

# 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	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	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	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	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)	Apple_22:a2:e8 (80:... 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	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)	Apple_22:a2:e8 (80:... 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	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	802.11	16	Request-to-send, Flags=.....
59	3.271366	HuaweiTe_b9:50:a8 (f8:23:b2:b9:50:a8) (...	HuaweiTe_b9:50:a8 (... 802.11	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	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	..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