

Informe Laboratorio 3

Sección 01

Alumno Felipe Andrés Mora Meneses
e-mail: felipe.mora_m@mail.udp.cl

Mayo de 2024

Índice

1. Descripción de actividades	2
2. Desarrollo (PASO 1)	3
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	4
2.4. Indica el tiempo que demoró en obtener la password	5
2.5. Descifra el contenido capturado	6
2.6. Describe como obtiene la url de donde descargar el archivo	8
3. Desarrollo (PASO 2)	8
3.1. Script para modificar diccionario original	9
3.2. Descripción del Comando	9
3.3. Resultado	9
3.4. Cantidad de passwords finales que contiene rockyou_mod.dic	10
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	14
4.7. Obtiene contraseña con pycrack	15

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)

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

```
wlx6466b31d7c78: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 64:66:b3:1d:7c:78 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

informatica@informatica-11:~$
```

Figura 1: interfaz de red wi-fi mediante el comando 'ifconfig'

```
informatica@informatica-11:~$ sudo airodump-ng wlx6466b31d7
[sudo] password for informatica:
iSorry, try again.
CH 7 ][ Elapsed: 2 mins ][ 2024-05-22 10:25
```

BSSID	PWR	Beacons	#Data, #/s	CH	MB	ENC CIPHER	AUTH	ESSID
48:D3:43:BE:A7:19	-90	2	0 0 1	130	WPA2 CCMP	PSK	VTR-7335606	
E6:AB:89:1C:85:38	-1	0	0 0 6	-1			<length: 0>	
B0:48:7A:D2:DD:74	-48	227	8792 0 9	54e	WEP WEP		WEP	
7A:C0:1A:74:67:0D	-54	149	14 0 6	180	WPA2 CCMP	PSK	crisstopher	
98:FC:11:86:B6:B9	-62	102	0 0 11	130	WPA2 CCMP	PSK	Telematica	
B0:1F:8C:E2:14:A0	-72	134	0 0 1	130	WPA3 CCMP	SAE	Sala Hibrida-UDP	
18:35:D1:90:C7:99	-69	8	0 0 11	130	WPA2 CCMP	PSK	VTR-6733269	
B0:1F:8C:E2:14:A6	-68	114	0 0 1	130	WPA3 CCMP	OWE	<length: 0>	
B0:1F:8C:E2:14:A7	-69	106	0 0 1	130	WPA2 CCMP	MGT	Administrativos-UDP	
E4:AB:89:04:63:6C	-90	3	0 0 11	130	WPA2 CCMP	PSK	BOYGENIUS	
B0:1F:8C:E2:14:A2	-69	126	0 0 1	130	WPA3 CCMP	OWE	<length: 0>	
B0:1F:8C:E2:14:A1	-67	128	0 0 1	130	OPN		Invitados-UDP	

Figura 2: se utiliza 'airodump-ng' para capturar el trafico de la interfaz de red wlx6466b31d7c78

La red del informante se destacaba en cuatro aspectos:

1. **Datos en la red:** En la red del informante habían más datos circulando que en el resto.
2. **Protocolo de encriptación:** El protocolo de encriptación usado en la red fue WEP.
3. **Canal de transmisión:** El canal de transmisión de la red fue 9.
4. **Conexión de "station con AP:** El ratio de transmisión/recepción entre un station y el AP era mayor en comparación a otras redes; además, el tráfico solo se generaba de un station al AP (y no de varios stations).

El primer aspecto mencionado fue el que destacó a la hora de identificar al informante para poder hacer la captura que se aprecia en la Figura 2.

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

Esto se explica por la probabilidad de colisión, la fórmula para la probabilidad de que al menos dos IVs colisionen (es decir, sean iguales) es:

$$P = 1 - \left(1 - \frac{1}{H}\right)^{\frac{N(N-1)}{2}}$$

donde:

- P es la probabilidad de colisión.
- H es el número total de IVs posibles.
- N es el número de paquetes capturados.

Para WEP, el espacio muestral H es 2^{24} , ya que los IVs tienen 24 bits.

Reemplazando $H = 2^{24}$ y $N = 5000$ en la fórmula:

$$H = 2^{24} = 16,777,216$$

$$N = 5000$$

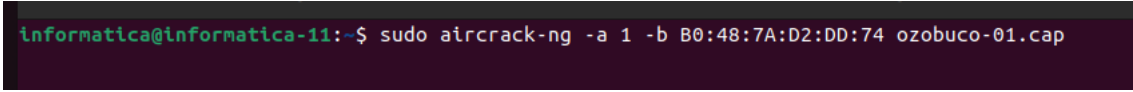
Implica qué:

$$P = 0,525 = 52,5 \%$$

Interpretación

Esto significa que con 5000 paquetes, la probabilidad de que al menos dos IVs colisionen es aproximadamente 52.5%. Este nivel de colisión es significativo porque en la criptografía WEP, la reutilización de IVs (debido a colisiones) es una debilidad que puede ser explotada para inferir la clave.

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



```
informatica@informatica-11:~$ sudo aircrack-ng -a 1 -b B0:48:7A:D2:DD:74 ozobuco-01.cap
```

Figura 3: Utilizando el BSSID de la red del informante, se ataca la red mediante aircrack-ng, El trafico de red capturado se encuentra en un archivo llamado 'ozobuco-01.cap'

2.4 Indica el tiempo que demoró en obtener la password 2 DESARROLLO (PASO 1)

```
CH 6 ][ Elapsed: 1 hour 8 mins ][ 2024-05-22 12:05 ][ fixed channel wlx6466b31d7c78: 3
```

BSSID	PWR	RXQ	Beacons	#Data, #/s	CH	MB	ENC	CIPHER	AUTH	ESSID
B0:48:7A:D2:DD:74	-82	0	4368	21624	2	9	54e	WEP	WEP	WEP

BSSID	STATION	PWR	Rate	Lost	Frames	Notes	Probes
B0:48:7A:D2:DD:74	10:27:F5:51:87:77	-37	1e- 1e	55	78138		WEP

Figura 4: Para poder obtener la key, fue necesario obtener 70.130 frames en un tiempo de 1 hora y 8 minutos de captura del trafico de red.

```
Aircrack-ng 1.6
[00:00:03] Tested 98431 keys (got 21494 IVs)
KB depth byte(vote)
0 1/ 8 12(28416) D5(28160) 46(27648) A0(26624) 0F(26368) 1D(26368) 97(26368) 90(26112) 30(25856) 08(25600) 40(25600) 42(25600)
1 1/ 8 34(27904) 09(27392) 41(27392) 19(27136) EE(26880) BA(26368) 34(26112) 11(25856) 97(25856) B2(25856) 21(25088) 35(25088)
2 5/ 23 56(26368) 6F(26112) 9F(26112) 0F(26112) 8D(25600) D2(25600) E2(25600) 5A(25344) 76(25344) 8A(25344) C3(25344) EA(25344)
3 5/ 6 78(26624) 4D(26368) 51(26368) 85(26368) 05(26112) 10(26112) 42(25856) A0(25856) 57(25600) 28(25344) 72(25344) D3(25344)
4 9/ 12 42(25600) 44(25344) A4(25344) 86(25088) 92(25088) 51(24832) 5C(24832) 8D(24832) B7(24832) BA(24832) FA(24832) C6(24832)

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

Informatica@Informatica-11:~$
```

Figura 5: Se obtuvo la llave 12:34:56:78:90 mediante el ataque de aircrack-ng

2.4. Indica el tiempo que demoró en obtener la password

```
felipe@IdeaPad-L340-15IRH:~$ time aircrack-ng /home/felipe/Descargas/
"FM.lab cripto3"/FM/ozobuco-01.cap
Reading packets, please wait...
```

Figura 6: Anteponiendo 'time' al comando utilizado, se obtienen los tiempos

```

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

real    0m6,478s
user    0m9,284s
sys     0m3,550s
felipe@IdeaPad-L340-15IRH:~$ █

```

Figura 7: En tiempo real se tardó 6,5 segundos, a nivel usuario tardó 9,3 segundos y a nivel de sistema tardó 3,5 segundos.

2.5. Descifra el contenido capturado

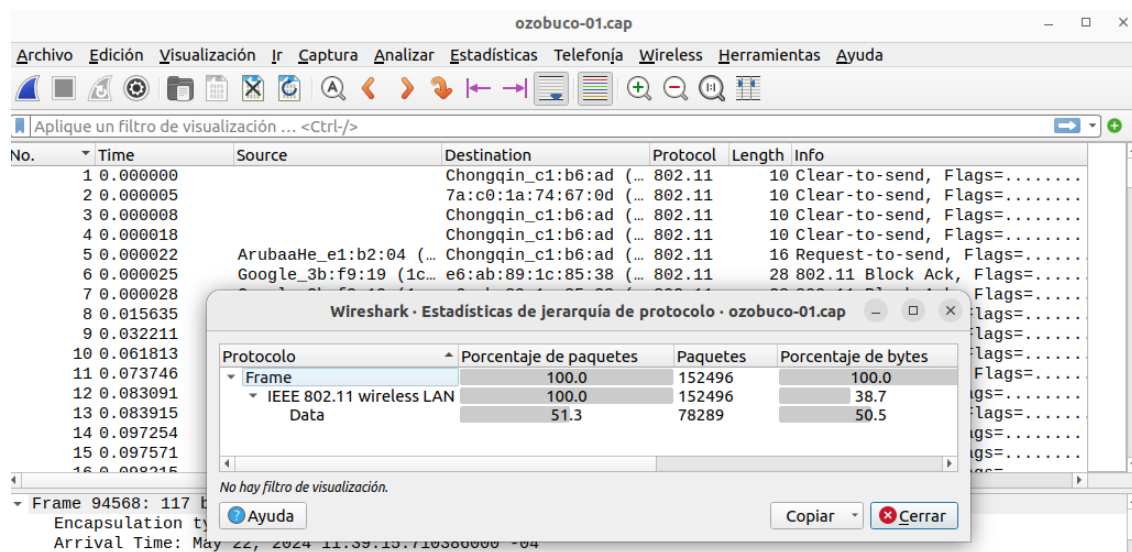


Figura 8: distribución de protocolos en la captura pre-desencriptación

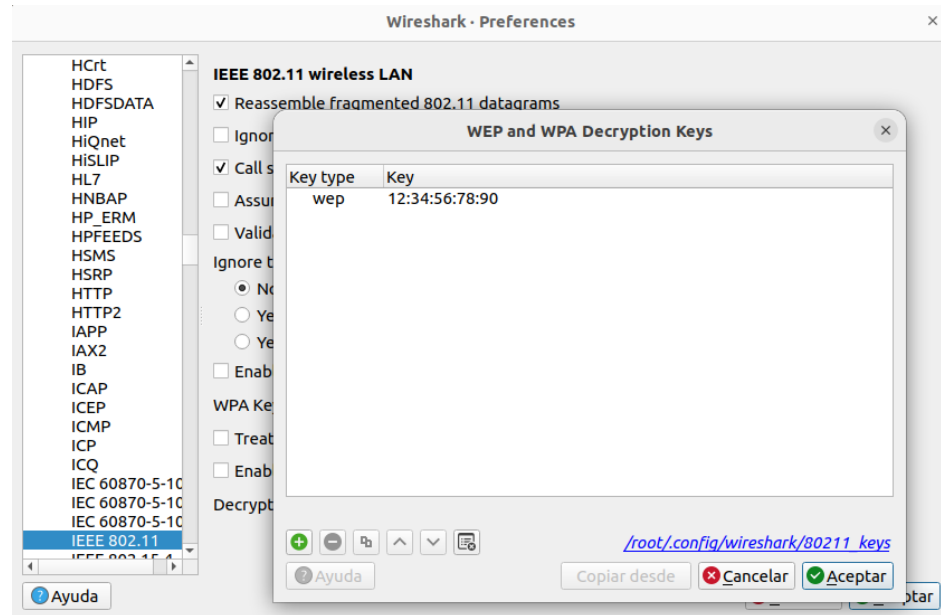


Figura 9: Mediante las preferencias de wireshark se agrega la key para descryptar el protocolo WEP con la key que se obtuvo del ataque por aircrack.

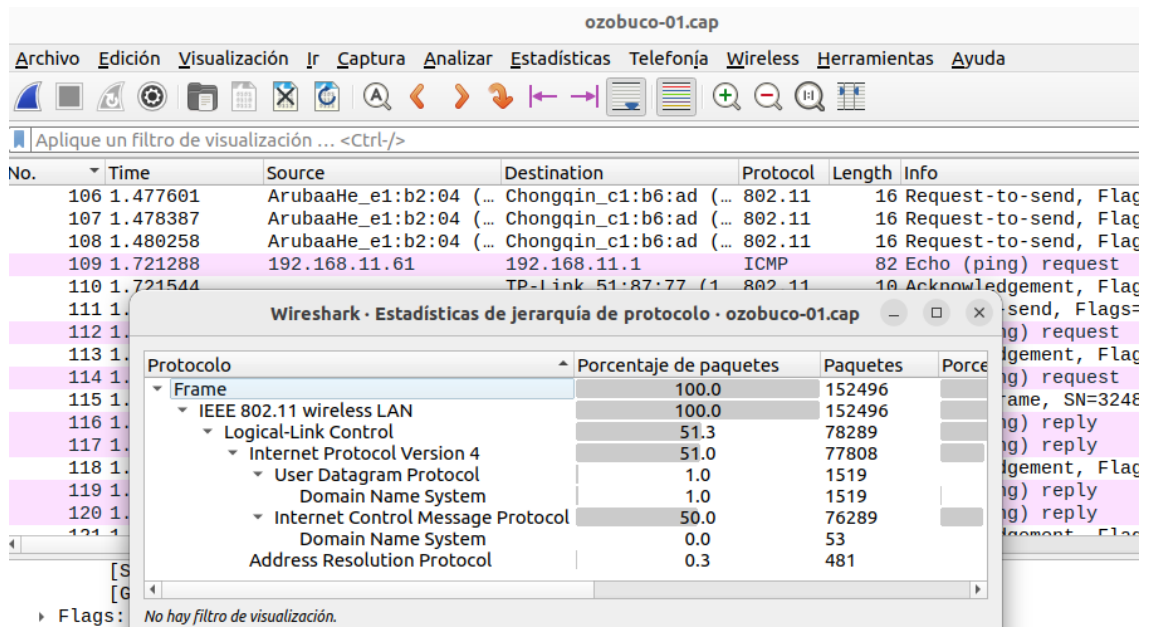
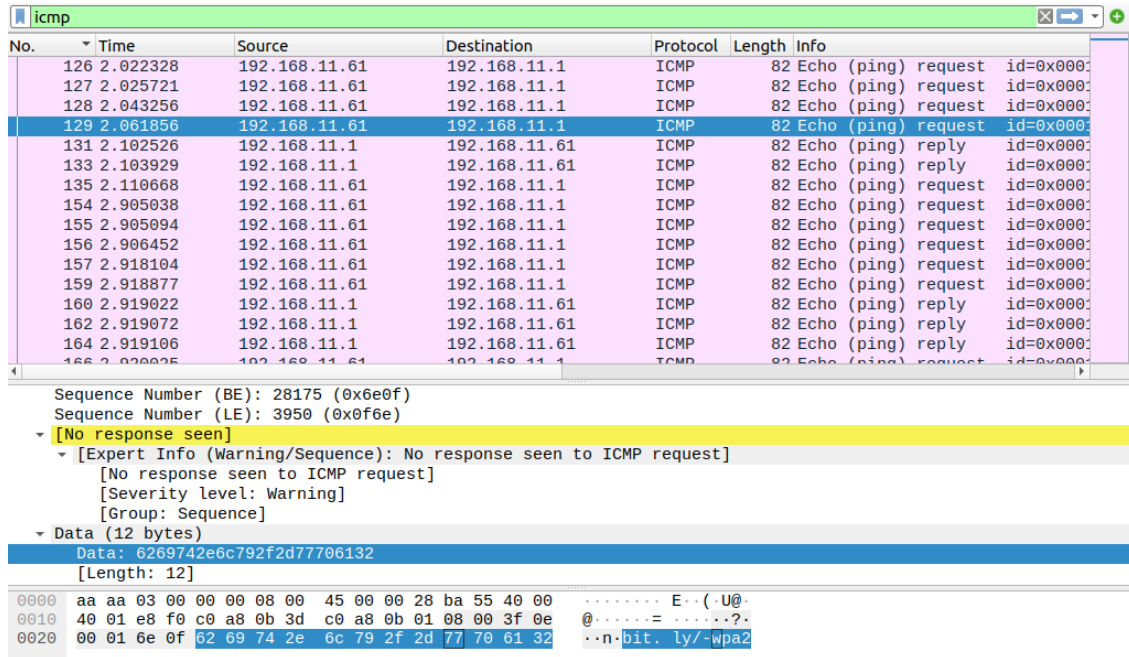


Figura 10: Distribución de protocolos de la captura de tráfico post-descryptación con la key obtenida.

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

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



No.	Time	Source	Destination	Protocol	Length	Info
126	2.022328	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
127	2.025721	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
128	2.043256	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
129	2.061856	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
131	2.102526	192.168.11.1	192.168.11.61	ICMP	82	Echo (ping) reply id=0x000:
133	2.103929	192.168.11.1	192.168.11.61	ICMP	82	Echo (ping) reply id=0x000:
135	2.110668	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
154	2.905038	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
155	2.905094	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
156	2.906452	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
157	2.918104	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
159	2.918877	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:
160	2.919022	192.168.11.1	192.168.11.61	ICMP	82	Echo (ping) reply id=0x000:
162	2.919072	192.168.11.1	192.168.11.61	ICMP	82	Echo (ping) reply id=0x000:
164	2.919106	192.168.11.1	192.168.11.61	ICMP	82	Echo (ping) reply id=0x000:
166	2.930025	192.168.11.61	192.168.11.1	ICMP	82	Echo (ping) request id=0x000:

Sequence Number (BE): 28175 (0x6e0f)
Sequence Number (LE): 3950 (0x0f6e)

[No response seen]
[Expert Info (Warning/Sequence): No response seen to ICMP request]
[No response seen to ICMP request]
[Severity level: Warning]
[Group: Sequence]

Data (12 bytes)
Data: 6269742e6c792f2d77706132
[Length: 12]

0000 aa aa 03 00 00 00 08 00 45 00 00 28 ba 55 40 00 E..U@.
0010 40 01 e8 f0 c0 a8 0b 3d c0 a8 0b 01 08 00 3f 0e @.....=.....?
0020 00 01 6e 0f 62 69 74 2e 6c 79 2f 2d 77 70 61 32 ..n:bit.ly/-wpa2

Figura 11: URL enviada por el informante encontrada en la data de un paquete ICMP request (seleccionado en azul)

3. Desarrollo (PASO 2)

3.1. Script para modificar diccionario original

```
felipe@IdeaPad-L340-15IRH:~$ grep -v '^[0-9]' /home/felipe/Descargas/"FM.lab cripto3"/rockyou.txt |
sed -e 's/^\([a-zA-Z]\)/\u\1;/s/$/0/' | tee /home/felipe/Descargas/"FM.lab cripto3"/rockyou_mod.dic
```

Figura 12: Script por terminal que modifica el diccionario 'rockyou.txt' y que retorna el archivo 'rockyou_mod.dic'.

3.2. Descripción del Comando

- **grep -v '^[0-9]'**: Filtra las líneas del archivo que **no** comienzan con un dígito. Esto excluye cualquier línea cuyo primer carácter sea un número.
- **sed -e 's/[a-zA-Z]/\u\1;/s/\$/0/':**
 - La primera parte del comando **sed** (**s/[a-zA-Z]/\u\1/**) convierte la primera letra de cada línea a mayúscula.
 - La segunda parte (**s/\$/0/**) agrega un 0 al final de cada línea.

tee: Redirige la salida modificada tanto a la terminal como al archivo especificado **rockyou_mod.dic**.

3.3. Resultado

El comando procesa el archivo **rockyou.txt** para excluir las líneas que comienzan con un número, capitaliza la primera letra de cada línea restante, agrega un 0 al final de cada línea y guarda el resultado en **rockyou_mod.dic**.

```
felipe@IdeaPad-L340-15IRH:~$ head /home/felipe/Descargas/"FM.lab cripto3"/rockyou.txt -n 10
123456
12345
123456789
password
iloveyou
princess
1234567
rockyou
12345678
abc123
felipe@IdeaPad-L340-15IRH:~$ head /home/felipe/Descargas/"FM.lab cripto3"/rockyou_mod.dic -n 10
Password0
Iloveyou0
Princess0
Rockyou0
Abc1230
Nicole0
Daniel0
Babygirl0
Monkey0
Lovely0
felipe@IdeaPad-L340-15IRH:~$ █
```

Figura 13: Comparación entre las primeras 10 palabras del diccionario original y el modificado.

3.4 Cantidad de passwords finales que contiene rockyou_mod4didDESARROLLO (PASO 3)

3.4. Cantidad de passwords finales que contiene rockyou_mod.dic

```
felipe@IdeaPad-L340-15IRH:~$ wc -l /home/felipe/Descargas/"FM.lab cripto3"/rockyou.txt
14344391 /home/felipe/Descargas/FM.lab cripto3/rockyou.txt
felipe@IdeaPad-L340-15IRH:~$ wc -l /home/felipe/Descargas/"FM.lab cripto3"/rockyou_mod.dic
11059584 /home/felipe/Descargas/FM.lab cripto3/rockyou_mod.dic
felipe@IdeaPad-L340-15IRH:~$
```

Figura 14: Comparación de número de contraseñas en el diccionario original con el modificado.

Se puede apreciar en la figura 13 que el diccionario ha sido modificado exitosamente, posteriormente en la figura 14 se utiliza el comando 'wc -l' tanto para el archivo original como para el modificado para contar las líneas que tiene cada uno y así sus contraseñas, resultando en que el original contiene 14.344.391 mientras que luego del filtro que se aplicó el modificado tiene 11.059.584, es decir que 3.284.807 contraseñas comenzaban con un dígito.

4. Desarrollo (Paso 3)

4.1. Obtiene contraseña con hashcat con potfile

```
felipe@IdeaPad-L340-15IRH:~/Descargas/FM.lab cripto3$ hashcat --potfile-path potfile.pot -m 22000 -a 0 handshake.22000 rockyou_mod.dic
hashcat (v6.2.5) starting
```

Figura 15: Ataque hashcat con potfile al archivo 'handshake.pcap' (o handshake.22000) usando el diccionario modificado rockyou_mod.dic

Inicialmente el comando 'hashcat' no funcionó con el archivo descargado de internet, handshake.pcapng, debido a una incompatibilidad en los plugins por lo que se modificó la captura de tráfico mediante el comando 'hcxpcapngtool -o handshake.22000 handshake.pcapng' creando así una copia con extensión 22000 ideal para el comando hashcat.

1. **Explicación del código -m 22000:** Especifica el modo de hash que se va a utilizar. En este caso, el modo 22000 indica que se utilizará el algoritmo de hash WPA/WPA2.
-a 0: Especifica el tipo de ataque que se va a realizar. En este caso, el valor 0 indica un ataque de fuerza bruta.

'rockyou_mod.dic': Es el nombre del archivo de diccionario que se utilizará para probar las contraseñas. Este archivo contiene una lista de posibles contraseñas que se probarán durante el ataque de fuerza bruta.

```

* Device #1: pthread-Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, 6870/13805 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 hit:

* Filename..: rockyou_mod.dic
* Passwords.: 11059577
* Bytes.....: 119975020
* Keyspace..: 11059577

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

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: handshake.22000
Time.Started.....: Wed May 29 05:10:25 2024 (0 secs)
Time.Estimated...: Wed May 29 05:10:25 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 8056 H/s (7.75ms) @ Accel:256 Loops:128 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 3817/11059577 (0.03%)
Rejected.....: 1769/3817 (46.35%)
Restore.Point....: 0/11059577 (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: 52c Util: 29%

```

Figura 16: Contraseña obtenida mediante hashcat con potfile, se obtuvo 'Security0' (en negro)

4.2. Nomenclatura del output

```

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

```

Figura 17: Output resultante de usar hashcat con potfile.

Sección Amarilla: Primer MIC del 4-Way handshake.

Sección Morada: MAC del Access Point.

Sección Verde: MAC de la Station (cliente).

Sección Azul: SSID de la red.

Sección Roja : Contraseña de la red.

4.3. Obtiene contraseña con hashcat sin potfile

```

felipe@IdeaPad-L340-15IRH:~/Descargas/FM.lab cripto3$ hashcat --potfile-disable -m 22000 -a 0
handshake.22000 rockyou_mod.dic
hashcat (v6.2.5) starting

OpenCL API (OpenCL 2.0 pocl 1.8 Linux, None+Asserts, RELOC, LLVM 11.1.0, SLEEF, DISTRO, POCL
_DEBUG) - Platform #1 [The pocl project]
=====
* Device #1: pthread-Intel(R) Core(TM) i5-9300H CPU @ 2.40GHz, 6870/13805 MB (2048 MB allocat
able), 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 hit:
* Filename...: rockyou_mod.dic
* Passwords..: 11059577
* Bytes.....: 119975020
* Keyspace...: 11059577

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

Session.....: hashcat
Status.....: Cracked
Hash.Mode.....: 22000 (WPA-PBKDF2-PMKID+EAPOL)
Hash.Target.....: handshake.22000
Time.Started....: Wed May 29 05:16:56 2024 (1 sec)
Time.Estimated...: Wed May 29 05:16:57 2024 (0 secs)
Kernel.Feature...: Pure Kernel
Guess.Base.....: File (rockyou_mod.dic)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....: 8709 H/s (7.07ms) @ Accel:32 Loops:1024 Thr:1 Vec:8
Recovered.....: 1/1 (100.00%) Digests
Progress.....: 2907/11059577 (0.03%)
Rejected.....: 1371/2907 (47.16%)

```

Figura 18: hashcat con potfile desactivado

En la figura 18 se aprecia el mismo resultado que en la figura 16, la diferencia es que en el primero caso se generó un archivo .pot que contiene las contraseñas encontradas, almacenando el string que se analizó en la nomenclatura, en cambio sin potfile solo se retorna por terminal.

4.4. Nomenclatura del output

Se aprecia nuevamente la misma salida de la figura 17, por lo que la nomenclatura no cambia, se aprecia al final del string la SSID de red y la contraseña 'Security0'

4.5. Obtiene contraseña con aircrack-ng

```

felipe@IdeaPad-L340-15IRH:~/Descargas/FM.lab cripto3$ aircrack-ng -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: Se utiliza aircrack-ng para conseguir la contraseña del archivo descargado de internet, handshake.pcapng, además se utiliza la libreria rockyou modificada para facilitar el trabajo.

Debido a que aircrack-ng no trabaja con archivos con extensión .pcapng, se tuvo que cambiar el formato del archivo a través del software de Wireshark para poder utilizar el comando que se aprecia en la figura 19.

```

Aircrack-ng 1.6

[00:00:00] 3418/9296207 keys tested (7521.44 k/s)

Time left: 20 minutes, 35 seconds                                0.04%

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

```

Figura 20: Resultado Aircrack con handshake.pcapng

Como se puede apreciar en la figura 20, se obtuvo nuevamente la contraseña 'Security0' mediante aircrack, comprobando como en los casos anteriores que el procedimiento ha sido exitoso.

4.7. Obtiene contraseña con pycrack

```
felipe@IdeaPad-L340-15IRH:~/Descargas/FM.lab cripto3/PyCrack$ python3 pywd.py
!!!Password Found!!!
Desired MIC1:          1813acb976741b446d43369fb96dbf90
Computed MIC1:         1813acb976741b446d43369fb96dbf90

Desired MIC2:          a349d01089960aa9f94b5857b0ea10c6
Computed MIC2:         a349d01089960aa9f94b5857b0ea10c6

Desired MIC2:          5cf0d63af458f13a83daa686df1f4067
Computed MIC2:         5cf0d63af458f13a83daa686df1f4067
Password:              Security0
felipe@IdeaPad-L340-15IRH:~/Descargas/FM.lab cripto3/PyCrack$
```

Figura 22: Contraseña obtenida mediante pycrack

Luego de obtener todos los parámetros de Wireshark se ejecuta el comando 'python3 pywd.py' dentro de la carpeta pycrack en donde también se encuentra handshake.pcap y el librería rockyou_mod.dic, retornando los MIC del handshake y la contraseña 'Security0' exitosamente.

Conclusiones y comentarios

esta experiencia proporcionó una valiosa lección sobre la vulnerabilidad de utilizar contraseñas cortas, que tienen una baja entropía. Además, fue una oportunidad importante para comprender el funcionamiento de la encriptación WEP y WPA2.

Durante el proceso, se pudo observar que la encriptación WEP es más susceptible a ataques debido a su dependencia de los Vectores de Inicialización (IVs), lo que la hace más fácil de comprometer en comparación con WPA2. Por otro lado, WPA2 presenta una mayor complejidad para ser vulnerado, ya que se basa en algoritmos de hash más robustos.

En conclusión, esta experiencia resaltó la importancia de utilizar contraseñas seguras y de comprender las diferentes capas de seguridad en las redes inalámbricas para proteger la información personal y empresarial.

Issues

Los problemas que dificultaron la experiencia fueron los siguientes:

1. **dificultad para obtener la contraseña por aircrack en la parte1:** esto debido a que este laboratorio lo recuperé en solitario y al haber poco trafico de red, aircrack tardó mucho más en recopilar paquetes y encontrar la key.
2. **incompatibilidad hashcat con la versión del plugin asociado a la captura:** el versionamiento es sumamente importante, ya que se tiene instalada la ultima versión de hashcat y la captura de trafico de red necesitaba actualizarse.

3. **identificar los parámetros para la configuración de pycrack:** posiblemente el proceso más largo de la experiencia fue indagar en distintos parámetros sin éxito, debido a un desconocimiento de los conceptos que representaban, pero después de investigar un poco pude identificar que el handshake se centraba en el protocolo eapol así como los datos que necesitaba.