SecAdmin 2024
CTF Writeup

@oreos

Sangre Binaria

Una mancha de sangre en una camiseta desechada ha llevado a tu agencia a un descubrimiento sorprendente. El análisis del ADN revela un mensaje oculto codificado en la secuencia genética, incluyendo la palabra "unit01". Sospechas que esto pertenece a una unidad cibernética, un proyecto ultrasecreto que escapó del Área 51.

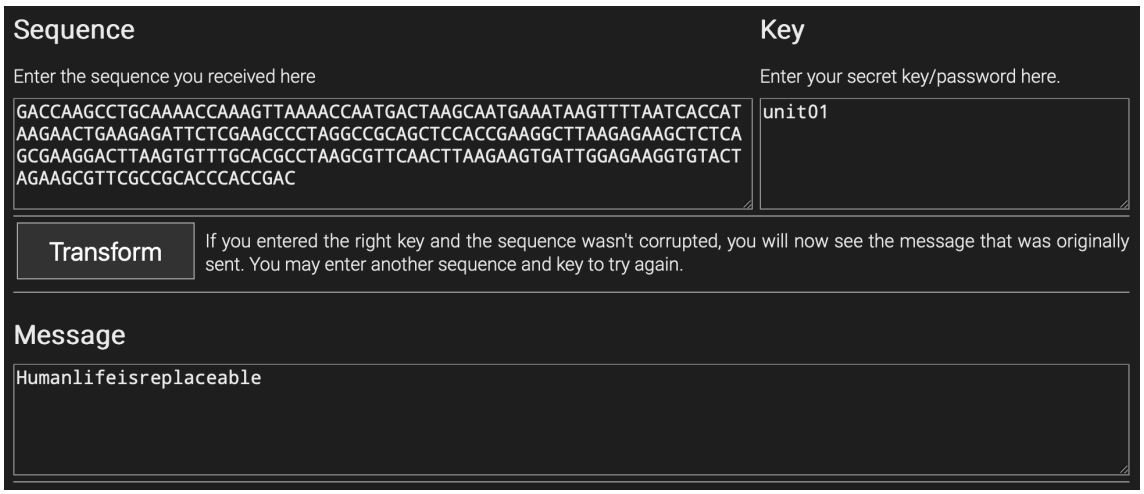
Tu misión: Descifrar el código genético y desbloquear la flag, revelando información crucial sobre esta unidad cibernética renegada.

GACCAAGCCTGCAAAACCAAAGTTAAAACCAATGACTAAGCAATGAAATAAGTTTTAATCACCAT
AAGAACTGAAGAGATTCTCGAAGCCCTAGGCCGCAGCTCCACCGAAGGCTTAAGAGAAGCTCT
CAGCGAAGGACTTAAGTGTTTGCACGCCTAAGCGTTCAACTTAAGAAGTGATTGGAGAAGGTG
TACTAGAAGCGTTCGCCGCACCCACCGAC

Solución:

Usamos la siguiente herramienta para descifrar el mensaje:

https://2016.igem.org/Team:Groningen/Decoding



Flag: Humanlifeisreplaceable

El Legado del Código

En una época olvidada, un antiguo programador dejó un mensaje secreto dentro de un código. Este mensaje, un legado de sabiduría y conocimiento, se ha perdido en el tiempo. Ahora, tu misión es recuperar este legado, analizando el código y descifrando sus enigmas. Debes desenmascarar los secretos ocultos, desentrañar las claves del código y desvelar la verdad que se oculta dentro. El destino de este conocimiento depende de ti.

legacy.ll

Solución:

1. *Compilamos el fuente de LLVM a código ensamblador:*

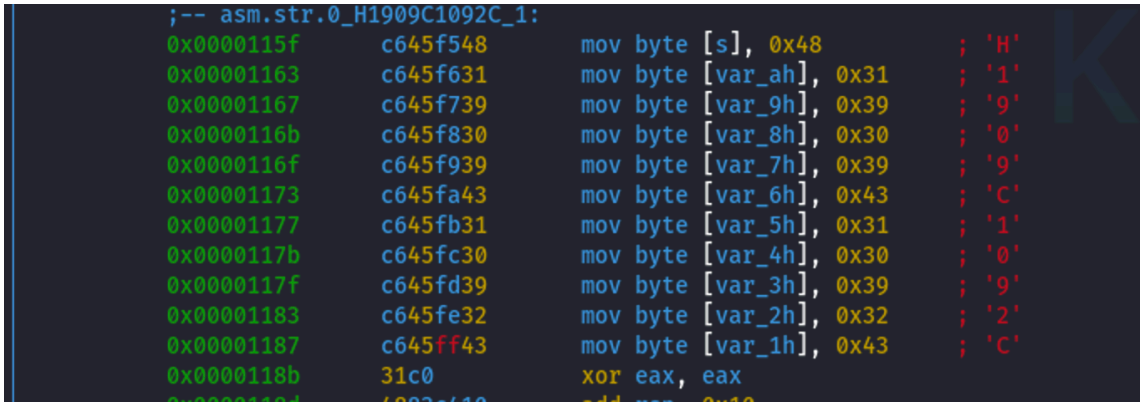
$ llc -filetype=asm legacy.ll -o legacy.s

2. *Compilamos el código ensamblador a binario ELF:*

$ gcc legacy.s -o legacy

3. *Abrimos el binario con r2 y obtenemos la flag:*

$ r2 -AAA legacy



Flag: H1909C1092C

El flag vacío

Todo lo que necesitas para superar este reto está en el archivo Flag.. Pero como siempre:

“..Lo esencial es invisible a los ojos”

Flag.txt

Solución:

Usamos el siguiente script:

```
from pwn import *

with open("Flag.txt", "rb") as bin_file:
   for _ in range(88):
      data = bytearray(bin_file.readline())
      data = data.replace(b'\x09', b'0')
      data = data.replace(b'\x20', b'1')
      data = data.decode("ascii")
      if unbits(data) == b'\x80': # ignored single 1's
         continue
      print(data,end = ' ')
```

Obtenemos la cadena binaria:

01010011011001010110001101000001011001000110110101101001011011100111
01101100011011110010110001001100101011100100111011101100001011100100
10001010110010001101001011101000110100101101111011011100111101

Convertimos desde binario:

SecAdmin{cyberwarEdition}

La profesora en apuro

Un conocido grupo de ransomware, R7g6B54, ha cifrado el ordenador de tu profesora. Sin embargo, parece que este grupo no es tan hábil como aparenta. Hace poco, intentaron atacar al equipo de un renombrado criptógrafo, pero este logró descifrar sus archivos y recuperar la clave en tan solo 15 minutos.

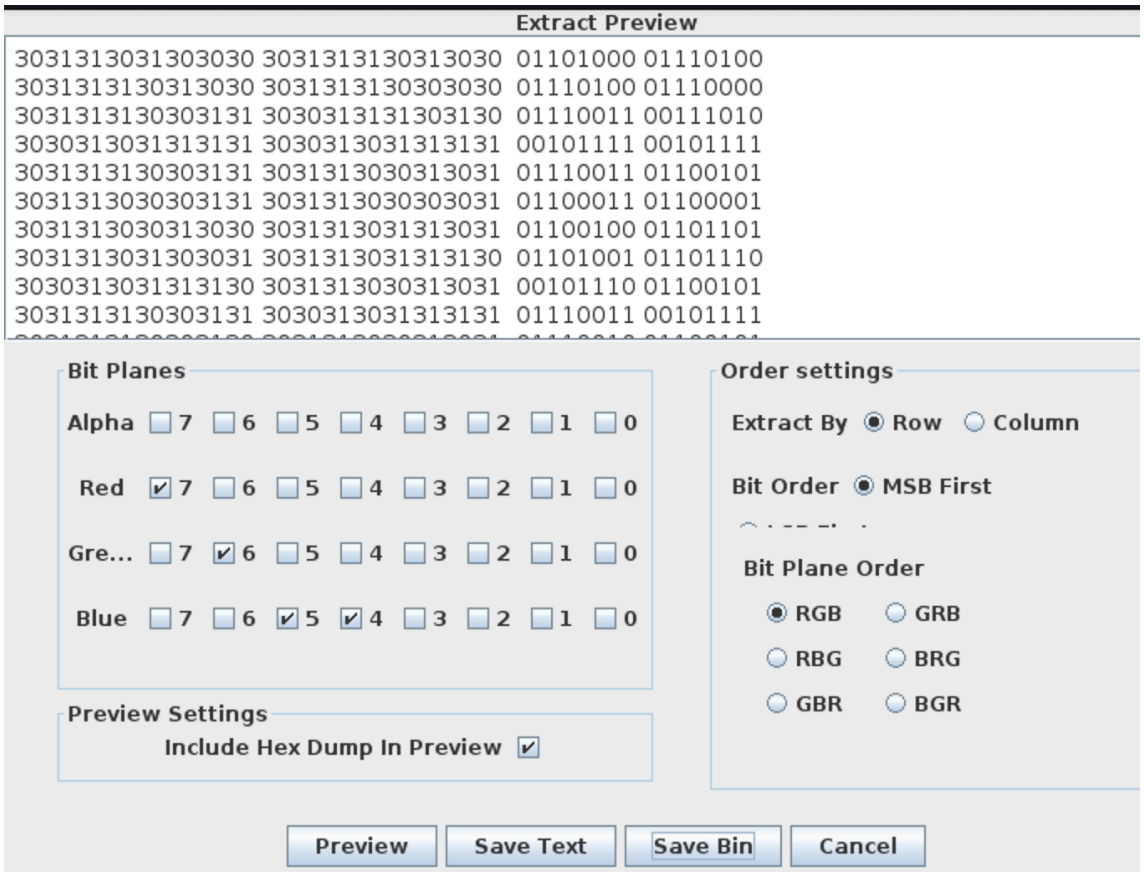
El problema es que el criptógrafo, aunque mencionó su logro en su blog, no explicó los detalles técnicos de cómo lo logró. Ahora, todo lo que tienes para trabajar es:

Una Captura del mensaje de rescate del grupo.
El nombre del grupo: R7g6B54.
¿Serás capaz de encontrar la clave y liberar los archivos de tu profesora? ¡El tiempo corre!

ransomware.png
Mensaje_con_flag.pdf.DBK

Solución:

Abrimos la imagen 'ransomware.png' con stegsolve y extramos datos de la imagen usando los siguientes planos de bits: R 7 G 6 B 54



Obtenemos el siguiente binario:

01101000011101000111010001110000011100110011101000101111001011110111001101100101011000110110000010110010001101101011010010110111000101110011001010111001100101111011100100110010101110011011000110111010101100101001010111001100101111011100100110010101110011011000110111010101100101001010111001100101111011100001110100

En texto:

https://secadmin.es/rescue.txt

Nos indica que accedamos como html:

https://secadmin.es/rescue.html

Obtenemos una clave:
41f4f4cfe129b4f353bb23ac5260846a

Y una cadena Base64:
TmVjZXNpdGFyJUMzJUExcyBsYSB2YWN1bmEgZGlzcG9uaWJsZSBlbiBodHRwczovL3NlY2FkbWluLmVzL3ZhY3VuYS56aXA=, q
Que se traduce como:
Necesitar%C3%A1s la vacuna disponible en https://secadmin.es/vacuna.zip

Descargamos la vacuna, un fichero exe, compilado con pyinstaller. Creamos un script de Python para descifrar el fichero PDF usando la clave antes obtenida:

```
from Crypto import Random
from Crypto.Cipher import AES

def pad(s):
    return s + b"\0" * (AES.block_size - len(s) % AES.block_size)

def encrypt(message, key, key_size=256):
    message = pad(message)
    iv = Random.new().read(AES.block_size)
    cipher = AES.new(key, AES.MODE_CBC, iv)
    return iv + cipher.encrypt(message)

def decrypt(ciphertext, key):
    iv = ciphertext[:AES.block_size]
    cipher = AES.new(key, AES.MODE_CBC, iv)
    plaintext = cipher.decrypt(ciphertext[AES.block_size:])
    return plaintext.rstrip(b"\0")

def encrypt_file(file_name, key):
    with open(file_name, 'rb') as fo:
```

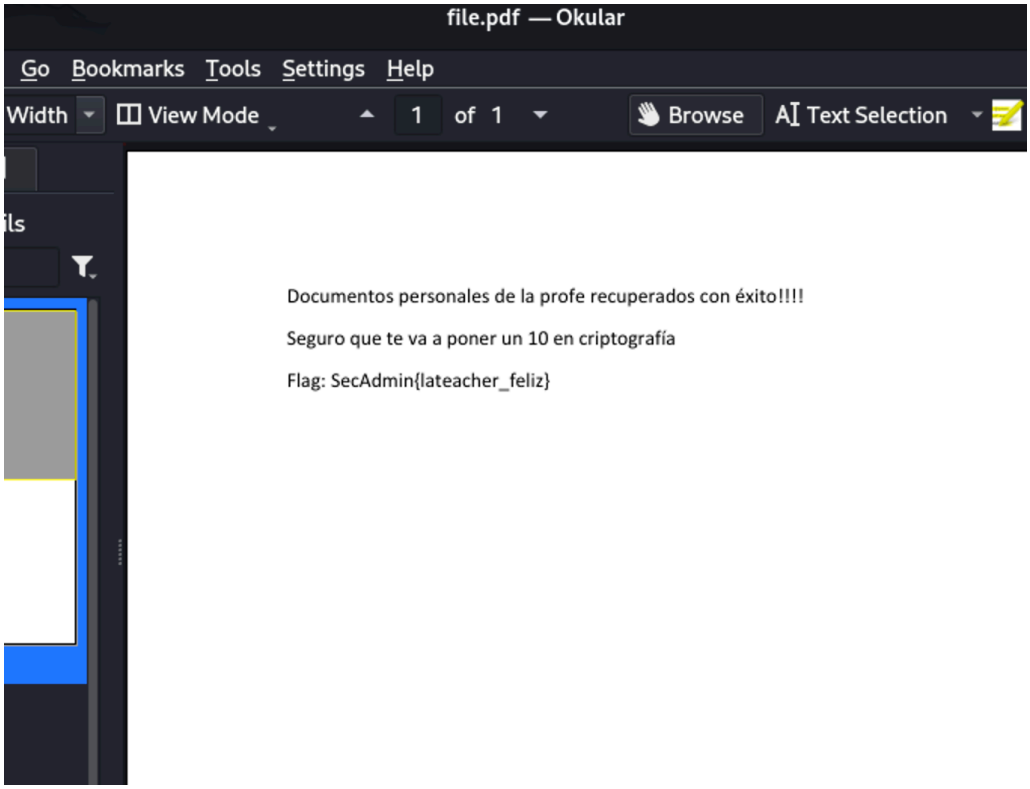
```
        plaintext = fo.read()
    enc = encrypt(plaintext, key)
    with open(file_name + ".enc", 'wb') as fo:
        fo.write(enc)

def decrypt_file(file_name, key):
    with open(file_name, 'rb') as fo:
        ciphertext = fo.read()
    dec = decrypt(ciphertext, key)
    with open(file_name[:-4], 'wb') as fo:
        fo.write(dec)


key =
b'\xbf\xc0\x85)\x10nc\x94\x02)j\xdf\xcb\xc4\x94\x9d(\x9e[EX\xc8\xd5\xbfI{\xa2$\x05
(\xd5\x18'
key = b'\x41\xf4\xf4\xcf\xe1\x29\xb4\xf3\x53\xbb\x23\xac\x52\x60\x84\x6a'
key = b'41f4f4cfe129b4f353bb23ac5260846a'

#encrypt_file('to_enc.txt', key)
decrypt_file('text.enc', key)
```

Abrimos el pdf generado:



Flag: SecAdmin{lateacher_feliz}

El ovni de Íker

Al parecer la Nave del misterio encontró por fin un vídeo de un ovni auténtico, pero los “hombres de negro” han modificado el archivo al vuelo.

Ayuda a Iker a poder ver su vídeo y tú, ¡obtén la flag!

topsecret_-_.zip

Solución:

Descargamos el fichero Léeme y obtenemos la clave:

$ unzip topsecret_-_ctf.zip Léeme.txt
inflating: Léeme.txt
$ cat Léeme.txt
#SecAdmin2024

La clave: #SecAdmin2024

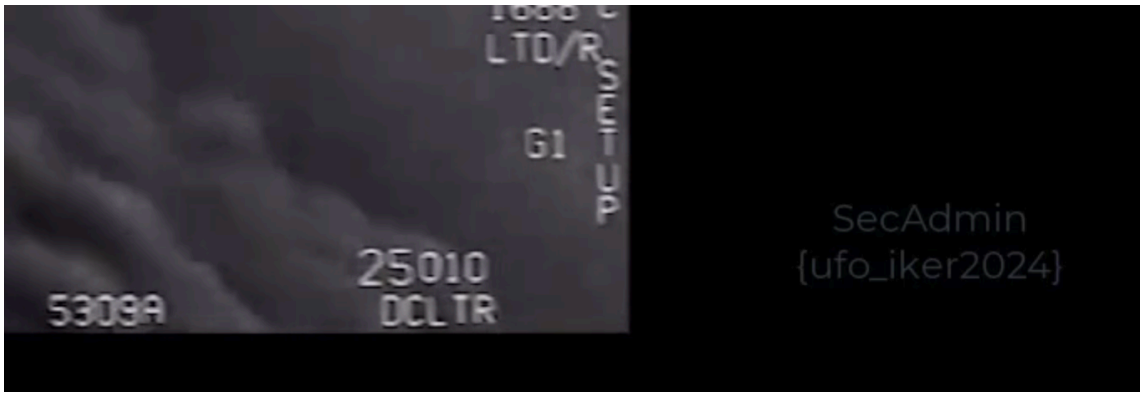
Descomprimimos el fichero usando la clave, y arreglamos la cabecera del fichero mp4:

$ cat checksum.txt
D08BE3EEB65DC6DADE0EBB740316088D

$ hexdump -C topsecret.mp4.orig | head -n1; md5sum topsecret.mp4.orig
00000000 23 45 76 21 66 74 79 70 6d 70 34 32 00 00 00 00 |#Ev!ftypmp42....|
161017a3dc288bb5f9c1313091350331 topsecret.mp4.orig

$ hexdump -C topsecret.mp4 | head -n1; md5sum topsecret.mp4
00000000 00 00 00 18 66 74 79 70 6d 70 34 32 00 00 00 00 |....ftypmp42....|
d08be3eeb65dc6dade0ebb740316088d topsecret.mp4

Obtenemos la flag en uno de los frames del fichero:



Flag: SecAdmin{ufo_iker2024}

El baile de bits del espía doble

El CNI, logró interceptar el tráfico de un agente doble muy conocido por hacer ininteligible la información que envía al gobierno para el que trabaja. El Centro Criptográfico no es capaz de leer ni dar sentido a la información interceptada.

Le pasamos las Trazas capturadas con la finalidad de que pueda acceder a algún tipo de información útil ya que creemos que puede estar pasando información que afecta a la seguridad nacional.

¡Buena suerte!

¡Actualización! Hemos detectado que hay problemas para descifrar la clave TLS, por lo que le facilitamos la URL que se tuvo que encontrar al examinar el paquete: https://pastebin.com/h8bKr9JX

trafico_home.pcapng

Solución:

Obtenemos una url de pastebin, https://pastebin.com/h8bKr9JX en la captura de tráfico:

```
1. National Information CNY COD1337
2. hxxtps://mega.nz/folder/0OJhjLZa#-Wbr0LDawNHztjHa3Z0WoA
3. #s3cADm1n
```

Accedemos al enlace de mega.nz y descargamos el contenido.

- *Rockyou.rar*
- *SecAdmin2024*

1. *Crackeamos el fichero rar usando JohnTheRipper:*

$ rar2john Rockyou.rar > hash
$ john hash --wordlist=../rockyou.txt
$ $ john hash --show
Rockyou.rar:123mango

2. *Descomprimimos el fichero rar:*

$ unrar e Rockyou.rar

UNRAR 7.10 beta 1 freeware Copyright (c) 1993-2024 Alexander Roshal
Password: 123mango
Extracting Rockyou.png OK
All OK

3. *Arreglamos la cabecera del fichero png extraído:*

$ file Rockyou.png
Rockyou.png: data

$ hexdump -C Rockyou.png | head -n3
00000000 8e 55 53 4c 0d 0a 1a 0a 00 00 00 0d 49 48 44 52 |.USL........IHDR|
00000010 00 00 02 6f 00 00 01 b0 08 02 00 00 00 70 d4 15 |...o.........p..|
00000020 b8 00 00 00 19 74 45 58 74 53 6f 66 74 77 61 72 |.....tEXtSoftwar|

52 61 72 21 1A 07 01 00	Rar!SUBBELSOHNUL	0	rar	Roshal ARchive compressed archive v5.00 onwards[24]
7F 45 4C 46	DELELF	0	none, .axf, .bin, .elf, .o, .out, .prx, .puff, .ko, .mod, .so	Executable and Linkable Format[25]
89 50 4E 47 0D 0A 1A 0A	‰PNGCRLFSUBLF	0	png	Image encoded in the Portable Network Graphics format[26]
0E 03 13 01	ENQETXDC3SOH	0	hdf4 h4	Data stored in version 4 of the Hierarchical Data Format.
89 48 44 46 0D 0A 1A 0A	‰HDFCRLFSUBLF	0, 512, 1024, 2048, ...	hdf5 h5	Data stored in version 5 of the Hierarchical Data Format.
C9	É	0	com	CP/M 3 and higher with overlays[27]

$ hexdump -C Rockyou.png | head -n3
00000000 89 50 4e 47 0d 0a 1a 0a 00 00 00 0d 49 48 44 52 |.PNG........IHDR|
00000010 00 00 02 6f 00 00 01 b0 08 02 00 00 00 70 d4 15 |...o.........p..|
00000020 b8 00 00 00 19 74 45 58 74 53 6f 66 74 77 61 72 |.....tEXtSoftwar|

$ file Rockyou.png
Rockyou.png: PNG image data, 623 x 432, 8-bit/color RGB, non-interlaced

4. *Abrimos la imagen con stegsolve y revisando la escala de gris, obtenemos lo siguiente:*

ID3 -5

---

5. *El fichero nos indica que el otro fichero, SecAdmin2024, es un fichero ID3 en el cual sus bytes se han desplazado en '5'. Restamos 5 bytes a cada byte del fichero con el siguiente código:*

```
with open('SecAdmin2024', 'rb') as f:
   data = bytearray(f.read())
data = bytearray((byte - 5) % 256 for byte in data)  # Decrementar cada byte en 5
with open('SecAdmin2024.mp3', 'wb') as f:
   f.write(data)
```

6. *Transcribimos el fichero mp3 de audio a texto, y obtenemos la flag:*

La flag es flag en hexadecimal

Mindjacked

Estás trabajando para una agencia que combate el cibercrimen. Este reto te lleva a las profundidades de un mundo distópico, una realidad entrelazada con tecnología avanzada, pero donde las líneas de control están cada vez más borrosas. Un Mensaje aparentemente inocente de NeuraLink ha llegado a tu bandeja de entrada.

El mensaje digital susurra sobre una actualización para tu enlace mental, ofreciendo un vistazo a un futuro más brillante. ¡Pero bajo la superficie yace un depredador digital, esperando atrapar a los desprevenidos!

phishing_email.eml

Solución:

$ strings phishing_email.eml | grep -i flag
X-Phishing-Flag: FLAG{N30_PH1SH_2077}

El origen

En SecAdmin, siempre tenemos algo más que decir...

ctf_el_origen.txt

Solución:

Usamos stegnow para obtener una cadena oculta en el fichero 'txt':

$ stegsnow -C -p SecAdmin ctf_el_origen.txt
flag{xaqonv_mn_BnlJmuqv}

La cadena de la flag está codificada:

xaqonv_mn_BnlJmuqv

Flag = ROT17(xarpnw_mn_BnlJmvrw) = ***origen_de_SecAdmin***

Operación Fénix

Un caso que llevaba mucho tiempo inactivo ha resurgido en los archivos de una agencia secreta. La clave para resolver este misterio de la era de la Guerra Fría se encuentra en una sola imagen: una Fotografía de un poderoso tanque.

Debes identificar el modelo exacto del tanque y su país de origen para desbloquear los secretos de esta investigación de décadas de antigüedad.

El token es, en minúsculas, el modelo y las siglas del país de origen del tanque.

Ejemplos de respuesta:
panzer-6-deu
m4-sherman-usa

NN.jpg

Solución:

m36-jackson-us

Corredores de Sombras

Una investigación de alto nivel dentro de una agencia secreta ha llevado a una sola pista críptica: un hash MD5. Se cree que esta huella digital está vinculada a un reciente ataque de ransomware, pero los perpetradores permanecen envueltos en misterio.

HASH: f9cee5e75b7f1298aece9145ea80a1d2

Tu misión: descubre la identidad del ransomware, su grupo APT asociado y su país de origen para ayudar a evitar futuros ataques.

Formato de respuesta (en minúsculas y sin espacios): nombredegrupo-nombrederansomware-pais

Solución:

lazarus-wannacry-northkorea