

Informe Laboratorio 3

Sección 02

Anselmo Pacheco

e-mail: anselmo.pacheco@mail.udp.cl

21 de Mayo de 2024

Índice

1. Descripción de actividades	2
2. Desarrollo (PASO 1)	2
2.1. En qué se destaca la red del informante del resto	2
2.2. Explica matemáticamente porqué se requieren más de 5000 paquetes para obtener la pass	3
2.3. Obtiene la password con ataque por defecto de aircrack-ng	3
2.4. Indica el tiempo que demoró en obtener la password	3
2.5. Descifra el contenido capturado	4
2.6. Describe como obtiene la url de donde descargar el archivo	4
3. Desarrollo (PASO 2)	6
3.1. Script para modificar diccionario original	6
3.2. Cantidad de passwords finales que contiene rockyou_mod.dic	6
4. Desarrollo (Paso 3)	6
4.1. Obtiene contraseña con hashcat con potfile	6
4.2. Nomenclatura del output	8
4.3. Obtiene contraseña con hashcat sin potfile	8
4.4. Nomenclatura del output	8
4.5. Obtiene contraseña con aircrack-ng	9
4.6. Identifica y modifica parámetros solicitados por pycrack	9
4.7. Obtiene contraseña con pycrack	12

1. Descripción de actividades

Su informante quiere entregarle la contraseña de acceso a una red, pero desconfía de todo medio para entregársela (aún no llega al capítulo del curso en donde aprende a comunicar una password sin que nadie más la pueda interceptar). Por lo tanto, le entregará un archivo que contiene un desafío de autenticación, que al analizarlo, usted podrá obtener la contraseña que lo permite resolver. Como nadie puede ver a su informante (es informante y debe mantener el anonimato), él se comunicará con usted a través de la redes inalámbricas y de una forma que solo usted, como experto en informática y telecomunicaciones, logrará esclarecer.

1. Identifique cual es la red inalámbrica que está utilizando su informante para enviarle información. Obtenga la contraseña de esa red utilizando el ataque por defecto de aircrack-ng, indicando el tiempo requerido para esto. Descifre el contenido transmitido sobre ella y descargue de Internet el archivo que su informante le ha comunicado a través de los paquetes que usted ha descifrado.

2. Descargue el diccionario de Rockyou (utilizado ampliamente en el mundo del pentesting). Haga un script que para cada string contenido en el diccionario, reemplace la primera letra por su letra en capital y agregue un cero al final de la password.

Todos los strings que comiencen con número toca eliminarlos del diccionario. Indique la cantidad de contraseñas que contiene el diccionario modificado debe llamarse rockyou_mod.dic A continuación un ejemplo de cómo se modifican las 10 primeras líneas del diccionario original.

3. A partir del archivo que descargó de Internet, obtenga la password asociada a la generación de dicho archivo. Obtenga la llave mediante un ataque por fuerza bruta.

Para esto deberá utilizar tres herramientas distintas para lograr obtener la password del archivo: hashcat, aircrack-ng, pycrack. Esta última, permite entender paso a paso de qué forma se calcula la contraseña a partir de los valores contenidos en el handshake, por lo que deberá agregar dichos valores al código para obtener la password a partir de ellos y de rockyou_mod.dic. Antes de ejecutar esta herramienta deberá deshabilitar la función RunTest().

Al calcular la password con hashcat utilice dos técnicas: una donde el resultado se guarda en el potfile y otra donde se deshabilita el potfile. Indique qué información retorna cada una de las 2 técnicas, identificando claramente cada campo.

Recuerde indicar los 4 mayores problemas que se le presentaron y cómo los solucionó.

2. Desarrollo (PASO 1)

2.1. En qué se destaca la red del informante del resto

La red del informante se destaca ya que el cifrado que esta tiene es de tipo WEP el cual es un cifrado que no se utiliza en la actualidad, junto a esto la red del informante se destaca

por enviar una mayor cantidad de trafico.

2.2. Explica matemáticamente porqué se requieren más de 5000 paquetes para obtener la pass

Para ejecutar un ataque de fuerza bruta exitosamente es crucial recopilar una cantidad significativa de paquetes de datos cifrados con la clave de red que se desea descifrar. A medida que se obtienen más paquetes cifrados, se puede inferir más información sobre la clave de red, lo que eventualmente permite descubrir la clave real.

Esto quiere decir que se requieren al menos 5000 paquetes de datos cifrados para llevar a cabo un ataque de fuerza bruta efectivo en una red WEP utilizando Airodump-ng. Sin embargo, la cantidad exacta de paquetes necesarios puede variar según la complejidad de la clave de red y la calidad de la señal inalámbrica. Y en algunos casos puede ser necesario capturar un número considerablemente mayor de paquetes para poder descifrar la clave de red.

2.3. Obtiene la password con ataque por defecto de aircrack-ng

En este paso a través de airodump se procede a capturar tráfico de red en la capa 8 y se procede a guardar en un archivo .cap

```

quitting...
telematica@informatica-04:~$ sudo airodump-ng -c 8 -w labora3 wlx6466b31e7745
09:47:28 Created capture file "labora3-02.cap".

CH 8 ][ Elapsed: 6 s ][ 2024-05-14 09:47

BSSID          PWR RXQ Beacons  #Data, #/s CH  MB  ENC CIPHER AUTH ESSID
B0:48:7A:D2:DD:74 -50  0      74      396  29  8   54e  WEP  WEP      WEP

BSSID          STATION          PWR  Rate  Lost  Frames  Notes  Probes
(not associated) 6C:C7:EC:FA:C3:A8 -91   0 - 1    0      2
(not associated) D0:39:57:0E:92:07 -75   0 - 6    0      1
(not associated) 36:F1:9F:34:F9:39 -91   0 - 1    0      2
(not associated) 0E:69:49:5D:37:5A -53   0 - 1    0      3      Kuref WiFi
(not associated) 56:B0:95:30:8B:AA -72   0 - 1    0      5
(not associated) 12:E1:F8:86:8B:D8 -89   0 - 1    0      1
(not associated) 5C:49:7D:79:D1:38 -91   0 - 1    0      2
(not associated) 0A:47:CA:9D:26:46 -92   0 - 1    0      1
(not associated) E6:A0:BC:73:76:28 -93   0 - 1    0      1
B0:48:7A:D2:DD:74 E0:0A:F6:3C:E0:91 -53  54e- 1e 1385  443      Wifi-sec

```

Figura 1: Captura de tráfico.

2.4. Indica el tiempo que demoró en obtener la password

A través del comando sudo aircrack-ng -b B0:48:7A:D2:DD:74 labora3-01.cap intenta descifrar la clave WEP de la red inalámbrica con la dirección MAC B0:48:7A:D2:DD:74 obtenida

al capturar el tráfico.

```

selenatic@informatica-04:~$ sudo aircrack-ng -b B8:4B:7A:D2:DD:74 labora3-01.cap
Reading packets, please wait...
Opening labora3-01.cap
Read 71919 packets.
Got 29827 out of 25000 IVsStarting PTW attack with 29827 ivs.tlal targets
Attack will be restarted every 5000 captured ivs.

Aircrack-ng 1.6

[00:00:00] Tested 18 keys (got 29827 IVs)

KB  depth  byte(vote)
0  0/ 2  12(38912) 7A(36864) 84(36896) 8B(36896) 7F(35840)
1  0/ 1  34(41216) 73(37376) AE(37376) 63(36608) B6(36608)
2  0/ 2  56(39168) 33(38656) 31(36352) ED(36352) 3C(35584)
3  1/ 3  36(38400) 14(37888) 45(36864) 52(35840) ED(35840)
4  0/ 2  90(38912) BC(37632) 09(36352) 1E(36896) 0C(35840)

KEY FOUND! [ 12:34:56:78:90 ]
Decrypted correctly: 100%

```

Figura 2: Obtención de la password.

2.5. Descifra el contenido capturado

Tal como se puede apreciar, en la captura anterior se obtiene la llave descifrada, es por esto que a continuación se procede a utilizar el siguiente comando, que permite realizar una captura que solo contenga paquetes con la password (12:34:56:78:90) obtenida.

```

selenatic@informatica-04:~$ sudo aircrack-ng -w 12:34:56:78:90 labora3-01.cap
Total number of stations seen 3
Total number of packets read 262150
Total number of WEP data packets 111178
Total number of WPA data packets 0
Number of plaintext data packets 1
Number of decrypted WEP packets 111178
Number of corrupted WEP packets 0
Number of decrypted WPA packets 0
Number of bad TKIP (WPA) packets 0
Number of bad CCMP (WPA) packets 0
selenatic@informatica-04:~$

```

Figura 3: Obtención de la password.

2.6. Describe como obtiene la url de donde descargar el archivo

A partir del análisis de la captura, se obtienen paquetes ICMP y partir de este mensaje se procede a buscar un decodificador de base 64 en internet para convertir el mensaje de base 64, con el objetivo de obtener la url deseada.

A partir de esto se obtiene lo siguiente:

2.6 Describe como obtiene la url de donde descargar el archivo DESARROLLO (PASO 1)

Decode from Base64 format
Simply enter your data then push the decode button.

Yml0Lmx5Ly13cGEy

For encoded binaries (like images, documents, etc.) use the file upload form a little further down on this page.

UTF-8 Source character set.

☐ Decode each line separately (useful for when you have multiple entries).

☒ Live mode OFF Decodes in real-time as you type or paste (supports only the UTF-8 character set).

< DECODE > Decodes your data into the area below.

bit.ly/wpa2

Figura 4: Convertir Base 64 a URL.

El link obtenido a partir de la url corresponde a un archivo llamado handshake.pcapng, que contiene lo siguiente:

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	802.11	123	Association Request, SN=2292, FN=0, Flags=....., SSID=VTR-164
2	0.000002	ee:de:67:8c:df:8b	ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA)	802.11	10	Acknowledgement, Flags=.....
3	0.002401	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	802.11	102	Association Response, SN=1184, FN=0, Flags=.....
4	0.002402	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b (b0:48:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....
5	0.007381	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	133	Key (Message 1 of 4)
6	0.009336	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b (b0:48:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....
7	0.017080	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	155	Key (Message 2 of 4)
8	0.017082	ee:de:67:8c:df:8b	ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA)	802.11	10	Acknowledgement, Flags=.....
9	0.017087	ee:de:67:8c:df:8b	ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA)	802.11	10	Clear-to-send, Flags=.....
10	0.050774	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	189	Key (Message 3 of 4)
11	0.060778	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b (b0:48:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....

Frame 1: 123 bytes on wire (984 bits), 123 bytes captured (984 bits)
IEEE 802.11 Association Request, Flags:
IEEE 802.11 Wireless Management

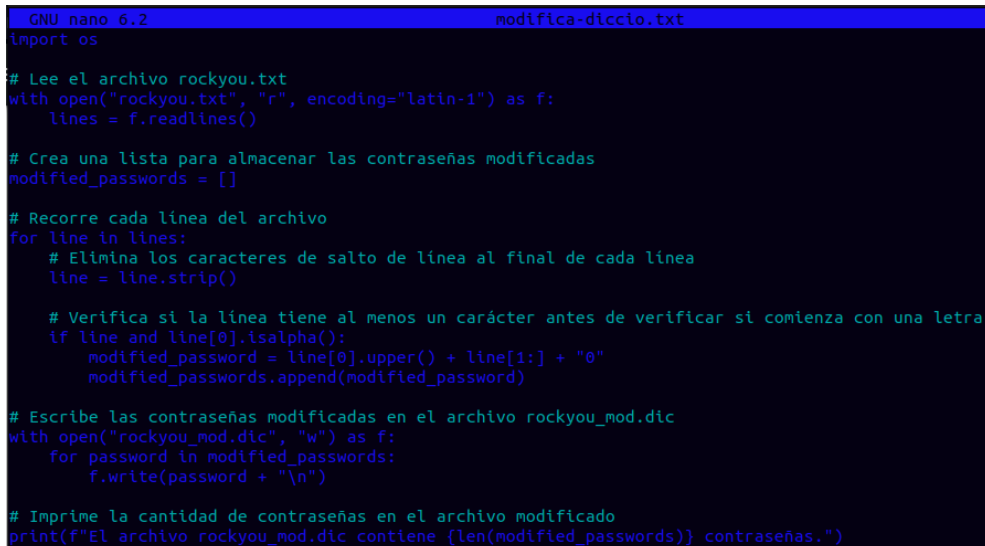
0000 00 00 3a 01 b0 48 7a d2 dc 18 ee de 67 8c df 8bHz.....g...
0010 b0 48 7a d2 dc 18 40 8f 31 04 01 00 00 0b 56 54Hz...@.1...VT
0020 52 2d 31 36 34 35 32 31 33 01 08 82 84 8b 96 0c R-1645213.....
0030 12 18 24 30 14 01 00 00 0f ac 04 01 00 00 0f ac ..\$.0.....
0040 04 01 00 00 0f ac 02 00 00 32 04 30 48 60 6c 3b2.0H'l;
0050 10 51 51 53 54 73 74 75 76 77 78 7c 7d 7e 7f 80 ..Q05Tstuvwx}~..
0060 82 7f 05 04 00 00 00 01 dd 07 00 50 f2 02 00 01P...
0070 00 dd 08 8c fd f0 01 01 02 01 00

Figura 5: URL obtenida.

3. Desarrollo (PASO 2)

3.1. Script para modificar diccionario original

El script utilizado para la modificación del diccionario original es el siguiente:



```
GNU nano 6.2 modifica-diccio.txt
import os

# Lee el archivo rockyou.txt
with open("rockyou.txt", "r", encoding="latin-1") as f:
    lines = f.readlines()

# Crea una lista para almacenar las contraseñas modificadas
modified_passwords = []

# Recorre cada línea del archivo
for line in lines:
    # Elimina los caracteres de salto de línea al final de cada línea
    line = line.strip()

    # Verifica si la línea tiene al menos un carácter antes de verificar si comienza con una letra
    if line and line[0].isalpha():
        modified_password = line[0].upper() + line[1:] + "0"
        modified_passwords.append(modified_password)

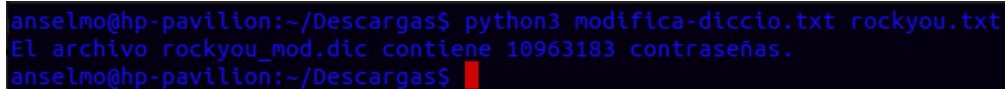
# Escribe las contraseñas modificadas en el archivo rockyou_mod.dic
with open("rockyou_mod.dic", "w") as f:
    for password in modified_passwords:
        f.write(password + "\n")

# Imprime la cantidad de contraseñas en el archivo modificado
print(f"El archivo rockyou_mod.dic contiene {len(modified_passwords)} contraseñas.")
```

Figura 6: Script modificado.

3.2. Cantidad de passwords finales que contiene rockyou_mod.dic

Para saber la cantidad de passwords que contiene el archivo modificado se debe ejecutar el código creado en Python.



```
anselmo@hp-pavilion:~/Descargas$ python3 modifica-diccio.txt rockyou.txt
El archivo rockyou_mod.dic contiene 10963183 contraseñas.
anselmo@hp-pavilion:~/Descargas$
```

Figura 7: Cantidad de contraseñas

4. Desarrollo (Paso 3)

4.1. Obtiene contraseña con hashcat con potfile

Previo a utilizar la herramienta hashcat se debe convertir la captura obtenida (handshake.pcap) a .hccapx, lo cual será realizado a partir del sitio <https://hashcat.net/cap2hashcat/>.

Luego se utilizará el siguiente comando que permitirá obtener lo solicitado: `hashcat -m 22000 158185-1716261282.hc22000 rockyou-mod.dic --potfile-path potfile.txt --force` Donde al ejecutar el comando se puede apreciar lo siguiente:

```

anselmo@hp-pavillon:~/Descargas$ hashcat -m 22000 158185_1716261282.hc22000 rockyou_mod.dic --potfile-path potfile.txt --force
hashcat (v6.2.5) starting

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.

=====
OpenCL API (OpenCL 2.0 pocl 1.8 Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: pthread-AMD A10-8700P Radeon R6, 10 Compute Cores 4C+6G, 4642/9348 MB (2048 MB allocatable), 4MCU

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 63

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename...: rockyou_mod.dic
* Passwords...: 10963183
* Bytes.....: 118845399
* Keyspace...: 10963176
* Runtime....: 2 secs

1813acb976741b446d43369fb96dbf90:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0
Session.....: hashcat

```

Figura 8: Hashcat con potfile

```

* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache built:
* Filename...: rockyou_mod.dic
* Passwords...: 10963183
* Bytes.....: 118845399
* Keyspace...: 10963176
* Runtime....: 2 secs

1813acb976741b446d43369fb96dbf90:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EPOL)
Hash.Target.....: 158185_1716261282.hc22000
Time.Started.....: Tue May 21 00:05:58 2024, (1 sec)
Time.Estimated...: Tue May 21 00:05:59 2024, (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 1763 H/s (8.93ms) @ Accel:128 Loops:128 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 2907/10963176 (0.03%)
Rejected.....: 1371/2907 (47.16%)
Restore.Point...: 1965/10963176 (0.02%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: Magandaako0 -> Dangerous0
Hardware.Mon.#1...: Temp: 76c Util: 95%

Started: Tue May 21 00:05:54 2024
Stopped: Tue May 21 00:06:01 2024
anselmo@hp-pavillon:~/Descargas$

```

Figura 9: Hashcat con potfile

```

anselmo@hp-pavillon:~/Descargas$ cat potfile.txt
15e1e0f80ed75380f627c6dc48207454b754983771ffc8031d89c5198d6fac76*5654522d31363435323133:Security0

```

Figura 10: Contraseña obtenida a partir del hashcat con potfile

4.2. Nomenclatura del output

En el output se puede observar una gran cantidad de parámetros distintos tales como características del dispositivo, la longitud máxima y mínima de la contraseña, la cantidad memoria para realizar el ataque, el diccionario que se utiliza indicando su nombre, cantidad de contraseñas y la cantidad de Bytes, en donde se puede notar Security0, que corresponde a la contraseña obtenida.

Además dicha contraseña se guarda en un potfile tal como se aprecia en la figura 10.

4.3. Obtiene contraseña con hashcat sin potfile

Para este caso se utilizará el comando `sudo hashcat m 22000 158185-1716261282.hc22000 rockyou_mod.dic --potfile-disable` que a diferencia del anterior es que los output son similares, sin embargo para este caso no se crea un archivo potfile que contenga la contraseña.

```

anselmo@hp-pavillon:~/Descargas$ hashcat -m 22000 158185_1716261282.hc22000 rockyou_mod.dic --potfile-disable
hashcat (v6.2.5) starting

OpenCL API (OpenCL 2.0 pocl 1.8 Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: pthread-AMD A10-8700P Radeon R6, 10 Compute Cores 4C+6G, 4642/9348 MB (2048 MB allocatable), 4MCU

Minimum password length supported by kernel: 8
Maximum password length supported by kernel: 63

Hashes: 1 digests; 1 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

Optimizers applied:
* Zero-Byte
* Single-Hash
* Single-Salt
* Slow-Hash-SIMD-LOOP

Watchdog: Temperature abort trigger set to 90c

Host memory required for this attack: 1 MB

Dictionary cache hit:
* Filename..: rockyou_mod.dic
* Passwords.: 10963176
* Bytes.....: 118845399
* Keyspace...: 10963176

1813acb976741b446d43369fb96dbf90:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (HFA-PBKDF2-PBKID+EAPOI)
Hash.Target.....: 158185_1716261282.hc22000
Time.Started.....: Tue May 21 08:08:47 2024 (1 sec)
Time.Estimated....: Tue May 21 08:08:48 2024 (0 secs)

```

Figura 11: Contraseña obtenida a partir del hashcat sin potfile

4.4. Nomenclatura del output

En base a la imagen anterior, por defecto, hashcat guarda el resultado de la operación en un archivo de texto en la carpeta de trabajo actual. Donde se identifica la contraseña, el modo de hash, el hash crackeado, las contraseñas del diccionario usadas y la cantidad de bytes, el tiempo corrido y otros parametros.

4.5. Obtiene contraseña con aircrack-ng

```

anselmo@hp-pavillon:~/Descargas$ sudo aircrack-ng -w rockyou_mod.dic handshake.cap
Reading packets, please wait...
Opening handshake.cap
Read 13 packets.

# BSSID      ESSID      Encryption
1  B0:48:7A:D2:DC:18  VTR-1645213  WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening handshake.cap
Read 13 packets.

1 potential targets

Aircrack-ng 1.6

[00:00:02] 2859/9285254 keys tested (1818.71 k/s)

Time left: 1 hour, 25 minutes, 3 seconds      0.03%

KEY FOUND! [ Security0 ]

Master Key   : 55 E1 E0 F0 8E 07 53 80 F6 27 C6 DC 48 20 74 54
              B7 54 98 37 71 FF C8 03 1D 89 C5 19 8D 6F AC 76

Transient Key : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

EAPOL HMAC   : 18 13 AC B9 76 74 1B 44 6D 43 36 9F B9 6D BF 90

```

Figura 12: Contraseña obtenida a través de aircrack-ng

4.6. Identifica y modifica parámetros solicitados por pyrcrack

En esta sección se procede a modificar el código pywd para trabajar con una captura de handshake previamente obtenida.

Se han modificado varios campos, como el SSID, aNonce, sNonce, apMac y cliMac, así como los valores de mic y data para los paquetes específicos. En este caso existen 3 valores data y 3 valores mic.

4.6 Identifica y modifica parámetros solicitados por pycrack 4 DESARROLLO (PASO 3)

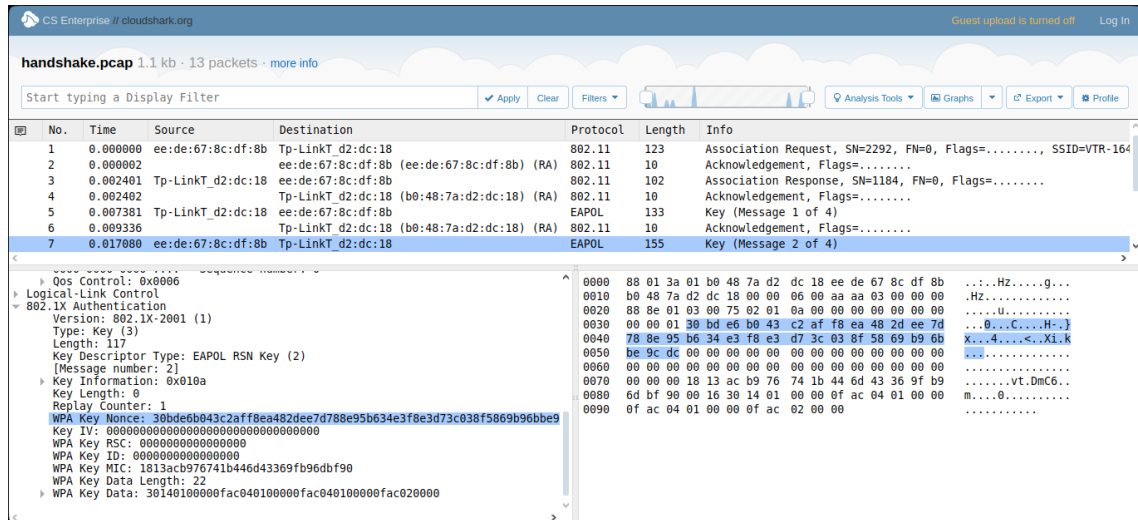
[illegible]

Figura 13: Código que modifica los campos requeridos

[illegible]

Figura 14: aNonce

4.6 Identifica y modifica parámetros solicitados por pycrack 4 DESARROLLO (PASO 3)



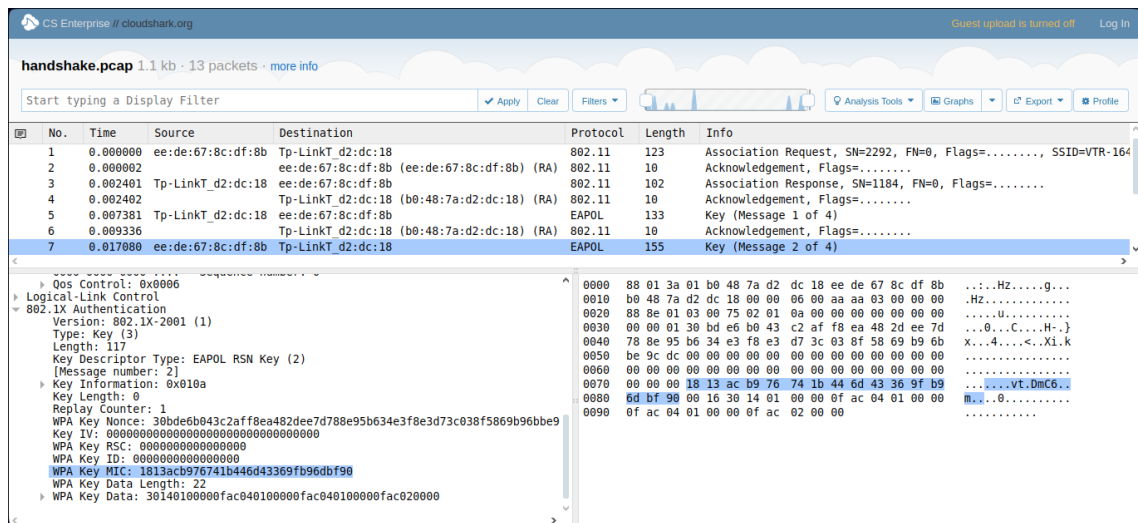
The image shows a Wireshark capture of a Wi-Fi handshake. The packet list on the left shows 7 packets. Packet 7 is selected, showing the details of the WPA Key Information. The sNonce field is highlighted in blue.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	802.11	123	Association Request, SN=2292, FN=0, Flags=....., SSID=VTR-164
2	0.000002	ee:de:67:8c:df:8b	ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA)	802.11	10	Acknowledgement, Flags=.....
3	0.002401	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	802.11	102	Association Response, SN=1184, FN=0, Flags=.....
4	0.002402	Tp-LinkT_d2:dc:18	Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....
5	0.007381	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	133	Key (Message 1 of 4)
6	0.009336	Tp-LinkT_d2:dc:18	Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....
7	0.017080	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	155	Key (Message 2 of 4)

Details of Packet 7 (EAPOL Key Message 2 of 4):

- QoS Control: 0x0000
- Logical-Link Control
- 802.1X Authentication
 - Version: 802.1X-2001 (1)
 - Type: Key (3)
 - Length: 117
 - Key Descriptor Type: EAPOL RSN Key (2)
 - [Message number: 2]
 - Key Information: 0x010a
 - Key Length: 0
 - Replay Counter: 1
 - WPA Key Nonce: 30bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9
 - Key IV: 00000000000000000000000000000000
 - WPA Key RSC: 00000000000000000000000000000000
 - WPA Key ID: 00000000000000000000000000000000
 - WPA Key MIC: 1813acb976741b446d43369fb96dbf90
 - WPA Key Data Length: 22
 - WPA Key Data: 301401000000fac0401000000fac0401000000fac020000

Figura 15: sNonce



The image shows a Wireshark capture of a Wi-Fi handshake, similar to Figure 15. The packet list on the left shows 7 packets. Packet 7 is selected, showing the details of the WPA Key Information. The Key MIC field is highlighted in blue.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	802.11	123	Association Request, SN=2292, FN=0, Flags=....., SSID=VTR-164
2	0.000002	ee:de:67:8c:df:8b	ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA)	802.11	10	Acknowledgement, Flags=.....
3	0.002401	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	802.11	102	Association Response, SN=1184, FN=0, Flags=.....
4	0.002402	Tp-LinkT_d2:dc:18	Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....
5	0.007381	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b	EAPOL	133	Key (Message 1 of 4)
6	0.009336	Tp-LinkT_d2:dc:18	Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....
7	0.017080	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	155	Key (Message 2 of 4)

Details of Packet 7 (EAPOL Key Message 2 of 4):

- QoS Control: 0x0000
- Logical-Link Control
- 802.1X Authentication
 - Version: 802.1X-2001 (1)
 - Type: Key (3)
 - Length: 117
 - Key Descriptor Type: EAPOL RSN Key (2)
 - [Message number: 2]
 - Key Information: 0x010a
 - Key Length: 0
 - Replay Counter: 1
 - WPA Key Nonce: 30bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9
 - Key IV: 00000000000000000000000000000000
 - WPA Key RSC: 00000000000000000000000000000000
 - WPA Key ID: 00000000000000000000000000000000
 - WPA Key MIC: 1813acb976741b446d43369fb96dbf90
 - WPA Key Data Length: 22
 - WPA Key Data: 301401000000fac0401000000fac0401000000fac020000

Figura 16: Key MIC I

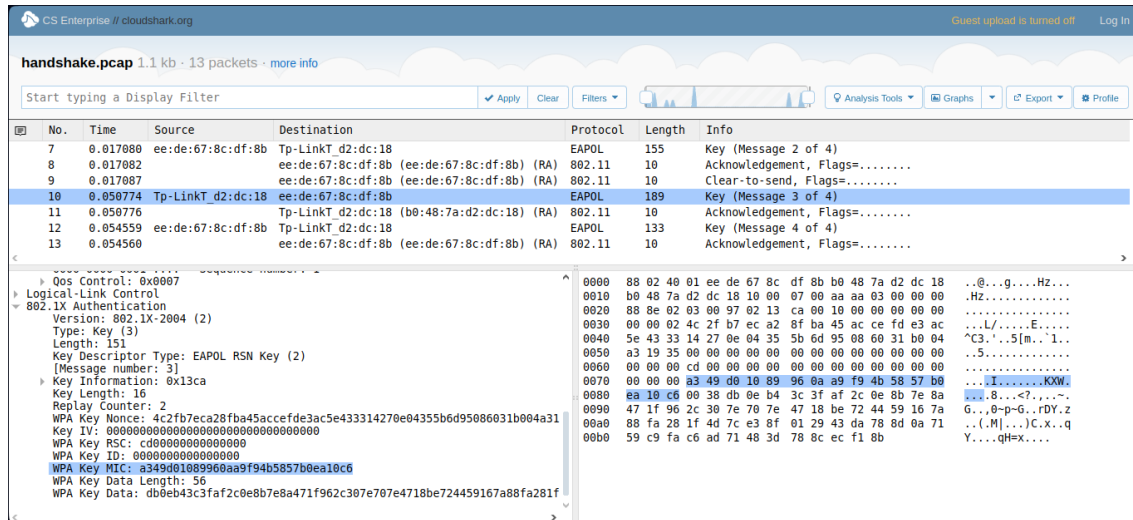


Figura 17: Key MIC II

4.7. Obtiene contraseña con pycrack

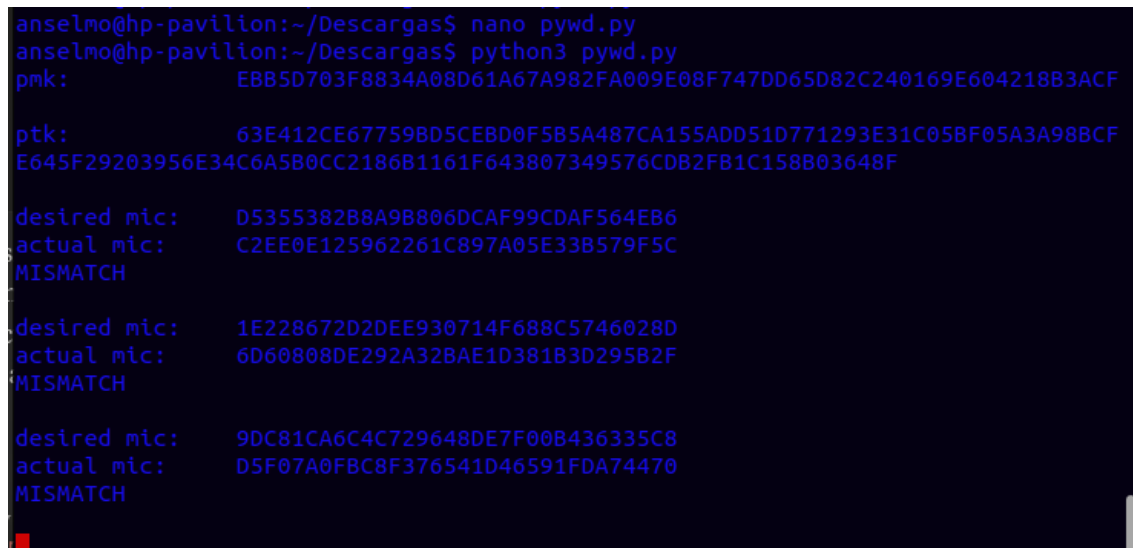


Figura 18: Contraseña obtenida a través de aircrack-ng

Conclusiones y comentarios

De esta experiencia se puede concluir que se cumplieron los objetivos propuestos entre los cuales de encontraba el realizar una serie de actividades utilizando diferentes herramientas con las cuales se permite comprender e identificar vulnerabilidades en las redes. Junto a esto se logro comprender el motivo por el cual ya no se utiliza cifrado WEP en la

actualidad. Esto es debido a que es una herramienta obsoleta, y que de esta manera se podría descifrar de manera sencilla.

Issues

Entre las problemáticas que se pueden encontrar se deben tener en cuenta las siguientes:

Al capturar el tráfico, puede que no se obtengan todos los paquetes necesarios, lo que resultará en datos incompletos para descifrar la contraseña.

Es por esto que se deben utilizar las herramientas proporcionadas de la mejor manera posible, considerando los factores que puedan generar fallos y de esta manera evitar errores y así garantizar una mejor captura de los paquetes.

Para esta experiencia se pudo notar un problema relacionado a La conversión de formatos de captura, como fue el caso al ejecutar cierto comando (pcapng). Es por esto que se tuvo que realizar la conversión al formato correspondiente (cap) para poder continuar con la ejecución. También se podría ver afectado en conversiones como fue el caso de .pcap a .hccapx, puede fallar o producir errores si no se realiza correctamente.

Para solucionar esta problemática es importante utilizar herramientas confiables, seguras y actualizadas para realizar la conversión necesaria con el menor riesgo de fallo posible.

Por otra parte se pueden presentar errores en la configuración del hashcat, debido al modo hash utilizado -m 22000 ya que al parecer se utiliza para redes WPA. Esto puede provocar fallas en el cifrado o retraso en la realización de las actividades propuestas.

Para disminuir el riesgo de estas acciones se debe realizar una lectura de las herramientas a utilizar, manteniendo la última versión actualizada para de esta manera, poder estar preparados a los diferentes escenarios a los que se puedan presentar en las experiencias de laboratorio.

Finalmente se puede mencionar que se debe contemplar una gestión eficiente de los resultados, ya que, una mala gestión de los archivos de salida y/o interpretación de los resultados puede llevar a confusión y/o pérdida de datos.

Es por esto que se deben ir realizando las actividades paso a paso, asignando nombres que se asimilen a la actividad que se está realizando, otra manera podría ser tomar apuntes de los resultados obtenidos y así tener un esquema de lo que se pretende obtener.