Informe Laboratorio 3

Sección x

Javier Ahumada
e-mail: javierigna.ahumada@mail.udp.cl

Mayo de 2023

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 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 rockyou_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

Se destaca sobre las otras redes por la excesiva data que se capturaba con la tarjeta de wi fi

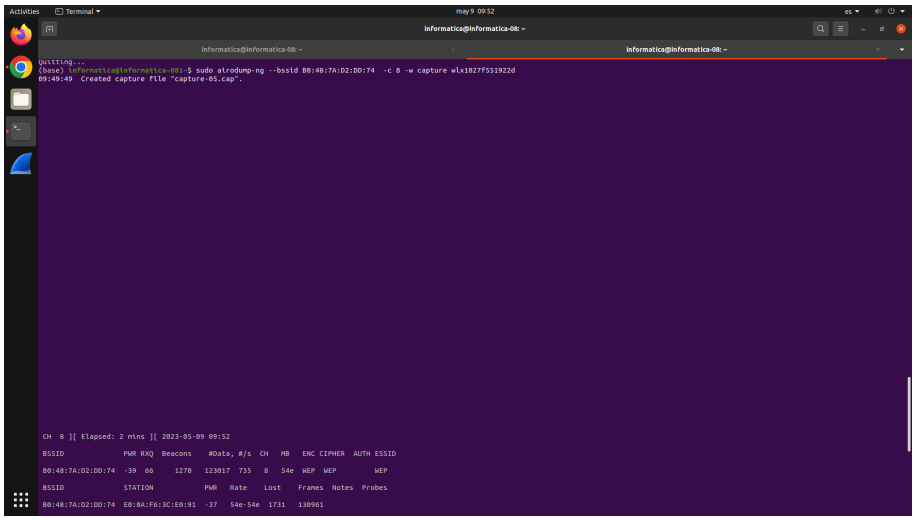

Figura 1: Airodump de la red WEP

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

Porque en ataques de diccionario se necesitan suficientes paquetes capturados para obtener la clave de cifrado. En el caso de WEP, se requieren alrededor de 5000 IVs para tener una probabilidad razonable de recuperar la clave

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

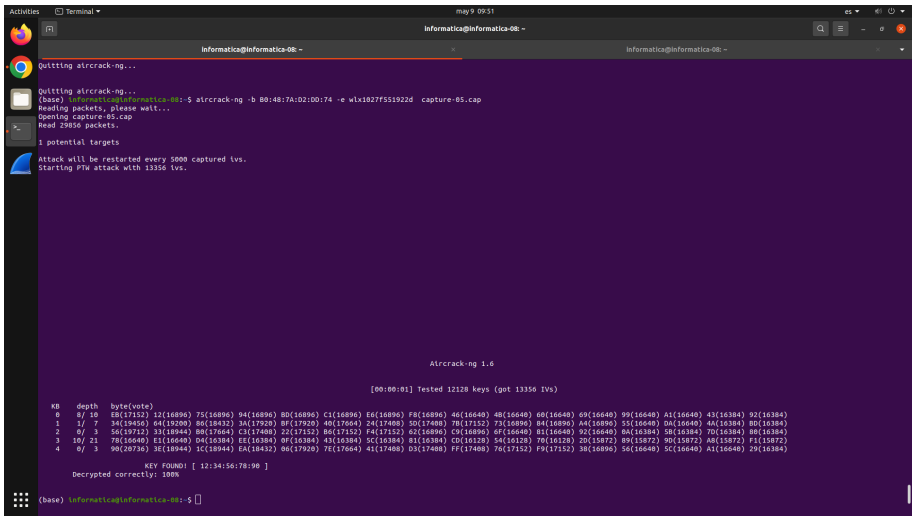

Figura 2: Dado el BSSID de WEP se obtiene la KEY 12:34:56:78:90

2.4. indica el tiempo que demoró en obtener la password

1 hora y 20 minutos

2.5. descifra el contenido capturado

```
javier@javier-Nitro-AN515-53:~/Escritorio/cripto/j/lab3$ airdecap-ng -w 12:34:56:78:90 capture-05.cap
Total number of stations seen            5
Total number of packets read        314811
Total number of WEP data packets    141542
Total number of WPA data packets         0
Number of plaintext data packets         2
Number of decrypted WEP  packets    141542
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 3: Dado el airodump capture-05.cap generado anteriormente y la KEY obtenida, se descifra el contenido y genera un nuevo archivo .cap

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

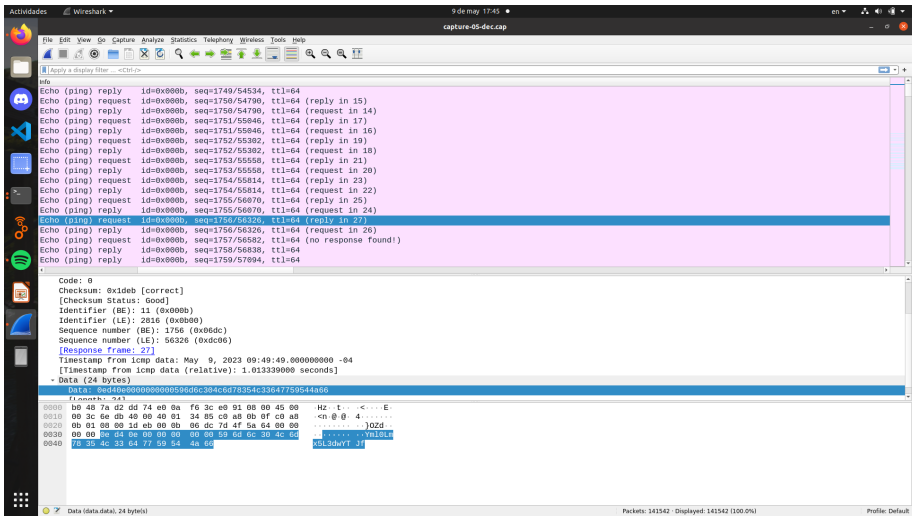

Figura 4: Se puede ver que en cada paquete del nuevo archivo generado se repiten los ultimos bytes.

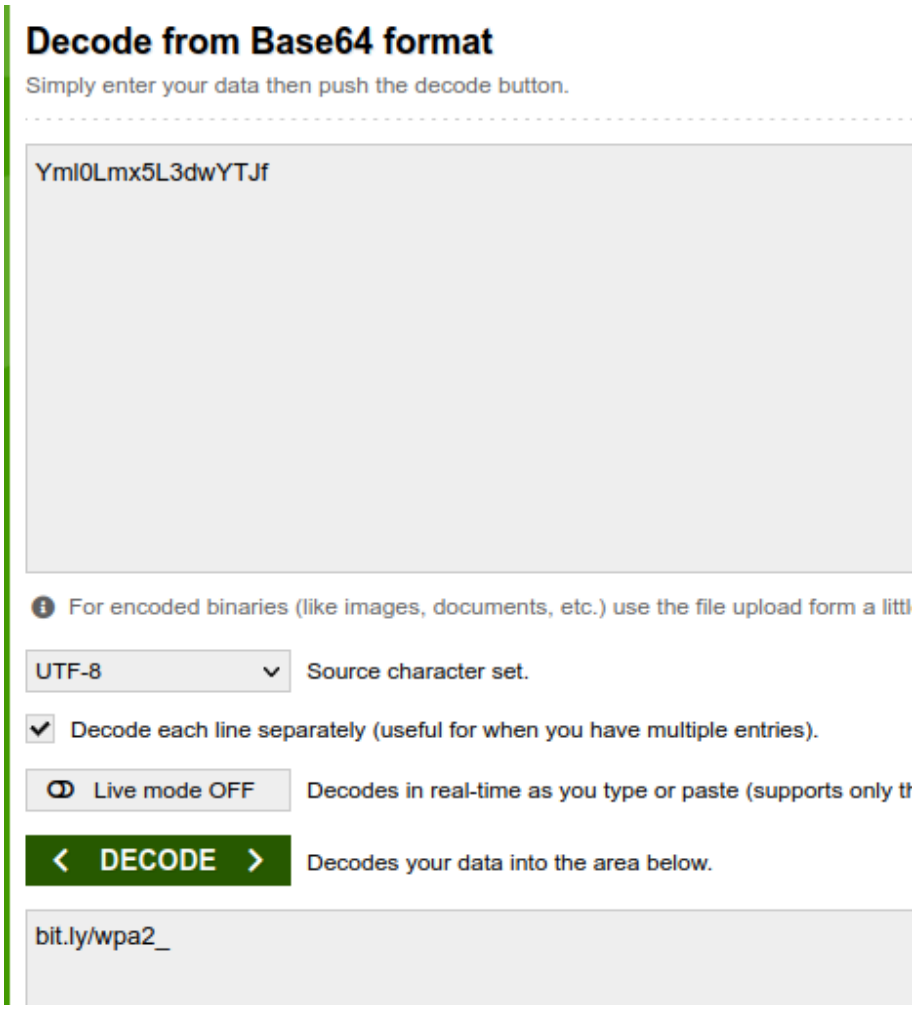

Figura 5: Luego se toma esa data y se decodifica en base 64 y se obtiene un link

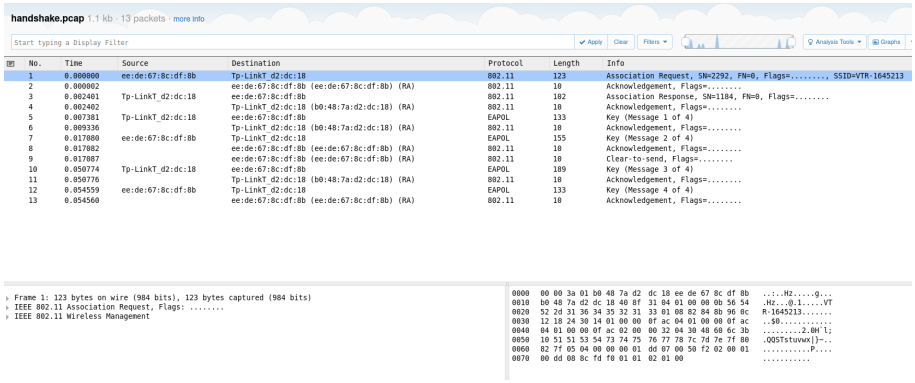

Figura 6: El archivo descargado es handshake.cap

3. Desarrollo (PASO 2)

3.1. indica script para modificar diccionario original

```python
# Abrir el archivo de texto para leer y crear un nuevo archivo para escribir
with open('rockyou.txt', 'r', encoding='ISO-8859-1') as archivo_origen, open('rockyou_mod.txt', 'w') as archivo_modificado:
    # Contador de cadenas modificadas
    contador = 0
    # Iterar sobre cada línea del archivo de origen
    for linea in archivo_origen:
        # Eliminar el carácter de nueva línea al final de la línea
        linea = linea.strip()
        # Comprobar si la línea está vacía
        if not linea:
            # Saltar esta línea porque está vacía
            continue
        # Comprobar si el primer carácter es un número
        if linea[0].isdigit():
            # Saltar esta línea porque el primer carácter es un número
            continue
        # Convertir el primer carácter en mayúscula y agregar un cero al final de la línea
        linea_modificada = linea[0].upper() + linea[1:] + '0'
        # Escribir la línea modificada en el archivo modificado
        archivo_modificado.write(linea_modificada + '\n')
        # Incrementar el contador de cadenas modificadas
        contador += 1

# Imprimir el número de cadenas modificadas
print(f'Se han modificado {contador} cadenas.')
```

Figura 7: Script genera rockyou$_m odtxtdondeserealizalosolicitado$

3.2. cantidad de passwords finales que contiene rockyou_mod.dic

```
javier@javier-Nitro-AN515-53:~/Escritorio/cripto/j/lab3$ sudo python3 script.py
Se han modificado 11059725 cadenas.
javier@javier-Nitro-AN515-53:~/Escritorio/cripto/j/lab3$
```

Figura 8: 11059725 passwords

4. Desarrollo (Paso 3)

4.1. obtiene contraseña con hashcat con potfile

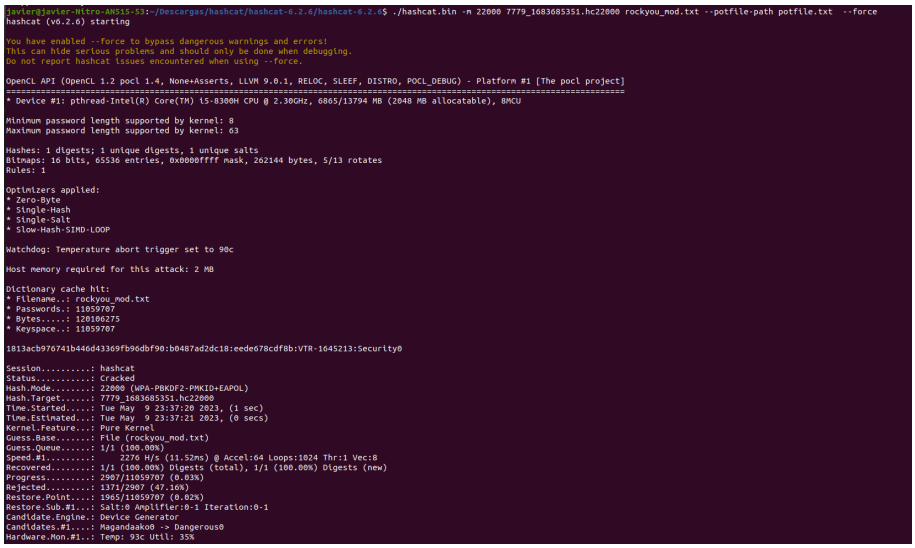

Figura 9: El archivo potfile se utiliza para almacenar los hashes que ya han sido descifrados, de modo que Hashcat no tenga que volver a descifrarlos

4.2. identifica nomenclatura del output

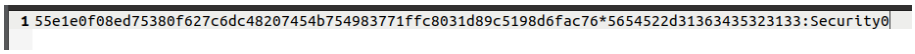

Figura 10: El output se divide en tres partes: el hash que ha sido crackeado, informacion adicional del hashcat y la contraseña encontrada Security0

4.3. obtiene contraseña con hashcat sin potfile

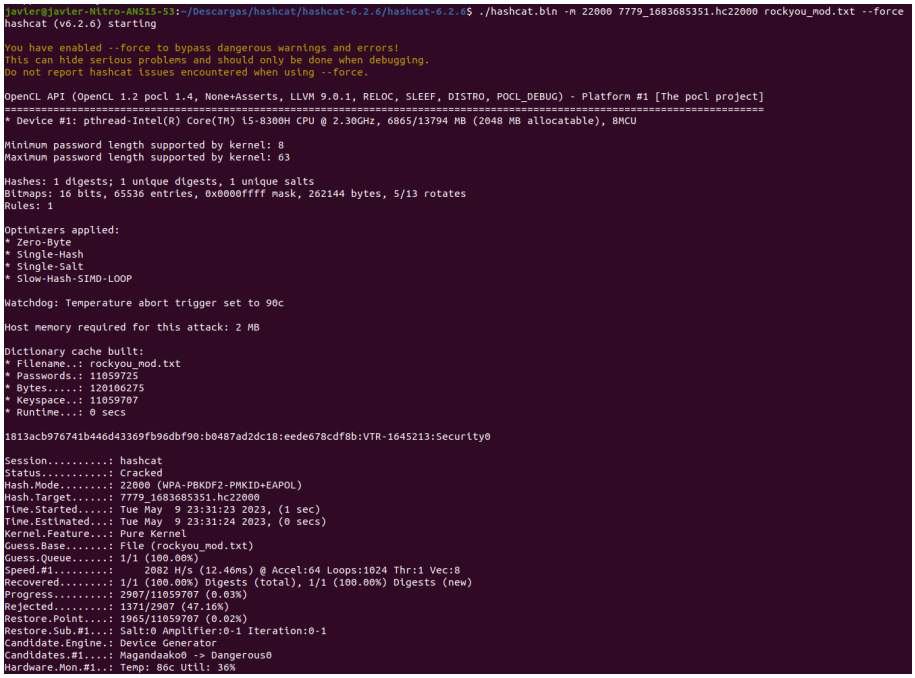

Figura 11:

4.4. identifica nomenclatura del output

En base a la imagen anterior, por defecto, hashcat guarda el resultado de la operación en un archivo de texto en la carpeta de trabajo actual. Donde se identifica la contraseña, el modo de hash, el hash crackeado, las contraseñas del diccionario usadas y la cantidad de bytes, el tiempo corrido y otros parametros.

4.5. obtiene contraseña con aircrack-ng

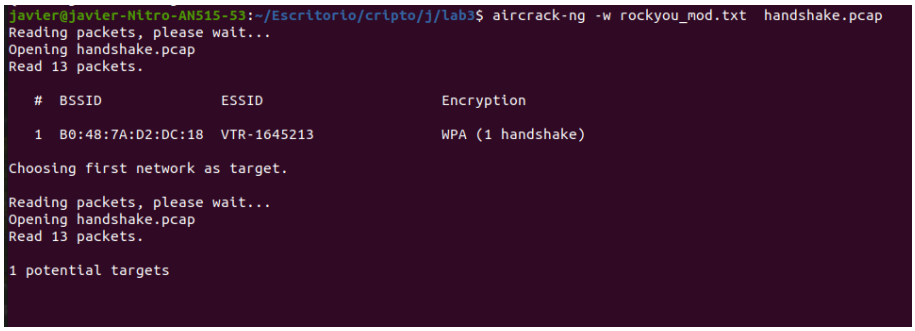

Figura 12: Ataque por fuerza bruta dado un diccionario y un handshake.pcap

```
                          Aircrack-ng 1.6

[00:00:00] 3385/9285363 keys tested (7100.67 k/s)

Time left: 21 minutes, 47 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

Transient Key  : 3C 1B 89 A6 31 30 BA 04 B6 59 D9 7E 65 BD D2 07
                 9E C6 8D 2A D6 EF 7F 9E A1 95 1C BC CC 62 A6 5D
                 CC 07 B2 E3 9D 12 99 A7 66 D4 3C D7 61 56 53 00
                 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

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

Figura 13:

4.6. identifica y modifica parámetros solicitados por pycrack

```
#RunTest()
#Read a file of passwords containing
#passwords separated by a newline
with open('rockyou_mod.txt') as f:
    S = []
    for l in f:
        S.append(l.strip())
#ssid name
ssid = "VTR-1645213"
#ANonce
aNonce = a2b_hex('4c2fb7eca28fba45accefde3ac5e433314270e04355b6d95086031b004a31935')
#SNonce
sNonce = a2b_hex("30bde6b043c2aff8ea482dee7d788e95b634e3f8e3d73c038f5869b96bbe9cdc")
#Authenticator MAC (AP)
apMac = a2b_hex("b0:48:7a:d2:dc:18")
#Station address: MAC of client
cliMac = a2b_hex("ee:de:67:8c:df:8b")
#The first MIC
mic1 = "1813acb976741b446d43369fb96dbf90"
#The entire 802.1x frame of the second handshake message with the MIC field set to all zeros
data1 = a2b_hex("30140100000fac040100000fac040100000fac020000")
#The second MIC
mic2 = "a349d01089960aa9f94b5857b0ea10c6"
#The entire 802.1x frame of the third handshake message with the MIC field set to all zeros
data2 = a2b_hex("db0eb43c3faf2c0e8b7e8a471f962c307e707e4718be724459167a88fa281f4d7ce38f012943da788d0a7159c9fac6ad71483d788cecf18b")
#The third MIC
mic3 = "9dc81ca6c4c729648de7f00b436335c8"
#The entire 802.1x frame of the forth handshake message with the MIC field set to all zeros
data3 = a2b_hex("0")
#Run an offline dictionary attack against the access point
TestPwds(S, ssid, aNonce, sNonce, apMac, cliMac, data1, data2, data3, mic1, mic2, mic3)
```

Figura 14: Se modifican los campos solicitados

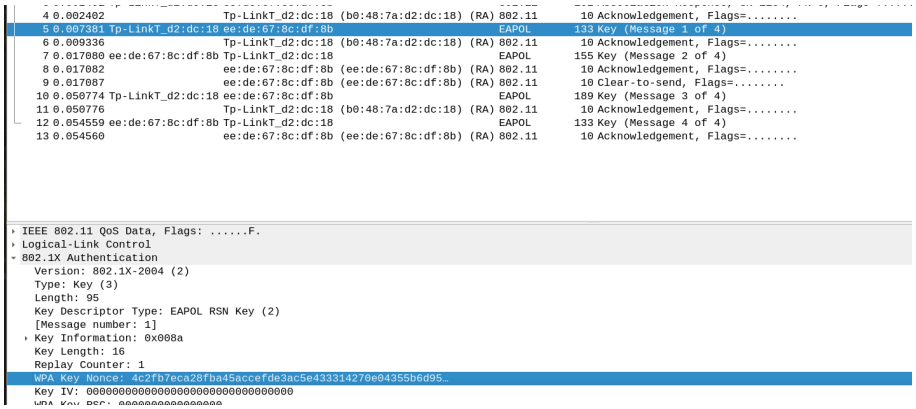

Figura 15: aNonce

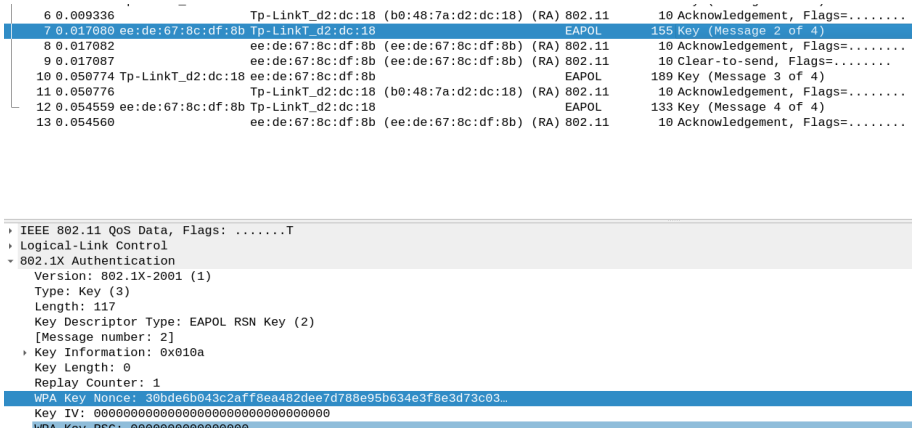

Figura 16: sNonce

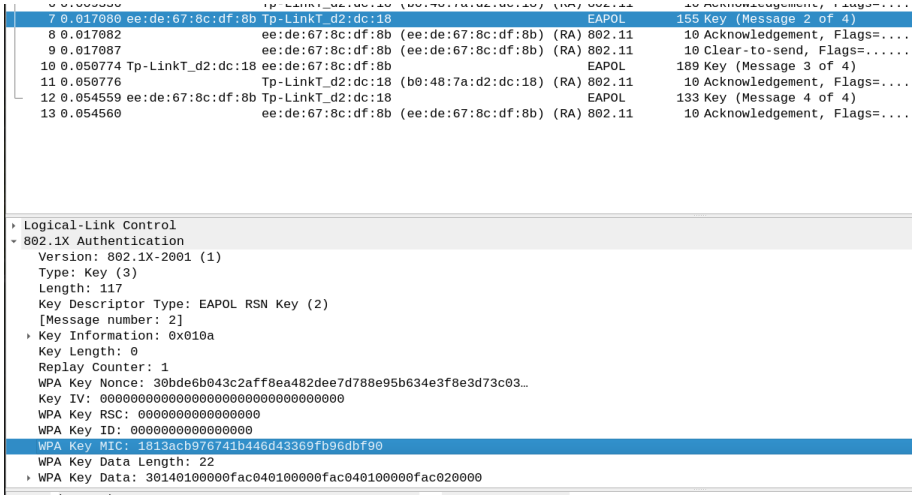

Figura 17: Key MIC 1

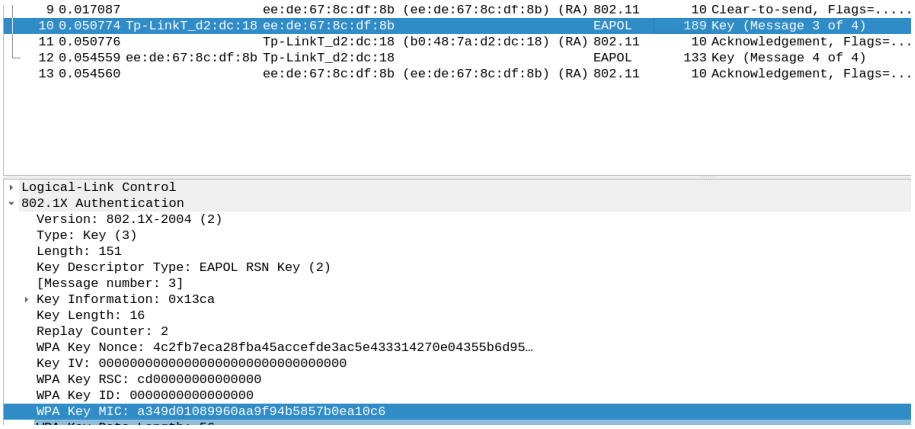

Figura 18: Key MIC 2

4.7. obtiene contraseña con pycrack

No logrado

Conclusiones y comentarios

Para concluir se comenta que hubieron percances al usar hashcat, debido al modo hash utilizado -m 22000 ya que al parecer se utiliza para redes WPA. Tambien respecto a drivers que no se encontraban actualizados en este PC referente a OpenCL se recurre a forzar el crackeo.