

Informe Laboratorio 3

Sección 3

Pablo Castro

e-mail: pablo.castrod@mail.udp.cl

22 Octubre de 2023

Índice

1. Descripción de actividades	2
2. Desarrollo (PASO 1)	2
2.1. identificar en qué se destaca la red del informante del resto	2
2.2. explica...5000 paquetes para obtener la pass	4
2.3. obtiene la password con ataque por defecto de aircrack-ng	5
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	6
3. Desarrollo (PASO 2)	7
3.1. indica script para modificar diccionario original	7
3.2. cantidad de passwords finales que contiene rockyou_mod.dic	8
4. Desarrollo (Paso 3)	9
4.1. obtiene contraseña con hashcat con potfile	9
4.2. identifica nomenclatura del output	10
4.3. obtiene contraseña con hashcat sin potfile	11
4.4. identifica nomenclatura del output	11
4.5. obtiene contraseña con aircrack-ng	11
4.6. identifica y modifica parámetros solicitados por pycrack	12
4.7. obtiene contraseña con pycrack	14

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, es primordial saber cómo los routers avisan a los dispositivos cercanos de su existencia. Esto lo hacen a través de paquetes beacon. Un paquete "Beacon" es un tipo de paquete que transmitido periódicamente por los puntos de acceso para anunciar la presencia de la red inalámbrica. Estos contienen información de la red, incluyendo: SSID, Dirección MAC, Tipo de Seguridad (WEP,WPA,WPA2,etc), entre otros. Para saber cuál es la red del informante, se debe hacer un escaneo con la interfaz en modo monitor. Este modo permitirá analizar todos los paquetes que se encuentren en el aire. Incluidos los beacons que debe emitir la red del informante.

Para escanear la red se utiliza el comando: `sudo tcpdump -i en0 -w /ruta/captura.pcap` Esto colocará mi interfaz de red `.en0` en modo monitor y comenzará a analizar paquetes. Finalmente se guardarán en `/rutaqueyoquiera/captura.pcap`. Para luego poder analizar con wireshark.

2.1 identificar en qué se destaca la red del informante del resto DESARROLLO (PASO 1)

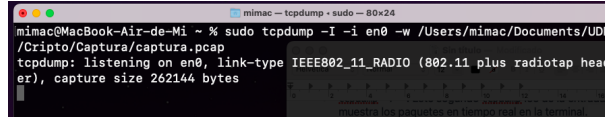


Figura 1: Inicio de Captura de Paquetes en modo monitor

Luego en wireshark, se utiliza el filtro: **wlan.fc.type_subtype eq 8** que permite mostrar los paquetes beacon que hay en la captura.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3785, Wm=, Flags=p....
2	0.000048	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3786, Wm=, Flags=p....
3	0.001177	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3785, Wm=, Flags=p....
4	0.003775	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3787, Wm=, Flags=p....
5	0.003791	Tp-LINK_42:00:74	Broadcast	802.11	95	Beacon frame, Seq=3802, Wm=, Flags=....., B1=00, SSID="WEP"
6	0.003838	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3788, Wm=, Flags=p....
7	0.003842	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3789, Wm=, Flags=p....
8	0.003853	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3790, Wm=, Flags=p....
9	0.003865	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3791, Wm=, Flags=p....
10	0.003905	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3792, Wm=, Flags=p....
11	0.003907	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3793, Wm=, Flags=p....
12	0.003934	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3794, Wm=, Flags=p....
13	0.003967	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3795, Wm=, Flags=p....
14	0.003979	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3796, Wm=, Flags=p....
15	0.003991	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3797, Wm=, Flags=p....
16	0.004003	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3798, Wm=, Flags=p....
17	0.004014	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3799, Wm=, Flags=p....
18	0.004026	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3800, Wm=, Flags=p....
19	0.004038	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3801, Wm=, Flags=p....
20	0.004050	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3802, Wm=, Flags=p....
21	0.004061	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3803, Wm=, Flags=p....
22	0.004073	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3804, Wm=, Flags=p....
23	0.004085	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3805, Wm=, Flags=p....
24	0.004113	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3806, Wm=, Flags=p....
25	0.004171	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3807, Wm=, Flags=p....
26	0.004185	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3808, Wm=, Flags=p....
27	0.004195	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3809, Wm=, Flags=p....
28	0.004225	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3810, Wm=, Flags=p....
29	0.004241	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3811, Wm=, Flags=p....
30	0.004252	Shanghai_480:00:09	IPNetwork_70	802.11	115	QoS Data, Seq=3812, Wm=, Flags=p....
31	0.004265	Shanghai_480:00:09	IPNetwork_70	802.11	115	QoS Data, Seq=3813, Wm=, Flags=p....
32	0.004296	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3814, Wm=, Flags=p....
33	0.004308	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3815, Wm=, Flags=p....
34	0.004325	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3816, Wm=, Flags=p....
35	0.004334	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3817, Wm=, Flags=p....
36	0.004346	Tp-LINK_42:00:74	Broadcast	802.11	95	Beacon frame, Seq=3820, Wm=, Flags=....., B1=00, SSID="WEP"
37	0.004462	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3820, Wm=, Flags=p....
38	0.004451	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3821, Wm=, Flags=p....
39	0.004540	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3822, Wm=, Flags=p....
40	0.004576	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3823, Wm=, Flags=p....
41	0.004608	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3824, Wm=, Flags=p....
42	0.004632	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3825, Wm=, Flags=p....
43	0.004651	Shanghai_480:00:09	Tp-LINK_42:00:74	802.11	22	QoS Data, Seq=3826, Wm=, Flags=p....

Figura 2: Paquetes capturados en wireshark

No.	Time	Source	Destination	Protocol	Length	Info
36	0.004346	Tp-LINK_42:00:74	Broadcast	802.11	95	Beacon frame, Seq=3820, Wm=, Flags=....., B1=00, SSID="WEP"

Figura 3: Filtro Activado

Con esto se logra identificar un único paquete con seguridad WEP, este es el beacon de la red del informante.

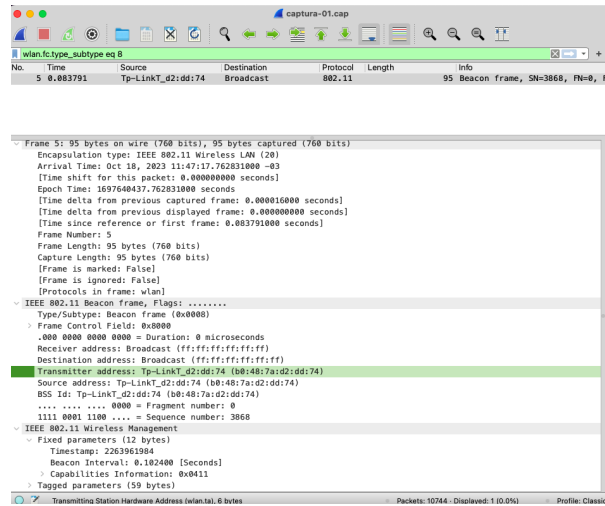


Figura 4: Datos del Paquete

2.2. explica...5000 paquetes para obtener la pass

Esto tiene que ver con el "ataque de cumpleaños", que es un clásico en la probabilidad. Esto dice que, dentro de un tamaño relativamente pequeño de personas, existen al menos 2 que comparten fecha de cumpleaños. Sólo se necesitan 23 personas en una habitación para que exista una probabilidad de que un 50 % comparta fecha. Al intentar hacer un ataque de fuerza bruta para descifrar una contraseña de WPA/WPA2, el "cumpleaños" sería equivalente a una colisión y la "persona" sería equivalente a un intento de adivinar la contraseña. La idea es que, después de capturar un cierto número de paquetes, existan 2 paquetes que compartan el mismo hash, formando así la coincidencia de las fechas de cumpleaños o las colisiones.

$$P \approx 1 - e^{-\frac{k^2}{2d+1}}$$

Figura 5: Fórmula del Ataque de Cumpleaños

Donde P es la probabilidad de encontrar una colisión, k es el número de intentos o elementos generados, d es el número de bits en la salida de la función hash. En este caso para WEP, de 128 bits, se asume que los primeros 24 bits son para el vector de inicialización (IV) y los 104 bits restantes son para la clave. Después de aproximadamente 1000 intentos (k=1000, d = 104), existe una probabilidad de colisión de un 97 %

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

La captura descifrada tiene varios paquetes ARP, DNS e ICMP. Se realizará un análisis por cada uno.

1. Mensajes ARP: Los mensajes ARP son desde el router (el router al cual se obtuvo la contraseña), solo a la dirección IP 192.168.11.37, puede ser un indicio de que ese es el equipo que usa el informante.

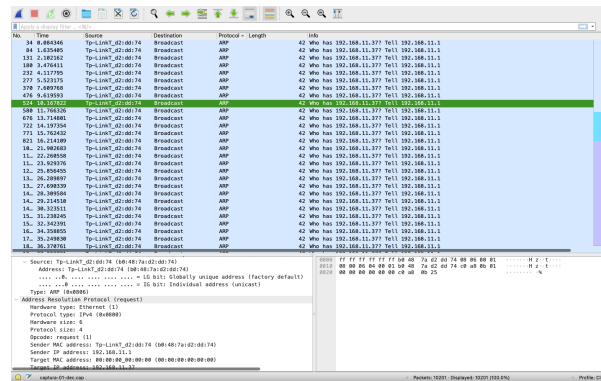


Figura 8: Mensajes ARP

2. Consultas DNS: La IP 192.168.11.37 (posible equipo del informante), realizó varias consultas a "debian.pool.ntp.org". Este dominio se asocia comúnmente con la sincronización de tiempo en sistemas Debian. No hay información relevante.

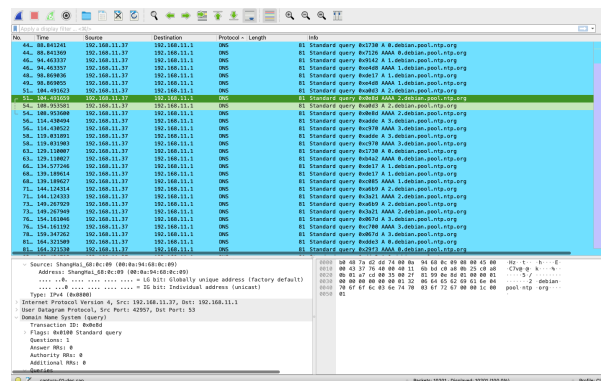


Figura 9: Mensajes DNS

3. Mensajes ICMP: Los mensajes ICMP capturados revelaron una actividad tipo ping request desde la IP 192.168.11.37 hacia el router (192.168.11.1). Todos los mensajes ICMP compartían el mismo ID, lo que indica que fueron parte de una misma secuencia de comando de ping. Además, en la carga útil de estos mensajes ICMP se incluye:

”bit.ly/wpa2_”. Lo cual lleva al enlace <https://www.cloudshark.org/captures/b5b39e1c51eb>. Que corresponde a una captura de red.

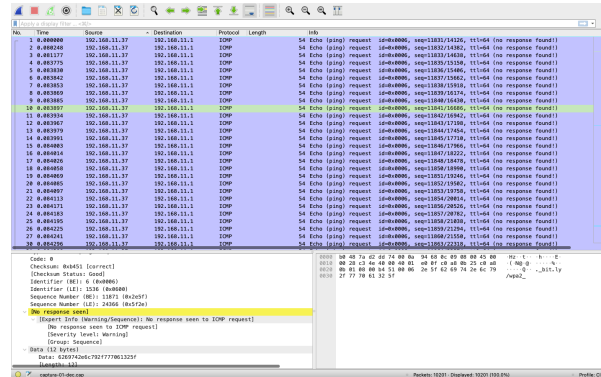


Figura 10: Mensajes ICMP

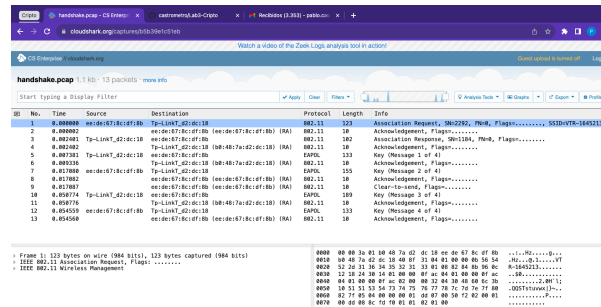


Figura 11: Página Obtenida

3. Desarrollo (PASO 2)

3.1. indica script para modificar diccionario original

El script de python que se usó es el siguiente:

```
def process_text_file(input_file_path, output_file_path="rockyou_mod.dic"):
    # Leer el archivo
    with open(input_file_path, 'r', encoding='ISO-8859-1') as file:
        content = file.read()
    # Dividir el contenido en strings
    strings = content.split()

    # Contar la cantidad total de strings antes de realizar cambios
```

3.2 cantidad de passwords finales que contiene rockyou_mod.dicDESARROLLO (PASO 2)

```
total_strings_before = len(strings)

# Filtrar los strings, eliminando aquellos que comiencen con un n mero
processed_strings = []
for string in strings:
    # Si el string comienza con un n mero, se ignora
    if string[0].isdigit():
        continue
    # Si no, se modifica y se agrega a la lista de strings procesados
    modified_string = string.capitalize() + '0'
    processed_strings.append(modified_string)

# Contar la cantidad de strings que fueron eliminados y la cantidad que
strings_deleted = total_strings_before - len(processed_strings)
strings_remaining = len(processed_strings)

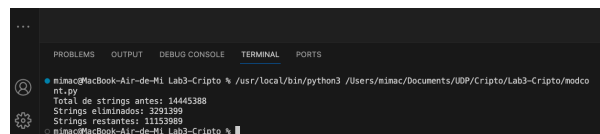
# Escribir los strings procesados en el nuevo archivo
with open(output_file_path, 'w') as file:
    file.write('\n'.join(processed_strings))

return total_strings_before, strings_deleted, strings_remaining

input_file_path = '/Users/mimac/Documents/UDP/Cripto/Lab3-Cripto/rockyou.txt'
total_before, deleted, remaining = process_text_file(input_file_path)
print(f'Total de strings antes: {total_before}')
print(f'Strings eliminados: {deleted}')
print(f'Strings restantes: {remaining}')
```

3.2. cantidad de passwords finales que contiene rockyou_mod.dic

Se obtuvieron los siguientes resultados:



```
mimac@MacBook-Air-de-MI Lab3-Cripto % /usr/local/bin/python3 /Users/mimac/Documents/UDP/Cripto/Lab3-Cripto/modco
nt.py
Total de strings antes: 14443388
Strings eliminados: 321399
Strings restantes: 11133989
mimac@MacBook-Air-de-MI Lab3-Cripto %
```

Figura 12: Resultados del Script

Que corresponde a 11.153.989 contraseñas. Estas se almacenan en un archivo ".dic".

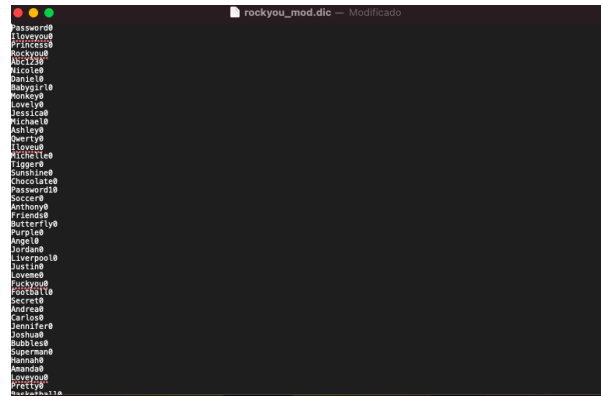


Figura 13: Contraseñas almacenadas

4. Desarrollo (Paso 3)

4.1. obtiene contraseña con hashcat con potfile

Para utilizar la captura en hashcat, es necesario primero transformar la captura a formato. Para eso se utiliza el convertidor web <https://hashcat.net/cap2hashcat/>. Con el archivo resultante quedando como "handshake.hc22000".

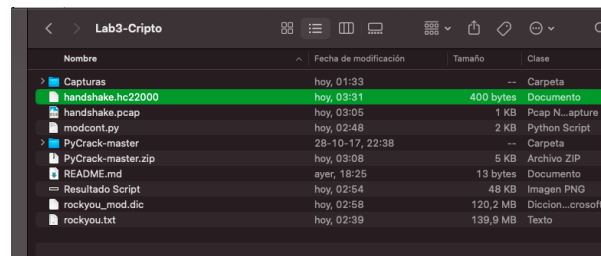


Figura 14: Captura en formato hc22000

El comando para la captura es **sudo hashcat -a 0 -m 22000 handshake.hc22000 rockyoumod.dic** Esto da la contraseña sin el potfile.

```

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPO)
Hash.Target.....: handshake.hc22000
Time.Started.....: Sun Oct 22 03:42:44 2023 (0 secs)
Time.Estimated.....: Sun Oct 22 03:42:44 2023 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#2.....: 2410 H/s (6.39ms) @ Accel:16 Loops:1024 Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%) Digests (new)
Progress.....: 2912/11153989 (0.03%)
Rejected.....: 1376/2912 (47.25%)
Restore.Point.....: 2806/11153989 (0.03%)
Restore.Sub.#2...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#2....: Iamthebest0 -> Anthony0
Hardware.Mon.SMC.: Fan0: 27%
Hardware.Mon.#2...: Temp: 41c

Started: Sun Oct 22 03:42:30 2023
Stopped: Sun Oct 22 03:42:45 2023
mimac@MacBook-Air-de-Mi Lab3-Cripto %

```

Figura 15: Resultado de HashCat

Ahora para ver el potfile, se debe utilizar el comando `cat ~/.hashcat/hashcat.potfile`
 Nota: a mi me pasó que no tenía hashcat almacenado en el lugar que debería estar, así que para que me funcionara este comando hice lo siguiente. En primer lugar, busqué donde estaba con el siguiente comando: `sudo find / -type f -name hashcat.potfile 2>/dev/null`. Luego, rearmé el comando original, quedando así: `cat /usr/local/Cellar/hashcat/6.2.6_1/share/hashcat/hashcat.potfile`

```

mimac@MacBook-Air-de-Mi Lab3-Cripto % cat ~/.hashcat/hashcat.potfile
cat: /Users/mimac/.hashcat/hashcat.potfile: No such file or directory
mimac@MacBook-Air-de-Mi Lab3-Cripto % sudo find / -type f -name hashcat.potfile
2>/dev/null
Password:
/usr/local/Cellar/hashcat/6.2.6_1/share/hashcat/hashcat.potfile
*Z*Z
zsh: suspended sudo find / -type f -name hashcat.potfile 2> /dev/null
mimac@MacBook-Air-de-Mi Lab3-Cripto % sudo cat /usr/local/Cellar/hashcat/6.2.6_1/share/hashcat/hashcat.potfile
55e1a0f08ed75380f627c6dc48207454b754983771ffc8031d89c5198d6fac76*5654522d3136343
5323133:Security0
mimac@MacBook-Air-de-Mi Lab3-Cripto %

```

Figura 16: Potfile

La contraseña encontrada es "Security0"

4.2. identifica nomenclatura del output

Nomenclatura del output:

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

1. **1813acb976741b446d43369fb96dbf90**: Este es el PMKID, una clave derivada de la contraseña de la red.

2. **b0487ad2dc18**: La dirección MAC del cliente.
3. **eede678cdf8b**: La dirección MAC del punto de acceso (router).
4. **VTR-1645213**: El SSID de la red, es decir, el nombre de la red WiFi.
5. **Security0**: La contraseña de la red WiFi que ha sido crackeada.

4.3. obtiene contraseña con hashcat sin potfile

Para obtener la contraseña sin el potfile, se puede ver analizando el output al realizar el comando `sudo hashcat -a 0 -m 22000 handshake.hc22000 rockyoumod.dic --show`

c

4.4. identifica nomenclatura del output

Son los mismos parámetros que al usar Potfile.

4.5. obtiene contraseña con aircrack-ng

Para realizar el ataque de fuerza bruta con aircrack hay que hacer lo mismo que con el ejercicio uno, la misma direccion mac y la nueva captura handshake.pcap. **aircrack-ng -b b0:48:7a:d2:dc:18 -w rockyoumod.dic handshake.pcap**

```

Aircrack-ng 1.7

[00:00:01] 2926/9351662 keys tested (4370.23 k/s)

Time left: 35 minutes, 39 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 : 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
              00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

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

mimac@MacBook-Air-de-Mi Lab3-Cripto %

```

Figura 18: Obtencion de contraseña con Aircrack

Se obtuvo la misma contraseña "Security0" con Aircrack

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

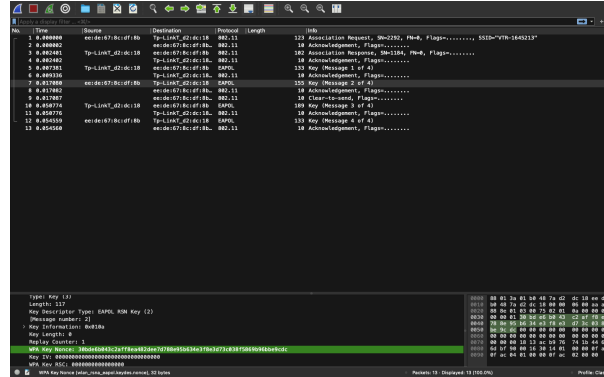


Figura 21: sNonce

ApMac se obtuvo con el Potfile, corresponde a ".e6de678cdf8b". Lo mismo con cliMac: "b0487ad2dc18".

Mic "n" se encuentra en el mensaje "n+1" del handshake. Por ejemplo, el Mic 1 se encuentra en el mensaje 2 del handshake.

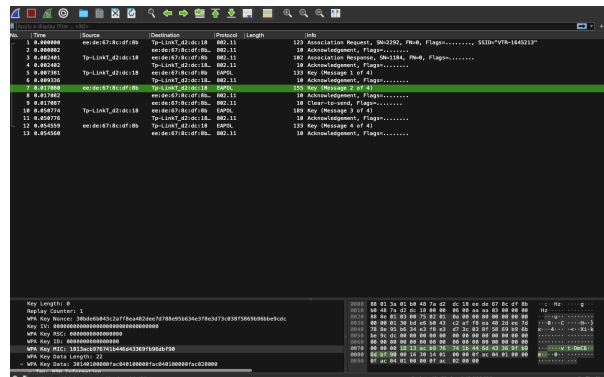


Figura 22: mic1

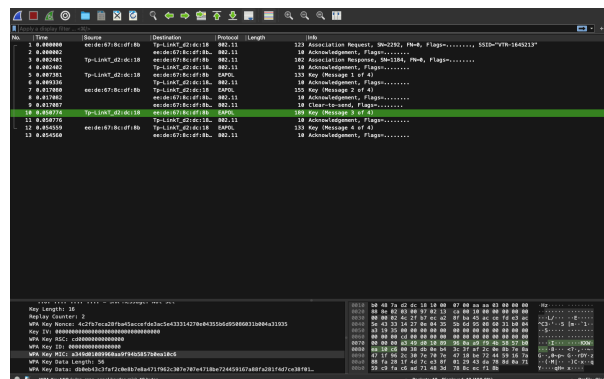


Figura 23: mic2

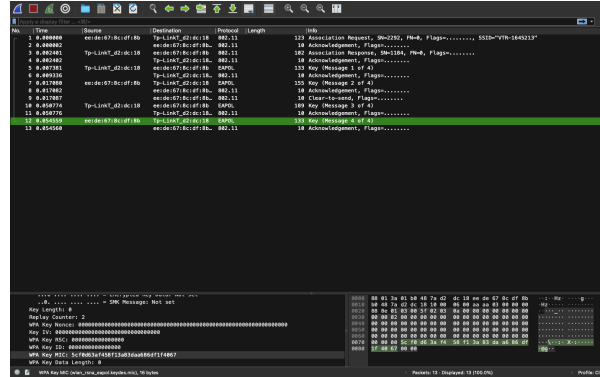


Figura 24: mic3

Lo mismo ocurre con los Data "n", sin embargo, estos son el mensaje completo en hex, y el lugar donde están los mic, se reemplazan con 0. Así quedaría todo finalmente:

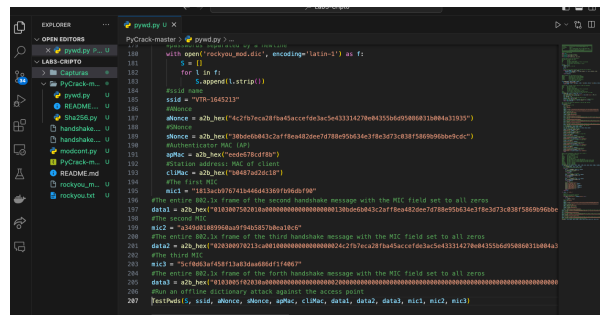


Figura 25: Pycrack con los datos correctos.

4.7. obtiene contraseña con pycrack

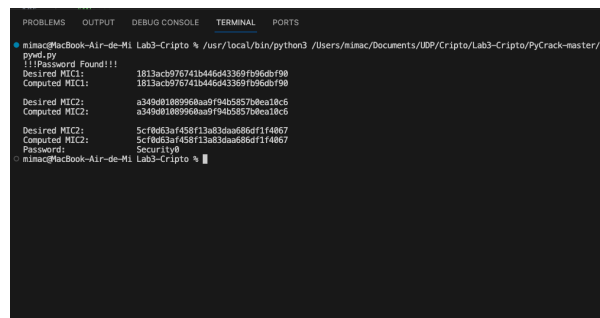


Figura 26: Obtención de contraseña con Pycrack

Conclusiones y comentarios

Con esta experiencia pude entender la importancia de la seguridad de redes Wifi, entendí por qué las redes WEP son las más inseguras y aprendí también a utilizar varias herramientas para poder obtener una contraseña. La verdad es que estas experiencias de laboratorio nunca dejan de sorprender.