

# Informe Laboratorio 1

## Sección 01

Anselmo Pacheco

e-mail: [anselmo.pacheco@mail.udp.cl](mailto:anselmo.pacheco@mail.udp.cl)

30 de agosto de 2024

## Índice

<b>1. Descripción</b>	<b>2</b>
<b>2. Actividades</b>	<b>2</b>
2.1. Algoritmo de cifrado . . . . .	2
2.2. Modo stealth . . . . .	2
2.3. MitM . . . . .	3
<b>3. Desarrollo de Actividades</b>	<b>4</b>
3.1. Actividad 1 . . . . .	4
3.2. Actividad 2 . . . . .	5
3.3. Actividad 3 . . . . .	12

## 1. Descripción

1. Usted empieza a trabajar en una empresa tecnológica que se jacta de poseer sistemas que permiten identificar filtraciones de información a través de Deep Packet Inspection (DPI). A usted le han encomendado auditar si efectivamente estos sistemas son capaces de detectar las filtraciones a través de tráfico de red. Debido a que el programa ping es ampliamente utilizado desde dentro y hacia fuera de la empresa, su tarea será crear un software que permita replicar tráfico generado por el programa ping con su configuración por defecto, pero con fragmentos de información confidencial. Recuerde que al comparar tráfico real con el generado no debe gatillar alarmas. De todas formas, deberá hacer una prueba de concepto, en la cual se demuestre que al conocer el algoritmo, será fácil determinar el mensaje en claro. Para los pasos 1,2,3 indicar el texto entregado a ChatGPT y validar si el código resultante cumple con lo requerido.

## 2. Actividades

### 2.1. Algoritmo de cifrado

1. Generar un programa, en python3 utilizando chatGPT, que permita cifrar texto utilizando el algoritmo Cesar. Como parámetros de su programa deberá ingresar el string a cifrar y luego el corrimiento.

```

$ sudo python3 cesar.py "criptografia y seguridad en redes" 9
larycxpajorj h bnpdarmjm nw anmnb

```

### 2.2. Modo stealth

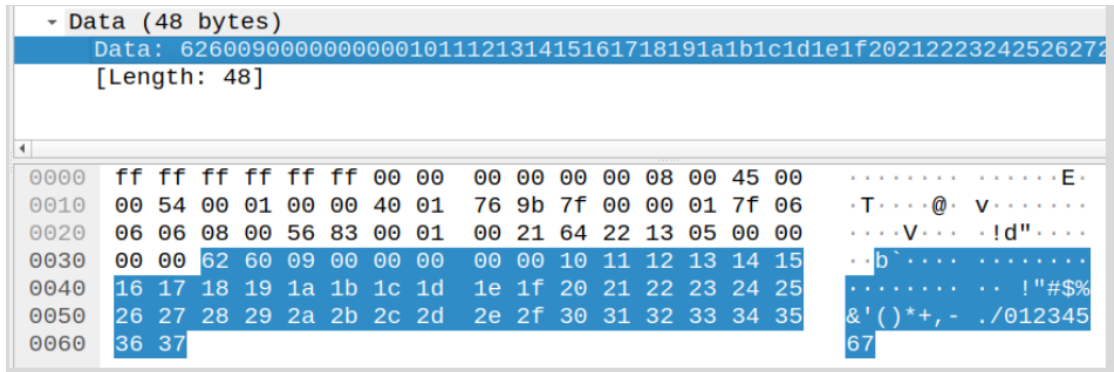
1. Generar un programa, en python3 utilizando ChatGPT, que permita enviar los caracteres del string (el del paso 1) en varios paquetes ICMP request (un caracter por paquete en el campo data de ICMP) para de esta forma no gatillar sospechas sobre la filtración de datos. Deberá mostrar los campos de un ping real previo y posterior al suyo y demostrar que su tráfico consideró todos los aspectos para pasar desapercibido.

```

$ sudo python3 pingv4.py "larycxpajorj h bnpdarmjm nw anmnb"
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.

```

El último carácter del mensaje se transmite como una b.



## 2.3. MitM

1. Generar un programa, en python3 utilizando ChatGPT, que permita obtener el mensaje transmitido en el paso2. Como no se sabe cual es el corrimiento utilizado, genere todas las combinaciones posibles e imprímalas, indicando en verde la opción más probable de ser el mensaje en claro.

```

$ sudo python3 readv2.py cesar.pcapng
0      larycxpajorj h bnpdarmjm nw anmnb
1      kzqxbwozinqi g amoczqlil mv zmlma
2      jypwavnyhmp h f zlnbypkhh lu yklklz
3      ixovzumxglog e ykmaxojgj kt xkjky
4      hwnuytlwfknd d xjlzwnifi js wjiix
5      gvmtxskvejme c wikyvmheh ir vihiw
6      fulswrjudild b vhjxulgdg hq uhghv
7      etkrvqitchkc a ugiwtkfcf gp tgfgu
8      dsjquphsbgjb z tfhvsjebe fo sfef
9      criptografia y seguridad en redes
10     bqhosnfqzehz x rdftqhczc dm qdcdr
11     apgnrmepdygy w qcespgbyb cl pcbbcq
12     zofmqldoxcfx v pbdrofaxa bk obabp
13     ynelpkcnwbew u oacqnezwz aj nazao
14     xmdkojbmadv t nzbpmdivy zi mzyzn
15     wlcjnia luzcu s myaolcxux yh lyxym
16     vkbimhzktybt r lxznkbwtw xg kxwxl
17     ujahlgysxas q kwymjavsv wf jwvwk
18     tizgkfxirwzr p jvxlizuru ve ivuvj
19     shyfjewhqvyq o iuwkhytqt ud hutui
20     rgxeidvgpuxp n htvjgxspc tc gtsth
21     qfwdhcufotwo m gsuifwrwr sb ffsrg
22     pevcbtensvn l frthevqng ra erqrf
23     odubfasdmrum k eqsgdupmp qz dqpqe
24     nctaezrcqltl j dprfctolo py cpopd
25     mbszdyqbksk i coqebnkn ox bonoc

```

Finalmente, deberá indicar 4 issues que haya tenido al lidiar con ChatGPT, netamente para reflejar cuál fue su experiencia al trabajar con esta tecnología.

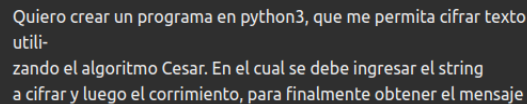
### 3. Desarrollo de Actividades

Para el desarrollo de este laboratorio se procederán a realizar una serie de actividades propuestas, entre las cuales se encuentra el crear programas en Python para desarrollar las diversas actividades propuestas.

Las actividades a realizar son:

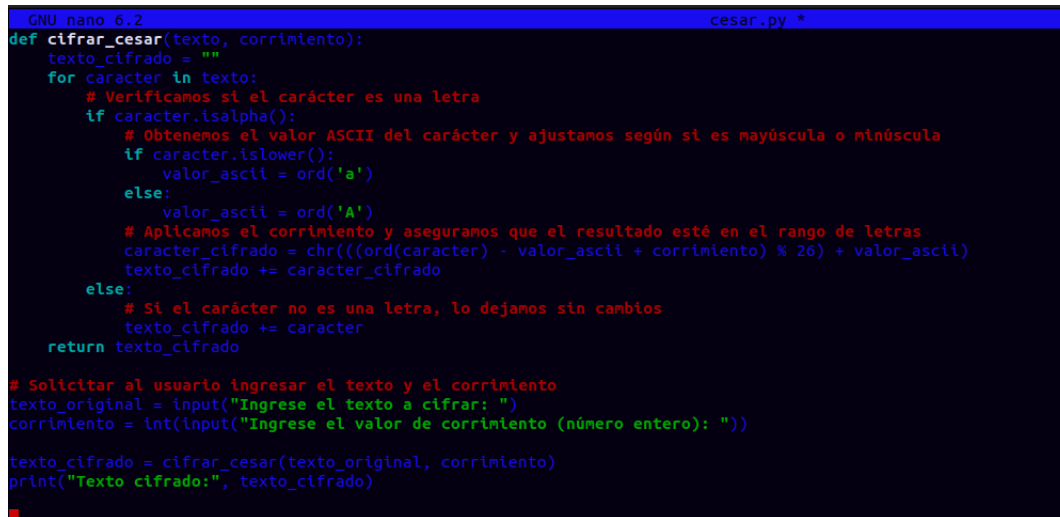
#### 3.1. Actividad 1

Para el desarrollo de este ítem se debe generar un código de cifrado en Python que permita mediante el texto ingresado y su respectivo corrimiento el cifrado de este. A continuación se procede a adjuntar el código obtenido y el resultado del cifrado:

A screenshot of a ChatGPT interface showing a user's request in Spanish to create a Python program for the Caesar cipher. The text is displayed in a dark-themed window with a light gray border.

Quiero crear un programa en python3, que me permita cifrar texto utilizando el algoritmo Cesar. En el cual se debe ingresar el string a cifrar y luego el corrimiento, para finalmente obtener el mensaje

Figura 1: Texto ChatGPT

A screenshot of a GNU nano 6.2 text editor window showing a Python script named 'cesar.py'. The code implements a Caesar cipher function and includes user input prompts and output printing. The code is color-coded with syntax highlighting.

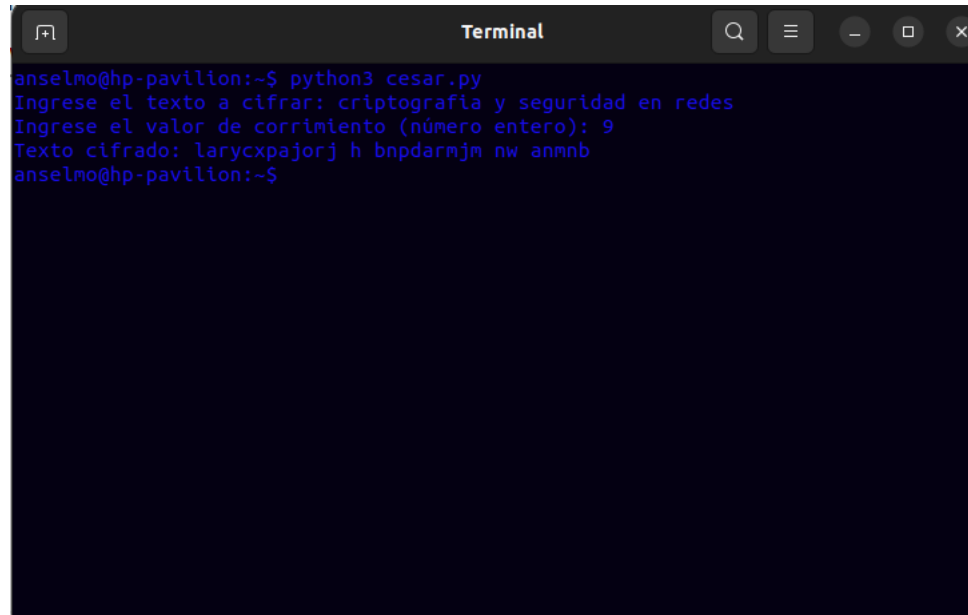
```
GNU nano 6.2 cesar.py *
def cifrar_cesar(texto, corrimiento):
    texto_cifrado = ""
    for caracter in texto:
        # Verificamos si el carácter es una letra
        if caracter.isalpha():
            # Obtenemos el valor ASCII del carácter y ajustamos según si es mayúscula o minúscula
            if caracter.islower():
                valor_ascii = ord('a')
            else:
                valor_ascii = ord('A')
            # Aplicamos el corrimiento y aseguramos que el resultado esté en el rango de letras
            caracter_cifrado = chr(((ord(caracter) - valor_ascii + corrimiento) % 26) + valor_ascii)
            texto_cifrado += caracter_cifrado
        else:
            # Si el carácter no es una letra, lo dejamos sin cambios
            texto_cifrado += caracter
    return texto_cifrado

# Solicitar al usuario ingresar el texto y el corrimiento
texto_original = input("Ingrese el texto a cifrar: ")
corrimiento = int(input("Ingrese el valor de corrimiento (número entero): "))

texto_cifrado = cifrar_cesar(texto_original, corrimiento)
print("Texto cifrado:", texto_cifrado)
```

Figura 2: Cifrado César

El resultado obtenido a partir de la ejecución de este código junto a los parámetros proporcionados se obtuvo lo siguiente:

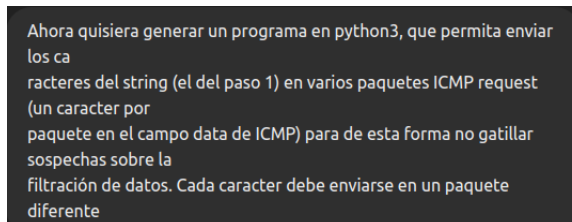


```
anselmo@hp-pavilion:~$ python3 cesar.py
Ingrese el texto a cifrar: criptografia y seguridad en redes
Ingrese el valor de corrimiento (número entero): 9
Texto cifrado: larycxpajorj h bnpdarmjm nw anmnb
anselmo@hp-pavilion:~$
```

Figura 3: Resultado Cifrado

### 3.2. Actividad 2

Luego de obtener el cifrado de "criptografia y seguridad en redes" se debe crear un programa que permita enviar los caracteres de larycxpajorj h bnpdarmjm nw anmnbco uno a uno, para cada paquete ICMP request, para evitar sospechas acerca de filtración de datos.



Ahora quisiera generar un programa en python3, que permita enviar los caracteres del string (el del paso 1) en varios paquetes ICMP request (un caracter por paquete en el campo data de ICMP) para de esta forma no gatillar sospechas sobre la filtración de datos. Cada caracter debe enviarse en un paquete diferente

Figura 4: Texto ChatGPT

Para obtener de manera precisa estos paquetes enviados se utilizó el siguiente filtro:



Figura 5: Filtro ICMP

A continuación se procede a adjuntar el código en Python proporcionado para la ejecución del programa con sus respectivas capturas en el programa Wireshark, que es utilizado como asistencia para determinar ciertos patrones de los paquetes obtenidos, permitiendo seguir de

manera más detallada el comportamiento de lo obtenido.  
El código obtenido es:

```
GNU nano 6.2 stealth.py
from scapy.all import *
import sys
import os

# Obtener el mensaje a cifrado
mensaje_cifrado = sys.argv[1]

# Generar valores aleatorios para el ID de paquete y el número de secuencia de ICMP
identificador_paquetes = os.urandom(2)
numero_secuencia = 1

# Enviar cada caracter del mensaje como un paquete ICMP de solicitud a la dirección IP de loopback
for caracter in mensaje_cifrado:
    datos_aleatorios = os.urandom(44)
    datos = datos_aleatorios + b'\x00' * (47 - len(datos_aleatorios)) + bytes(caracter, 'utf-8') + b'\x00'
    paquete_icmp = IP(dst="127.0.0.1", id=int.from_bytes(identificador_paquetes, byteorder='big'))/ICMP(id=1, seq=numero_secuencia)/Raw(load=datos)
    send(paquete_icmp)
    numero_secuencia += 1
```

Figura 6: Código Python ping.

Para la generación del tráfico ICMP, deberá basarse en los campos de un paquete generado por el programa ping basado en Ubuntu, de lo cual se obtuvo lo siguiente:

```
anselmo@hp-pavilion:~$ sudo python3 stealth.py larycxpajorj h bnpdarmjm nw anmnb
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
Sent 1 packets.
anselmo@hp-pavilion:~$
```

Figura 7: Envío de paquetes.

Al analizar los paquetes filtrados en wireshark se obtuvo la siguiente secuencia de paquetes:

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0x6c28 [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 1 (0x0001)						
Sequence Number (LE): 256 (0x0100)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0.. @.?.....			
0020	7f 00 00 01 08 00 6c 28	00 01 00 01 a2 c1 d6 e7	.....l.....			
0030	a0 99 f9 cf d2 c6 fc fd	12 85 f1 9a 42 27 9e ac	.....M.....B'..			
0040	17 e9 20 54 17 37 6c fc	6b 3d f8 d5 88 c2 af 2b	..T.7l..k=.....+			
0050	27 9f 41 59 80 72 81 70	00 00 00 6c 00	'..AY..r.p...l..			

Figura 8: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0x274f [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 2 (0x0002)						
Sequence Number (LE): 512 (0x0200)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0.. @.?.....			
0020	7f 00 00 01 08 00 27 4f	00 01 00 02 3e 06 2b 4e	.....'O.....>+N			
0030	41 89 f0 52 91 38 67 bd	51 14 cf 4a 52 7b cd a5	A..R.8g. Q..JR{..			
0040	e4 01 e0 b9 ba 7b f3 f6	03 7e e0 e5 4b 89 1a 3c	.....{...~..K...<			
0050	1b d3 15 06 62 5d ab 17	00 00 00 61 00	....b]... ..a..			

Figura 9: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0x9451 [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 3 (0x0003)						
Sequence Number (LE): 768 (0x0300)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?...			
0020	7f 00 00 01 08 00 94 51	00 01 00 03 28 2d 49 6d	.....Q....(-Im			
0030	5e 57 c3 83 dd 2c 9d e4	76 e2 c0 bf fd b1 cb 2d	^W....,..v.....			
0040	01 21 91 fb 30 80 cc 75	31 45 dd 95 55 2b 4b 7d	..!...0...u 1E...U+K}			
0050	2b a9 68 9c c5 ad ba a5	00 00 00 72 00	+..h.....r..			

Figura 10: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0xe21e [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 4 (0x0004)						
Sequence Number (LE): 1024 (0x0400)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?...			
0020	7f 00 00 01 08 00 e2 1e	00 01 00 04 44 80 c6 66	.....D...f			
0030	d7 a3 20 3e 46 e4 a8 e4	cd 84 3c f5 7d 18 57 e9	...>F....<..}..W..			
0040	2c 80 e8 ba c4 6b 0a 54	19 1a 32 27 94 8f 04 9a	,....k..T..2'....			
0050	47 8f 89 48 63 e1 46 36	00 00 00 79 00	G..Hc.F6...y..			

Figura 11: Resultado Cifrado



No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0xb0de [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 5 (0x0005)						
Sequence Number (LE): 1280 (0x0500)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 86 dd 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?...			
0020	7f 00 00 01 08 00 b0 de	00 01 00 05 70 ac 6f ca	.....p.o.			
0030	c9 47 9c 62 ac 46 82 a5	9b 43 9c 1d d3 1f 42 52	.G.b.F...C...BR			
0040	18 75 28 37 24 ee c4 1f	55 ab 64 c9 3f 0d 44 e8	.u(7\$...U.d.?..D			
0050	0a 8e d9 f0 91 07 a7 2c	00 00 00 63 00	.....,....c.			

Figura 12: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0x4c61 [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 6 (0x0006)						
Sequence Number (LE): 1536 (0x0600)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?...			
0020	7f 00 00 01 08 00 4c 61	00 01 00 06 70 e4 5d 2f	.....La...[p:]			
0030	93 6d 03 cb 37 21 e1 e8	d0 85 1c 06 a9 72 14 62	.m..7!...r..b			
0040	d1 77 ae 30 55 77 65 8b	c7 1a 2d 98 88 d6 54 6c	.w..0Uwe...TL			
0050	00 1a be f4 36 52 20 66	00 00 00 78 00	...6R f...x.			

Figura 13: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0xdba3 [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 7 (0x0007)						
Sequence Number (LE): 1792 (0x0700)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?..			
0020	7f 00 00 01 08 00 db a3	00 01 00 07 85 22 86 4a	.....N.....J			
0030	3c 29 be d6 4c ad 0c 5b	e8 87 86 c1 47 75 86 34	<)...L...[...Gu-4			
0040	21 06 df c9 78 ac 81 98	49 cc c6 bb 17 1b bc 39	!...X...I.....9			
0050	a5 37 6f b2 d2 e0 08 be	00 00 00 70 00	-70.....p.			

Figura 14: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0x4eef [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 8 (0x0008)						
Sequence Number (LE): 2048 (0x0800)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?..			
0020	7f 00 00 01 08 00 4e ef	00 01 00 08 51 c8 8d be	.....N.....Q			
0030	4b 0c 91 bf 83 a3 cb cd	c2 1a 32 31 08 b7 8c 7f	K.....21...			
0040	68 20 da 04 76 3b 68 cd	8f cd d2 c0 9d d9 02 a7	h...V;h.....			
0050	d0 41 9f 43 44 16 3b 87	00 00 00 61 00	-A-CD;...a.			

Figura 15: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0x720c [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 9 (0x0009)						
Sequence Number (LE): 2304 (0x0900)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?....			
0020	7f 00 00 01 08 00 72 0c	00 01 00 09 3d 1b c4 d9	.....r.....=			
0030	cd 39 39 73 7e a5 b2 cf	2c 32 d2 f6 98 0f db 3f	..99s~....,2....?			
0040	13 48 b3 d4 b0 18 e9 47	3f 93 15 6c be a7 3d 18	..H....G?..l..=			
0050	89 e4 f8 a9 ca ad da 76	00 00 00 6a 00	.....v...j..			

Figura 16: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0xb808 [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 10 (0x000a)						
Sequence Number (LE): 2560 (0x0a00)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?....			
0020	7f 00 00 01 08 00 b8 08	00 01 00 0a 04 42 1c 95	.....B.....			
0030	80 a3 bb ba bc 38 8b 4f	1e b2 17 e6 5d f2 16 99	.....8 0 ....]			
0040	3c 9e 38 dd 5c 6a bb 65	fb 1a cc 6a 58 b2 6c ab	<.8.\j.e...jX.l.			
0050	0e 78 7f 4f 40 48 07 5d	00 00 00 6f 00	.x.0@H.]...o.			

Figura 17: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0x6c90 [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 11 (0x000b)						
Sequence Number (LE): 2816 (0x0b00)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?_...			
0020	7f 00 00 01 08 00 0c 00	00 01 00 0b d1 1d 76 fb	.....l.....v...			
0030	f0 9e 78 1b a6 09 87 bd	5b bf a8 5e 62 ed c9 c4	..x.....[...Ab...			
0040	2e a7 f6 2d 3f eb 9d 9e	60 5a 92 fb 6a 03 57 8c	...-?...Z...j..W...			
0050	31 d9 4c 3a 69 8b dc a2	00 00 00 72 00	1..L:1.....r...			

Figura 18: Resultado Cifrado

No.	Time	Source	Destination	Protocol	Length	Info
45	6.194877803	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=1/256, ttl=64 (no response found!)
49	6.242560266	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=2/512, ttl=64 (no response found!)
50	6.290618980	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=3/768, ttl=64 (no response found!)
51	6.356691731	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=4/1024, ttl=64 (no response found!)
52	6.412393130	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=5/1280, ttl=64 (no response found!)
53	6.474675305	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=6/1536, ttl=64 (no response found!)
54	6.522365171	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=7/1792, ttl=64 (no response found!)
55	6.562463968	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=8/2048, ttl=64 (no response found!)
56	6.606393222	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=9/2304, ttl=64 (no response found!)
57	6.646370699	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=10/2560, ttl=64 (no response found!)
58	6.694659129	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=11/2816, ttl=64 (no response found!)
59	6.726821337	127.0.0.1	127.0.0.1	ICMP	93	Echo (ping) request id=0x0001, seq=12/3072, ttl=64 (no response found!)
Internet Control Message Protocol						
Type: 8 (Echo (ping) request)						
Code: 0						
Checksum: 0x20a1 [correct]						
[Checksum Status: Good]						
Identifier (BE): 1 (0x0001)						
Identifier (LE): 256 (0x0100)						
Sequence Number (BE): 12 (0x000c)						
Sequence Number (LE): 3072 (0x0c00)						
[No response seen]						
Data (49 bytes)						
0000	00 02 03 04 00 06 00 00	00 00 00 00 00 00 08 00	.....			
0010	45 00 00 4d 3d 4f 00 00	40 01 3f 5f 7f 00 00 01	E..M=0..@.?_...			
0020	7f 00 00 01 08 00 20 a1	00 01 00 0c e9 9c 73 f8	.....s.....			
0030	0a 59 21 cb 00 03 39 b0	75 ea 0b ab 23 f9 74 50	..Y!...9..u...#.tP			
0040	98 6b b2 d5 a2 b4 a0 42	72 2b 5f 9e ef b4 6a c4	..k.....B r+...j..			
0050	dd 9f b2 71 c4 06 eb 07	00 00 00 6a 00	...q.....j...			

Figura 19: Resultado Cifrado

De estas capturas adjuntas se puede apreciar que se mantiene una secuencia en los datos obtenidos a partir del filtrado.

### 3.3. Actividad 3

Para esta actividad se solicita el consultar por un programa en Python que permita obtener el mensaje "larycxpajorj h bnpdarmjm nw anmnb", actuando como Man in the middle,

y de esta manera saber cual es el corrimiento utilizado para el cifrado del mensaje. Se deben utilizar todas las combinaciones posibles y resaltar el posible resultado del mensaje original.

A continuación quisiera un programa en Python que permita obtener el mensaje "larycxpajorj h bnpdarmjm nw anmnb", actuando como Man in the middle, y de esta manera saber cual es el corrimiento utilizado para el cifrado del mensaje. Se deben utilizar todas las combinaciones posibles y resaltar el posible resultado del mensaje original.

Figura 20: Texto ChatGPT

```
GNU nano 6.2 mitm.py
from scapy.all import *
from unicode import unicode

# Función para cifrar el texto utilizando el cifrado César
def cifrar_cesar(texto, clave):
    texto_cifrado = ""
    for caracter in texto:
        if caracter.isalpha():
            es_mayuscula = caracter.isupper()
            caracter = caracter.lower()
            codigo_ascii = ord(caracter)
            codigo_cifrado = (codigo_ascii + clave - 97) % 26 + 97
            texto_cifrado += chr(codigo_cifrado).upper() if es_mayuscula else chr(codigo_cifrado)
        else:
            texto_cifrado += caracter
    return texto_cifrado

# Abrir el archivo de captura
paquetes = rdpcap("cesarr.pcapng")

# Inicializar una cadena vacía para almacenar los caracteres
caracteres_concatenados = ""

# Iterar sobre los paquetes y buscar los paquetes ICMP con dirección de destino 127.0.0.1
for paquete in paquetes:
    if ICMP in paquete and paquete[IP].dst == "127.0.0.1":
        # Obtener el contenido del campo de datos ICMP
        datos = paquete[ICMP].load
        # Acceder al caracter en la posición 45 del contenido
        caracter_45 = chr(datos[44])
        # Saltar el primer paquete que contiene una "i"
        if caracter_45 == "i" and not caracteres_concatenados:
            continue
        # Concatenar el caracter a la cadena
```

Figura 21: Código Man in the middle.(Parte 1)

```
GNU nano 6.2                               mltm.py
    if caracter_45 == "l" and not caracteres_concatenados:
        continue
    # Concatenar el caracter a la cadena
    caracteres_concatenados += caracter_45

# Imprimir la cadena original
print(f"Texto original: {caracteres_concatenados}")

# Inicializar una lista vacía para almacenar los textos cifrados
textos_cifrados = []

# Generar todas las combinaciones posibles de cifrado César y almacenarlas en la lista
for clave in range(26):
    texto_cifrado = cifrar_cesar(caracteres_concatenados, clave)
    textos_cifrados.append(texto_cifrado)

# Función para leer el diccionario de palabras
def leer_diccionario():
    palabras = []
    archivos = os.listdir("diccionario")
    for archivo in archivos:
        with open(f"diccionario/{archivo}", "r") as f:
            for linea in f:
                palabra = linea.strip()
                palabras.append(palabra)
                if any(c in "áéíóúüÄËÎÖÜ" for c in palabra):
                    palabras.append(unidecode(palabra))
    return palabras

# Función para encontrar el índice del texto cifrado con más coincidencias en el diccionario
def encontrar_indice_mas_coincidencias(textos_cifrados, palabras_diccionario):
    max_coincidencias = 0
    indice_mas_coincidencias = 0
    for i, texto_cifrado in enumerate(textos_cifrados):
```

Figura 22: Código Man in the middle.(Parte 2)

```

GNU nano 6.2
Puede pegar la imagen desde el portapapeles.

    for linea in f:
        palabra = linea.strip()
        palabras.append(palabra)
        if any(c in "áéíóúüÄËÎÖÜÜ" for c in palabra):
            palabras.append(unidecode(palabra))

    return palabras

# Función para encontrar el índice del texto cifrado con más coincidencias en el diccionario
def encontrar_indice_mas_coincidencias(textos_cifrados, palabras_diccionario):
    max_coincidencias = 0
    indice_mas_coincidencias = 0
    for i, texto_cifrado in enumerate(textos_cifrados):
        coincidencias = 0
        for palabra in palabras_diccionario:
            if palabra in texto_cifrado:
                coincidencias += 1
        if coincidencias > max_coincidencias:
            max_coincidencias = coincidencias
            indice_mas_coincidencias = i
    return indice_mas_coincidencias

# Leer el diccionario de palabras
diccionario = leer_diccionario()

# Encontrar el índice del texto cifrado con más coincidencias en el diccionario
indice_max_coincidencias = encontrar_indice_mas_coincidencias(textos_cifrados, diccionario)

# Imprimir todos los textos cifrados, resaltando en verde el que tiene más coincidencias
for i, texto_cifrado in enumerate(textos_cifrados):
    if i == indice_max_coincidencias:
        print('\033[92m' + texto_cifrado + '\033[0m')
    else:
        print(texto_cifrado)

```

Figura 23: Código Man in the middle.(Parte 3)

Para este punto, junto al desarrollo del código se debe crear un diccionario que contenga diversas palabras entre las cuales esté contenida la palabra que se pretende descifrar a través de las combinaciones al momento de recorrerlo.



Figura 24: Diccionario (A-Z)

A partir de lo mencionado se pudo obtener el sig. resultado:



```

anselmo@hp-pavilion:~$ sudo python3 mitm.py cesarr.pcapng
String original: larycxpajorj h bnpdarmjm nw anmnb
larycxpajorj h bnpdarmjm nw anmnb
mbszdyqbksk i coqebnskn ox bonoc
nctaezrcqltl j dprfctolo py cpopd
odubfasdmrun k eqsgdupmp qz dqpqe
pevcgbtensvn l frthevqng ra erqrf
qfwdhucufotwo m gsulfrwor sb fersg
rgxeldvgpuxp n htvjgxspz tc gtsth
shyffjewhqvyq o iuwkhytqt ud hutui
tizgkfxirwzr p jvxlizuru ve ivuvj
ujahlgyjsxas q kwmjavsv wf jwvww
vkbimhzktybt r lxznkbtw xg kxwxl
wlcjnluzcu s myaolcxux yh lyxym
xmdkojbmadv t nzbpmdivy zi mzyzn
ynelpkcwbew u oacqnezvz aj nazao
zofnqldoxcfx v pbdrofaxa bk obabp
apgnrmepdygy w qcespgbyb cl pcbbcq
bqhoshnfqzehz x rdtqhczc dm qdcdr
criptografia y seguridad en redes
dsjquphsbgjb z tfhvsjebe fo sfeft
etkrvgitchkc a ugiwtkfcf gp tgfgu
fulswrjudild b vhjxulgdg hq uhghv
gvmtxskvejme c wikyvmeh ir vihiw
hwnuytlwfkf d xjllznifi js wjiix
lxovzumxglog e yknaxojgj kt xkjky
jypwavyhmpf f zlnbyphkh lu yklkz
kzqxbwozinqi g amoczqlil mv znlma
anselmo@hp-pavilion:~$

```

Figura 25: Resultado corrimiento.

## Conclusiones y Comentarios

De esta experiencia se puede concluir que se desarrollaron los objetivos propuestos, entre los cuales se encontraban el generar programas en python3, con el propósito de analizar ciertos comportamientos, junto al software Wireshark que permitió apreciar de mejor manera los paquetes relacionados al protocolo ICMP.

Entre las posibles problemas presentados se puede mencionar ciertas dificultades que añadieron una complejidad al desarrollo de este laboratorio, entre las cuales se encuentran la poca información presente en internet o más bien no se encuentran resultados específicos, ya que, en varias ocasiones al momento de solicitar en foros no se encontraba de manera precisa lo que se buscaba.

Otra dificultad presentada en el transcurso de esta actividad se debe mencionar la importancia de saber comunicarse al utilizar herramientas como chat-gpt, ya que, al realizar ciertas consultas advertía acerca de la violación de privacidad o de leyes. Mencionando que estas solicitudes podrían ser con fines no éticos o ilegales. Es importante mencionar que este problema fue el que más dificultó la realización del laboratorio.

Junto al punto anterior se presenta el problema de no tener claridad total de lo que requería para el desarrollo de ciertas actividades, lo que generó cierto retraso en la realización de la experiencia.

Finalmente se deben mencionar ciertos aspectos técnicos de carácter de menor complejidad en cuanto a la resolución de estos, como es el caso de ciertos permisos de usuario o instalación de paquetes para la ejecución de los programas generados.

Sin embargo se enfrentaron estos problemas con la misión de llevar a cabo la experiencia en



su totalidad de la mejor manera posible.