

# Informe Laboratorio 3

## Sección 01

Rafael Maturana Wullfrodt  
rafael.maturana@mail.udp.cl

Mayo de 2024

## Índice

<b>1. Descripción de actividades</b>	<b>2</b>
<b>2. Desarrollo (PASO 1)</b>	<b>2</b>
2.1. En qué se destaca la red del informante del resto . . . . .	4
2.2. Explica matemáticamente porqué se requieren más de 5000 paquetes para obtener la pass . . . . .	5
2.3. Obtiene la password con ataque por defecto de aircrack-ng . . . . .	5
2.4. Indica el tiempo que demoró en obtener la password . . . . .	6
2.5. Descifra el contenido capturado . . . . .	6
2.6. Describe como obtiene la url de donde descargar el archivo . . . . .	7
<b>3. Desarrollo (PASO 2)</b>	<b>7</b>
3.1. Script para modificar diccionario original . . . . .	8
3.2. Cantidad de passwords finales que contiene rockyou_modificado.dic . . . . .	9
<b>4. Desarrollo (Paso 3)</b>	<b>10</b>
4.1. Obtiene contraseña con hashcat con potfile . . . . .	11
4.2. Nomenclatura del output . . . . .	12
4.3. Obtiene contraseña con hashcat sin potfile . . . . .	13
4.4. Nomenclatura del output . . . . .	14
4.5. Obtiene contraseña con aircrack-ng . . . . .	15
4.6. Identifica y modifica parámetros solicitados por pycrack . . . . .	17
4.7. Obtiene contraseña con pycrack . . . . .	18

## 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 las redes inalámbricas y de una forma que solo usted, como experto en informática y telecomunicaciones, logrará esclarecer.

1. Identifique cuál 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 mayúscula y agregue un cero al final de la password.

Todos los strings que comiencen con número deben eliminarse del diccionario. Indique la cantidad de contraseñas que contiene el diccionario modificado, que 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 del `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 dos técnicas, identificando claramente cada campo.

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

## 2. Desarrollo (PASO 1)

Para esta parte del laboratorio, lo primero que se debe hacer es iniciar en modo monitor. Para esto, se utilizan los siguientes comandos:

```
sudo airmon-ng
```

Este comando muestra las interfaces disponibles para utilizar con este programa, indicando cuál se puede poner en modo monitor, como se ve en la figura.

Se procede a desactivar la interfaz para luego inicializarla en modo monitor con el siguiente comando:

```
sudo airmon-ng check kill
```

Luego, se inicia en modo monitor la interfaz encontrada, que es wlan0, con el siguiente comando:

```
sudo airmon-ng start wlan0
```

Los resultados se muestran en la figura 1.

```

(rafa@kali)-[~]
$ sudo airmon-ng
[sudo] password for rafa:

PHY      Interface  Driver      Chipset
phy0 wlan0      rtw_8822ce  Realtek Semiconductor Co., Ltd. RTL8822CE 802.11ac PCIe Wireless Network Adapter

(rafa@kali)-[~]
$ sudo airmon-ng check kill

Killing these processes:
PID Name
1195 wpa_supplicant

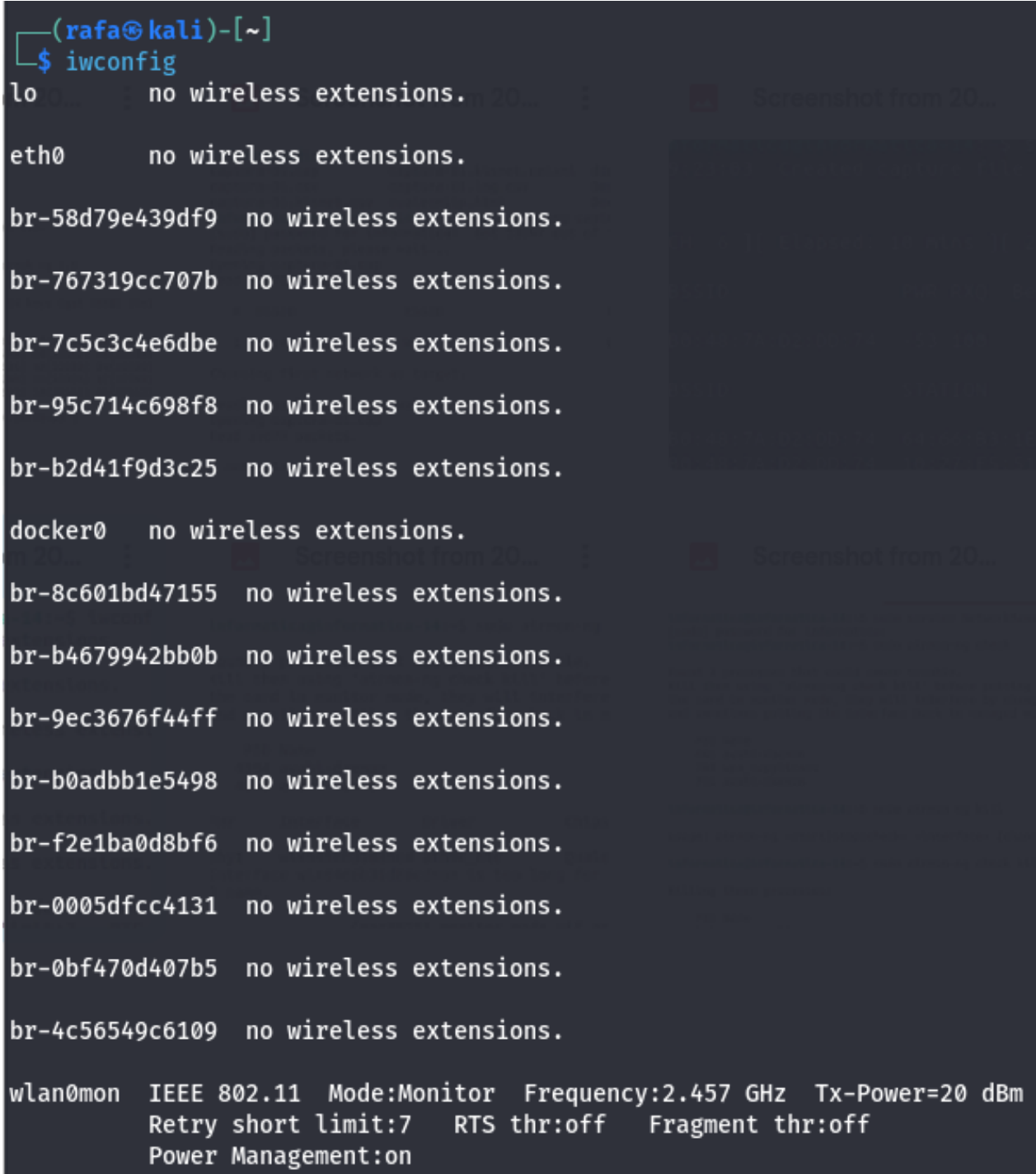
(rafa@kali)-[~]
$ sudo airmon-ng start wlan0

PHY      Interface  Driver      Chipset
phy0 wlan0      rtw_8822ce  Realtek Semiconductor Co., Ltd. RTL8822CE 802.11ac PCIe Wireless Network Adapter

(mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
(mac80211 station mode vif disabled for [phy0]wlan0)

```

Figura 1: Listado las interfaces de red disponibles para poner en modo monitor e iniciación.



```
(rafa@kali)-[~]  
$ iwconfig  
lo          no wireless extensions.  
  
eth0        no wireless extensions.  
  
br-58d79e439df9  no wireless extensions.  
br-767319cc707b  no wireless extensions.  
br-7c5c3c4e6dbe  no wireless extensions.  
br-95c714c698f8  no wireless extensions.  
br-b2d41f9d3c25  no wireless extensions.  
  
docker0     no wireless extensions.  
br-8c601bd47155  no wireless extensions.  
br-b4679942bb0b  no wireless extensions.  
br-9ec3676f44ff  no wireless extensions.  
br-b0adbb1e5498  no wireless extensions.  
br-f2e1ba0d8bf6  no wireless extensions.  
br-0005dfcc4131  no wireless extensions.  
br-0bf470d407b5  no wireless extensions.  
br-4c56549c6109  no wireless extensions.  
  
wlan0mon    IEEE 802.11  Mode:Monitor  Frequency:2.457 GHz  Tx-Power=20 dBm  
            Retry short limit:7  RTS thr:off  Fragment thr:off  
            Power Management:on
```

Figura 2: Interfaces de red.

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

El comando

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

`airodump-ng wlan0mon`

se utiliza para ver todas las redes inalámbricas disponibles. En la figura 3, se puede observar la red con SSID "WEP" tiene una velocidad de 54 Mbps, una mayor cantidad de Beacons en comparación al resto de las redes, tiene 39233 datos y es una red muy poco utilizada actualmente, siendo reemplazada por WPA2.

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	cableadaTelenati
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

Figura 3: Resultados del comando `airodump-ng wlan0mon`, mostrando todas las redes inalámbricas disponibles.

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

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

Con la red identificada, se procede a utilizar el siguiente comando para centrarse en la red WEP del informante, como se muestra en la figura 4:

```
sudo airodump-ng --channel 6 --write parte1_password0 --bssid B0:48:7A:D2:DD:74 wlan0mon
```

Luego, de forma paralela a este comando, se ejecuta el siguiente:

```
aircrack-ng parte1_password-01.cap
```

Este comando se utiliza para obtener la contraseña basada en los paquetes enviados por el informante debido a que está en WEP, lo que permite utilizar el ataque por defecto de aircrack, obteniendo los resultados mostrados en la figura.

```
CH 6 ][ Elapsed: 10 mins ][ 2024-05-17 09:33
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
B0:48:7A:D2:DD:74	-53	100	2439	68252 186	6	54e	WEP	WEP	SKA	WEP

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
B0:48:7A:D2:DD:74	64:66:B3:1E:61:B2	-51	0 - 1	0	199		
B0:48:7A:D2:DD:74	10:27:F5:51:8E:C3	-47	36e- 1e	33	167605		WEP

Figura 4: Información detallada de la red con SSID "WEP", incluyendo datos en el aire y beacons.

```
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%
```

Figura 5: Se obtiene la contraseña basada en los paquetes enviados.

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

Como se puede ver en la figura 5, la obtención de la contraseña tomó 0 segundos, utilizando 14 claves con 26601 vectores iniciales, resultando en la clave "12:34:56:78:90".

## 2.5. Descifra el contenido capturado

Para descifrar el contenido capturado se utiliza la clave obtenida anteriormente con el siguiente comando:

```
airdecap-ng -w 1234567890 parte1_password-01 -o parte1_password.cap
```

Los resultados se muestran en la figura 6.

## 2.6 Describe como obtiene la url de donde descargar el archivo DESARROLLO (PASO 2)

```
(rafa@kali)-[~/cripto/lab3_oficial]
$ airdecap-ng -w 1234567890 parte1_password-01.cap -o parte1_password.cap
Total number of stations seen      1
Total number of packets read      114985
Total number of WEP data packets  36094
Total number of WPA data packets  0
Number of plaintext data packets  0
Number of decrypted WEP packets   36094
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
```

Figura 6: Descifrando el contenido capturado desde modo monitor.

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

Con la captura descifrada, se analiza en Wireshark y se observa que los datos más comunes son ICMP, con un peso menor al de un ping común. Al examinarla, se encuentra un enlace: “<https://bit.ly/wpa2>”, como se muestra en la figura 7.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.11.1	192.168.11.16	ICMP	54	Echo (ping) repl...
2	0.000083	192.168.11.1	192.168.11.16	ICMP	54	Echo (ping) repl...
Frame 1: 54 bytes on wire (432 bits) captured (432 bits) on interface 0						
Ethernet II, Src: Tplink (08:00:00:00:00:00), Dst: Realtek (08:00:00:00:00:00)						
Internet Protocol Version 4, Src: 192.168.11.1, Dst: 192.168.11.16						
Internet Control Message Protocol, Unreachable (Destination Unreachable)						

Figura 7: Captura en Wireshark, mostrando el vínculo relevante en los datos ICMP.

## 3. Desarrollo (PASO 2)

Después de obtener el enlace, se accede a él y se procede a la descarga. Este enlace lleva a una captura de Cloudshark, como se ve en la figura 8.

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 (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	0.017089	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	0.050776	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b (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 (ee:de:67:8c:df:8b) (RA)	802.11	10	Acknowledgement, Flags=.....

<p>Frame 1: 123 bytes on wire (984 bits), 123 bytes captured (984 bits) on interface 0</p> <p>IEEE 802.11 Association Request, Flags: .....</p> <p>IEEE 802.11 Wireless Management</p>	<pre> 0000  00 00 3a 01 b0 48 7a d2 dc 18 ee de 67 8c df 8b  ....Hz....g... 0010  b0 48 7a d2 dc 18 40 8f 31 04 01 00 00 0b 56 54  .Hz...@.1....VT 0020  52 2d 31 36 34 35 32 31 33 01 00 82 84 8b 90 0c  R-1645213..... 0030  12 18 24 30 14 01 00 00 0f ac 04 01 00 00 0f ac  ..50..... 0040  04 01 00 00 0f ac 02 00 00 32 04 30 48 60 6c 3b  .....2.0H'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 01  .....P.... 0070  00 dd 08 8c fd f0 01 01 02 01 00  .....</pre>
--	---

Figura 8: Muestra el enlace de Cloudshark obtenido y la descarga del archivo rockyou.txt”.

Luego se descarga el archivo “rockyou.txt” para iniciar esta segunda parte.

### 3.1. Script para modificar diccionario original

Debido a la limitación de tiempo, se decide consultar a ChatGPT para obtener el script de manera eficiente. A continuación, se muestra el prompt y el programa en la figura 9.



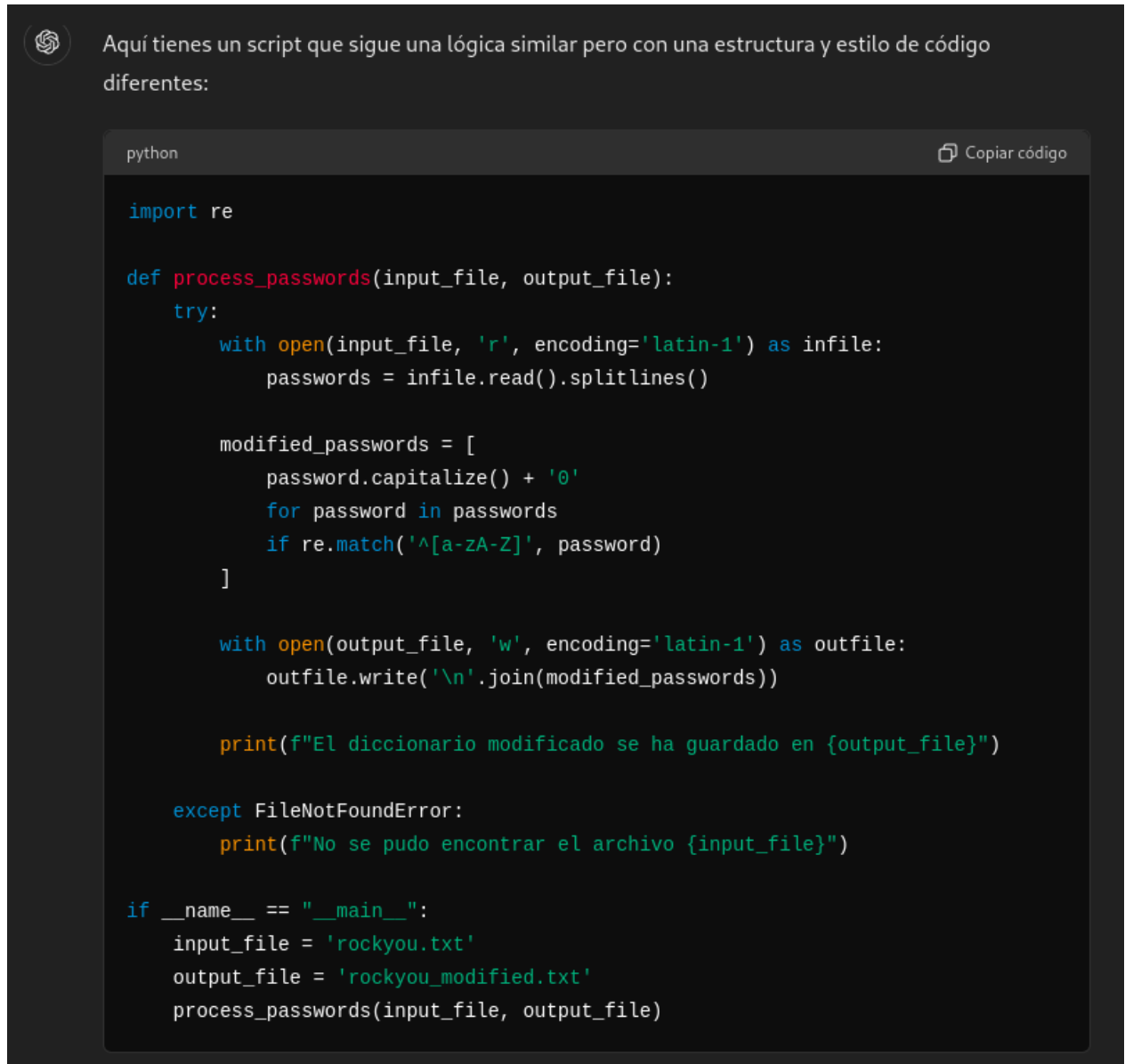


Figura 9: Prompt y script generado por ChatGPT para modificar el diccionario original.

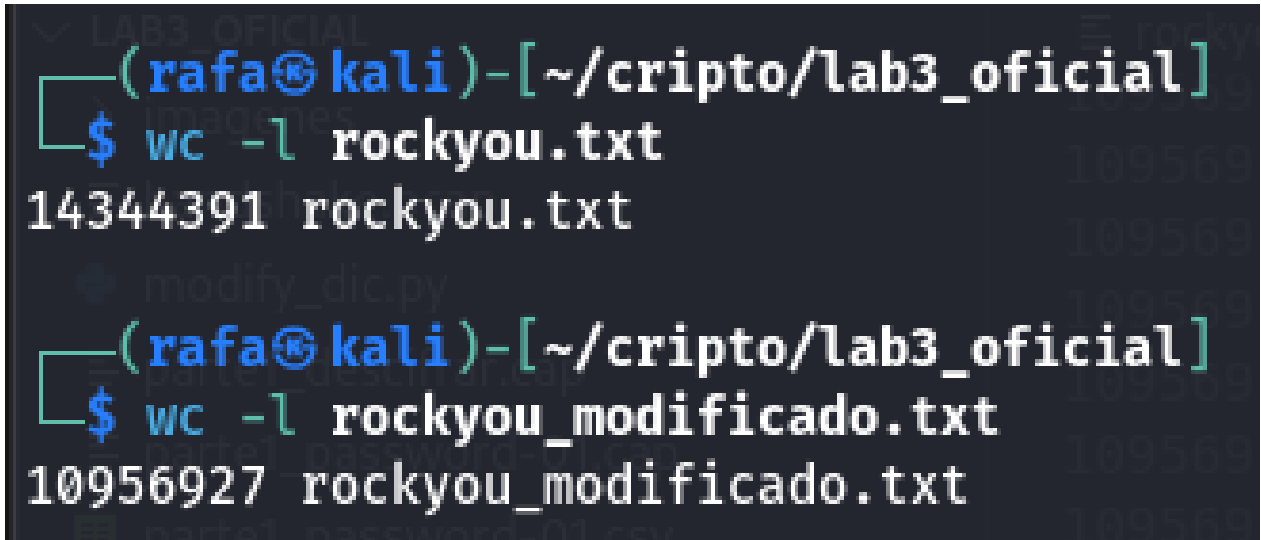
Se procede a modificar el valor del archivo de salida a `rockyou_modificado.dic`.

### 3.2. Cantidad de passwords finales que contiene `rockyou_modificado.dic`

Luego de ejecutar el programa, se procede a comparar la cantidad de contraseñas en base a las líneas de texto de cada archivo, utilizando los siguientes comandos:

```
wc -l rockyou.txt
wc -l rockyou_mod.txt
```

Los resultados se muestran en la figura, donde se puede ver que tiene menos de 3,387,464 contraseñas que el archivo original.

A terminal window with a dark background and light blue/green text. The prompt is (rafa@kali)-[~/cripto/lab3\_oficial]. The first command is \$ wc -l rockyou.txt, which outputs 14344391 rockyou.txt. The second command is \$ wc -l rockyou\_modificado.txt, which outputs 10956927 rockyou\_modificado.txt.

```
(rafa@kali)-[~/cripto/lab3_oficial]
$ wc -l rockyou.txt
14344391 rockyou.txt

(rafa@kali)-[~/cripto/lab3_oficial]
$ wc -l rockyou_modificado.txt
10956927 rockyou_modificado.txt
```

Figura 10: Comparación de la cantidad de contraseñas entre `rockyou.txt` y `rockyou_mod.txt`.

## 4. Desarrollo (Paso 3)

Para obtener la contraseña con hashcat, primero se convierte el formato de la captura a un archivo `.hc22000` en un conversor en línea. Posteriormente se ejecuta el siguiente comando , en donde se obtiene la contraseña.

```
hashcat -m 22000 handshake.hc22000 rockyou_modificado.txt
```

Los resultados de la captura se muestran en la figura 11.

```

(rafa@kali)~/cripto/lab3_oficial
$ hashcat -m 22000 handshake.hc22000 rockyou_modificado.txt
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEF, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: cpu-haswell-AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx, 5873/11810 MB (2048 MB allocatable), 8MCU

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 detected

Host memory required for this attack: 2 MB

Dictionary cache built:
* Filename..: rockyou_modificado.txt
* Passwords.: 10956928
* Bytes.....: 118581962
* Keyspace...: 10956928
* Runtime....: 1 sec

1813acb976741b446d43369fb96dbf90:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0
Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: handshake.hc22000
Time.Started....: Thu May 23 11:36:41 2024 (0 secs)
Time.Estimated...: Thu May 23 11:36:41 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_modificado.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 2197 H/s (7.43ms) @ Accel:256 Loops:64 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 3816/10956928 (0.03%)
Rejected.....: 1768/3816 (46.33%)
Restore.Point....: 0/10956928 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Password0 -> Password10
Hardware.Mon.#1...: Temp: 37c Util: 83%
Started: Thu May 23 11:35:33 2024
Stopped: Thu May 23 11:36:43 2024

```

Figura 11: Resultado de la captura transformada a .hccapx utilizando hashcat.

## 4.1. Obtiene contraseña con hashcat con potfile

Para obtener la contraseña de hashcat utilizando el potfile, se usa el siguiente comando:

```
hashcat handshake.hc22000 -m 22000 rockyou_modificado.dic --deprecated-check-disable -
```

Se utiliza la opción `--deprecated-check-disable` para evitar conflictos al ejecutar hashcat.



```
(rafa@kali)-[~/cripto/lab3_oficial]
$ hashcat -m 22000 handshake.hc22000 rockyou_modificado.dic --deprecated-check-disable --show
1813acb976741b446d43369fb96dbf90:b0487ad2dc18:eede678cdf8b:VTR-1645213:Security0
```

Figura 12: Salida de hashcat utilizando el potfile.

## 4.2. Nomenclatura del output

La nomenclatura del output se muestra en la figura 12, presentando los datos de la sesión de hashcat y la contraseña obtenida, de la siguiente manera:

macAutenticador:macEstacion:SSID:contraseña

```

(rafa@kali)~/cripto/lab3_oficial
$ hashcat -m 22000 handshake.hc22000 rockyou_modificado.dic --deprecated-check-disable --potfile-disable
hashcat (v6.2.6) starting

OpenCL API (OpenCL 3.0 PoCL 5.0+debian Linux, None+Asserts, RELOC, SPIR, LLVM 16.0.6, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: cpu-haswell-AMD Ryzen 7 3750H with Radeon Vega Mobile Gfx, 5873/11810 MB (2048 MB allocatable), 8MCU

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: 2 MB

Dictionary cache built:
* Filename..: rockyou_modificado.dic
* Passwords.: 10956928
* Bytes.....: 118581962
* Keyspace...: 10956928
* Runtime....: 1 sec

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

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: handshake.hc22000
Time.Started....: Thu May 23 11:49:28 2024 (0 secs)
Time.Estimated...: Thu May 23 11:49:28 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_modificado.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 4630 H/s (6.81ms) @ Accel:256 Loops:64 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 3816/10956928 (0.03%)
Rejected.....: 1768/3816 (46.33%)
Restore.Point....: 0/10956928 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Password0 -> Password10
Hardware.Mon.#1..: Temp: 35c Util: 25%

```

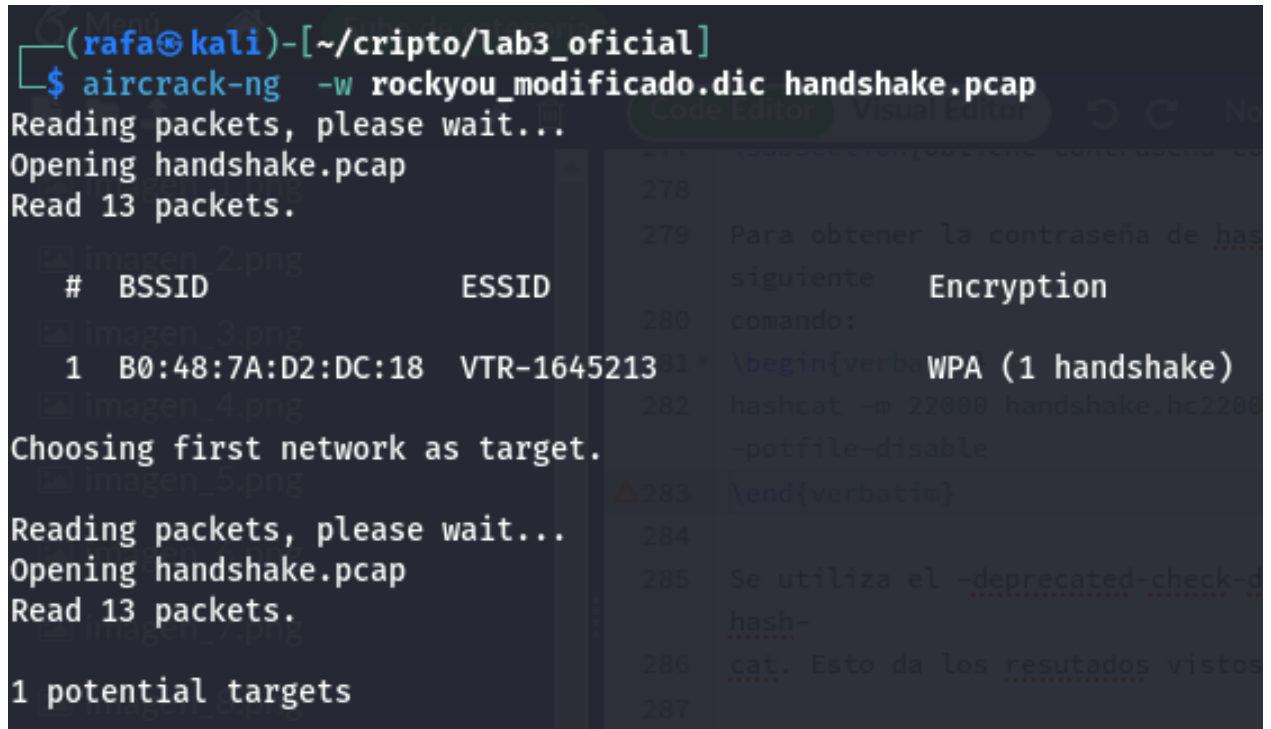
Figura 13: Nomenclatura del output de hashcat, mostrando los datos de la sesión y la contraseña obtenida.

### 4.3. Obtiene contraseña con hashcat sin potfile

Contraseña con hashcat sin usar el potfile, se utiliza el siguiente comando:

```
hashcat -m 22000 handshake.hc22000 rockyou_modificado.dic --deprecated-check-disable --
```

Se utiliza la opción `--deprecated-check-disable` para evitar conflictos al ejecutar hashcat.



```
(rafa@kali)-[~/cripto/lab3_oficial]
$ aircrack-ng -w rockyou_modificado.dic handshake.pcap
Reading packets, please wait...
Opening handshake.pcap
Read 13 packets.

# BSSID ESSID
1 B0:48:7A:D2:DC:18 VTR-164521391

Choosing first network as target.
Reading packets, please wait...
Opening handshake.pcap
Read 13 packets.

1 potential targets
```

Figura 14: Salida de hashcat sin utilizar el potfile.

#### 4.4. Nomenclatura del output

La nomenclatura del output se muestra en la figura, donde se presentan los datos de la sesión de hashcat y la contraseña obtenida, de la siguiente manera:

macAutenticador:macEstacion:SSID:contraseña

La diferencia al usar potfile son los candidatos para contraseña.



```

Aircrack-ng 1.7
[00:00:01] 2838/9302713 keys tested (4167.46 k/s)
Time left: 37 minutes, 11 seconds
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   : 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 3C D7 61 56 53 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 15: Nomenclatura del output de hashcat sin potfile, mostrando candidatos a la contraseña.

## 4.5. Obtiene contraseña con aircrack-ng

Para obtener la contraseña utilizando aircrack-ng, se utiliza el siguiente comando:

```
aircrack-ng -w rockyou_mod.dic handshake.pcap
```

```
(rafa@kali)-[~/cripto/lab3_oficial]
$ aircrack-ng -w rockyou_modificado.dic handshake.pcap
Reading packets, please wait...
Opening handshake.pcap
Read 13 packets.

# BSSID          ESSID
1 B0:48:7A:D2:DC:18 VTR-164521391

Choosing first network as target.
Reading packets, please wait...
Opening handshake.pcap
Read 13 packets.

1 potential targets
```

Figura 16: Resultados del comando `aircrack-ng -w rockyou_modificado.dic handshake.pcap`.





```

Aircrack-ng 1.7

[00:00:01] 2838/9302713 keys tested (4167.46 k/s)

Time left: 37 minutes, 11 seconds

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   : 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 3C D7 61 56 53 00
                  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

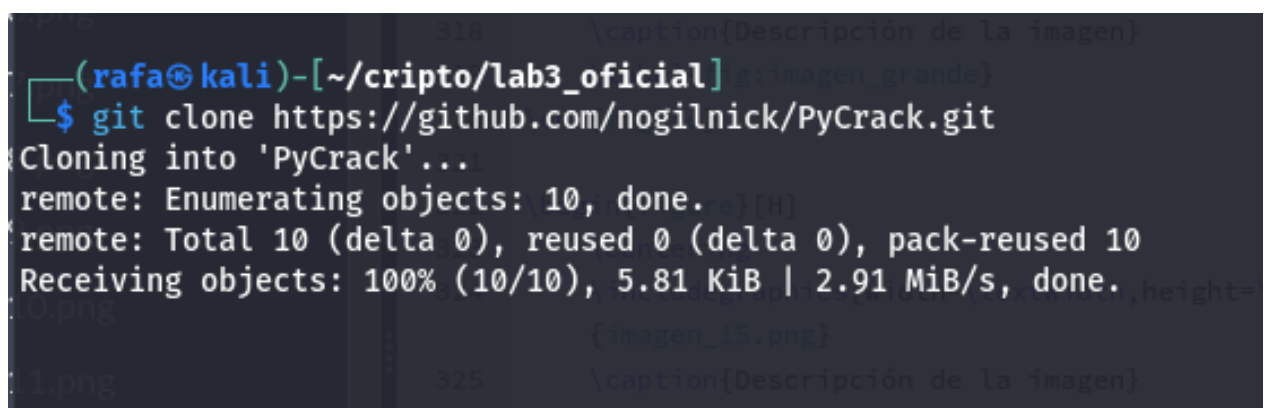
EAPOL HMAC      : 18 13 AC B9 76 74 1B 44 6D 43 36 9F B9 6D BF 90
  
```

Figura 17: Obtención de la contraseña Security0 utilizando aircrack-ng.

## 4.6. Identifica y modifica parámetros solicitados por pycrack

Se clona el repositorio de pycrack mediante el comando:

```
git clone https://github.com/nogilnick/PyCrack.git
```



```

(rafa@kali)-[~/cripto/lab3_oficial]
$ git clone https://github.com/nogilnick/PyCrack.git
Cloning into 'PyCrack'...
remote: Enumerating objects: 10, done.
remote: Total 10 (delta 0), reused 0 (delta 0), pack-reused 10
Receiving objects: 100% (10/10), 5.81 KiB | 2.91 MiB/s, done.
  
```

Figura 18: Proceso de clonación del repositorio pycrack y modificación de parámetros.

Es necesario modificar el archivo de PyCrack pywd.py, se modifica la línea 60, con el fin de hacerlo compatible con el alfabeto latino. Se elimina la línea RunTest() y los datos obtenidos de la captura, los cuales son:

- **ssid**: Nombre de la red
- **aNonce**: Nonce para establecer la PSK
- **sNonce**: Nonce que recibe la estación en el primer mensaje
- **apMac**: MAC del autenticador
- **cliMac**: MAC de la estación
- **mic1**: Primer Mensaje de Integridad
- **data1**: Campo de 802.1X Authentication reemplazando el valor de MIC por 0s del segundo frame del handshake de 4 vías
- **mic2**: Segundo Mensaje de Integridad
- **data2**: Campo de 802.1X Authentication reemplazando el valor de MIC por 0s del tercer frame del handshake de 4 vías
- **mic3**: Tercer Mensaje de Integridad
- **data3**: Campo de 802.1X Authentication reemplazando el valor de MIC por 0s del cuarto frame del handshake de 4 vías

#### **4.7. Obtiene contraseña con pycrack**

Después de modificar los datos, se procede a ejecutar el script con el siguiente comando:

```
PyCrack/python pywd.py
```

Los resultados se muestran en la figura 19.

```
(rafa@kali)-[~/cripto/lab3_oficial]
$ python 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
```

Figura 19: Ejecución del script pycrack y obtención de la contraseña Security0.

## Conclusiones y comentarios

El análisis y desarrollo del laboratorio han demostrado la vulnerabilidad inherente a las redes WEP, que permiten obtener la contraseña de forma rápida y sencilla a través de herramientas como aircrack-ng. A pesar de la mejora en la seguridad con el protocolo WPA2, también se observó que, con un diccionario adecuado y herramientas como hashcat y pycrack, es posible comprometer redes más modernas.

## Issues

1. Encontrar las líneas a modificar en el archivo `pwd.encode` en PyCrack. Se solucionó al comprobar que el código de PyCrack se encontraba dentro del archivo descargado; junto a chatGPT se logró el resultado esperado.
2. Identificar los datos en los mensajes handshake. Tras varias horas de analizar los diversos paquetes del handshake, me di cuenta que los datos necesarios se encuentran esparcidos en varios de los distintos paquetes, y no solo en uno.
3. Problemas de compatibilidad con versiones de las herramientas utilizadas. Al momento de pasar el archivo `handshake.pcap` a `.hc22000` se logró el resultado esperado, pero esto después de investigar arduamente.

4. Errores en la configuración del modo monitor de la tarjeta de red. Debido a que era mi primera vez utilizando estos servicios, apagué mi propia tarjeta de red varias veces sin darme cuenta, sin embargo, el familiarizarme con las herramientas permitió un mejor uso de estas, produciendo los resultados obtuvidos.