Quick and easy fake WiFi access point in Kali

I'm working on a project at the moment that requires me to observe traffic from an iOS/Android app to various external IPs.

The easiest way to do this is to setup a fake WiFi access point and use Wireshark to sniff the traffic. This is very easy in Kali Linux.

1. Connect the Kali box to the Internet

On my machine, this is as simple as connecting to my WiFi network "DoingAJob5G" using the built-in wireless card on my x220. I use the GUI provided with Kali.

Using *ifconfig* I can see that this adapter is called *wlan0*.

You could use wired Ethernet, then in all likelihood this will be *eth0* instead.

2. Connect an external WiFi adapter that is supported by hostapd

I'm using a USB TP-LINK TL-WN722N which is using an Atheros AR9271 chipset. These are cheap (£8-£10), powerful and reliable.

I suspect many USB WiFi adapters are compatible with *hostapd*, unfortunately I can't see a clear source documenting which ones.

Check it works by connecting to any network using Kali's GUI. This will save you hassle later if there are any driver or hardware issues.

3. Bring up the new wireless interface.

Use *ifconfig -a* to see the new wireless interface name:

```
wlan3     Link encap:Ethernet  HWaddr c0:4a:00:1e:64:fd
          BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

wlan3   Link encap:Ethernet HWaddr c0:4a:00:1e:64:fd

BROADCAST MULTICAST MTU:1500 Metric:1

RX packets:0 errors:0 dropped:0 overruns:0 frame:0

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:0 (0.0 B) TX bytes:0 (0.0 B)

Bring this up as the gateway for your new wireless network. I am using 10.0.0.1/24 simply to avoid any chance of confusion with my internal NATed 192.168.0.1/24 network.

```
root@kali:~# ifconfig wlan3 10.0.0.1/24 up
root@kali:~# ifconfig wlan3
wlan3     Link encap:Ethernet  HWaddr c0:4a:00:1e:64:fd
          inet addr:10.0.0.1  Bcast:10.0.0.255
Mask:255.255.255.0
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
```

root@kali:~# ifconfig wlan3 10.0.0.1/24 up

root@kali:~# ifconfig wlan3

wlan3   Link encap:Ethernet HWaddr c0:4a:00:1e:64:fd

inet addr:10.0.0.1 Bcast:10.0.0.255 Mask:255.255.255.0

```
UP BROADCAST MULTICAST  MTU:1500  Metric:1

RX packets:0 errors:0 dropped:0 overruns:0 frame:0

TX packets:0 errors:0 dropped:0 overruns:0 carrier:0

collisions:0 txqueuelen:1000

RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)
```

4. Configure and run DHCP and DNS services

DHCP assigns IP addresses when clients connect, and DNS provides resolution of names to IPs.

Most wireless clients expect DHCP by default, so it is convenient to run a DHCP server. You can manually set IP addresses, but it’s really easier to do DHCP.

Running our own DNS server means that we can easily intercept and alter DNS queries, which can assist in setting up *man-in-the-middle* attacks.

A piece of software called *dnsmasq* does both DHCP and DNS and is very simple to setup.

First, install *dnsmasq*:

Next, create a config file *dnsmasq.conf* as follows:

```
interface=wlan3
dhcp-range=10.0.0.10,10.0.0.250,12h
dhcp-option=3,10.0.0.1
dhcp-option=6,10.0.0.1
server=8.8.8.8
log-queries
log-dhcp
```

```
interface=wlan3
```

```
dhcp-range=10.0.0.10,10.0.0.250,12h

dhcp-option=3,10.0.0.1

dhcp-option=6,10.0.0.1

server=8.8.8.8

log-queries

log-dhcp
```

This is about as simple as it gets. Only listen on *wlan3*, our additional wireless adapter. Hand out DHCP addresses from 1*0.0.0.10-10.0.0.250*. DHCP option 3 is the gateway, DHCP option 6 is the DNS server – both of these should be set to our *wlan3* IP of *10.0.0.1*. *server* specifies upstream DNS servers that will handle most DNS queries – I have provided Google's DNS server of *8.8.8.8*. Finally, log DNS queries and DHCP requests – this just makes it easier to check everything is working.

We also want to create a file *fakehosts.conf* to allow us to spoof certain DNS requests:

This will cause the *dnsmasq* DNS server to respond with *10.0.0.9* to any request for *neohub.co.uk*.

We then need to bring *dnsmasq* up. I want it to run with output to stderr, so this is done as follows:

```
dnsmasq -C dnsmasq.conf -H fakehosts.conf -d
```

dnsmasq -C dnsmasq.conf -H fakehosts.conf -d

5. Configure and run hostapd

Next, we need to get our wireless adapter to run as a access point.

hostapd allows us to do this.

Install *hostapd:*

Create a config file *hostapd.conf:*

```
interface=wlan3
driver=nl80211
ssid=Kali-MITM
channel=1
```

interface=wlan3

driver=nl80211

ssid=Kali-MITM

channel=1

Again – really simple. Use our additional wireless adapter *wlan3* with the nl80211 drivers (which seem to cover pretty much all modern adapters than can be APs), set the SSID to *Kali-MITM* and set the channel to 1. There is no encryption etc. but I really don't need or want it for sniffing traffic.

Then start *hostapd*:

```
root@kali:~# hostapd ./hostapd.conf
Configuration file: ./hostapd.conf
Failed to update rate sets in kernel module
Using interface wlan3 with hwaddr c0:4a:00:1e:64:fd and ssid
'Kali-MITM'
```

root@kali:~# hostapd ./hostapd.conf

Configuration file: ./hostapd.conf

Failed to update rate sets in kernel module

Using interface wlan3 with hwaddr c0:4a:00:1e:64:fd and ssid 'Kali-MITM'

6. Setup routing for the access point

You want a very simple setup at the moment – act as a basic NAT gateway between *wlan3* and *wlan0*.

Without going into any detail, the following commands will set this up:

```
sudo sysctl -w net.ipv4.ip_forward=1
sudo iptables -P FORWARD ACCEPT
sudo iptables --table nat -A POSTROUTING -o wlan0 -j MASQUERADE
```

sudo sysctl -w net.ipv4.ip_forward=1

sudo iptables -P FORWARD ACCEPT

sudo iptables --table nat -A POSTROUTING -o wlan0 -j MASQUERADE

At this stage, you should now be able to connect to *Kali-MITM*, get an IP address, and start using the Internet.