Informe Laboratorio 3

Sección 2

Alumno Matias Herrera
e-mail: matias.herrera2@mail.udp.cl

Mayo de 2024

Índice

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

Al momento de dejar en modo monitoreo la antena wifi para obtener la red del informante se procede a utilizar el comando **sudo airodump-ng wlan0mon -W Intersec.pcap**. El

comando **sudo** hace que se ejecute en proceso administrador, **wlan0mon** es el nombre de la interfase en modo monitor, el comando **-W** se utiliza para crear un archivo cap de wireshark para realizar el análisis de paquetes de la red del informante, los resultados se observan en la figura 1.

Como se puede observar en la figura 1 en las primeras lineas tenemos la interface llamada **WEP** el cual es el nombre de la red del informante esto se sabe ya que fue informado a que la red del informante estaría constan mente entregando trafico de red y se puede demostrar a través de la tabla de monitoreo la cual cuenta con un **#Data** de 22104 paquetes obtenidos de dicha red.

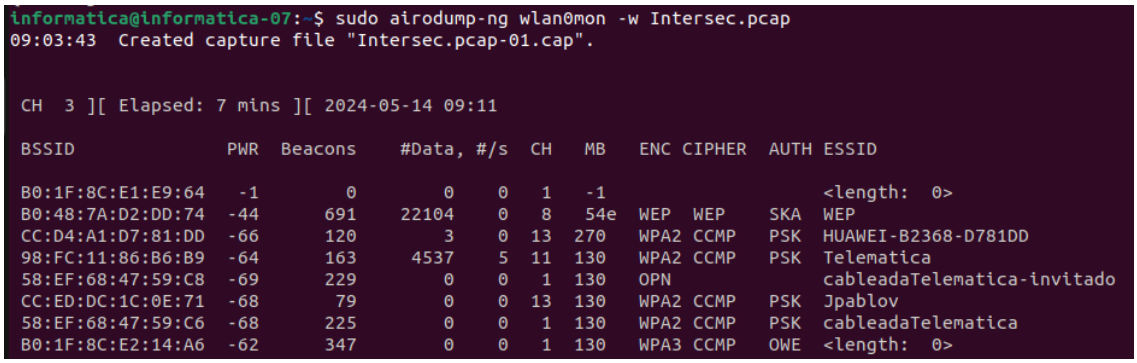

Figura 1: Captura de interfaces de red en modo monitor

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

Sabemos que Aircrack-ng utiliza el algoritmo RC4 con un vector de inicialización de 24 bits lo que permite $2^{24} = 16{,}777{,}216$ combinaciones posibles, los IVs se transmiten en texto plano en cada paquete, y su objetivo es evitar la reutilización de la misma clave de cifrado para los diferentes paquetes. Se requieren tener suficientes paquetes, esto puede implicar que algunos IVs se repitan lo que conlleva colisiones estas pueden servirnos para reducir la clave de la red del informante. La probabilidad de que no haya colisiones en n IVs puede ser aproximada utilizando el problema del cumpleaños. Para n IVs:

$$P(sincolisiones) \approx \exp\left(-\frac{n^2}{2 \cdot 2^{24}}\right)$$

Para que la probabilidad de al menos una colisión sea significativa (por ejemplo, 50 %), se necesita:

$$n \approx \sqrt{2 \cdot 2^{24} \cdot \ln(2)} \approx 4823$$

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

Luego de obtener los paquetes de la red del informante en un archivo .cap se hace uso del comando **aircrack-ng Intersec.pcap-01.cap** esto lo que realiza nos entrega el listado de las redes obtenidas en la sección 2.1 como se puede apreciar en esta figura 2. Luego de obtener la lista observamos que el puesto 78 es la red del informante llamada **WEP** como se puede observar en la figura 3, como se puede apreciar tenemos 22104 IVs para realizar el ataque por defecto de aircrack-ng. Luego de seleccionar la red de la lista aircrack-ng realiza el proceso de descriptado RC4 de los paquetes obtenidos para finalmente obtener la clave de la red como se puede observar en la figura 4.

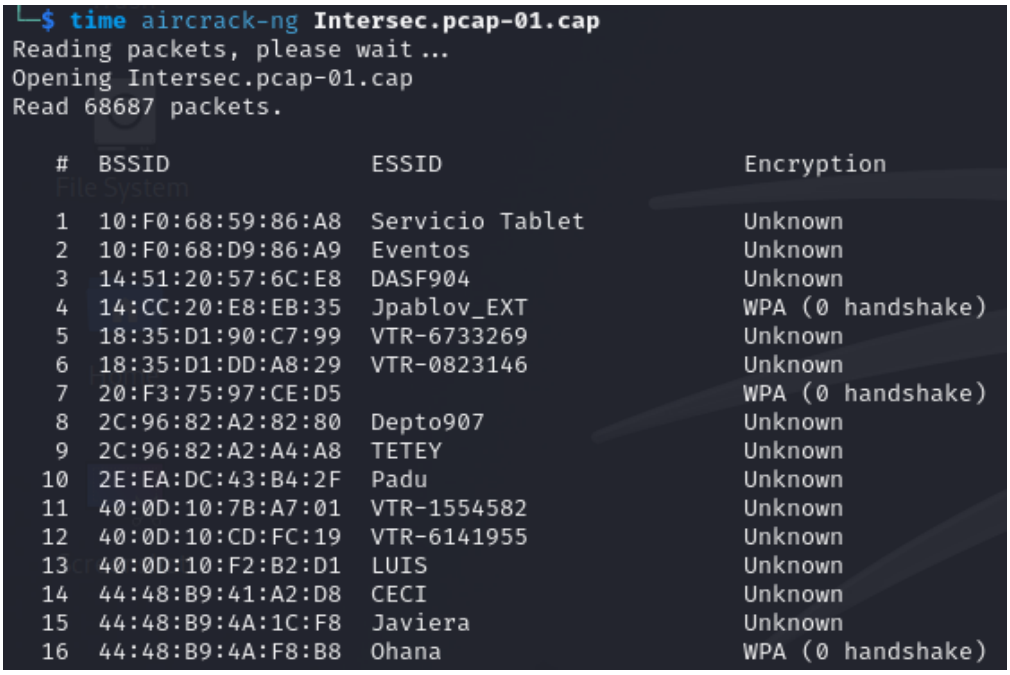

Figura 2: Listado redes obtenidas

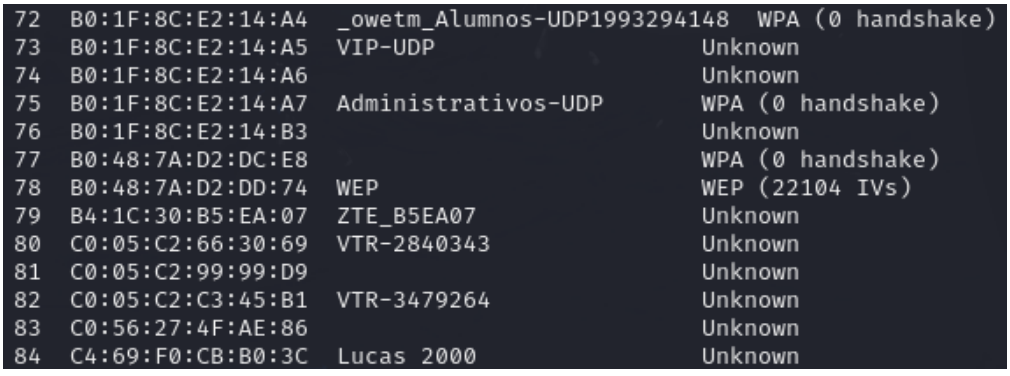

Figura 3: Red del informante y su información obtenida

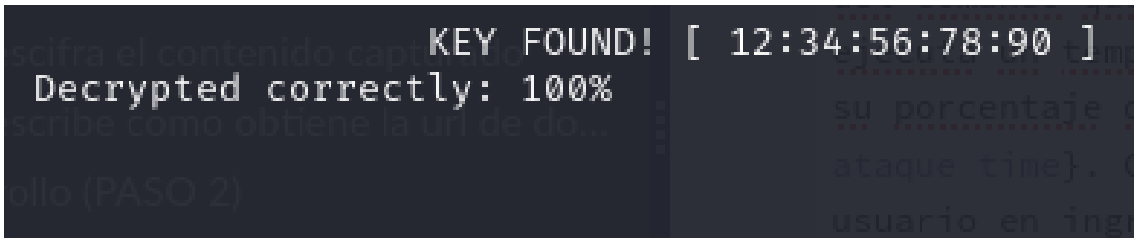

Figura 4: Ataque por defecto aircrack-ng

2.4. Indica el tiempo que demoró en obtener la password

Podemos tomar el tiempo que le toma obtener la password a través del comando time anteponiéndolo antes del comando que queremos ejecutar en este caso **time aircrack-ng Intersec.pcap-01.cap** este ejecuta un temporizador que nos entrega el tiempo de ejecución real, el tiempo del user, del sistema y su porcentaje de uso en cpu para el comando como se puede apreciar en la figura 5. Como se puede observar tenemos un tiempo real el cual considera cuanto se tarda el usuario en ingresar los datos de 0.94 segundos, un tiempo de usuario de 0.08 segundos y un tiempo de sistema de 0.03 segundos.

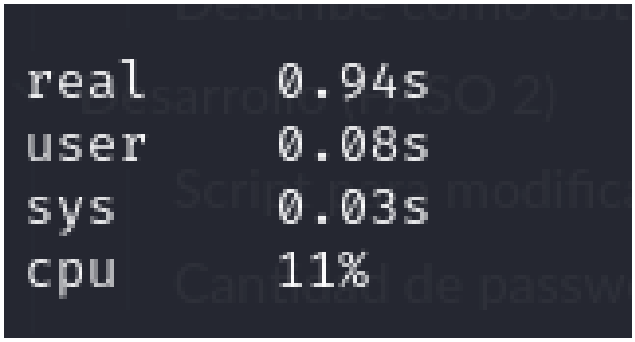

Figura 5: Tiempo de ejecución de ataque por defecto Aircrack-ng

2.5. Descifra el contenido capturado

Ahora que tenemos la key procedemos a descifrar el archivo de paquetes anteriormente utilizado para obtener solo los paquetes de la red del informante a través del comando **sudo airdecap-ng -w 12:34:56:78:90 Intersec.pcap-01.cap** este retorna un archivo con el nombre del archivo mas **-dec**, el resultado de esto en consola es lo que se puede apreciar en la figura 6

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

Revisando el archivo **Intersec.pacap-01-dec.cap** podemos revisar los parámetros en la capa de Internet (Capa 3) podemos observar la data de estos paquetes donde tenemos la

```
┌──(kali㉿kali)-[~]
└─$ sudo airdecap-ng -w 12:34:56:78:90 Intersec.pcap-01.cap
[sudo] password for kali:
Total number of stations seen            26
Total number of packets read          68687
Total number of WEP data packets      22104
Total number of WPA data packets       4424
Number of plaintext data packets          6
Number of decrypted WEP  packets      22104
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: Comando airdecap con la clave

trama con el nombre de una url llamada **bin.ly/-wpa2** como se puede observar en la figura 7, el paquete de tipo ICMP donde su data es de 12 bytes como se observa en la trama descifrada en tabla ASCII.

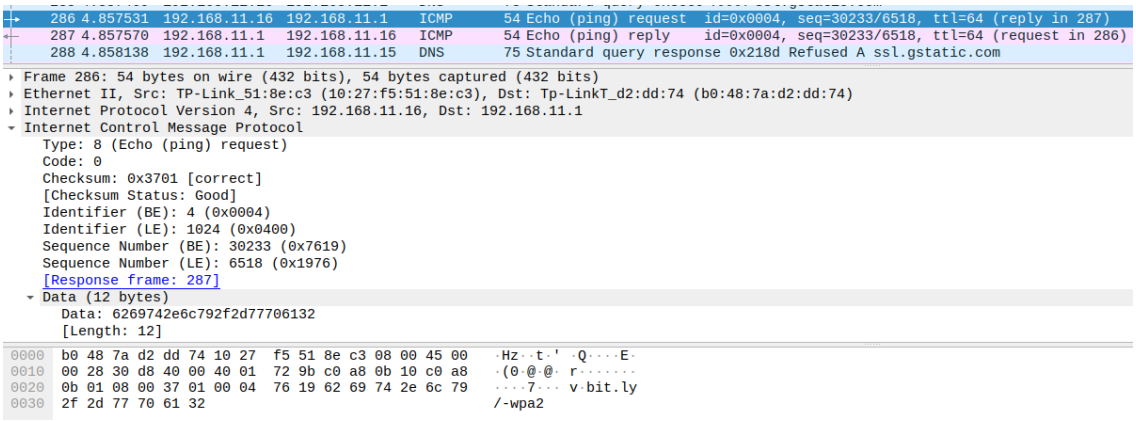

Figura 7: Paquetes descifrados y su información

3. Desarrollo (PASO 2)

3.1. Script para modificar diccionario original

Para realizar este script utilizamos los siguiente datos como se pueden observar en la figura 8. El código cuenta con el dato Original que es el archivo **rockyou.txt** y el archivo modificado que se llamara **rockyou_mod.dic**, luego se define los comandos sed para realizar las modificaciones, la primera expresión elimina todas las líneas que comienzan con un dígito numérico del archivo original, la segunda convierte el primer carácter de cada línea restante a mayúsculas y la ultima agrega el carácter ”0.ªl final de cada línea y todo es redirigido al archivo modificado, luego definimos un valor line_count que ocupa el comando wc -l (word count) y finalmente se ingresa a través de consola la cantidad de contraseñas obtenidas.

```
#!/bin/bash

original_file="rockyou.txt"
modified_file="rockyou_mod.dic"

sed '/^[0-9]/d; s/^\(.\)\(.*\)/\U\1\E\2/' "$original_file" | sed 's/$/0/' > "$modified_file"

line_count=$(wc -l < "$modified_file")

echo "La cantidad de contraseñas en el diccionario modificado es: $line_count"

```

Figura 8: Script de modificación Rockyou.txt

3.2. Cantidad de passwords finales que contiene rockyou_mod.dic

Finalmente luego de realizar las modificaciones obtuvimos 11.059.789 passwords como se observa en la figura 9.

```
┌──(kali㉿kali)-[~]
└─$ ./Modificador.sh
La cantidad de contraseñas en el diccionario modificado es: 11059798
```

Figura 9: Resultado script modificador

4. Desarrollo (Paso 3)

4.1. Obtiene contraseña con hashcat con potfile

Luego de obtener la url del handshake del paso 1, descargamos el archivo .pcap para realizar los ataques de fuerza bruta y luego de tener las biblioteca de password modificada del paso 2, se necesita el archivo handshake.pcap y necesitamos convertir en formato **.22000** el cual es el nuevo formato para trabajar con hashcat, esto se realiza con el comando **hcxpcapngtool -o handshake.22000 handshake.pcap** luego de convertir el archivo en el nuevo formato se procede a realizar el ataque de fuerza bruta con hashcat guardando en un archivo **.pot** a travez del comando 'hashcat -m 22000 handshake.22000 rockyou_mod.dic –potfile-path=handshake.pot' al finalizar en el proceso como se observa en la figura 10. Como resultado obtuvimos una password ingresado en el nuevo archivo handshake.pot como se muestra en la figura 11.

```
┌──(kali㉿kali)-[~]
└─$ hashcat -m 22000 handshake.22000 rockyou_mod.dic --potfile-path=handshake.pot
hashcat (v6.2.6) starting
```

Figura 10: Comando hashcat

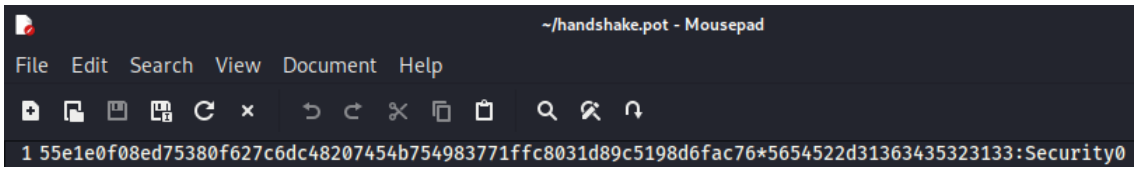

Figura 11: Resultado hashcat

4.2. Nomenclatura del output

La nomenclatura del output obtenido comienza con el PMKID (Pairwise Master Key Identifier) que es un identificador utilizado en el proceso de autenticación WPA/WPA2, luego separado con un asterisco tenemos SSID codificado en hexadecimal (ESSID) que es el nombre de la red el cual es 'VTR-1645213' y finalmente separado en dos puntos se entrega la contraseña encontrada para la red del informante.

4.3. Obtiene contraseña con hashcat sin potfile

Para la obtención de contraseña con el mismo programa pero sin solicitar un archivo potfile funciona de la siguiente manera utilizaremos el mismo archivo convertido en el punto anterior y realizamos el ataque de fuerza bruta a través del siguiente comando 'hashcat -m 22000 handshake.22000 rockyou_mod.dic –potfile-disable'. Luego de ingresar el comando por consola obtuvimos una sola password valida la cual se puede observar en la figura 12

Figura 12: Resultado hashcat sin potfile

4.4. Nomenclatura del output

La nueva nomenclatura a la hora de usar este comando nos entrega el resultado por consola el cual tiene la siguiente estructura:

- PMKID: Es un id usado en la autenticación WPA/WPA2. Se utiliza para verificar la autenticidad del handshake y la contraseña descubierta.
- BSSID: Esta es la dirección MAC del punto de acceso de la red WiFi.
- MAC del Cliente: Esta es la dirección MAC del cliente que se conecta a la red.
- ESSID: Nombre de la red WiFi en texto plano.
- Contraseña: Es la contraseña descubierta para la red.

4.5. Obtiene contraseña con aircrack-ng

Para obtener la contraseña con aircrack-ng necesitamos tener la dirección MAC del punto de acceso a la red, luego de eso se procede utilizar el siguiente comando 'aircrack-ng -w rockyou_mod.dic -b b0:48:7a:d2:dc:18 handshake.pcap' el resultado del comando es el siguiente como se observa en la figura 13. Se puede corroborar que la password obtenido es la misma obtenida en los anteriores ataques con otro programas.

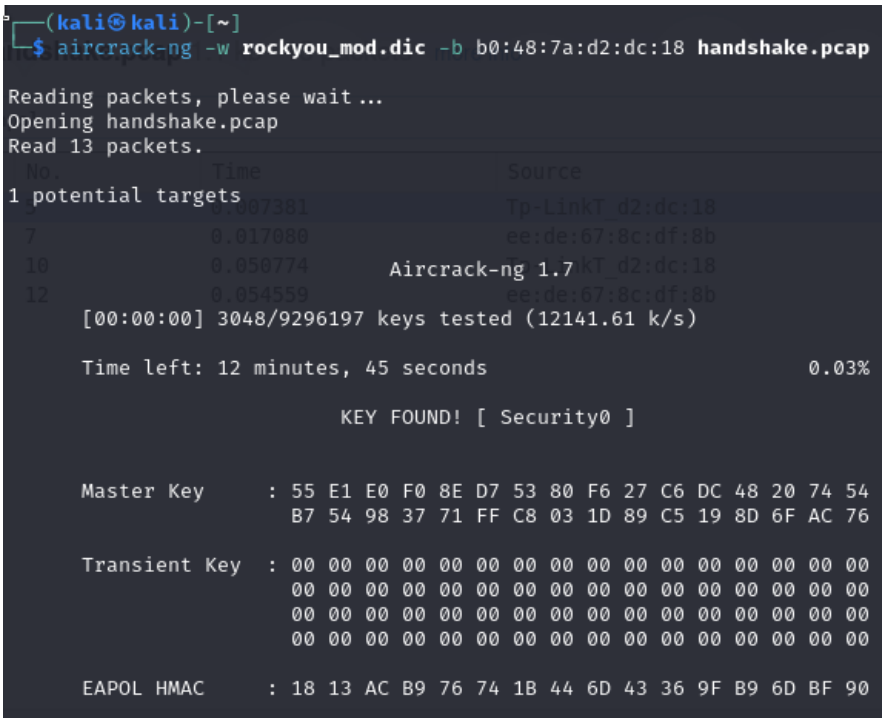

Figura 13: Ataque aircrack-ng

4.6. Identifica y modifica parámetros solicitados por pycrack

Los datos a modificar para realizar la obtención de contraseña son las siguientes:

- La dirección en memoria del diccionario a utilizar (rockyou_mod.dic).
- SSID: Al observar el primera paquete de la captura de handshake podemos encontrarla la cual es 'VTR-1645213'.
- aNonce: Numero aleatorio que se utiliza para establecer la PTK, la cual utilizamos del primer paquete del primer mensaje del protocolo 'eapol' con la etiqueta 'WPA Key Nonce'.
- SNonce: Ultima información que el punto de acceso necesita para calcular el PTK esta se encuentra en el segundo mensaje del protocolo 'eapol'.
- APMac: Dirección MAC del punto de acceso, el cual es 'b0487a2ddc18' en hexadecimal.
- CliMac: Dirección MAC del cliente la cual es 'eede678cdf8b'.
- mics: El MIC es un valor calculado que se utiliza para garantizar la integridad y autenticidad de los mensajes en una red WPA/WPA2. En este caso lo tenemos para cada paquete, se necesitan los 3 últimos paquetes del protocolo 'eapol'.
- Data: La data es toda la trama 802.1x de los 3 últimos paquetes del protocolo.

los datos cambiados se pueden observar en las figuras 14 y 15

```
115 if __name__ == "__main__":
116
117     #Read a file of passwords containing
118     #passwords separated by a newline
119     with open('rockyou_mod.dic') as f:
120         S = []
121         for l in f:
122             S.append(l.strip())
123     #ssid name
124     ssid = "VTR-1645213"
125     #ANonce
126     aNonce = a2b_hex('4c2fb7eca28fba45accefde3ac5e433314270e04355b6d95086031b004a31935')
127     #SNonce
128     sNonce = a2b_hex("30bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9cdc")
129     #Authenticator MAC (AP)
130     apMac = a2b_hex("b0487a2ddc18")
131     #Station address: MAC of client
132     cliMac = a2b_hex("eede678cdf8b")
```

Figura 14: Variables modificadas de Pycrack parte 1

4.7. Obtiene contraseña con pycrack

Conclusiones y comentarios

Como conclusion podemos determinar que la experiencia se pudo realizar sin tantos conflictos pudimos cubrir el objetivo de la experiencia que se centraba realizar el desafio de

```
133     #The first MIC
134     mic1 = "1813acb976741b446d43369fb96dbf90"
135     #The entire 802.1x frame of the second handshake message with the MIC field set to all zeros
136     data1 =
    a2b_hex("0103007502010a0000000000000000000130bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9cdc00
    0000000000000000000000000000000000000000000000000000000000000001813acb976741b446d43369fb96dbf90001630140100000
    fac040100000fac040100000fac020000")
137     #The second MIC
138     mic2 = "a349d01089960aa9f94b5857b0ea10c6"
139     #The entire 802.1x frame of the third handshake message with the MIC field set to all zeros
140     data2 =
    a2b_hex("020300970213ca0010000000000000000024c2fb7eca28fba45accefde3ac5e433314270e04355b6d95086031b004a3193500
    0000000000000000000000000000cd0000000000000000000000000000a349d01089960aa9f94b5857b0ea10c60038db0eb43c3f-
    af2c0e8b7e8a471f962c307e707e4718be724459167a88fa281f4d7ce38f012943da788d0a7159c9fac6ad71483d788cecf18b")
141     #The third MIC
142     mic3 = "5cf0d63af458f13a83daa686df1f4067"
143     #The entire 802.1x frame of the forth handshake message with the MIC field set to all zeros
144     data3 =
    a2b_hex("0103005f02030a0000000000000000002000000000000000000000000000000000000000000000000000000000000000
    0000000000000000000000000000000000000000000000000000000000000005cf0d63af458f13a83daa686df1f40670000")
145     #Run an offline dictionary attack against the access point
146     TestPwds(S, ssid, aNonce, sNonce, apMac, cliMac, data1, data2, data3, mic1, mic2, mic3)
```

Figura 15: Variables modificadas de Pycrack parte 2

autenticacion y obtener la clave secreta del informante desconfiado de manera anonima, comprobamos con diferentes metodos como obtener la constraseña, pero esto aun asi ha sido ineficiente. ¿Quien me asegura que solo yo pude obtener la contraseña y no otras personas con los mismos conocimientos que los mios? realizando los mismos procesos que realizamos en dicha experiencia.

Issues

Dentro de las dificultades observadas a lo largo de esta experiencia son los siguientes:

1. Una de las primeras dificultades que se experimento fue entender el funcionamiento de los comandos airodump-ng para que cuando se solicite hacer un escaneo de la red tomara solo el canal correspondiente a la red del informante que correspondía a la red 8 lo cual provoco mucha demora a la hora de posteriormente utilizar el archivo para realizar un ataque por defecto con aircrack-ng ya que teníamos demasiadas redes que no eran relevantes para la experiencia, esto se soluciono simplemente con filtrar o crear un nuevo archivo donde solo los paquetes de la red del informante estuvieran en el.

2. Existieron dificultades a la hora de utilizar el comando 'sed' para la parte 2 debido a tener bago conocimiento de expresiones regulares para utilizar de manera eficiente las consultas para realizar las modificaciones al archivo 'Rockyou.txt', como solucion fue entender el funcionamiento de las expresiones regulares para entender cual era su funcionamiento para poder realizar las modificaciones.

3. Una de las problemáticas que mas me tomo tiempo entender fue a la hora de entender por que no podía obtener la contraseña del handshake a través de hashcat y esto se debía a la forma en como hashcat necesita obtener los datos del '.pcap' y esto se soluciona haciendo la conversión de formato de '.pcap' a '.22000' luego de realizar dicha conversión hashcat no me genero problemas a la hora de obtener la contraseña.

4. Finalmente la ultima de las dificultades que se experimentaron en esta experiencia fue entender los conceptos utilizados y el reconocimiento de los parámetros que me entregaban cada ítem del paso 3 y sus nomenclaturas respectivas, las resoluciones realizadas fue investigar como funcionaban cada una de ella a través de sus documentaciones y entender que datos entregaban sus outputs.