Una-al-mes: Misión#005 - Hispasec

Misión#005

Información personal:

Nombre: Thomas A. Anderson
Fecha de nacimiento: 11 de Marzo del 1962
Trabajo: Programador
Empresa: Metacortex

Misión:

Nivel: Medio

Introducción:

¡Neo, tenemos un problema! Han secuestrado a Morfeo y no sabemos donde lo pueden tener. Necesitamos que investigues y descubras su localización para rescatarle. La única pista que tenemos es una URL que conseguimos. ¿Serás capaz de encontrarle?

Información adicional:

URL conseguida: http://34.253.233.243/search/localizacion.php

Tip: La flag es el nombre del sitio donde se encuentra con el formato UAM{Localización}.

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

Tip3: El archivo ".zip" se descomprime con "123mango".

Tip4: Hay una flag trampa la cuál no tiene localización.

Accedo a la URL que nos facilita el reto:
http://34.253.233.243/search/localizacion.php

Y veo que me redirige a la página:
http://34.253.233.243/search/index.php

Mirando el código fuente de "localización.php", no me da muchas pistas.

No todo es lo que parece...

Por tanto, hago la petición a "localizacion.php" pero evitando la redirección a "index.php", para ello hago uso de curl:

```
curl -X GET http://34.253.233.243/search/localizacion.php
```

Veo que en la respuesta, vienen dos enlaces a Google drive:

```
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 </br>
```

Archivo 1: Es una imagen de Morfeo.

https://goo.gl/K1dcbG

Archivo 2: Un script en Python

https://drive.google.com/open?id=1CAz5xxsf9YxGlSWDgOVURsvFmT6A1Swn

```
#!/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)
```

Lo primero que hago es analizar el script de Python para ver qué espera recibir. Veo que espera recibir una cadena de 18 caracteres y que luego tras hacer unas transformaciones a cada uno de ellos, nos devuelve el resultado.
El carácter “f” es “://” lo cual me puede dar una pista de que puede ser una URL. Por tanto puedo pensar que los caracteres “abcde” podrían corresponder a “https”.

Tras este análisis y teniendo en mente las posibles pistas tras analizar el script, procedo a buscar información en la imagen.

Lo primeo que hago es usar el comando `strings` de Linux para ver si arroja algo de información:

```
strings morfeo.jpg
```

Algunas pistas que encuentro:

```
<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>
```

Veo que se ha utilizado la herramienta "`exiftool`". Por tanto procedo a analizar la imagen con esa herramienta para ver sus `metadatos`:

```
exiftool morfeo.jpg
```

```
ExifTool Version Number         : 10.10
File Name                       : morfeo.jpg
Directory                       : .
File Size                       : 33 kB
File Modification Date/Time     : 2018:03:16 11:11:46+01:00
File Access Date/Time           : 2018:03:16 12:15:09+01:00
File Inode Change Date/Time     : 2018:03:16 12:06:17+01:00
File Permissions                : rw-rw-r--
File Type                       : JPEG
File Type Extension             : jpg
MIME Type                       : image/jpeg
JFIF Version                    : 1.01
Resolution Unit                 : None
X Resolution                    : 1
Y Resolution                    : 1
XMP Toolkit                     : Image::ExifTool 10.75
Creator                         : Pass:UAM
Image Width                     : 425
Image Height                    : 465
Encoding Process                : Baseline DCT, Huffman coding
Bits Per Sample                 : 8
Color Components                : 3
Y Cb Cr Sub Sampling            : YCbCr4:2:0 (2 2)
Image Size                      : 425x465
Megapixels                      : 0.198
```

No consigo obtener más información útil que la arrojada anteriormente con el comando strings.

Investigo sobre herramientas de estaganografía y finalmente hago uso de "steghide", con la que consigo un texto cifrado haciendo uso del "pass" encontrado en los metadatos, que era "UAM".

```
steghide extract -sf morfeo.jpg
```

Este es el texto cifrado que obtengo:

```
AABBBBAABBBAABBABBBBBAABA AABBAABBBAABBBA AABBAABABB AABABABABABAAAABAAABAAABA
```

También pruebo con un decodificador online y me da el mismo resultado:
https://futureboy.us/stegano/decode.pl

No es la cadena de 18 caracteres que espera recibir el script de Python. Investigo y veo que está codificado con "Bacon". Hago uso de un decodificador online:
https://mothereff.in/bacon
Y obtengo esta cadena:
HTTPS GOO GL FKQRC

Esto tiene ya una longitud de 18 caracteres, empieza por https y por tanto lo puedo usar en el script de Python. Con Python 2 no me funciona, me da un error, así que hay que utilizar Python 3.
Guardo el script en un archivo "uam.py" y lo ejecuto:

```
python uam.py
Introduce la información que hayas sacado de la imagen: HTTPS GOO GL FKQRC
https://goo.gl/FkQRc2
```

Me devuelve una URL para descargar el archivo morfeo.zip

Para descomprimir el archivo, pide una contraseña que viene en el enunciado del reto "123mango" y obtengo un archivo de 2,1GB llamado "morfeo.dmp".

Investigo sobre el formato y veo que corresponde a un archivo de volcado que contiene la razón y ubicación específica en memoria en el que la aplicación dejó de funcionar súbitamente.
Aquí se puede leer más al respecto: https://techlandia.com/formato-dmp-info_290450/

La mayoría de herramientas que encuentro son para Windows. Como no tengo ese sistema operativo, hago uso del comando strings de Linux para hacer búsquedas sobre todo ese archivo de texto.

Con esta búsqueda encuentro el flag falso:
```
strings morfeo.dmp | grep UAM -A10 -B10
```
Lo que hace es buscar en todo el archivo la palabra "UAM" y muestra las 10 líneas anteriores y las 10 posteriores al patrón encontrado.

```
<html>
<head>
<title>UAM FLAG</title>
</head>
<body>
<h1>UAM{N3O_i5_4_G0D}</h1>
</body>
</html>
```

Tras hacer varias búsquedas más con diferentes patrones y viendo que buscando la palabra "UAM" no me da resultados, busco por "<html>" por ver si el flag sigue el mismo formato que la flag falsa y nos da alguna pista válida de la ubicación.

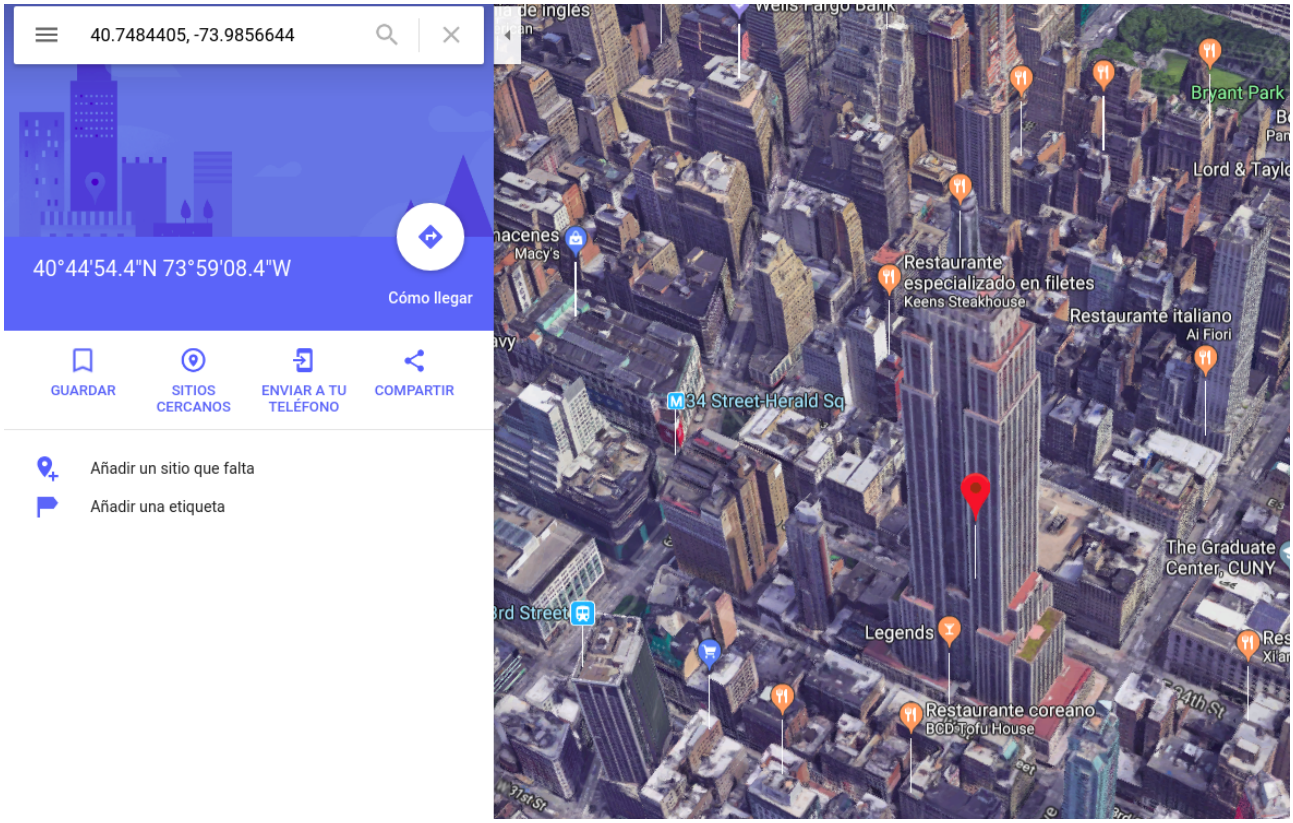
```
strings morfeo.dmp | grep "<html>" -A10 -B10
```

Cual es mi sorpresa, que en los resultados de búsqueda me aparece este archivo HTML:

```
<html>
<head>
<title>Coordenadas de Morfeo</title>
</head>
<body>
<h1>40.7484405, -73.9856644</h1>
</body>
</html>
```

Tras introducir las coordenadas en Google maps, encuentro la ubicación de Morfeo:

https://www.google.es/maps/place/40%C2%B044'54.4%22N+73%C2%B059'08.4%22W/@40.7495802,-73.9875527,16.5z/data=!4m5!3m4!1s0x0:0x0!8m2!3d40.7484405!4d-73.9856644?hl=es



¡ Es el Empire State Building !

Por tanto el flag es: **UAM{Empire_State_Building}**

Rafa Martos
@elbuenodefali