Descripción

Nombre: Misión#006

Fecha de liberación: 15 de Abril del 2018

Dificultad: Fácil (según los retadores)

Información personal:

Nombre: Beatrix Michelle Kiddo
Año de nacimiento: 1976
Trabajo: Ex Asesina
Afiliación: Antiguamente en 'Deadly Viper Assasination Squad'

Misión:

Introducción:

La flag escondida en esta prueba te va a dar a escoger entre dos opciones. Esperemos que escojas bien, sino vas a recibir las consecuencias ...

Información adicional:

URL conseguida: goo.gl/YUNxSu

Esta vez seremos "La Mamba Negra" una ex-asesina, después de su incidente en la capilla juró vengarse y nosotros deberemos ayudarle en su misión.

Objetivo

Formato de flag: UAM{flag}

Herramientas utilizadas

Chrome (66.0.3359.106) https://www.google.com/chrome/
file (5.33) https://github.com/file/file // http://freshmeat.sourceforge.net/projects/file/
strings (2.30)
curl (7.59) https://github.com/curl/curl // https://curl.haxx.se/
hashID | hash-identifier (3.1.4) https://github.com/psypanda/hashID
https://md5online.org
https://29a.ch/photo-forensics/#forensic-magnifier
https://gchq.github.io/CyberChef/

Resumen:

Comenzamos por visitar la página de la misión donde se nos entrega la url como única pista.

Visitamos la url acortada de google (https://goo.gl/YUNxSu) con el navegador la cual hace una redirección al Drive de Google en:
https://drive.google.com/file/d/1J2mMiIwqZ_pgRBEUEccyepJrBPH3c_FA/view

Descargamos en la parte superior derecha de la pantalla o utilizamos con curl la descarga del archivo llamado *bill2.jpg* que nos arroja un fotograma de la película donde se muestra a la actriz Uma Thurman como el personaje de Beatrix:

```
$ curl -L -o bill2.jpg
'https://drive.google.com/uc?export=download&id=1J2mMiIwqZ_pgRBEUEccyepJ
rBPH3c_FA'
```

Al descargar el archivo *bill2.jpg* su análisis inicial nos arroja:

```
$ file bill2.jpg
bill2.jpg: JPEG image data, JFIF standard 1.01, resolution (DPI),
density 72x72, segment length 16, baseline, precision 8, 1200x630,
frames 3

$ strings bill2.jpg
JFIF
...
aa81a304ea2a25ad2947a03062c05fdf
```

Podemos observar que nos arroja un hash (una cadena de 32 bits). Para poder identificarlo lanzamos la tool hash-identifier:

```
$ hashid aa81a304ea2a25ad2947a03062c05fdf
Analyzing 'aa81a304ea2a25ad2947a03062c05fdf'
[+] MD2
[+] MD5
[+] MD4
[+] Double MD5
[+] LM
[+] RIPEMD-128
[+] Haval-128
[+] Tiger-128
[+] Skein-256(128)
[+] Skein-512(128)
```

```
[+] Lotus Notes/Domino 5
[+] Skype
[+] Snefru-128
[+] NTLM
[+] Domain Cached Credentials
[+] Domain Cached Credentials 2
[+] DNSSEC(NSEC3)
[+] RAdmin v2.x
```

Optamos por el hash más común MD5 y nos ayudamos de la herramienta online https://md5online.org/md5-decrypt.html y obtenemos:

Found : goo.gl/4kxSs7
(hash = aa81a304ea2a25ad2947a03062c05fdf)

Hemos obtenido una url acortada de Google (https://goo.gl/4kxSs7) redireccionándonos a https://drive.google.com/file/d/18nlxec8n1ziQJmSXOMfa1e4IoG1FsPRA/view a un archivo llamado *kill-bill-movie.png* imagen de nuevo de nuestra protagonista Beatrix:

```
$ curl -L -o kill-bill-movie.png
'https://drive.google.com/uc?export=download&id=18nlxec8n1ziQJmSXOMfa1e4
IoG1FsPRA'
```

Al descargar el archivo *kill-bill-movie.png* su análisis inicial nos arroja:

```
$ file kill-bill-movie.png
kill-bill-movie.png: PNG image data, 1920 x 1080, 8-bit/color RGBA,
non-interlaced
```

Comprobamos strings y binwalk sin arrojarnos nada significativo ni que oculte ningún otro archivo diferente a la imagen :

```
$ binwalk kill-bill-movie.png

DECIMAL       HEXADECIMAL     DESCRIPTION
--------------------------------------------------------------------------------
--------
0             0x0             PNG image, 1920 x 1080, 8-bit/color RGBA,
non-interlaced
180           0xB4            Zlib compressed data, best compression
```

Documentandose en Google y observando que no esconde ningún archivo, nos enfrentamos a un reto de esteganografía.

Lanzamos varias tools bastante utilizadas en esteganografía como *steghide*, *stegosuite* y *openstego* sin obtener ningún resultado.

Tomando la orientación que la esteganografía podría ser más visual utilizamos la herramienta online https://29a.ch/photo-forensics/#forensic-magnifier

Jugando con el cursor por encima de la imagen de Beatrix detectamos un código Morse a los pies de la protagonista, obteniendo la siguiente cadena.

```
...- ..- ..-. -. . ----- ... -..- - --. -..- ..-. --.- .--- ..-.
-- -... ..-. ----. ...- -..- .---- .--- .-.. - - -... ----.
```

Añadir que con la tool en Java de StegSolve 1.3 by Caesum también pudimos ver la misma cadena.

Decodificando nos da el siguiente string:

VUFNE0SXTGXFQJFMBF9VX1JLTTB9

Analizamos el siguiente string en https://md5hashing.net/hash_type_checker arrojandonos que lo identifica como base64

```
$ echo -n VUFNE0SXTGXFQJFMBF9VX1JLTTB9 | base64 -d
UAMD'Le'@'L_U_RKM0}
```

La decodificación nos da un resultado muy cercano al formato de la flag que es UAM{flag} .

A partir de aquí viene un punto de inflexión y de comprobar que el resultado de decodificar un MORSE siempre será un resultado en MAYÚSCULAS pero una cadena de base64 varía dependiendo de las mayúsculas y las minúsculas (Case Sensitive).

Nos ayudamos de una navaja suiza online CyberChef:

https://gchq.github.io/CyberChef/#recipe=From_Base64('A-Za-z0-9%2B/%3D',true)&input=VlVGTkUwU1hUR1hGUUpGTUJGOVZYMUpMVFRCOQ

Que nos permite jugar con el string de base64 hasta conseguir una flag dentro de formato y con sentido quedando el base64 de esta forma:

```
VUFNe0sxTGxfQjFMbF9vX1JlTTB9
```

Y la solución

La flag es: **UAM{K1LI_B1LI_o_ReM0}**