Mission 5

http://unaaldia.hispasec.com/2018/03/una-al-mes-mision005.html

Parece que han atrapado a Morfeo y debemos averiguar dónde.

Cuando entramos en http://34.253.233.243/search/localizacion.php

vemos que hace una redirección a http://34.253.233.243/search/index.php

Si miramos el código de la página antes de que nos redirija, encontramos el texto al final del html:

Para continuar deberéis sacar X información del primer archivo (la cuál está encriptada) y pasársela al segundo archivo: </br>**archivo 1: https://goo.gl/K1dcbG** </br>**archivo 2: https://drive.google.com/open?id=1CAz5xxsf9YxGlSWDgOVURsvFmT6A1Swn**

El archivo 1

https://goo.gl/K1dcbG

Nos descargamos la imagen y con un simple *head* vemos que entre los metadatos hay cosas interesantes:

```
$ head morfeo.jpg
ÿØÿà JFIF ÿÛ
                :http://ns.adobe.com/xap/1.0/<?xpacket begin='' id='W5M0MpCehiHzreSzNTczkc9d'?>
<x:xmpmeta xmlns:x='adobe:ns:meta/' x:xmptk='Image::ExifTool 10.75'>
<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#'>

 <rdf:Description rdf:about=''
  xmlns:dc='http://purl.org/dc/elements/1.1/'>
  <dc:creator>
   <rdf:Seq>
    <rdf:li>Pass:UAM</rdf:li>
   </rdf:Seq>
```

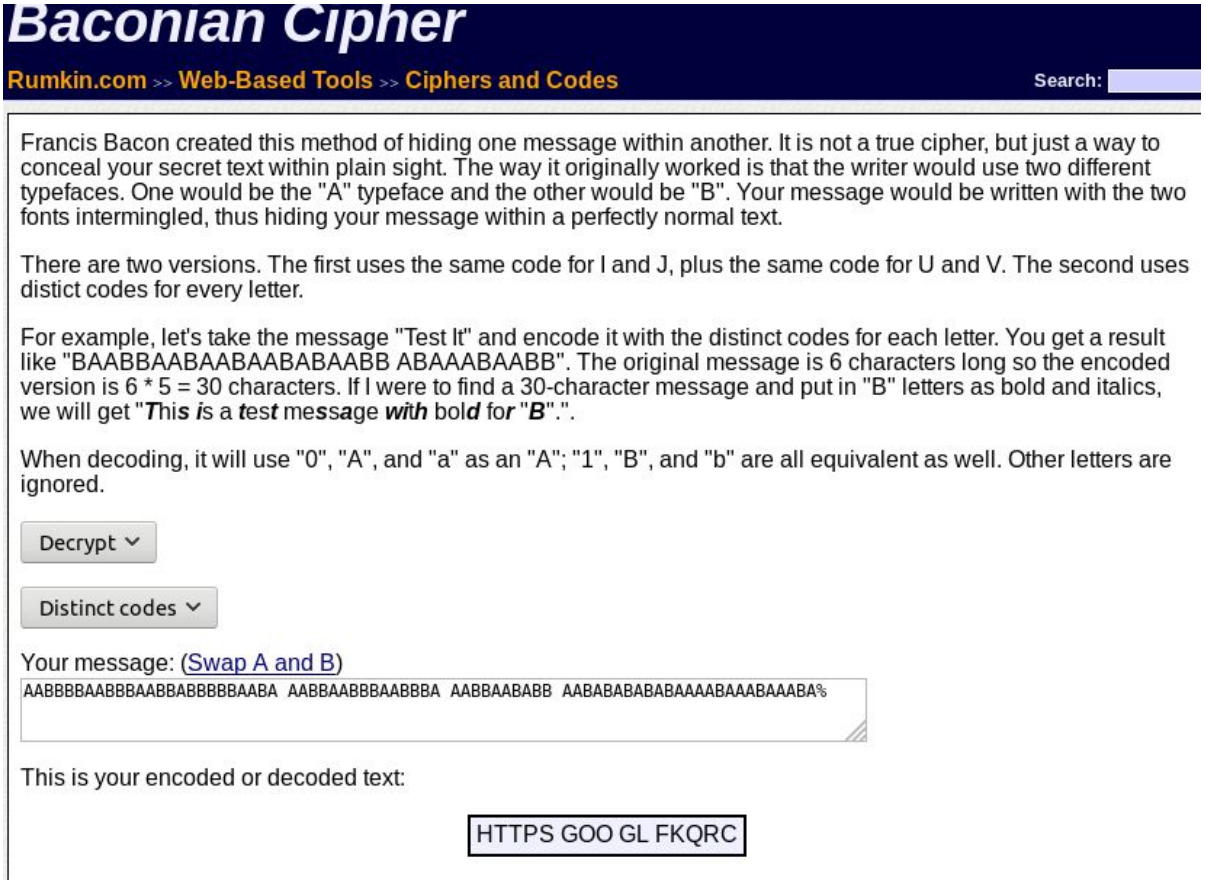
Con Steghide podemos averiguar lo que esconde la imagen, y como es lógico, probamos con la pass UAM

```
$ steghide extract -sf morfeo.jpg -p UAM
anotó los datos extraídos e/"morf.txt".
```

El jpg escondía este texto:

```
$ cat morf.txt
AABBBBAABBBAABBABBBBBAABA AABBAABBBAABBBA AABBAABABB AABABABABABAAAABAAABAAABA
```

Está codificado en Baconian y uso cualquier utilidad online:



El archivo 2

https://drive.google.com/open?id=1CAz5xxsf9YxGISWDgOVURsvFmT6A1Swn

```
#!/usr/bin/python3

string = input("Introduce la información que hayas sacado de la imagen: ")

a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r = string

a = a.lower()
b = b.lower()
c = c.lower()
d = d.lower()
e = e.lower()
f = '://'
g = g.lower()
h = h.lower()
i = i.lower()
j = '.'
k = k.lower()
l = l.lower()
m = '/'
n = n.upper()
o = o.lower()
p = p.upper()
q = q.upper()
r = r.lower()
s = '2'

print (a + b + c + d + e + f + g + h + i + j + k + l + m + n + o + p + q + r + s)
```

Un script en python que nos ayuda a construir la url de verdad, transformando a mayúsculas y minúsculas como corresponde. Además añade un 2 de postre :)

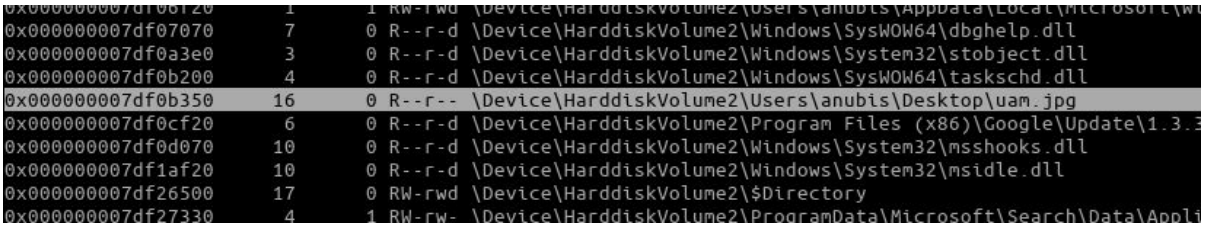
Esta es la URL finalmente https://goo.gl/FkQRc2

El archivo zip tiene password pero no lo conocemos, y tras un rato probando diccionarios sin éxito, con john the ripper trato de hacer brute force… llevaba 30 minutos funcionando cuando nos informaron los admins del CTF por el canal de Telegram que esperaban que funcionara con algunos diccionarios típicos, pero al probar que no sacaba el password optaron por darnos el password a todos: 123mango

Lo descomprimimos usando ese password y extraemos el archivo morfeo.dmp

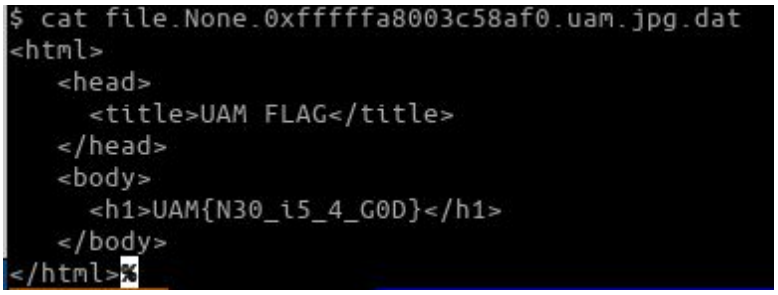
Parece un dump de memoria así que vamos a usar volatility para explorarlo. Después de probar algunas cosas miro sus archivos, buscando los que contengan ‘uam’...

volatility -f morfeo.dmp --profile=Win7SP1x64 filescan



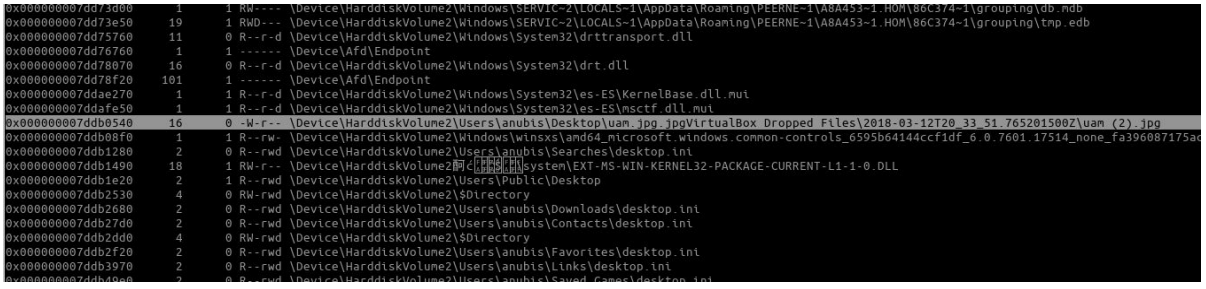
Extraigo el archivo.

volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles -Q 0x000000007df0b350 --name -D files/



Parece que encontramos un flag, no? La maldita trampa… Envié la flag por email pensando que lo había conseguido pero justo leo en telegram el recordatorio... nos piden un lugar, no esto :(

Pero hay más archivos:

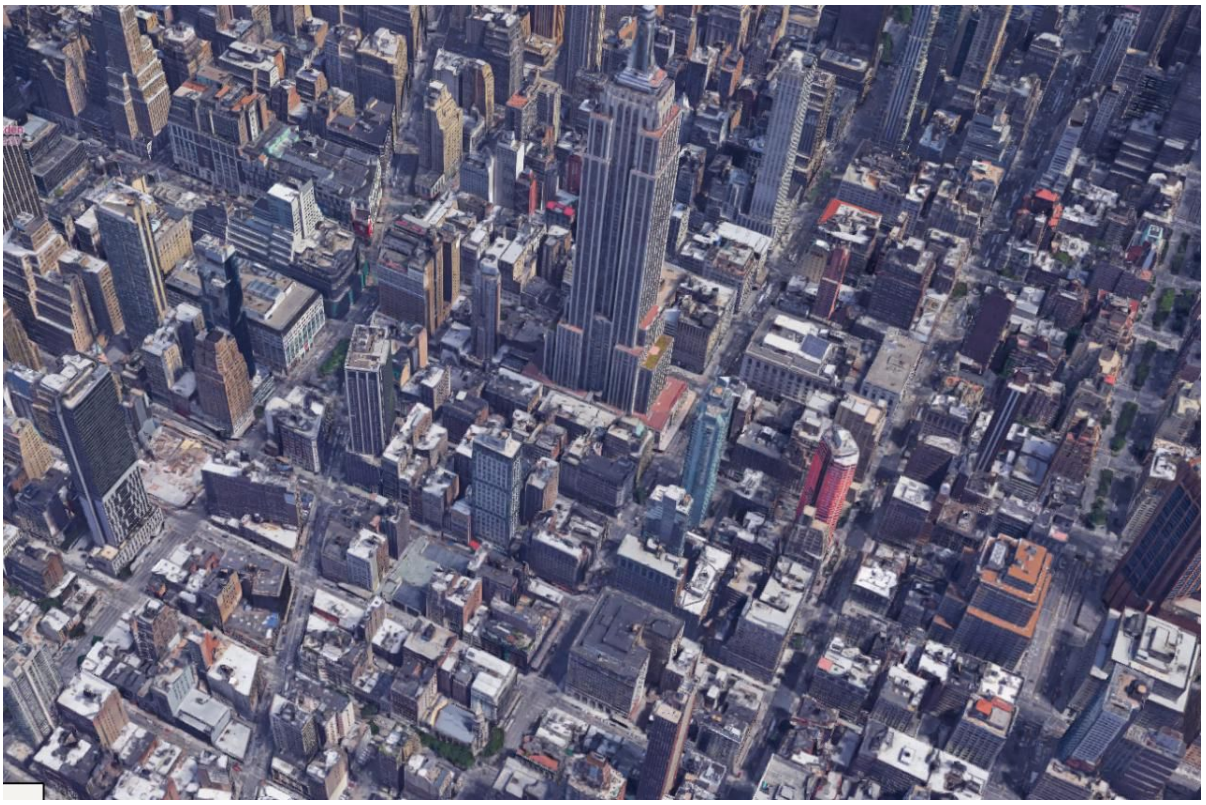


$ volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles -Q 0x000000007ddb0540 --name -D files

Y este sí tiene algo interesante

```
$ cat file.None.0xfffffa8003bb1f10.uam\ \(2\).jpg.dat
<html>
        <head>
                <title>Coordenadas de Morfeo</title>
        </head>
        <body>
                <h1>40.7484405, -73.9856644</h1>
        </body>
</html>%
```

Vamos a maps y vemos que....



Morfeo está en el Empire State Building!

Y como nos advertían en el tip…

```
Tip2: El nombre del sitio en la flag es con "_" en lugar de espacios.
```

Tienen a Morfeo en el **UAM{Empire_State_Building}**!!!

Saludos!

Herramientas usadas:

Firefox
7-zip
steghide
volatility

José Ángel Sánchez (j0n3)
@_j0n3