José Ignacio de Miguel González.

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.

Cargamos lo primero la URL, y vemos que cambia de localización.php a index.php.

Vamos a ver la petición a localizacion.php con BURP a ver qué devuelve: efectivamente devuelve un 200 con un redirect:

```
HTTP/1.1 200 OK
Date: Thu, 15 Mar 2018 16:24:38 GMT
Server: Apache/2.4.25 (Debian)
Vary: Accept-Encoding
Content-Length: 2993
Connection: close
Content-Type: text/html; charset=UTF-8

<html>
        <head>
                <meta http-equiv="refresh" content="0; URL=http://34.253.233.243/search/index.php" />
                <title>Archivo de localización encriptado</title>
```

Si embargo, al final del fichero nos da esta información:

```
</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=1CAz5xxsf9YxGlSWDgOVURsvEmT6A1Swn </br>
```

Procedemos a descargar los dos ficheros.

El primero es una imagen: Morfeo.jpg. Si lo descargamos y consultamos sus metadatos, vemos que aparece una Pass: UAM

```
XMP Toolkit                     : Image::ExifTool 10.75
Creator                         : Pass:UAM
Image Width                     : 425
```

Si procedemos a buscar información dentro del fichero, con la aplicación steghide, nos pide un salvoconducto. Probamos esta contraseña “UAM” y nos saca un fichero:

```
nacho@kali:~$ steghide --extract -sf morfeo.jpg
Anotar salvoconducto:
anot los datos extra dos e/"morf.txt".
nacho@kali:~$
```

Al editar el fichero morf.txt nos muestra:

```
nacho@kali:~$ more morf.txt
AABBBBAABBBAABBABBBBBAABA AABBAABBBAABBBA AABBAABABB AABABABABABAAAABAAABAAABA
```

Esto es una codificación binaria, en vez de 1 y 0, con A y B. Existe una codificación llamada Bacon’s Cipher que basa su alfabeto cambiando los 1 y 0 por los caracteres A y B. En internet sacamos el alfabeto y vamos traduciendo:

https://en.wikipedia.org/wiki/Bacon's_cipher

Letter	Code	Binary	Letter	Code	Binary
A	aaaaa	00000	N	abbab	01101
B	aaaab	00001	O	abbba	01110
C	aaaba	00010	P	abbbb	01111
D	aaabb	00011	Q	baaaa	10000
E	aabaa	00100	R	baaab	10001
F	aabab	00101	S	baaba	10010
G	aabba	00110	T	baabb	10011
H	aabbb	00111	U	babaa	10100
I	abaaa	01000	V	babab	10101
J	abaab	01001	W	babba	10110
K	ababa	01010	X	babbb	10111
L	ababb	01011	Y	bbaaa	11000
M	abbaa	01100	Z	bbaab	11001

Letra a letra nos saca la cadena final:

```
AABBB_BAABB_BAABB_ABBBB_BAABA AABBA_ABBBA_ABBBA AABBA_ABABB AABAB_ABABA_BAAAA_BAAAB_AAABA
    h    t    t    p    s    g    o    o    g    l    f    k    q    r    c
https goo gl fkqrc
```

Esta es la clave decodificada. Ahora el otro fichero que nos derivó la página principal es un código Python que espera una entrada:

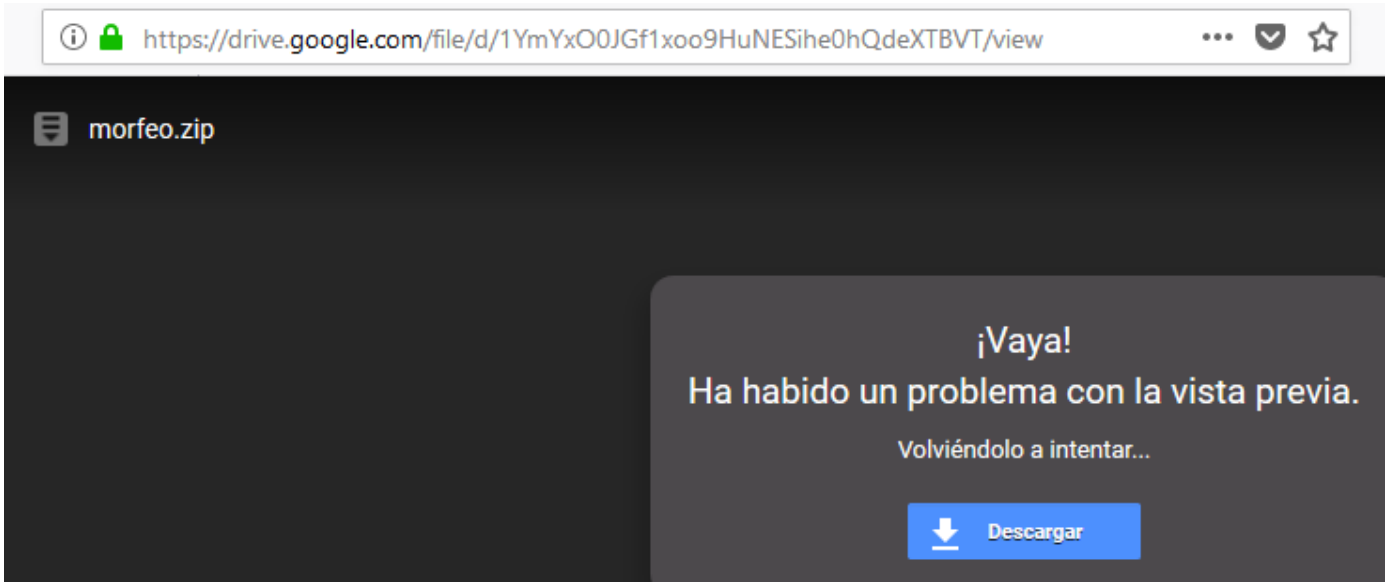
web.py

```
#!/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 convertimos a un fichero Python y procedemos a ejecutarlo. Nos pide una cadena de entrada y metemos la que hemos sacado del anterior fichero:

```
nacho@kali:~$ python3 pythonUAM.py
Introduce la información que hayas sacado de la imagen: https goo gl fkqrc
https://goo.gl/FkQRc2
nacho@kali:~$
```

Nos proporciona una url, que nos lleva a un fichero ZIP, que descargamos:



Ahora lo descomprimimos con la clave que nos dieron “123mango”, en mi caso no ha hecho falta ya la fuerza bruta. Aparece un fichero de 2 GB con nombre “morfeo.dmp”.

```
00000000h: 50 41 47 45 44 55 36 34 0F 00 00 00 B1 1D 00 00 ; PAGEDU64....±...
00000010h: 00 70 18 00 00 00 00 00 00 00 00 00 80 FA FF FF ; .p..........€úÿÿ
00000020h: 90 8E 85 02 00 F8 FF FF 90 AB 83 02 00 F8 FF FF ;  Ž…..øÿÿ «ƒ..øÿÿ
00000030h: 64 86 00 00 01 00 00 00 4D 41 54 54 50 41 47 45 ; d†......MATTPAGE
00000040h: 4D 4F 4F 4E 00 00 00 00 53 4F 4C 53 00 00 00 00 ; MOON....SOLS....
00000050h: 4D 4F 4F 4E 00 00 00 00 53 4F 4C 53 00 00 00 00 ; MOON....SOLS....
00000060h: 50 41 47 45 50 41 47 45 50 41 47 45 50 41 47 45 ; PAGEPAGEPAGEPAGE
00000070h: 50 41 47 45 50 41 47 45 50 41 47 45 50 41 47 45 ; PAGEPAGEPAGEPAGE
00000080h: 80 70 B3 01 80 FA FF FF 02 00 00 00 50 41 47 45 ; €p³.€úÿÿ....PAGE
00000090h: 8E FF 07 00 00 00 00 00 01 00 00 00 00 00 00 00 ; Žÿ..............
000000a0h: 9E 00 00 00 00 00 00 00 00 01 00 00 00 00 00 00 ; ž...............
000000b0h: F0 FE 07 00 00 00 00 00 50 41 47 45 50 41 47 45 ; ðþ......PAGEPAGE
```

El fichero empieza por la palabra PAGEDU64, que según la documentación es un “crash dump” de un sistema Windows de 64 bits:

64-bit crash dumps

Crash dumps from 64-bit systems begin with a `_DMP_HEADER64` structure. The Signature field must be "PAGEDU64" for volatility to consider it valid.

Lanzamos un “volatility -f morfeo.dmp crashinfo” y efectivamente nos confirma que es un dump completo:

```
DumpType              Full Dump
SystemTime            2018-03-12 20:35:20 UTC+0000
SystemUpTime          0:02:17.018554
```

Ahora procedemos a buscar los ficheros que hay contenidos dentro del dump, a ver si vemos algo que nos valga. De los 2100 ficheros filtramos por los que contengan UAM, y salen estos tres, con sus correspondientes direcciones de memoria:

```
0x000000007df0b350  \Device\HarddiskVolume2\Users\anubis\Desktop\uam.jpg

0x000000007ddb0540  \Device\HarddiskVolume2\Users\anubis\Desktop\uam.jpg.jpgVirtualBox Dropped
Files\2018-03-12T20_33_51.765201500Z\uam (2).jpg

0x000000007ddc5070
\Device\HarddiskVolume2\Users\anubis\AppData\Roaming\Microsoft\Windows\Recent\uam (2).lnk
```

Ahora vamos a extraerlos del dump, con el comando:

volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles -dumpdir=filesDump -Q Dirección de memoria:

```
nacho@kali:~$ volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles --dump-dir=filesDump -Q 0x00
0000007ddb0540
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x7ddb0540   None   \Device\HarddiskVolume2\Users\anubis\Desktop\uam.jpg.jpgVirt
ualBox Dropped Files\2018-03-12T20_33_51.765201500Z\uam (2).jpg
nacho@kali:~$ volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles --dump-dir=filesDump -Q 0x00
0000007ddc5070
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x7ddc5070   None   \Device\HarddiskVolume2\Users\anubis\AppData\Roaming\Microso
ft\Windows\Recent\uam (2).lnk
nacho@kali:~$ volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles --dump-dir=filesDump -Q 0x00
0000007df0b350
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x7df0b350   None   \Device\HarddiskVolume2\Users\anubis\Desktop\uam.jpg
nacho@kali:~$
```

Ya está extraídos, listamos el directorio filesDump:

```
nacho@kali:~/filesDump$ ls -la
total 20
drwxr-xr-x  2 nacho nacho 4096 mar 15 18:47 .
drwxr-xr-x 54 nacho nacho 4096 mar 15 18:46 ..
-rw-r--r--  1 nacho nacho 4096 mar 15 18:46 file.None.0xfffffa8003bb1f10.dat
-rw-r--r--  1 nacho nacho 4096 mar 15 18:47 file.None.0xfffffa8003bc8f10.dat
-rw-r--r--  1 nacho nacho 4096 mar 15 18:47 file.None.0xfffffa8003c58af0.dat
nacho@kali:~/filesDump$
```

Si los vamos editando:

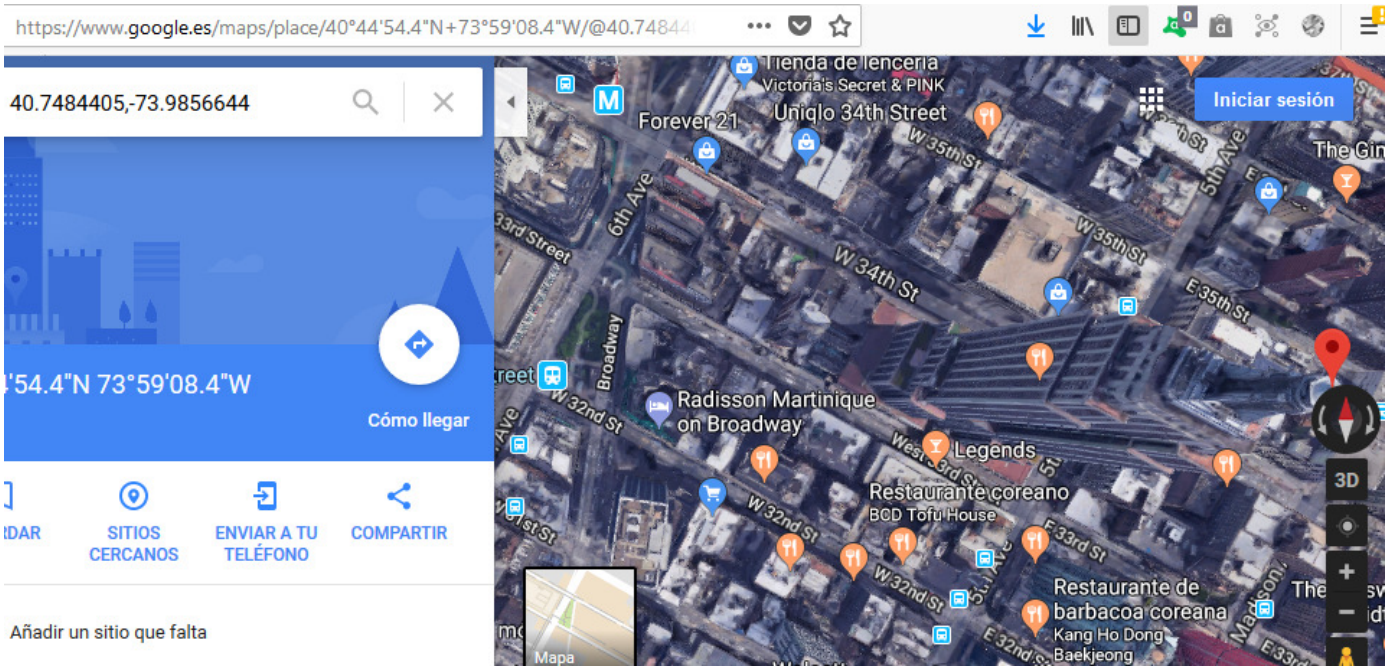
- Este es la flag falsa:

```
-rw-r--r--  1 nacho nacho 4096 mar 15 18:47 file.None.0xfffffa8003c58af0.dat
nacho@kali:~/filesDump$ more file.None.0xfffffa8003c58af0.dat
<html>
        <head>
                <title>UAM FLAG</title>
        </head>
        <body>
                <h1>UAM{N30_i5_4_G0D}</h1>
        </body>
</html>
```

- En este está las coordenadas buenas:

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

Las buscamos en el Google maps y sale el Empire State Building:



Por tanto, la Flag debe ser UAM{Empire_State_Building}