Informe Laboratorio 3

Sección 1

Alumno Rayen Millaman
e-mail: rayen.millaman@mail.udp.cl

Mayo de 2024

Índice

1. Descripción de actividades **2**

2. Desarrollo (PASO 1) **3**
2.1. En qué se destaca la red del informante del resto 3
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 4
2.4. Indica el tiempo que demoró en obtener la password 5
2.5. Descifra el contenido capturado 5
2.6. Describe como obtiene la url de donde descargar el archivo 5

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) **7**
4.1. Obtiene contraseña con hashcat con potfile 7
4.2. Nomenclatura del output 8
4.3. Obtiene contraseña con hashcat sin potfile 9
4.4. Nomenclatura del output 9
4.5. Obtiene contraseña con aircrack-ng 10
4.6. Identifica y modifica parámetros solicitados por pycrack 11
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

Antes de iniciar el proceso, se deben realizar las configuraciones necesarias para que la tarjeta de red Wi-Fi pueda capturar el intercambio de información en las redes existentes. Una vez finalizada esta etapa, se procede a mostrar las redes Wi-Fi activas en el área de cobertura del módem, utilizando la interfaz wlan0mon centrada solo en el canal 6 que es donde se presenta la red deseada.

```
informatica@informatica-14:~$ sudo airodump-ng wlan0mon --channel 6


 CH  6 ][ Elapsed: 1 min ][ 2024-05-17 09:19

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

 C6:BC:FB:43:F2:E8   -1   0        0        0    0   6   -1                    <length:  0>
 B0:48:7A:D2:DD:74  -53 100      746    39233  375   6   54e  WEP  WEP    SKA  WEP
 96:EC:A2:EE:6B:1F  -65   0      177       82    2   6  130   WPA2 CCMP   PSK  <length: 15>
 58:EF:68:47:59:C8  -75  22      491        0    0   6  130   OPN              cableadaTelemati
 58:EF:68:47:59:C6  -78  20      416        0    0   6  130   WPA2 CCMP   PSK  cableadaTelemati
 44:48:B9:DA:85:C8  -82   7      215        0    0   6  130   WPA2 CCMP   PSK  movistar2,4GHZ_D
 82:45:6B:0D:79:DA  -83   7      142       36    0   6  130   WPA2 CCMP   PSK  <length: 15>
 E4:AB:89:04:D3:34  -83   0       26        0    0   6  130   WPA2 CCMP   PSK  Himeko
 10:F0:68:99:86:A9  -84   0        5        0    0   6  130   WPA2 CCMP   PSK  DTI
 F8:5B:3B:4E:C0:53  -84   7       59        1    0   6  130   WPA2 CCMP   PSK  Xomi Pia
 18:35:D1:48:EB:39  -84   0       34        0    0   6  130   WPA2 CCMP   PSK  VTR-5376275
 20:AA:4B:31:A2:D4  -84   0       13        1    0   6  130   WPA2 CCMP   PSK  OF-CCFI
 10:F0:68:59:86:A8  -85   0       27        0    0   6  130   WPA2 CCMP   PSK  Servicio Tablet
 10:F0:68:99:86:A8  -86   0        0        0    0   6  130   WPA2 CCMP   PSK  Administrativos
 E6:AB:89:B4:8F:FE  -87   6       48        0    0   6  130   WPA2 CCMP   PSK  Hector
 10:F0:68:59:86:A9  -87   0        0        0    0   6  130   WPA2 CCMP   PSK  WiFi UCEN Admin
 10:F0:68:D9:86:A8  -87   3        0        3    0   6  130   WPA2 CCMP   PSK  Wifi_Ucentral
Quitting...
informatica@informatica-14:~$ sudo airodump-ng wlan0mon --channel 6 B0:48:7A:D2:DD:74 -w captura
```

Figura 1: Redes existentes capturadas.

En base a la Figura 7 se puede obtener que la red del informante se destaca por el uso de cifrada antigua WEP, a comparación de las demás que tienen cifrados WPA2. Además el tráfico generado por esta red tiene mayor procesamiento de datos sobre las demás y utiliza un proceso de autentificación SKA para compartir la clave secreta. Por lo que la red del informante corresponde a ESSID=WEP.

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

Se requiere un gran número de paquetes, ya que al intentar obtener la contraseña por fuerza bruta, la probabilidad de éxito aumenta en función de la cantidad de caracteres disponibles. Por ejemplo, para una contraseña de 3 dígitos que solo incluya caracteres alfabéticos, existen:

$26^3 = 17576$ posibles combinaciones
Luego, si solo se tiene 5000 paquetes la probabilidad para encontrar la contraseña mediante

fuerza bruta es considerablemente menor, en este caso si se aplica la ley de distribución binomial, es igual a:

$$\text{Probabilidad} = \binom{5000}{1} * (1 - 1/17576)^1 * (1 - 1/17{,}576)^{5000-1} = 5{,}63 * 10^{-12}$$

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

Ya identificado el nombre de la red del informante, se inicia la captura de paquetes con uso del siguiente comando para obtener el handshake donde se encuentra la clave entre el cliente y el punto de acceso inalámbrico.

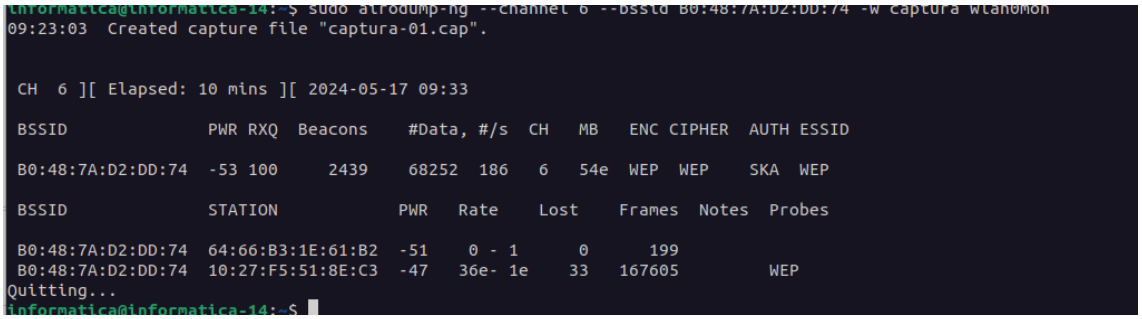

Figura 2: Comando para captura de tráfico.

Una vez captura e tráfico de red, se guarda los datos en una captura en dónde se encuentra la contraseña, mostrada por el siguiente comando:

Figura 3: Parametros obtenidos.

2.4. Indica el tiempo que demoró en obtener la password

Para obtener el tiempo tardado, se emplea el uso de time en la linea de comandos mostrado a continuación(real 0.106s):

```
Reading packets, please wait...
Opening captura-01.cap
Read 118368 packets.

1 potential targets

Attack will be restarted every 5000 captured ivs.


                                Aircrack-ng 1.6


                    [00:00:00] Tested 14 keys (got 26601 IVs)

   KB    depth   byte(vote)
    0    0/  1   12(36608) 8F(33792) 50(33024) A7(33024) 54(32512) 77(32256)
    1    0/ 13   B1(33280) 51(32768) EC(32768) E4(32512) 40(32000) BC(31744)
    2    0/  1   56(37376) AB(34304) 80(33536) 40(33280) 94(33280) 13(32768)
    3    0/  1   78(36096) C2(33280) CB(33280) 10(33024) 1E(32768) FC(32768)
    4    0/  1   90(37120) 89(33280) 28(32512) 53(32512) A5(32512) CC(32512)

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



real    0m0,106s
user    0m0,008s
sys     0m0,000s
```

Figura 4: Contraseña obtenida con su tiempo respectivo.

2.5. Descifra el contenido capturado

Una vez obtenida la llave decifrada, se hace uso de airdecap enviando como parametros la clave ya obtenida. Comando utilizado para visualizar de mejor manera la captura obtenida.

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

Después de descifrar el contenido capturado, se procede a analizar la información obtenida enviada por el informante en dónde todos los paquetes tenian protocolo ICMP con el mensaje del informante oculto en la data, la cual tenía contenida una URL correspondiente a bit.ly/-wpa2. Esta URL dirige a un archivo que contiene la captura del handshake en formato handshake.pcap.

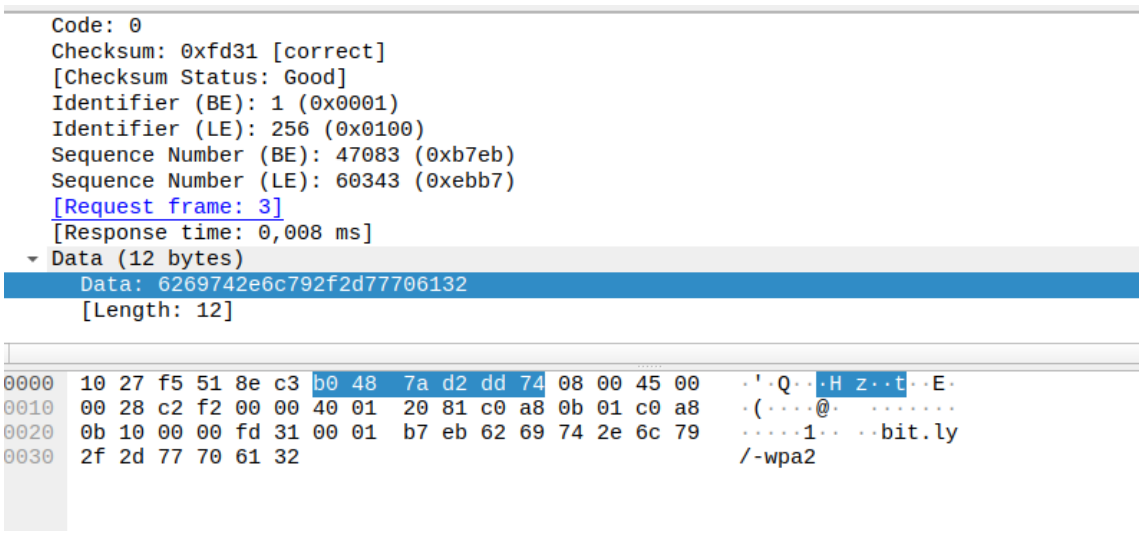

Figura 5: Captura obtenida.

3. Desarrollo (PASO 2)

3.1. Script para modificar diccionario original

El siguiente script utiliza comandos sed para procesar el archivo rockyou.txt, modificando su contenido y creando un nuevo archivo denominado 'rockyou_mod.dic'. El script realiza tres pasos principales: primero, elimina las entradas cuyo primer carácter es un dígito; luego, convierte la primera letra de las entradas restantes a mayúsculas; y finalmente, agrega un '0' al final de cada entrada.

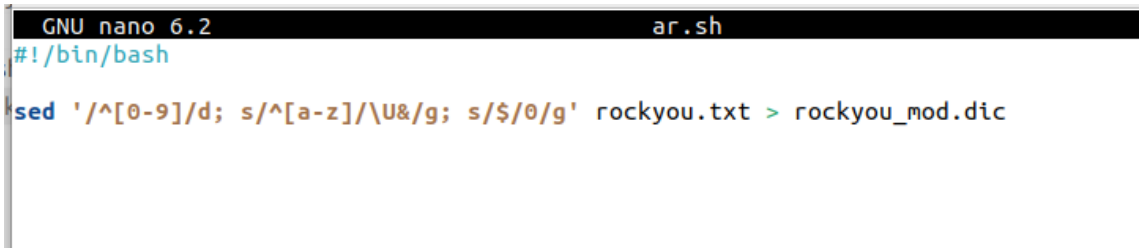

Figura 6: Script con codigo sed.

3.2. Cantidad de passwords finales que contiene rockyou_mod.dic

Luego se realiza un conteo en ambos archivos, teniendo una cantidad de 14,344,391 en el archivo original y un total de 11,059,798 (incluyendo contraseñas con caracteres especiales).

```
rayen@hp:~/Documents$ nano ar.sh
rayen@hp:~/Documents$ ./ar.sh
rayen@hp:~/Documents$ ls
ar.sh  hola  rockyou_mod.dic  rockyou.txt
rayen@hp:~/Documents$ wc -l rockyou.txt
14344391 rockyou.txt
rayen@hp:~/Documents$ wc -l rockyou_mod.dic
11059798 rockyou_mod.dic
rayen@hp:~/Documents$
```

Figura 7: Cantidad de contraseñas.

4. Desarrollo (Paso 3)

4.1. Obtiene contraseña con hashcat con potfile

Para la recuperación de contraseñas por fuerza bruta, se implementa la herramienta hashcat junto con un potfile, que permite llevar un registro detallado de las contraseñas descifradas correctamente, para ello se utilizara el archivo obtenido en el paso 1 (captura pcap) y el diccionario de contraseñas modificado, obtenido en el paso 2. Para la compatibilidad con hash se hara una converción de la captura al formato hc22000.
Luego se ejecuta el siguiente comando, el cual guarda las contraseñas correctas a un archivo hola.txt

```
rayen@hp:~/Documents$ hashcat -m 22000 archivo.hc22000 rockyou_mod.dic --potfile-path hola.txt
hashcat (v6.2.6-851-g6716447df) starting

OpenCL API (OpenCL 3.0 ) - Platform #1 [Intel(R) Corporation]
============================================================
* Device #1: Intel(R) UHD Graphics, 3377/6819 MB (1704 MB allocatable), 23MCU

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: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 940 MB

Dictionary cache built:
* Filename..: rockyou_mod.dic
```

Figura 8: Comando hashcat.

4.2. Nomenclatura del output

Esta salida de la Fígura 8 detalla información sobre el progreso del ataque, como la cantidad de hashes que se han procesado, la cantidad de contraseñas que se han encontrado y la velocidad a la que se está ejecutando el ataque.

Mientras que a la fígura a continuación muestra detalles de la respuesta al ataque

```
Dictionary cache built:
* Filename..: rockyou_mod.dic
* Passwords.: 11059798
* Bytes.....: 119977317
* Keyspace..: 11059791
* Runtime...: 1 sec

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

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target......: archivo.hc22000
Time.Started.....: Thu May 23 19:58:15 2024 (7 secs)
Time.Estimated...: Thu May 23 19:58:22 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou_mod.dic)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     6961 H/s (6.27ms) @ Accel:128 Loops:4 Thr:16 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 72676/11059791 (0.66%)
Rejected.........: 25572/72676 (35.19%)
Restore.Point....: 0/11059791 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Password0 -> Tasha210

Started: Thu May 23 19:57:50 2024
Stopped: Thu May 23 19:58:23 2024
```

Figura 9: Output hashcat obtenido.

Mientras que a la Fígura **??** muestra detalles de la respuesta al ataque, que parece estar dividida en 5 partes, limitadas por 'doble punto' ':'
Respuesta: 1813acb976741b446d43369fb96dbf90:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0

1. **1 parte;1813acb976741b446d43369fb96dbf90**
   Parametro que puede pertenecer a la contraseña generada por algoritmo hash.
2. **2 parte; b0487ad2dc18**
   Parametro posiblemente indicativo de la 'sal' agregada para dificultar el hash.
3. **3 parte ; eede678cdf8b:VTR-1645213**
   Parametro PMKID, correspondiente al identificador único que se utiliza para verificar y establecer conexiones seguras entre un punto de acceso inalámbrico (AP) y cliente.
4. **4 parte ; VTR-1645213**
   Nombre de la red atacada por uso de hash.

5. **5 parte ; Security0**
   Contraseña de la red a atacar.

4.3. Obtiene contraseña con hashcat sin potfile

Luego, se realiza el ataque pero sin uso de un archivo potfile evitando el reguistro de contraseñas exitosas, para ello dejamos esta inhabilitada en el comando.

```
rayen@hp:~/Documents$ hashcat -m 22000 archivo.hc22000 rockyou_mod.dic --potfile-disable
hashcat (v6.2.6-851-g6716447df) starting

OpenCL API (OpenCL 3.0 ) - Platform #1 [Intel(R) Corporation]
============================================================
* Device #1: Intel(R) UHD Graphics, 3377/6819 MB (1704 MB allocatable), 23MCU

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: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 940 MB
```

Figura 10: Comando hashcat sin potfile.

4.4. Nomenclatura del output

De igual modo que en el uso con potfile, se obtiene los mismos parametros obtenidos al realizar el ataque, cambiando solo en la velocidad de ejecució.
Para el tiempo de inicio, con potfile se tardó más tiempo en comenzar el proceso de crackeo (7 segundos) debido a que hashcat tuvo que buscar el hash objetivo en el potfile, mientras que sin potfile comenzo el proceso más rápido (2 segundos), ya que se inicio directamente el ataque.
Ádemas el número de loops es menor con potfile (4) que sin él (32) demostrando que hashcat necesitó menos iteraciones sobre el potfile para encontrar el hash objetivo.

```
Dictionary cache hit:
* Filename..: rockyou_mod.dic
* Passwords.: 11059791
* Bytes.....: 119977317
* Keyspace..: 11059791

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

Session..........: hashcat
Status...........: Cracked
Hash.Mode........: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target......: archivo.hc22000
Time.Started.....: Thu May 23 20:01:47 2024 (2 secs)
Time.Estimated...: Thu May 23 20:01:49 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.......: File (rockyou_mod.dic)
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     6682 H/s (6.66ms) @ Accel:8 Loops:32 Thr:32 Vec:1
Recovered........: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.........: 10106/11059791 (0.09%)
Rejected.........: 4218/10106 (41.74%)
Restore.Point....: 0/11059791 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Password0 -> Bigbaby0

Started: Thu May 23 20:01:44 2024
Stopped: Thu May 23 20:01:50 2024
```

Figura 11: Comando hashcat sin potfile.

4.5. Obtiene contraseña con aircrack-ng

Con uso de la siguiente herramienta aircrack-ng se realiza el ataque por fuerza bruta utilizando los mismos herramientas que en el paso anterior (uso de la captura y el diccionario de contraseñas)

```
ayen@hp:~/Documents$ aircrack-ng handshake.pcap -w rockyou_mod.dic
eading packets, please wait...
pening handshake.pcap
ead 13 packets.

  #  BSSID              ESSID                     Encryption

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

hoosing first network as target.

eading packets, please wait...
pening handshake.pcap
ead 13 packets.

 potential targets


                      Aircrack-ng 1.7 rev 9af4082f

    [00:00:00] 2737/9296197 keys tested (11440.83 k/s)
```

Figura 12: Comando aircrack-ng.

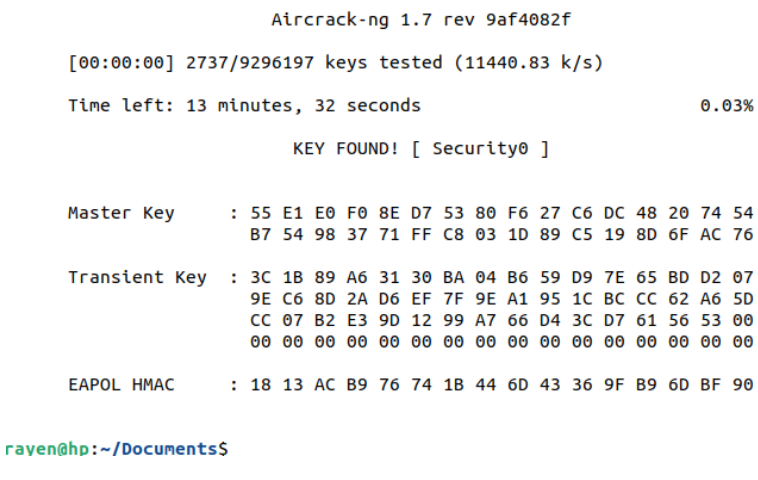

Figura 13: Resultados obtenidos con aircrack-ng.

Los resultados obtenidos al ejecutar el comando, muestra información sobre el tiempo estimado en realizar el ataque y revelando aspectos sobre la clave:

1. **Clave Maestra:** clave obtenida en formato hexadecimal.
2. **Transient Key:** clave de corta duración generada a partir de la Master Key y utilizada para cifrar y descifrar datos durante la comunicación.
3. **EAPOL HMAC:** valor de autenticación utilizado para verificar la integridad y autenticidad de los mensajes EAPOL.

Finalmente la contrase˜na obtenida es Security0 asociado a la red VTR-1645213.

4.6. Identifica y modifica parámetros solicitados por pycrack

Con uso de la herramienta pycrack, se modifica el archivo pywd.py de prueba para poder realizar un ataque por fuerza bruta, cambiando parametros correspondientes a la ruta del archrivo(diccionario de contraseña), nombre de la red ssid(VTR-1645213), los valores unicos sNonce y aNonce correspondientes de establecer la clave temporal, obtenidos en los paquetes 5 y 7 en el apartado WPA KEY NONCE. Por ultimo se modifican los datos de mic1, mic2 y mic3 tambien definidos en la captura en Wireshark, obtenido a partir de los datos y las claves compartidas entre el cliente y el punto de acceso, en el apartado Transmitter Address paro los paquetes de protocolo EAPOL, correspondientes a los paquetes 5,7 y 10 respectivamente.

```
174
175 if __name__ == "__main__":
176
177     #RunTest()
178     #Read a file of passwords containing
179     #passwords separated by a newline
180     with open('../rockyou_mod.dic') as f:
181         S = []
182         for l in f:
183             S.append(l.strip())
184     #ssid name
185     ssid = "VTR-1645213"
186     #ANonce
187     aNonce = a2b_hex('4c2fb7eca28fba45accefde3ac5e433314270e04355b6d95086031b004a31935')
188     #SNonce
189     sNonce = a2b_hex("30bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9cdc")
190     #Authenticator MAC (AP)
191     apMac = a2b_hex("b0487ad2dc18")
192     #Station address: MAC of client
193     cliMac = a2b_hex("eede678cdf8b")
194     #The first MIC
195     mic1 = "1813acb976741b446d43369fb96dbf90"
196     #The entire 802.1x frame of the second handshake message with the MIC field set to all zeros
197     data1 = a2b_hex("0103007502010a000000000000000000130bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c03
198     #The second MIC
199     mic2 = "a349d01089960aa9f94b5857b0ea10c6"
200     #The entire 802.1x frame of the third handshake message with the MIC field set to all zeros
201     data2 = a2b_hex("020300970213ca001000000000000000024c2fb7eca28fba45accefde3ac5e433314270e04355b6d95
202     #The third MIC
203     mic3 = "5cf0d63af458f13a83daa686df1f4067"
204     #The entire 802.1x frame of the forth handshake message with the MIC field set to all zeros
205     data3 = a2b_hex("0103005f02030a000000000000000000020000000000000000000000000000000000000000000000
206     #Run an offline dictionary attack against the access point
207     TestPwds(S, ssid, aNonce, sNonce, apMac, cliMac, data1, data2, data3, mic1, mic2, mic3)
```

Figura 14: Resultados obtenidos con aircrack-ng.

4.7. Obtiene contraseña con pycrack

Luego de haber modificado los parámetros necesarios para realizar el ataque por fuerza bruta, se procede a ejecutar el archivo pywd.py obteniendo con ello la contraseña.

```
rayen@hp:~/Documents$ iconv -f iso-8859-1 -t utf-8 rockyou_mod.dic > rockyouu_mod.dic
rayen@hp:~/Documents$ python3 PyCrack/pywd.py
!!!Password Found!!!
Desired MIC1:           1813acb976741b446d43369fb96dbf90
Computed MIC1:          1813acb976741b446d43369fb96dbf90

Desired MIC2:           a349d01089960aa9f94b5857b0ea10c6
Computed MIC2:          a349d01089960aa9f94b5857b0ea10c6

Desired MIC2:           5cf0d63af458f13a83daa686df1f4067
Computed MIC2:          5cf0d63af458f13a83daa686df1f4067
Password:               Security0
rayen@hp:~/Documents$
```

Figura 15: Resultados obtenidos con aircrack-ng.

Conclusiones y comentarios

En esta experiencia de laboratorio, se llevó a cabo un proceso completo para obtener la contraseña de una red inalámbrica. Desde la identificación de la red hasta la obtención de la contraseña, se demostró la importancia de la seguridad en las redes Wi-Fi y la vulnerabilidad de los sistemas que utilizan cifrado WEP.

Para realizar el ataque por fuerza bruta, se utilizaron varias herramientas como Hashcat, Aircrack-ng y PyCrack, adaptando un diccionario de contraseñas comunes al formato requerido y eliminando contraseñas que comenzaban con números.
En cuanto a las herramientas utilizadas, se puede mencionar que Hashcat es potente para realizar ataques de fuerza bruta, mientras que Aircrack-ng es más efectiva para cifrados antiguos como WEP y WPA, pero puede ser menos eficiente para cifrados más modernos y seguros. Por otro lado, PyCrack depende de bibliotecas Python, lo que puede implicar configuraciones adicionales y menor eficiencia en comparación con herramientas más especializadas.

Issues

Problematicas encontradas:
Al realizar la modificación del archivo, en primera instancia no realicé la conversión para que no se contaran los caracteres especiales. Esto causó un problema al intentar usar pycrack, ya que no se podía leer correctamente el archivo, su solución fue realizar una modificación en la escritura.

Un problema fue en el análisis y comparativas de los datos. Esto, debido a que los parámetros obtenidos para las distintas funciones de fuerza bruta, quizás no se implementarón a mayor comlejidad ni se realizó un análisis exhaustivo para interpretar adecuadamente los resultados, aunque se implementó a un modo básico obteniendo parametros iniciales.

Un problema se centra en la información necesaria para realizar la actividad 1, específicamente en la información sobre los comandos y cómo ejecutarlos correctamente. La solución consistió en aprender en el momento cómo utilizar estos comandos, utilizando un chat con GPT.

Finalmente, otro problema se centra en el manejo rápido de comandos y el uso eficiente de recursos. Se presentan dificultades al no poder realizar estas acciones sin tener que recurrir a consultas en Google o al uso de inteligencia artificial. Ante esto, se decidió otorgar mayor tiempo en la implementación y ejecución dentro del informe para mejorar esta habilidad.