

# Informe Laboratorio 3

## Sección 2

Claudio Lopez  
e-mail: claudio.lopez1@mail.udp.cl

Mayo de 2023

## Índice

<b>1. Descripción de actividades</b>	<b>2</b>
<b>2. Desarrollo (PASO 1)</b>	<b>2</b>
2.1. Identificar en qué se destaca la red del informante del resto . . . . .	2
2.2. Explica matemáticamente por qué 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 . . . . .	4
2.5. descifra el contenido capturado . . . . .	5
2.6. describe como obtiene la url de donde descargar el archivo . . . . .	5
<b>3. Desarrollo (PASO 2)</b>	<b>6</b>
3.1. indica script para modificar diccionario original . . . . .	6
3.2. cantidad de passwords finales que contiene rockyou_mod.dic . . . . .	7
<b>4. Desarrollo (Paso 3)</b>	<b>8</b>
4.1. obtiene contraseña con hashcat con potfile . . . . .	8
4.2. identifica nomenclatura del output . . . . .	8
4.3. obtiene contraseña con hashcat sin potfile . . . . .	8
4.4. identifica nomenclatura del output . . . . .	9
4.5. obtiene contraseña con aircrack-ng . . . . .	9
4.6. identifica y modifica parámetros solicitados por pycrack . . . . .	10
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 RockyouLinks to an external site. (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.
3. Todos los strings que comiencen con número toca eliminarlos del diccionario. Indique la cantidad de contraseñas que contiene el diccionario modificado debe llamarse rock-you\_mod.dic A continuación un ejemplo de cómo se modifican las 10 primeras líneas del diccionario original.

## 2. Desarrollo (PASO 1)

### 2.1. Identificar en qué se destaca la red del informante del resto

Para identificar la red, se instaló aircrack-ng, con el fin de acceder al modo monitor y realizar las siguientes actividades, para esto se utilizó el siguiente comando **sudo apt-get install aircrack-ng**. Posteriormente, se identificó la interfaz de la tarjeta wifi del equipo, el cual es **wlp2s0** y con el comando **sudo airmon-ng start wlp2s0**. Luego se utilizó la herramienta **Airodump-ng** con el siguiente comando **sudo airodump-ng wlp2s0mon -w cripto2**, con el fin de capturar paquetes y guardarlos en un archivo .cap, el resultado fue el referenciado en la figura 1.

## 2.2 Explica matemáticamente por qué se requieren más de 5000 paquetes para obtener la pass

BSSID	PWR	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
E6:AB:89:1C:85:38	-1	0	0	0	6	-1				<length: 0>
FE:68:46:F4:AC:A9	-41	439	1279	4	6	130	WPA2	CCMP	PSK	iPhone de Kevin Cabrera
B0:48:7A:D2:DD:74	-47	603	20896	0	8	54e	WEP	WEP	SKA	WEP
7C:95:F3:C0:76:95	-53	52	0	0	1	195	OPN			VIP
7C:95:F3:C0:76:97	-54	75	0	0	1	195	WPA2	CCMP	PSK	Hybrid Rooms
7C:95:F3:C0:76:91	-54	83	296	0	1	195	OPN			WiFi_AlumnosUDP
7C:95:F3:C0:76:93	-54	77	0	0	1	195	OPN			EIT
7C:95:F3:C0:76:94	-53	69	0	0	1	195	OPN			WiFi_AdministrativosUDP
7C:95:F3:C0:76:90	-53	85	1	0	1	195	WPA2	CCMP	PSK	wifi_anipass
7C:95:F3:C0:76:96	-54	80	0	0	1	195	OPN			WiFi_InvitadosUDP
CC:ED:DC:1C:0E:71	-56	348	255	0	13	130	WPA2	CCMP	PSK	Myhome
98:FC:11:B6:B6:B9	-53	245	6847	17	6	130	WPA2	CCMP	PSK	Telematica
58:EF:68:47:59:C8	-59	248	22	0	11	130	OPN			cableadaTelematica-Invitado
58:EF:68:47:59:C6	-63	242	22	0	11	130	WPA2	CCMP	PSK	cableadaTelematica
8A:D8:1B:C6:83:E9	-63	328	0	0	2	195	WPA2	CCMP	PSK	<length: 0>
8A:D8:1B:C6:83:E9	-63	336	23	0	2	195	WPA2	CCMP	PSK	FAMILIAGL_EXT
CC:D4:A1:D7:81:DD	-70	128	0	0	4	270	WPA2	CCMP	PSK	HUAWEI-B2368-D781DD
B4:1C:30:B5:EA:07	-68	89	2	0	10	130	WPA2	CCMP	PSK	ZTE_B5EA07
26:96:82:26:A7:5E	-68	96	1	0	11	130	WPA2	CCMP	PSK	Martin1
E8:DE:27:B2:7E:E8	-67	94	5	0	10	54	WPA2	CCMP	PSK	Clinica-gym
C0:05:C2:E3:09:41	-66	325	12	0	1	130	WPA2	CCMP	PSK	CAFM
48:D3:43:B7:0C:61	-70	153	0	0	1	130	WPA2	CCMP	PSK	VTR-9108176-24
50:17:FF:3B:23:67	-71	14	30	0	6	195	WPA2	CCMP	PSK	Hybrid Rooms
00:15:6D:72:C5:1D	-70	40	0	0	8	130	WPA2	CCMP	PSK	(P) \$6.000 wifi.toesca@gmail.com
E4:AB:89:0C:85:38	-72	4	0	0	6	130	WPA2	CCMP	PSK	Juan Pablo
AC:F8:CC:1D:60:60	-76	74	6	0	1	130	WPA2	CCMP	PSK	VTR-8492879
E4:AB:89:70:2F:74	-73	44	12	0	10	130	WPA2	CCMP	PSK	HUAWEI-B2368-702F74
E4:AB:89:67:33:90	-73	134	0	0	1	270	WPA2	CCMP	PSK	Otakus depa
48:D3:43:61:91:89	-74	78	4	0	1	130	WPA2	CCMP	PSK	VTR-8198506

Figura 1: Captura de las redes a través de airodump.

Para finalizar esta sección se logró identificar la red del informante, la cual es **WEP**, se llegó a esta conclusión debido a que es la única red con encriptación WEP y además genera demasiado tráfico en comparación a las otras redes.

## 2.2. Explica matemáticamente por qué se requieren más de 5000 paquetes para obtener la pass

Se requieren más de 5000 paquetes debido a que WEP utiliza vectores de inicialización (IVs) de 24 bits, el problema que puede tener esto es la repetición de estos vectores en una red activa. como es de 24 bits, se tiene que el resultado de la probabilidad de encontrar vectores iguales es igual a lo mostrado en la ecuacion 1.

$$1 - [(2^{24} - 1)/2^{24}]. \quad (1)$$

Por lo tanto, el número esperado de IVs repetidos es de Probabilidad de IVs repetidos \* Número total de paquetes capturados, para este caso el resultado se muestra en la ecuacion 2.

$$[1 - ((2^{24} - 1)/2^{24})] * 5000 \quad (2)$$

Por lo tanto, entre más paquetes, hay más probabilidad de encontrar un IV repetido.

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

Para obtener la password con ataque por defecto de aircrack, se utilizó el siguiente comando **sudo aircrack-ng -b B0:48:7A:D2:DD:74 crito2.cap**, como se muestra en la

figura 2.

```
Aircrack-ng 1.6

[00:00:01] Tested 11580 keys (got 18756 IVs)

KB  depth  byte(vote)
0   0/ 3    12(26624) 91(24320) CC(24064) 1A(23552) 92(22784) 9C(22784) 88(22528) 85(22272) BF(22272)
1   3/ 24    34(23552) FB(23296) 5A(22784) 11(22784) 51(22528) 62(22528) 88(22528) DD(22528) F2(22528)
2   0/ 1     56(32512) BC(24064) 9A(23808) B6(23552) 4E(23040) FA(23040) 0F(22784) A4(22784) 36(22528)
3  10/ 16    78(23040) A6(23040) B1(23040) FD(23040) 67(22784) 9B(22784) A1(22528) D3(22528) 09(22016)
4   8/ 11    59(22784) 61(22528) A4(22528) 01(22272) 23(22272) 5A(22272) 4A(22016) 68(22016) 7B(22016)

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

claudiolpz@claudiolpz-HP-ProBook-640-G2:~/Desktop$
```

Figura 2: Búsqueda de la password de la red WEP.

La password que se encontró fue la siguiente **12:34:56:78:90**

## 2.4. indica el tiempo que demoró en obtener la password

Como indica la figura 2, el tiempo que se demoró para poder encontrar la Key fue de 1 segundo.

## 2.5. descifra el contenido capturado

Para descifrar se utiliza el comando `sudo airdecap-ng -w 12:34:56:78:90 cripto2-01.cap`, una vez ejecutado este comando, se crea un archivo con los paquetes descifrados, el resultado se muestra en la figura 3.

1	0.000000	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65124/25854, ttl=64 (repl...
2	0.000103	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65124/25854, ttl=64 (requ...
3	0.000142	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65125/26110, ttl=64 (repl...
4	0.000873	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65125/26110, ttl=64 (requ...
5	0.001500	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65126/26366, ttl=64 (repl...
6	0.002043	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65126/26366, ttl=64 (requ...
7	0.002376	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65127/26622, ttl=64 (repl...
8	0.002916	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65127/26622, ttl=64 (requ...
9	0.003140	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65128/26878, ttl=64 (repl...
10	0.003795	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65128/26878, ttl=64 (requ...
11	0.004064	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65129/27134, ttl=64 (repl...
12	0.004476	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65129/27134, ttl=64 (requ...
13	0.004747	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65130/27390, ttl=64 (repl...
14	0.005287	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65130/27390, ttl=64 (requ...
15	0.005605	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65131/27646, ttl=64 (repl...
16	0.005969	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65131/27646, ttl=64 (requ...
17	0.006556	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65132/27902, ttl=64 (repl...
18	0.006884	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65132/27902, ttl=64 (requ...
19	0.007512	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65133/28158, ttl=64 (no r...
20	0.009870	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65134/28414, ttl=64 (repl...
21	0.010425	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65134/28414, ttl=64 (requ...
22	0.010667	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65135/28670, ttl=64 (repl...
23	0.011123	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65135/28670, ttl=64 (requ...
24	0.011733	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65136/28926, ttl=64 (repl...
25	0.012109	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65136/28926, ttl=64 (requ...
26	0.012354	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65137/29182, ttl=64 (no r...
27	0.020245	192.168.11.15	192.168.11.1	ICMP	74 Echo (ping) request	id=0x000b, seq=65139/29694, ttl=64 (repl...
28	0.020948	192.168.11.1	192.168.11.15	ICMP	74 Echo (ping) reply	id=0x000b, seq=65139/29694, ttl=64 (requ...

  

Frame 13: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)	0000	b0 48 7a d2 dd 74 e0 0a f6 3c e0 91 08 00 45 00	H...
Ethernet II, Src: e0:0a:f6:3c:e0:91 (e0:0a:f6:3c:e0:91), Dst: Tp-LinkT_d2:dd:74 (b0:48:7a:d2:dd:74)	0010	00 3c 66 37 40 00 40 01 3d 29 c0 a8 0b 0f c0 a8	<f7@: =).....
Internet Protocol Version 4, Src: 192.168.11.15, Dst: 192.168.11.1	0020	0b 01 08 00 38 1d 00 0b fe 6a 7a 4f 5a 64 00 00	...8...jz0Zd...
Internet Control Message Protocol	0030	00 00 25 13 00 00 00 00 00 00 59 6d 0c 30 4c 0d	...m10m
Type: 8 (Echo (ping) request)	0040	78 35 4c 33 64 77 59 54 4a 68	5L3dwYT Jf
Code: 0			
Checksum: 0x381d [correct]			
[Checksum Status: Good]			
Identifier (BE): 11 (0x000b)			
Identifier (LE): 2810 (0x0b00)			
Sequence number (BE): 65130 (0xfe6a)			
Sequence number (LE): 27390 (0x6afe)			
[Response frame: 14]			
Timestamp from icmp data: May 9, 2023 09:49:46.000000000 -04			
[Timestamp from icmp data (relative): 0.682863000 seconds]			
Data (24 bytes)			
Data: 63130b00000000000000596d0c304c0d78354c33647759544a68			
[Length: 24]			

Figura 3: Captura de paquetes, descifrados.

## 2.6. describe como obtiene la url de donde descargar el archivo

Se observa que los paquetes de la captura del punto anterior, el texto que se repite es **Yml0Lmx5L3dwYTJf**, luego se procedió a decodificar el texto, el cual está en base 64 y se obtuvo el siguiente link acortado **http://bit.ly/wpa2\_**, el cual lleva al link **https://www.cloudshark.org/captures/b5b39e1c51eb**. Este punto se puede evidenciar en las imágenes 3, 4 y 5.

**Base64**

Ym10Lmx5L3dwYTJf

**Decode Base64 to ASCII**

**Text**

bit.ly/wpa2\_

The result of Base64 decoding will appear here

Figura 4: Decode de Base64 a ASCII.

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-1645213
2	0.000002		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 (b0:40:7a: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 (b0:40:7a: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) (RA)	802.11	10	Acknowledgement, Flags=.....
9	0.017087		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	0.050776		Tp-LinkT_d2:dc:18 (b0:40:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....
12	0.054559	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	133	Key (Message 4 of 4)
13	0.054560		ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA)	802.11	10	Acknowledgement, Flags=.....

Figura 5: Página enviada por el informante.

### 3. Desarrollo (PASO 2)

#### 3.1. indica script para modificar diccionario original

Se logró realizar el script mostrado en la figura 6 para modificar el diccionario.

### 3.2 cantidad de passwords finales que contiene rockyou\_mod.dicDESARROLLO (PASO 2)

```
script.py > ...
1  import re
2
3  def modificar_diccionario(diccionario):
4      diccionario_modificado = diccionario.copy()
5      for clave, valor in diccionario.items():
6          if not re.match(r'^[a-zA-Z]', valor):
7              del diccionario_modificado[clave]
8          else:
9              diccionario_modificado[clave] = valor.capitalize() + '0'
10     return diccionario_modificado
11
12 def guardar_diccionario_modificado(diccionario, nombre_archivo):
13     with open(nombre_archivo, 'w') as archivo:
14         contador = 0
15         for valor in diccionario.values():
16             archivo.write(valor + '\n')
17             contador += 1
18     print(f"Se guardaron {contador} strings en el archivo {nombre_archivo}")
19
20
21 # Carga el diccionario desde un archivo de texto
22 def cargar_diccionario(nombre_archivo):
23     diccionario = {}
24     with open(nombre_archivo, 'r', encoding='latin-1') as archivo:
25         for linea in archivo:
26             linea = linea.strip()
27             diccionario[linea] = linea
28     return diccionario
29
30
31 # Nombre del archivo de entrada
32 archivo_entrada = "rockyou.txt"
33 # Nombre del archivo de salida modificado
34 archivo_salida = "rockyou_mod.dic"
35
36 # Cargar el diccionario original
37 diccionario_original = cargar_diccionario(archivo_entrada)
38
39 # Modificar el diccionario
40 diccionario_modificado = modificar_diccionario(diccionario_original)
41
42 # Guardar el diccionario modificado en un archivo
43 guardar_diccionario_modificado(diccionario_modificado, archivo_salida)
44
```

Figura 6: Script para modificar el diccionario.

### 3.2. cantidad de passwords finales que contiene rockyou\_mod.dic

Se logró obtener un total de **10956580** contraseñas en el diccionario. Esto se puede evidenciar en la figura 7.

```
● claudio@claudio-B365-M-AORUS-ELITE:~/Escritorio/cripto/lab3$ ./bi
Se guardaron 10956580 strings en el archivo rockyou_mod.dic
○ claudio@claudio-B365-M-AORUS-ELITE:~/Escritorio/cripto/lab3$
```

Figura 7: Ejecución del script.

## 4. Desarrollo (Paso 3)

### 4.1. obtiene contraseña con hashcat con potfile

Para este punto se utilizó aircrack para pasar el archivo handshake.pcap a un archivo con la extensión hccapx, el comando utilizado fue el siguiente: **aircrack-ng handshake.pcap -j converted1**. Luego se instaló hashcat a través de GitHub, luego se utilizó el comando **hashcat -m 2500 --deprecated-check-disable converted1.hccapx rockyou\_mod.dic -o /home/claudio/Escritorio/cripto/lab3/hashcat.potfile**, con el fin de obtener la contraseña con potfile. Se obtuvo el siguiente resultado (figura 8).

The screenshot shows a terminal window with the following output from hashcat:

```

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: 1323 MB

Dictionary cache built:
+ Filename.: rockyou_mod.dic
+ Passwords.: 10956580
+ Bytes.....: 118595095
+ Keyspace.: 10956574
+ Runtime...: 0 secs

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 2500 (WPA-EAPOL-PBKDF2)
Hash.Target.....: VTR-1645213 (AP:b0:48:7a:d2:dc:18 STA:ee:de:67:8c:df:8b)
Time.Started.....: Tue May 9 21:56:03 2023 (0 secs)
Time.Estimated...: Tue May 9 21:56:03 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 300.9 kH/s (9.88ms) @ Accel:8 Loops:512 Thr:128 Vec:1
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 39122/10956574 (0.36%)
Rejected.....: 14546/39122 (37.18%)
Restore.Point...: 0/10956574 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1...: Password0 -> Buggaboo0
Hardware.Mon.#1...: Temp: 59c Fan: 0% Util: 29% Core:1935MHz Mem:5750MHz Bus:16

Started: Tue May 9 21:56:01 2023
Stopped: Tue May 9 21:56:04 2023

```

Overlaid on the terminal is a text editor window titled "hashcat.potfile" showing the following content:

```

1 b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0

```

Figura 8: Obtención de contraseña con potfile.

### 4.2. identifica nomenclatura del output

En el archivo hashcat.potfile visto en la figura 8 se describen los siguientes campos, MAC del AP, MAC del cliente, SSID y contraseña (Security0).

### 4.3. obtiene contraseña con hashcat sin potfile

Para este punto se utilizó el comando **hashcat -m 2500 --deprecated-check-disable converted1.hccapx rockyou\_mod.dic** y se obtuvo el siguiente resultado (figura 9)



```
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: 1323 MB

Dictionary cache built:
* Filename..: rockyou_mod.dic
* Passwords.: 10956580
* Bytes.....: 118595095
* Keyspace..: 10956574
* Runtime...: 0 secs

b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 2500 (WPA-EAPOL-PBKDF2)
Hash.Target.....: VTR-1645213 (AP:b0:48:7a:d2:dc:18 STA:ee:de:67:8c:df:8b)
Time.Started.....: Tue May 9 21:57:37 2023 (0 secs)
Time.Estimated...: Tue May 9 21:57:37 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 308.9 kH/s (9.72ms) @ Accel:64 Loops:256 Thr:32 Vec:1
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 75676/10956574 (0.69%)
Rejected.....: 26524/75676 (35.05%)
Restore.Point....: 0/10956574 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Password0 -> Swimming!0
Hardware.Mon.#1..: Temp: 61c Fan: 0% Util: 7% Core:1935MHz Mem:5750MHz Bus:16

Started: Tue May 9 21:57:35 2023
Stopped: Tue May 9 21:57:38 2023
```

Figura 9: Obtención de contraseña sin potfile.

#### 4.4. identifica nomenclatura del output

Se puede observar que al no utilizar potfile, este no se guarda en un archivo. Pero el output es mostrado en la consola, el punto importante para esta actividad está abajo del campo **Dictionary cache built**, se obtiene MAC del AP, MAC del cliente, SSID y contraseña (Security0).

#### 4.5. obtiene contraseña con aircrack-ng

En este punto se procedió a obtener la contraseña con la herramienta aircrack-ng, el comando utilizado fue el siguiente: **aircrack-ng -w rockyou\_mod.dic -b b0:48:7a:d2:dc:18**

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

**handshake.pcap -l resultado.txt.** La contraseña encontrada es la misma que en los puntos anteriores (Security0), el resultado se muestra en la imagen 10.

```
claudio@claudio-B365-M-AORUS-ELITE:~/Escritorio/crpto/lab3$ aircrack-ng -w rockyou_mod.dic -b b0:48:7a:d2:dc:18 handshake.pcap -l resultado.txt
Reading packets, please wait...
Opening handshake.pcap
Read 13 packets.

1 potential targets

Aircrack-ng 1.6

[00:00:00] 3069/9301501 keys tested (15659.69 k/s)

Time left: 9 minutes, 53 seconds                                0.03%

KEY FOUND! [ Security0 ]

Master Key      : 55 E1 E0 F0 0E D7 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   : FD FF 61 91 F1 F3 26 71 48 23 D6 DE 05 C0 B2 88
                  DF 64 B2 3C 18 89 A6 31 30 BA 04 B6 59 D9 7E 65
                  BD D2 07 9E C6 8D 2A D6 EF 7F 9E A1 95 1C BC CC
                  62 A6 5D CC 07 B2 E3 9D 12 99 A7 66 D4 ED 3C D7

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

Figura 10: Obtención de contraseña a través de aircrack.

#### 4.6. identifica y modifica parámetros solicitados por pycrack

Para este punto se descargó el código de Python Pycrack, además se modificaron los campos ssid ("VTR-1645213"), aNonce (WPA Key Nonce del AP), sNonce (WPA Key Nonce del cliente), apMac (MAC del AP), cliMac (MAC del cliente), mic1 (WPA Key MIC del segundo paquete y para los siguientes mic se repite esta instrucción, pero con los datos de los próximos dos paquetes), data1 (Campo completo del 802.1X Authentication, pero con los bytes del mic en seteados en cero, cabe destacar que este paso se repite para los dos paquetes siguientes), mic2, data2, mic3, data3. Estos datos se obtienen por la captura del handshake descargada. Los parámetros se muestran en la figura 11 y además en la figura 12 se evidencia los parametros del paquete 2, el cual se utiliza en el mic1 y data1.

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

[illegible]

Figura 11: Código modificado en Pycrack.

Time	Source	Destination	Protocol	Length	Info
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	(... 802.11	10	Acknowledgement, Flags=.....
9.0.017087		ee:de:67:8c:df:8b	(... 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)
11.0.050776		Tp-LinkT_d2:dc:18	(... 802.11	10	Acknowledgement, Flags=.....
12.0.054559	ee:de:67:8c:df:8b	Tp-LinkT_d2:dc:18	EAPOL	133	Key (Message 4 of 4)
13.0.054560		ee:de:67:8c:df:8b	(... 802.11	10	Acknowledgement, Flags=.....
Frame 7: 155 bytes on wire (1240 bits), 155 bytes captured (1240 bits)					
IEEE 802.11 QoS Data, Flags: .....T					
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: 30bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c03...					
Key IV: 00000000000000000000000000000000					
WPA Key RSC: 0000000000000000					
WPA Key ID: 0000000000000000					
WPA Key MIC: 1813acb976741b446d43369fb96dbf9e					
WPA Key Data Length: 22					
WPA Key Data: 30140100000fac040100000fac040100000fac020000					

Figura 12: Captura de Handshake.

## 4.7. obtiene contraseña con pycrack.

Una vez identificamos y modificamos los campos se procedió a ejecutar el código, dando el resultado mostrado en la figura 13.

```
• claudio@claudio-B365-M-AORUS-ELITE:~/Escritorio/cripto/lab3/PyCrack-master$ /bin/python3
!!!Password Found!!!
Desired MIC1:      1813acb976741b446d43369fb96dbf90
Computed MIC1:     1813acb976741b446d43369fb96dbf90

Desired MIC2:      a349d01089960aa9f94b5857b0ea10c6
Computed MIC2:     a349d01089960aa9f94b5857b0ea10c6

Desired MIC2:      5cf0d63af458f13a83daa686df1f4067
Computed MIC2:     5cf0d63af458f13a83daa686df1f4067
Password:          Security0
```

Figura 13: Resultado de Pycrack.

## Conclusiones y comentarios

Para el ámbito de romper contraseñas se puede evidenciar que existen varias herramientas, en el caso de este laboratorio se logró trabajar con tres de estas herramientas (aircrack-ng, hashcat y pycrack).

Cabe destacar que unos de los problemas presentados en este trabajo fue en al emplear por segunda vez hashcat, esto es debido a que para volver a utilizar esta herramienta con los mismos parámetros había que borrar los logs y los archivos que este generaba.