal of a wireless card. For pen testing purposes it is to the best benefit to have a card set to the maximum supporting power.

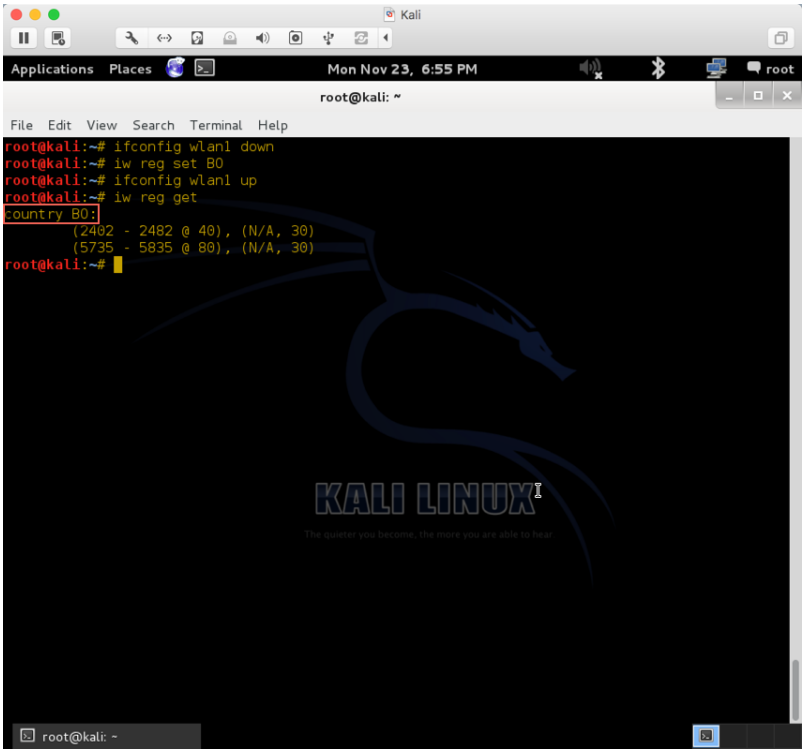
To get the current region

```
iw reg get
```



To change the region thus, the transmission power of the card

```
ifconfig <interface> down
iw reg set <region code>
ifconfig <interface> up
iw reg get
```



A comprehensive list of region codes can be retrieved here:
https://en.wikipedia.org/wiki/ISO_3166-1_alpha-2

Changing the operation mode

Typically, wireless cards are set to managed mode, so they can function as clients to infrastructure based networks. Monitor mode allows cards to read all traffic including packets that originate from non-associated networks.

To set the card in monitor mode one can rely on the tool `airmon-ng` of the aircrack suite

```
airmon-ng start <interface>
```

Changing the mac address

It is possible to change the MAC address of the NIC card

```
Ifconfig <interface> down
macchanger –m <new mac address> <interface>
Ifconfig <interface> up
```

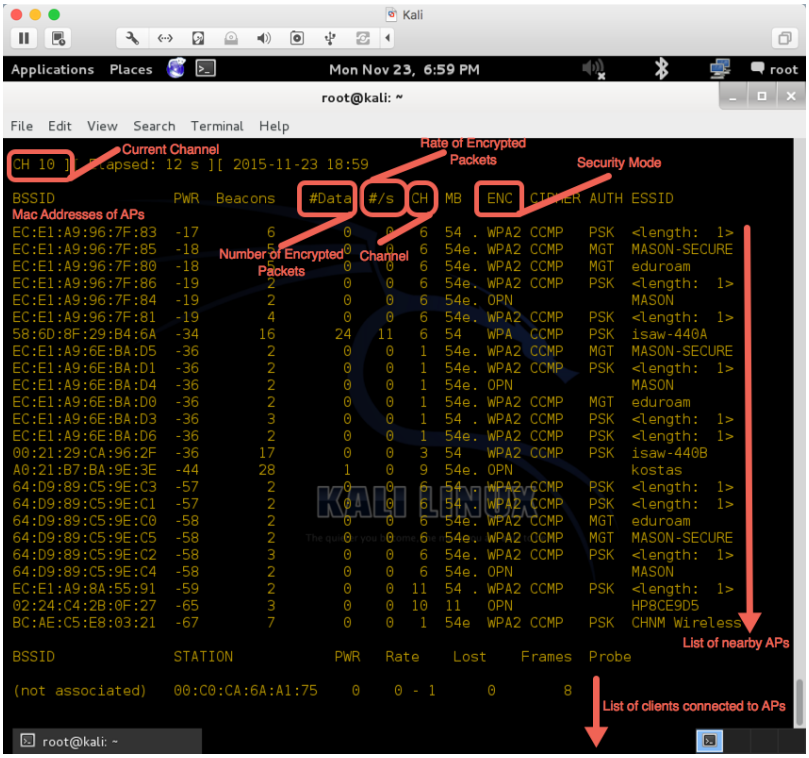
Analyzing Traffic

When a wireless card is set in monitor mode it captures all packets from the air interface. It is possible with the right tools to view, analyze and store these packets.

The airodump-ng tool

To view a list of all the APs in the area and the STAs connected to each one

```
airodump-ng <interface in monitor mode>
```



Note: by default, airodump-ng forces the card to hop among channels. Keep in mind that to achieve this, the card spends only a portion of time on each channel. However, when listening to a channel all packets transmitted to the rest of the channels will evade the monitoring.

To restrain the monitoring to a specific channel

```
airodump-ng <interface in monitor mode> -c <number of desired channel>
```

This is usually done when the attacker has located the victim AP or STA and wishes to capture as many packets as possible for further analysis.

Airodump has the capability of saving all packets on the disk.

```
airodump-ng <interface in monitor mode> -c <number of desired channel>
-w <name of file>
```

Note that airodump-ng saves packets only relevant to WEP key cracking or pen testing. Therefore, the created file will not contain all the packets in the channel.

For more information on the capabilities of airodump-ng tool visit: http://www.aircrack-ng.org/doku.php?id=airodump-ng

The Wireshark tool

It is possible to associate Wireshark's output with a wireless network interface thus, gaining insight to the packets of the live capture.

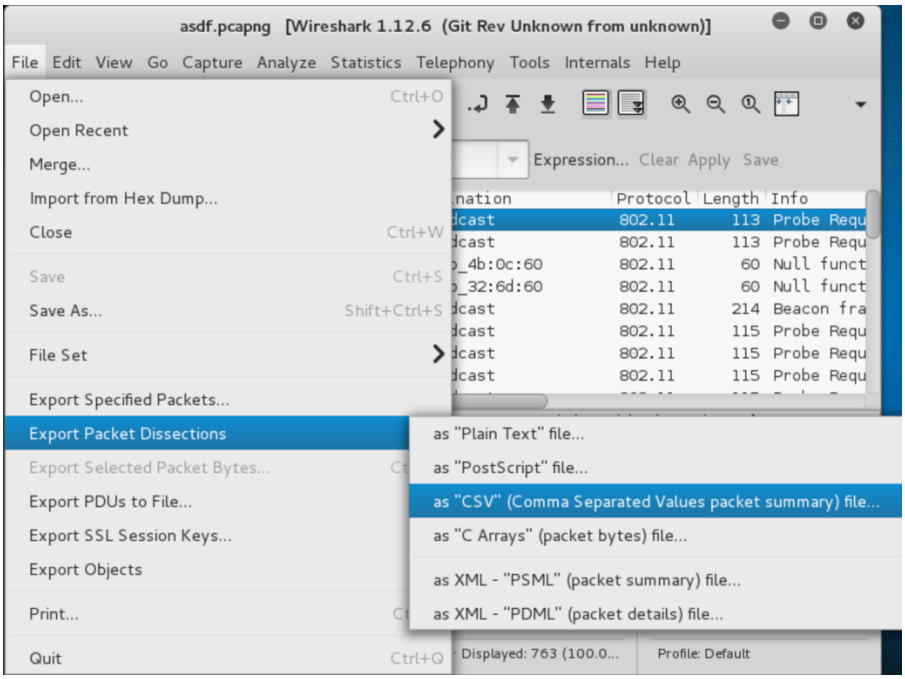
Moreover, one can apply different kinds of filters regarding various fields of the packets (e.g. their type and subtype). This can be done by inserting the mnemonic and the desired value in the filter input field. Alternatively, filtering can be achieved by locating a packet with a desired attribute and setting it as an example filter. Moreover, it is possible to combine multiple filters by applying the standard C operators (e.g., ==, !=, >, <=, !, &&, || etc.).

Some of the most important filters for wireless capture can be retrieved from here:
https://www.wireshark.org/docs/dfref/w/wlan.html
https://www.wireshark.org/docs/dfref/w/wlan_mgt.html

The subtype codes of 802.11 frames can be retrieved here:
https://supportforums.cisco.com/document/52391/80211-frames-starter-guide-learn-wireless-sniffer-traces

The traffic captured with Wireshark can be saved as a binary file (pcap) or another file type including textual formats (e.g., CSV). This is useful for processing with conventional tools and methods.

To do that in Wireshark one simply can choose File->Export Packet Dissections-> as "CSV".



Availability Attacks

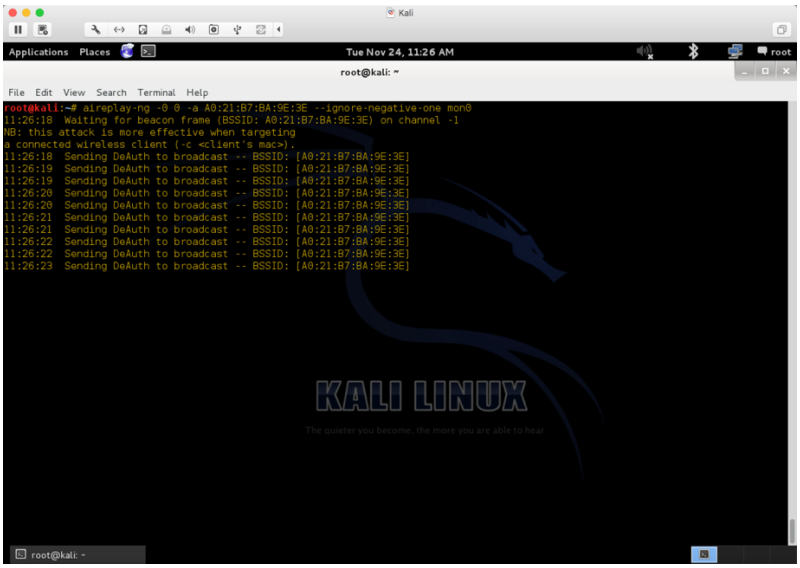
It is possible to reduce the availability of a wireless network or cause denial-of-service (DoS) against specific clients by forging and transmitting specific management (in most cases) frames. This steams from the fact that in 802.11 networks management frames are transmitted unencrypted.

Deauthentication attack

This attack is based on the transmission of deauthentication frames. It is considered the easiest and most effective way of creating a DoS attack against all or specific clients of the network.

The aircrack suite has tools that automate this process. To unleash a deauthentication attack against all clients connected to a specific AP, first one has to know the MAC address of the victim AP. This can be easily done via airodump-ng or wireshark. Then, by using the -0 (or --deauth) option of the aireplay-ng tool one can cause a flood of deauthentication frames to be transmitted.

```
aireplay-ng --ignore-negative-one -0 <packets to be sent> -a <AP MAC Address> <interface in monitor mode>
```



Notice that you can insert 0 instead of a predefined number of packets and the process will carry on indefinitely.

Another tool that can unleash a deauthetication attack is `mdk3`. Actually, the specific tool follows a deadlier methodology (but at the same time more obvious to intrusion detection systems) for this attack. To execute a deauthentication attack with mdk3

mdk3 <interface> d