Una al Mes (Mission 005)

@percu

20/03/18

Contenido

INTRODUCCIÓN

Puesto que es de mis primeros CTF, este *write-up* tiene un enfoque más práctico que técnico.

El motivo es tener una futura referencia de las herramientas utilizadas, así como las deducciones que he ido aplicando cuando me he quedado estancado.

Por lo tanto, más que la explicación de los pasos realizados para la resolución del CTF, este *write-up* es un *time-line* de como he ido solucionado los diferentes retos propuestos, aunque también identifico los palos de ciego que he ido dando mientras me quedaba estancado en cada uno de los retos y como le daba la vuelta para encontrar su solución.

Empecemos…

MISIÓN

En la URL http://34.253.233.243/mission5.php encontramos la información:

Mission#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 dónde 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.

RESOLUCIÓN

PARTE 1 - WEB

Al acceder a la url http://34.253.233.243/search/localizacion.php solamente encontramos un gif de Neo con la frase "No todo es lo que parece…":

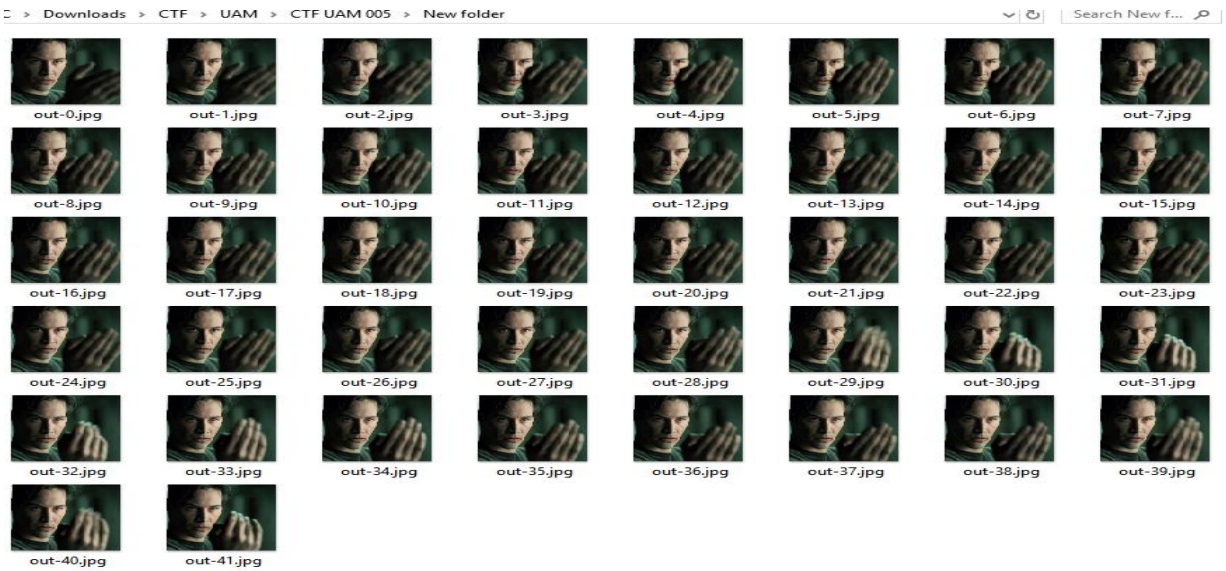
En principio parece un reto de esteganografía, por lo que procedemos a descargar el gif i revisarlo.

Con la utilidad '*steghide'* miramos si hay información escondida, pero no encontramos nada:

```
 > steghide info neo.gif
steghide: the file format of the file "neo.gif" is not supported.
```

Con el comando '*convert'* de *ImageMagick* obtenemos todas las imágenes del gif, pero no encontramos nada… salvo a Neo saludando en todas las posiciones posibles:

```
#: convert -coalesce neo.gif out.jpg
```



Miramos las propiedades de los archivos '*.jpg' obtenidos, pero sin encontrar nada de interés.

Se parecia que al cargar la url hace previamente una llamada, por lo que vamos a intentar descargar la url principal con el comando 'wget':

```
#: wget http://34.253.233.243/search/localizacion.php
```

Vemos que el código .php obtenido tiene como título "Archivo de localización encriptado" y después de muchos tags *
* hay escondidas dos *url*'s y la indicación de que tenemos que sacar información del primer archivo para pasársela al segundo

```
        <br>
</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=1CAz5xxsf9YxGlSWDgOVURsvFmT6A1Sw
n </br>
 >
```

En la primera url https://goo.gl/K1dcbG obtenemos el Archivo 1 morfeo.jpg

Y en la segunda url https://drive.google.com/open?id=1CAz5xxsf9YxGlSWDgOVURsvFmT6A1Swn el fichero web.py:

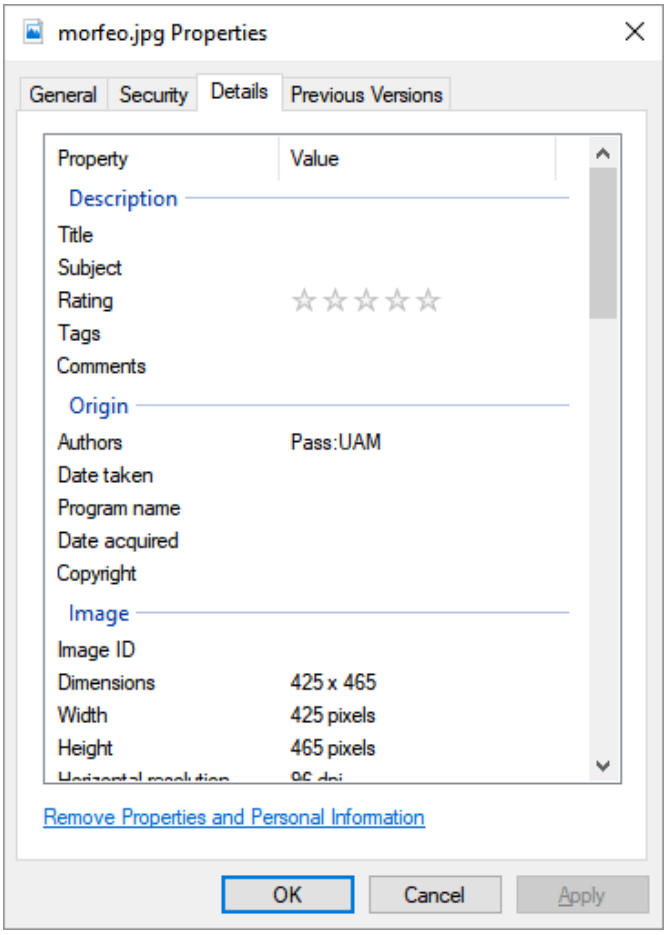
```
#!/usr/bin/python3
string = input("Introduce la información que hayas sacado de la imagen: ")

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)
```

PARTE 2 - ESTEGANOGRAFIA

Analizando el fichero 'morfeo.jpg' con la utilidad '*steghide'* vemos que tiene información oculta y nos pide contraseña:

```
 > steghide info morfeo.jpg
"morfeo.jpg":
  format: jpeg
  capacity: 1.3 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
steghide: could not extract any data with that passphrase!
 > _
```

Si miramos sus propiedades vemos que la tiene en los metadatos del fichero:



Ponemos dicho password y vemos que tiene un fichero 'morfeo.txt' oculto en la imagen:

```
 > steghide info morfeo.jpg
"morfeo.jpg":
  format: jpeg
  capacity: 1.3 KB
Try to get information about embedded data ? (y/n) y
Enter passphrase:
  embedded file "morf.txt":
    size: 78.0 Byte
    encrypted: rijndael-128, cbc
    compressed: yes
```

Extraemos dicho fichero y miramos su contenido:

```
 > rsteghide extract -sf morfeo.jpg
Enter passphrase:
wrote extracted data to "morf.txt".
 > more morf.txt
AABBBBAABBBAABBABBBBBAABA AABBAABBBAABBBA AABBAABABB AABABABABABAAAABAAABAAABA
```

Esta cadena de caracteres es la que tendremos que pasarle al script Python 'web.py' descargado anteriormente tal y como nos indicaban…

Al ejecutar el script en Python nos pide la información extraída de la imagen anterior y si ponemos el contenido del fichero '*morf.txt*' nos responde diciendo que tenemos que poner 18 caracteres:

```
>python web.py
Introduce la informacion que hayas sacado de la imagen: AABBBBAABBBAABBABBBBBAABA AABBAABBBAABBBA AABBAABABB AABABABABABAAAABAAABAAABA
Traceback (most recent call last):
  File "web.py", line 5, in <module>
    a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r = string
ValueError: too many values to unpack (expected 18)
```

Observando el script se ve como el sexto carácter lo sustituye por "://", el décimo por ".", el treceavo por "/" y le añade un "2" al final. Parece que sea el formato de una url, así que nos fijaremos en los primeros 5 caracteres:

```
a = a.lower()
b = b.lower()
c = c.lower()
d = d.lower()
e = e.lower()
f = '://'
```

Podemos deducir que el primer carácter tiene que ser una 'H', el segundo y tercero una 'T', el cuarto una 'P' i el quinto una 'S'.

A partir del contenido del fichero '*morf.txt*' vemos que las cadenas formadas por las letras 'A' y 'B' forman 4 grupos, los mismos que hay en el script en Python.

Con el primer grupo (AABBBBAABBBAABBABBBBBAABA) debemos formar 5 caracteres, así que dividimos la cadena en 5 trozos iguales:

AABBB, BAABB, BAABB, ABBBB, BAABA

Ya vemos que el segundo y tercer grupo son iguales, por lo que el patrón deducido de https parece ser correcto. Solo falta averiguar valores…

Si convertimos a binario suponiendo que la 'B' tiene valor '1' y la 'A' valor '0' obtenemos:

7, 19, 19, 15, 18

Si tomamos el abecedario con la letra A=0, nos queda H, T, T, P, S dando la cadena correcta.

Solamente nos queda transformar cada grupo de 5 letras del fichero '*morf.txt*' a binario y buscar su valor en el abecedario:

GRUPO 1	GRUPO 2	GRUPO 3	GRUPO 4
AABBB - 7 - h BAABB - 19 - t BAABB - 19 - t ABBBB - 15 - p BAABA - 18 - s	AABBA - 6 - g ABBBA - 14 - o ABBBA - 14 - o	AABBA - 6 - G ABABB - 11 - L	AABAB - 5 - f ABABA - 10 - k BAAAA - 16 - q BAAAB - 17 - r AAABA - 2 - c

Le pasamos la cadena obtenida (https goo gl fkqrc) al script Python:

```
>python web.py
Introduce la información que hayas sacado de la imagen: https goo gl fkqrc
https://goo.gl/FkQRc2
```

Y obtenemos la url: https://goo.gl/FkQRc2

PARTE 3 – ANÁLISIS FORENSE

En la dirección anterior https://goo.gl/FkQRc2 nos podemos descargar un fichero de nombre '*morfeo.zip*' comprimido, del cual nos dan la contraseña en un *tip* (123Mango)

Dentro de ese fichero comprimido hay otro de nombre morfeo.dmp que parece ser un volcado de memoria. Lo analizamos con la herramienta *'volatility'*

```
>volatility -f morfeo.dmp imageinfo
          Suggested Profile(s) : Win8SP0x64, Win81U1x64, Win10x64_14393, Win2012R2x64_18340, Win10x64,
Win2016x64_14393, Win2012R2x64, Win2012x64, Win8SP1x64_18340, Win10x64_10586, Win8SP1x64, Win10x64_15063
(Instantiated with Win10x64_15063)
                     AS Layer1 : SkipDuplicatesAMD64PagedMemory (Kernel AS)
                     AS Layer2 : WindowsCrashDumpSpace64 (Unnamed AS)
                     AS Layer3 : FileAddressSpace (/mnt/c/morfeo.dmp)
                      PAE type : No PAE
                           DTB : 0x187000L
                          KDBG : 0xf800028040a0L
          Number of Processors : 1
     Image Type (Service Pack) : 1
                KPCR for CPU 0 : 0xfffff80002805d00L
             KUSER_SHARED_DATA : 0xfffff78000000000L
           Image date and time : 2018-03-12 20:35:20 UTC+0000
     Image local date and time : 2018-03-12 21:35:20 +0100
```

Vemos que es de un sistema Windows de 64bits. Procedemos a mirar con qué profile examinarla con el plugin '*kdbgscan'*, que lo instancia con todos los *profiles* posibles (recorto salida de pantalla porque es extensa):

```
>volatility -f morfeo.dmp kdbgscan
**************************************************
Instantiating KDBG using: Kernel AS Win10x64_14393 (6.4.14393 64bit)
Offset (V)                    : 0xf800028040a0
Offset (P)                    : 0x28040a0
KdCopyDataBlock (V)           : 0xf8000271bc04
Block encoded                 : No
Wait never                    : 0xb79180109e1676c0
Wait always                   : 0x84f0f5ea6bc9180
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win10x64_14393
Version64                     : 0xf80002804068 (Major: 15, Minor: 7601)
Service Pack (CmNtCSDVersion) : 1
Build string (NtBuildLab)     : 7601.17514.amd64fre.win7sp1_rtm.
PsActiveProcessHead           : 0xfffff8000283ab90 (37 processes)
PsLoadedModuleList            : 0xfffff80002858e90 (139 modules)
KernelBase                    : 0xfffff80002613000 (Matches MZ: True)
Major (OptionalHeader)        : 6
Minor (OptionalHeader)        : 1
KPCR                          : 0xfffff80002805d00 (CPU 0)
**************************************************
Instantiating KDBG using: Kernel AS Win10x64 (6.4.9841 64bit)
Offset (V)                    : 0xf800028040a0
Offset (P)                    : 0x28040a0
KdCopyDataBlock (V)           : 0xf8000271bc04
Block encoded                 : No
Wait never                    : 0xb79180109e1676c0
Wait always                   : 0x84f0f5ea6bc9180
KDBG owner tag check          : True
Profile suggestion (KDBGHeader): Win10x64
Version64                     : 0xf80002804068 (Major: 15, Minor: 7601)
Service Pack (CmNtCSDVersion) : 1
Build string (NtBuildLab)     : 7601.17514.amd64fre.win7sp1_rtm.
PsActiveProcessHead           : 0xfffff8000283ab90 (37 processes)
PsLoadedModuleList            : 0xfffff80002858e90 (139 modules)
KernelBase                    : 0xfffff80002613000 (Matches MZ: True)
Major (OptionalHeader)        : 6
Minor (OptionalHeader)        : 1
KPCR                          : 0xfffff80002805d00 (CPU 0)
```

Vemos que el '*Build String*' es '*7601.17514.amd64fre.win7sp1_rtm.*', así que lo abriremos con el *profile* de '*Win7SP1x64*'.

Miramos los procesos en memoria con el plugin '*pstree*':

```
> volatility -f morfeo.dmp --profile=Win7SP1x64 pstree
Volatility Foundation Volatility Framework 2.6
Name                                                 Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0xfffffa8003805460:explorer.exe                     1996   1956     36    877 2018-03-12 20:33:29 UTC+0000
. 0xfffffa80038d37d0:VBoxTray.exe                    1792   1996     15    155 2018-03-12 20:33:29 UTC+0000
. 0xfffffa8003be6060:DumpIt.exe                      1208   1996      6     93 2018-03-12 20:35:18 UTC+0000
 0xfffffa80018cc060:wininit.exe                       412    368      3     75 2018-03-12 20:33:16 UTC+0000
. 0xfffffa800341c270:lsass.exe                       528    412      8    700 2018-03-12 20:33:18 UTC+0000
. 0xfffffa80033a46e0:lsm.exe                         536    412     10    144 2018-03-12 20:33:18 UTC+0000
. 0xfffffa800337fb30:services.exe                    508    412     10    186 2018-03-12 20:33:16 UTC+0000
.. 0xfffffa800350bb30:svchost.exe                    916    508     29    545 2018-03-12 20:33:19 UTC+0000
... 0xfffffa80036fb060:dwm.exe                      1976    916      4     71 2018-03-12 20:33:29 UTC+0000
.. 0xfffffa80037e0a70:taskhost.exe                  1924    508     11    172 2018-03-12 20:33:29 UTC+0000
.. 0xfffffa80034d1740:svchost.exe                    804    508     22    508 2018-03-12 20:33:19 UTC+0000
... 0xfffffa8003556b30:audiodg.exe                   316    804      7    125 2018-03-12 20:33:19 UTC+0000
.. 0xfffffa8003c54060:mscorsvw.exe                  2972    508      5     60 2018-03-12 20:35:38 UTC+0000
.. 0xfffffa800382c510:SearchIndexer.                2256    508     12    601 2018-03-12 20:33:35 UTC+0000
... 0xfffffa800395f7d0:SearchFilterHo               2344   2256      4     77 2018-03-12 20:33:36 UTC+0000
... 0xfffffa80039482a0:SearchProtocol               2324   2256      7    224 2018-03-12 20:33:36 UTC+0000
.. 0xfffffa8003963b30:wmpnetwk.exe                  2448    508     15    427 2018-03-12 20:33:37 UTC+0000
.. 0xfffffa8003494060:VBoxService.ex                 700    508     13    118 2018-03-12 20:33:19 UTC+0000
.. 0xfffffa800352ab30:svchost.exe                    960    508     40    887 2018-03-12 20:33:19 UTC+0000
... 0xfffffa8003806b30:taskeng.exe                  1984    960      6     82 2018-03-12 20:33:29 UTC+0000
.. 0xfffffa8003acb4a0:svchost.exe                   2644    508     10    354 2018-03-12 20:33:38 UTC+0000
.. 0xfffffa80035924a0:svchost.exe                    868    508     19    469 2018-03-12 20:33:20 UTC+0000
.. 0xfffffa80035d1b30:spoolsv.exe                   1232    508     15    290 2018-03-12 20:33:20 UTC+0000
.. 0xfffffa80036b3060:svchost.exe                   1380    508     22    292 2018-03-12 20:33:22 UTC+0000
.. 0xfffffa80035c6740:svchost.exe                   1132    508     17    462 2018-03-12 20:33:20 UTC+0000
.. 0xfffffa80034a85f0:svchost.exe                    752    508      8    276 2018-03-12 20:33:19 UTC+0000
.. 0xfffffa80036205f0:svchost.exe                   1268    508     21    319 2018-03-12 20:33:20 UTC+0000
.. 0xfffffa800346f4b0:svchost.exe                    636    508     10    358 2018-03-12 20:33:19 UTC+0000
... 0xfffffa8003bbcab0:WmiPrvSE.exe                 2828    636      8    123 2018-03-12 20:34:10 UTC+0000
... 0xfffffa80037e7630:dllhost.exe                  3020    636     10    167 2018-03-12 20:33:58 UTC+0000
... 0xfffffa8003b136b0:WmiPrvSE.exe                 2780    636      8    117 2018-03-12 20:33:38 UTC+0000
 0xfffffa8003415060:csrss.exe                        376    368      9    426 2018-03-12 20:33:16 UTC+0000
 0xfffffa80018c5040:System                             4      0     91    512 2018-03-12 20:33:05 UTC+0000
. 0xfffffa800295cb30:smss.exe                        268      4      2     29 2018-03-12 20:33:06 UTC+0000
 0xfffffa80018cb5b0:csrss.exe                        424    404      7    209 2018-03-12 20:33:16 UTC+0000
. 0xfffffa8003754060:conhost.exe                    1280    424      2     56 2018-03-12 20:35:18 UTC+0000
 0xfffffa800337c330:winlogon.exe                     464    404      6    114 2018-03-12 20:33:16 UTC+0000
> _
```

Y también un '*psscan'* para ver los que había antes en ejecución:

```
> volatility -f morfeo.dmp --profile=Win7SP1x64 psscan
Volatility Foundation Volatility Framework 2.6
Offset(P)          Name                PID   PPID PDB               Time created                   Time exited
------------------ ---------------- ------ ------ ----------------- ------------------------------ ------------------------------
0x000000007d9b6910 DumpIt.exe         2672   1996 0x000000006e5ec000 2018-03-12 20:34:05 UTC+0000   2018-03-12 20:34:53 UTC+0000
0x000000007d9b6910 DumpIt.exe         2672   1996 0x000000006e5ec000 2018-03-12 20:34:05 UTC+0000   2018-03-12 20:34:53 UTC+0000
0x000000007d9f4060 mscorsvw.exe       2972    508 0x0000000062479000 2018-03-12 20:35:38 UTC+0000
0x000000007dae7040 System                4      0 0x0000000000187000 2018-03-12 20:33:05 UTC+0000
0x000000007daed5b0 csrss.exe           424    404 0x000000002c0ef000 2018-03-12 20:33:16 UTC+0000
```

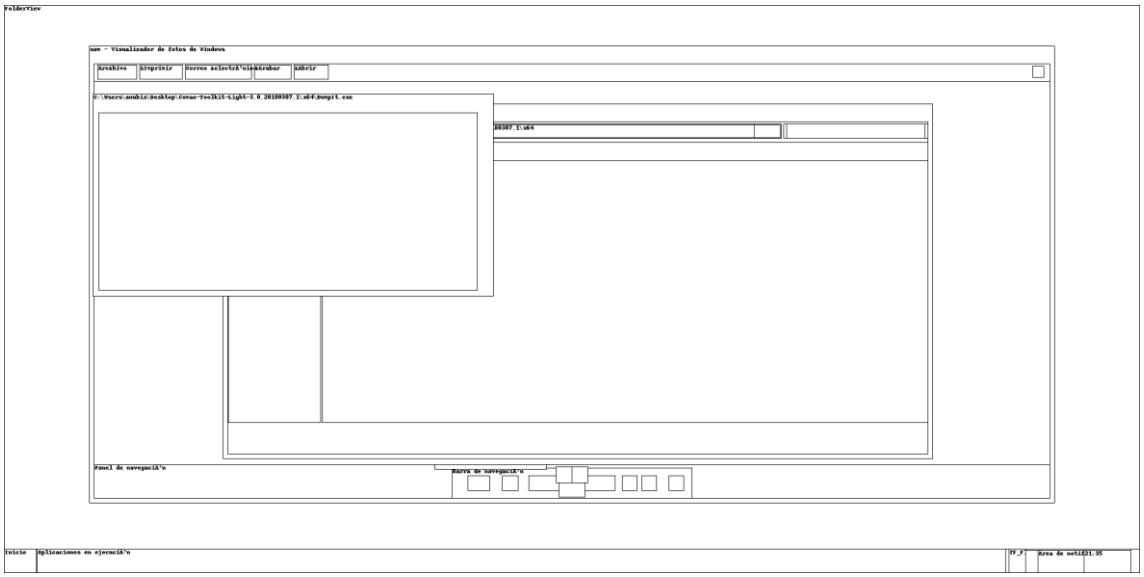
Detectamos que se había ejecutado anteriormente dos veces el proceso DumpIt.exe

Analizando los procesos no se detecta ningún proceso que llame la atención.

Intentaremos obtener una reconstrucción del escritorio:

```
>volatility -f morfeo.dmp --profile=Win7SP1x64 screenshot -D /output/
```

En la reconstrucción observamos que tiene abierto el "*Visualizador de fotos de Windows*" y la foto de nombre "*uam*", así que ya sabemos por dónde tirar:



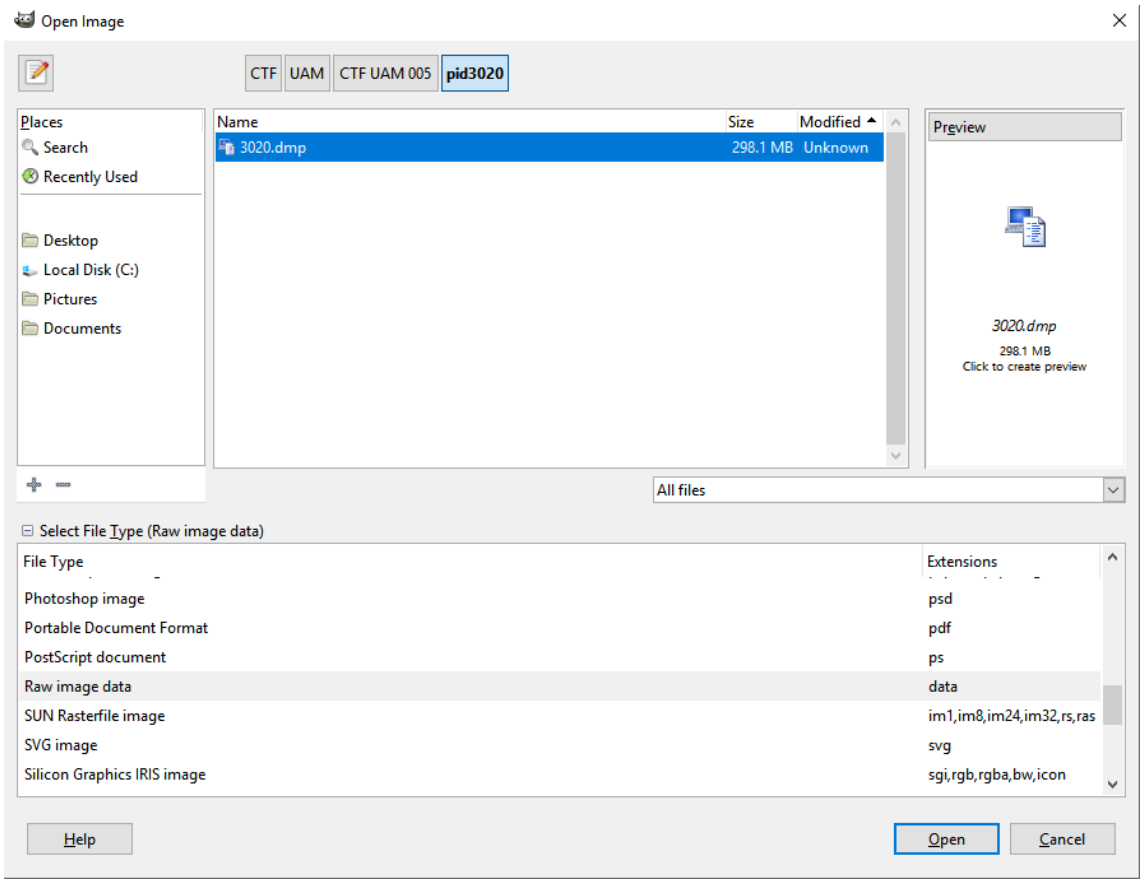
Vamos a intentar reconstruir esta imagen, sabiendo que el proceso es '*dllhost.exe*', así que obtenemos su PID (3020):

```
 > volatility -f morfeo.dmp --profile=Win7SP1x64 pslist --name dllhost.exe
Volatility Foundation Volatility Framework 2.6
Offset(V)          Name                    PID   PPID   Thds     Hnds   Sess  Wow64 Start
------------------ -------------------- ------ ------ ------ -------- ------ ------ ------------------------------
-------------
0xfffffa80037e7630 dllhost.exe            3020    636     10      167      1      0 2018-03-12 20:33:58 UTC+0000
```

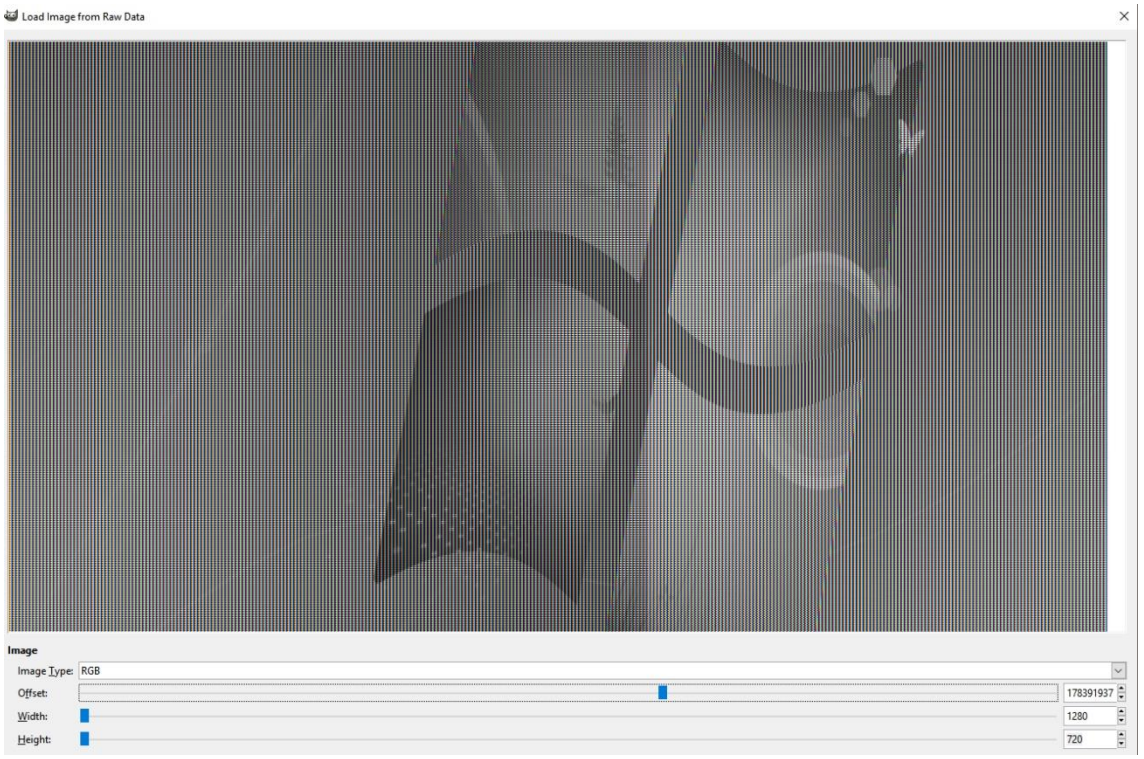
Volcamos todos los datos del proceso para su posterior análisis

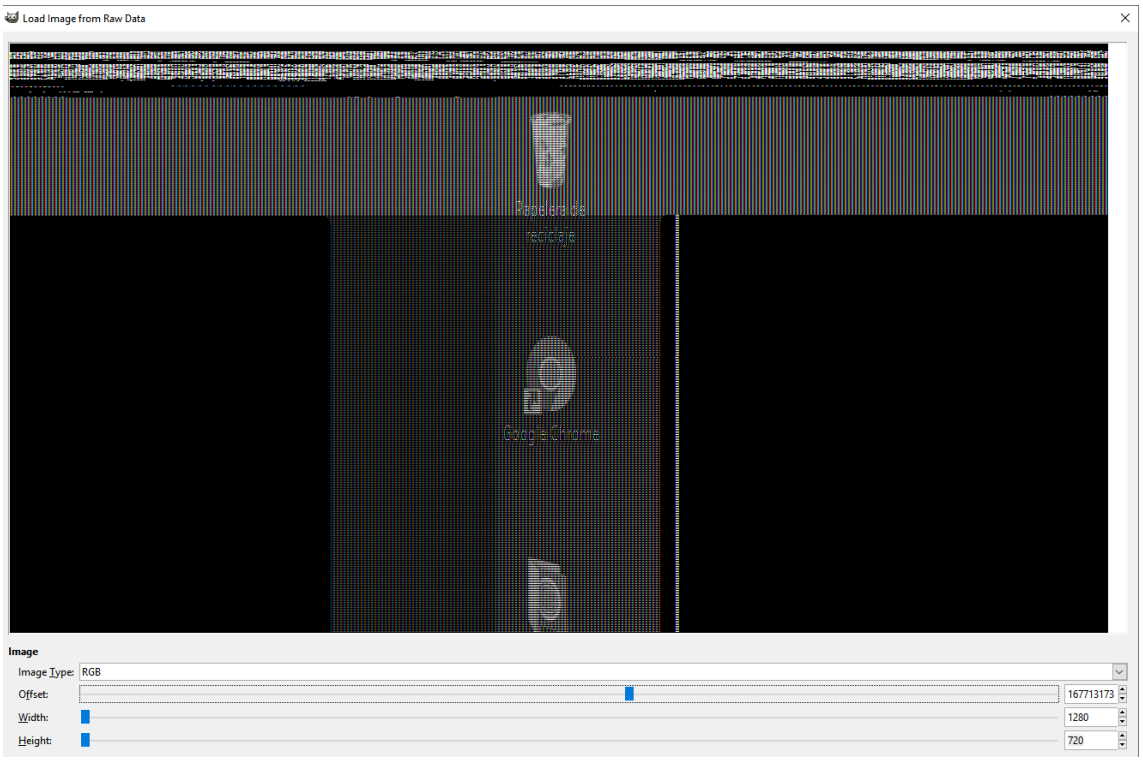
```
>volatility -f morfeo.dmp --profile=Win7SP1x64 memdump -p 3020 --dump ./pid3020/
Volatility Foundation Volatility Framework 2.6
************************************************************************
Writing dllhost.exe [  3020] to 3020.dmp
```

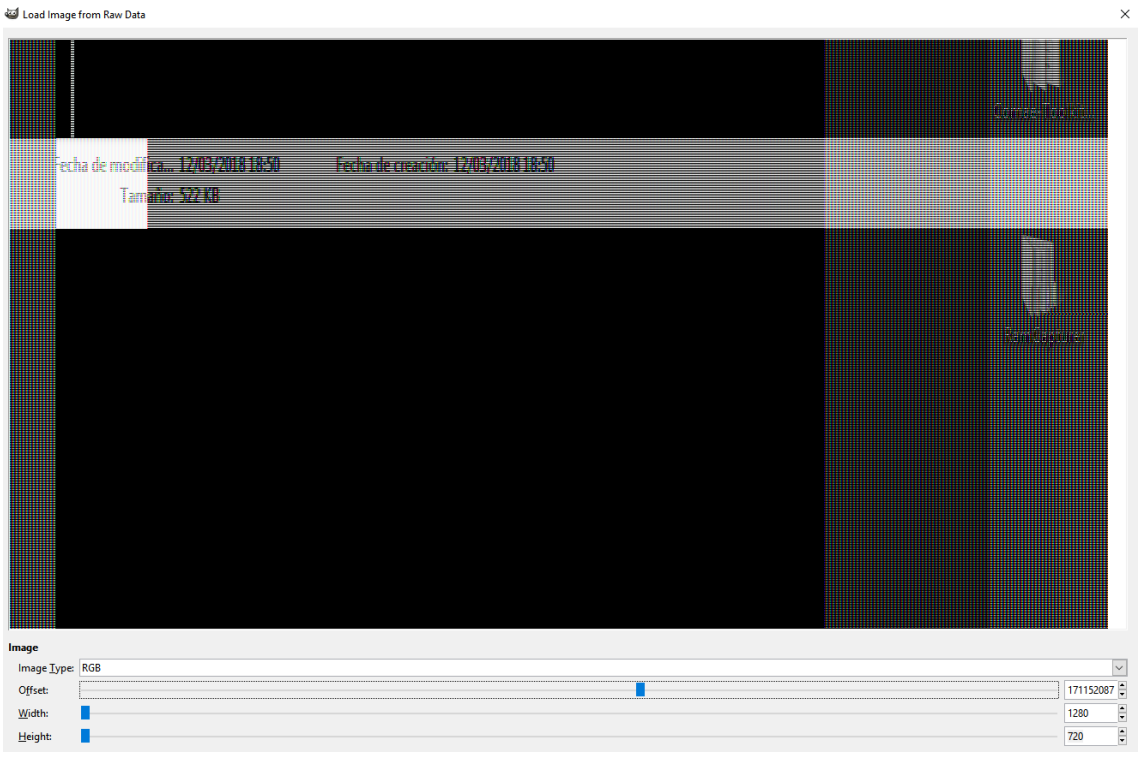
Obtenido el Dump del proceso, intentaremos regenerar con el Gimp el escritorio. Para eso abrimos el fichero '*.dmp*' generado como imagen Raw. Ponemos la resolución a 1280 x 720 y vamos mirando a ver si podemos encontrar un '*screenshot'* de la pantalla:



Solamente podemos regenerar un par de imágenes con el fondo de pantalla y algunos iconos, pero nada que nos sirva:




Load Image from Raw Data
Papelera de reciclaje
Google Chrome
Image
Image Type: RGB
Offset: 167713173
Width: 1280
Height: 720


Load Image from Raw Data
Fecha de modifica... 12/03/2018 18:50
Fecha de creación: 12/03/2018 18:50
Tamaño: 522 KB
Image
Image Type: RGB
Offset: 171152087
Width: 1280
Height: 720

Vamos a mirar si hay algo en el portapapeles. La única sesión que tiene algo es la 1, pero tampoco sabemos avanzar a partir de aquí ya que no sabemos su formato ('*Unknow*')

```
>volatility -f morfeo.dmp --profile=Win7SP1x64 wndscan
Volatility Foundation Volatility Framework 2.6
**************************************************
WindowStation: 0x7e597d30, Name: WinSta0, Next: 0x0
SessionId: 1, AtomTable: 0xfffff8a005d4e110, Interactive: True
Desktops: Default, Disconnect, Winlogon
ptiDrawingClipboard: pid - tid -
spwndClipOpen: 0x0, spwndClipViewer: 0x0
cNumClipFormats: 5, iClipSerialNumber: 2
pClipBase: 0xfffff900c1b81da0, Formats: Unknown choice 3248575072,Unknown
choice 0,Unknown choice 3261317800,Unknown choice 3222142992,Unknown
choice 5000
**************************************************
```

Los siguientes procesos también están involucrados en las tareas de copiar y pegar:

```
> volatility -f morfeo.dmp --profile=Win7SP1x64 wintree | grep
CLIPBRDWNDCLASS
Volatility Foundation Volatility Framework 2.6
.#10084  explorer.exe:1996 CLIPBRDWNDCLASS
.#10122  explorer.exe:1996 CLIPBRDWNDCLASS
.#1013c  VBoxTray.exe:1792 CLIPBRDWNDCLASS
.#10206  explorer.exe:1996 CLIPBRDWNDCLASS
.#10084  explorer.exe:1996 CLIPBRDWNDCLASS
.#10122  explorer.exe:1996 CLIPBRDWNDCLASS
.#1013c  VBoxTray.exe:1792 CLIPBRDWNDCLASS
.#10206  explorer.exe:1996 CLIPBRDWNDCLASS
```

Así que callejón sin salida. Volvemos al PID 3020 (dllhost.exe) y observamos que al realizar la opción '*pstree'*, éste cuelga del proceso explorer.exe (PID 1966).

Realizamos el mismo procedimiento anterior para ver si podemos extraer un screenshot para el proceso explorer.exe pero obtenemos el mismo resultado.

Procederemos al volcado de los ficheros cacheados en memoria del PID 1966 (explorer.exe) a ver si obtenemos alguna pista. Por cada fichero, se extrae la sección de datos del archivo, la sección de imagen y la '*SharedCacheMap*'.

```
> volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles -n -p 1996 -D ./pid1996/
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0xfffffa8002db6210   1996   \Device\HarddiskVolume2\Windows\es-ES\explorer.exe.mui
DataSectionObject 0xfffffa8002db6950   1996   \Device\HarddiskVolume2\Windows\System32\es-ES\setupapi.dll.mui
DataSectionObject 0xfffffa800381b8d0   1996   \Device\HarddiskVolume2\Windows\System32\es-ES\shell32.dll.mui
DataSectionObject 0xfffffa80038a36a0   1996   \Device\HarddiskVolume2\Windows\System32\es-ES\msctf.dll.mui
DataSectionObject 0xfffffa800384cdd0   1996   \Device\HarddiskVolume2\Windows\Fonts\StaticCache.dat
SharedCacheMap 0xfffffa800384cdd0   1996   \Device\HarddiskVolume2\Windows\Fonts\StaticCache.dat
DataSectionObject 0xfffffa80037d92d0   1996   \Device\HarddiskVolume2\Windows\winsxs\amd64_microsoft.windows.c..-controls.resources_
6595b64144ccf1df_6.0.7600.16385_es-es_103af8cc43d0a688\comctl32.dll.mui
DataSectionObject 0xfffffa80038ac920   1996   \Device\HarddiskVolume2\Windows\System32\es-ES\msutb.dll.mui
DataSectionObject 0xfffffa80038c5b40   1996   \Device\HarddiskVolume2\Windows\System32\es-ES\explorerframe.dll.mui
DataSectionObject 0xfffffa80038c68b0   1996   \Device\HarddiskVolume2\Windows\System32\es-ES\authui.dll.mui
DataSectionObject 0xfffffa80038d0530   1996   \Device\HarddiskVolume2\Windows\System32\es-ES\propsys.dll.mui
DataSectionObject 0xfffffa8002dc8070   1996   \Device\HarddiskVolume2\Users\anubis\AppData\Local\Microsoft\Windows\Explorer\thumbcac
he_idx.db
DataSectionObject 0xfffffa8002c4ed10   1996   \Device\HarddiskVolume2\Users\anubis\AppData\Local\Microsoft\Windows\Explorer\thumbcac
he_256.db
DataSectionObject 0xfffffa8003bb7f20   1996   \Device\HarddiskVolume2\Users\anubis\AppData\Local\Microsoft\Windows\Explorer\thumbcac
he_sr.db
DataSectionObject 0xfffffa8003201810   1996   \Device\HarddiskVolume2\Users\anubis\AppData\Local\Microsoft\Windows\Explorer\thumbcac
he_32.db
DataSectionObject 0xfffffa80034cf070   1996   \Device\HarddiskVolume2
DataSectionObject 0xfffffa80037df340   1996   \Device\HarddiskVolume2\Users\anubis\AppData\Local\Microsoft\Windows\Explorer\thumbcac
he_1024.db
```

Revisando algunos de ellos vemos que hay ficheros del tipo *thumbcache* (que al abrirlos solo nos muestran iconos), y también vemos algún index.dat. Al abrirlo nos muestra información de la ubicación de un fichero '*uam.jpg*' ubicado en el escritorio del usuario *anubis*:

```
00004fe0:  efbe adde efbe adde efbe adde efbe adde efbe adde efbe adde  :......................
00004ff8:  efbe adde efbe adde 5552 4c20 0200 0000 50a8 afa2 2aba d301  :........URL ....P...*...
00005010:  50a8 afa2 2aba d301 874c 558e 0000 0000 0000 0000 0000 0000  :P...*....LU............
00005028:  0000 0000 0000 0000 6000 0000 6800 0000 fe00 1010 0000 0000  :........`...h...........
00005040:  0100 2000 a000 0000 1400 0000 0000 0000 6c4c 558e 0100 0000  :.. ............LLU.....
00005058:  0000 0000 0000 0000 0000 0000 efbe adde 5669 7369 7465 643a  :................Visited:
00005070:  2061 6e75 6269 7340 6669 6c65 3a2f 2f2f 433a 2f55 7365 7273  : anubis@file:///C:/Users
00005088:  2f61 6e75 6269 732f 4465 736b 746f 702f 7561 6d2e 6a70 6700  :/anubis/Desktop/uam.jpg.
000050a0:  1000 0200 0000 0010 0000 0000 0100 0000 0000 0000 efbe adde  :......................
000050b8:  efbe adde efbe adde efbe adde efbe adde efbe adde efbe adde  :......................
000050d0:  efbe adde efbe adde efbe adde efbe adde efbe adde efbe adde  :......................
000050e8:  efbe adde efbe adde efbe adde efbe adde efbe adde efbe adde  :......................
00005100:  5552 4c20 0200 0000 60bb 11a8 46b9 d301 60bb 11a8 46b9 d301  :URL ....`...F...`...F...
00005118:  864c d774 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000  :.L.t....................
00005130:  6000 0000 6800 0000 fe00 1010 0000 0000 0100 2000 ac00 0000  :`...h.............. ....
00005148:  1400 0000 0000 0000 6b4c d774 0200 0000 0000 0000 0000 0000  :........kL.t............
00005160:  0000 0000 efbe adde 5669 7369 7465 643a 2061 6e75 6269 7340  :........Visited: anubis@
00005178:  6669 6c65 3a2f 2f2f 433a 2f55 7365 7273 2f61 6e75 6269 732f  :file:///C:/Users/anubis/
00005190:  446f 776e 6c6f 6164 732f 5261 6d43 6170 7475 7265 722e 7a69  :Downloads/RamCapturer.zi
000051a8:  7000 adde 1000 0200 0000 0010 0000 0000 0200 0000 0000 0000  :p.......................
000051c0:  efbe adde efbe adde efbe adde efbe adde efbe adde efbe adde  :......................
000051d8:  efbe adde efbe adde efbe adde efbe adde efbe adde efbe adde  :......................
000051f0:  efbe adde efbe adde efbe adde efbe adde 5552 4c20 0200 0000  :................URL ....
00005208:  5016 4673 41ba d301 5016 4673 41ba d301 874c 40a4 0000 0000  :P.FsA...P.FsA....L@.....
00005220:  0000 0000 0000 0000 0000 0000 0000 0000 6000 0000 6800 0000  :................`...h...
00005238:  fe00 1010 0000 0000 0100 2000 a000 0000 1400 0000 0000 0000  :.......... ............
00005250:  6c4c 40a4 0200 0000 0000 0000 0000 0000 0000 0000 efbe adde  :lL@.....................
00005268:  5669 7369 7465 643a 2061 6e75 6269 7340 6669 6c65 3a2f 2f2f  :Visited: anubis@file:///
00005280:  433a 2f55 7365 7273 2f61 6e75 6269 732f 4465 736b 746f 702f  :C:/Users/anubis/Desktop/
00005298:  7561 6d2e 6a70 6700 1000 0200 0000 0010 0000 0000 0200 0000  :uam.jpg.................
000052b0:  0000 0000 efbe adde efbe adde efbe adde efbe adde efbe adde  :......................
```

Así que buscamos ficheros con extensión '*.jpg*' o nombre '*uam*':

```
> volatility -f morfeo.dmp --profile=Win7SP1x64 filescan | grep jpg
Volatility Foundation Volatility Framework 2.6
0x000000007ddb0540     16      0 -W-r-- \Device\HarddiskVolume2\Users\anubis\Desktop\uam.jpg.jpgVirtualBox Dropped Files\2018-03
-12T20_33_51.765201500Z\uam (2).jpg
0x000000007dedea70     11      0 R--r-- \Device\HarddiskVolume2\Users\anubis\AppData\Roaming\Microsoft\Windows\Themes\Transcoded
Wallpaper.jpg
0x000000007df0b350     16      0 R--r-- \Device\HarddiskVolume2\Users\anubis\Desktop\uam.jpg
0x000000007e1c2940     16      0 -W---- \Device\HarddiskVolume2\$Recycle.Bin\S-1-5-21-3107480389-3703109444-792598018-1001\$IGKW
FUM.jpg
> volatility -f morfeo.dmp --profile=Win7SP1x64 filescan | grep uam
Volatility Foundation Volatility Framework 2.6
0x000000007ddb0540     16      0 -W-r-- \Device\HarddiskVolume2\Users\anubis\Desktop\uam.jpg.jpgVirtualBox Dropped Files\2018-03
-12T20_33_51.765201500Z\uam (2).jpg
0x000000007ddc5070      2      0 RW-rw- \Device\HarddiskVolume2\Users\anubis\AppData\Roaming\Microsoft\Windows\Recent\uam (2).ln
k
0x000000007df0b350     16      0 R--r-- \Device\HarddiskVolume2\Users\anubis\Desktop\uam.jpg
```

Ya parece que lleguemos al final. Vemos que hay un fichero '*uam.jpg*' en el escritorio y otro fichero '*uam (2).lnk*' que es un acceso directo.

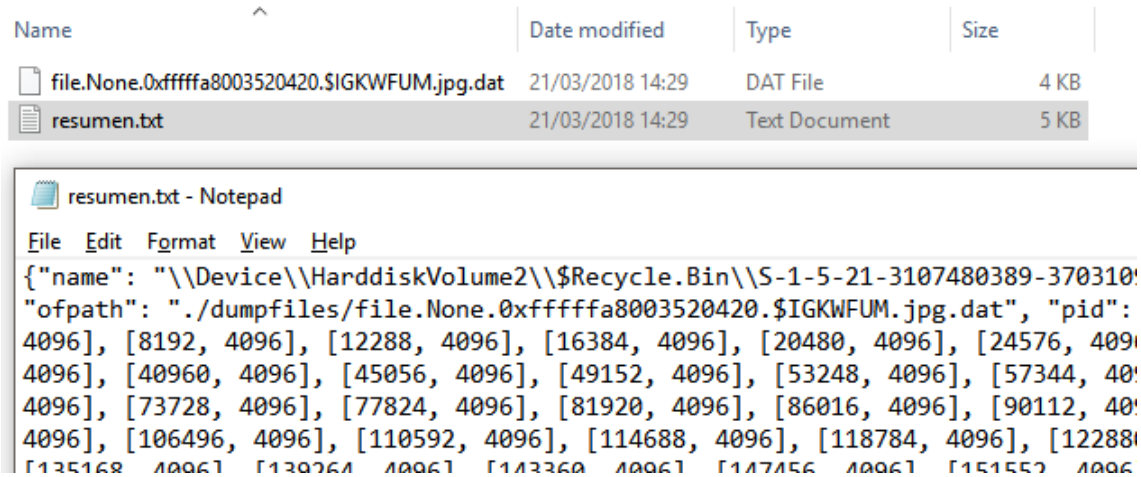
Intentamos extraerlos, pero no extrae nada:

```
>volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles -r jpg$ -i -n -u
-S resumen.txt -D ./
```

Intentamos poniendo el offset físico:

```
> volatility -f morfeo.dmp --profile=Win7SP1x64 dumpfiles -Q 0x000000007e1c2940 -n -u -S ./dumpfiles/resumen.txt -D ./du
mpfiles/
Volatility Foundation Volatility Framework 2.6
DataSectionObject 0x7e1c2940   None   \Device\HarddiskVolume2\$Recycle.Bin\S-1-5-21-3107480389-3703109444-792598018-1001
\$IGKWFUM.jpg
```

Tampoco conseguimos descargar el fichero para poderlo abrir:



Probaremos con la opción '*mftparser'* que escanea potenciales entradas de MFT (ficheros) en memoria:

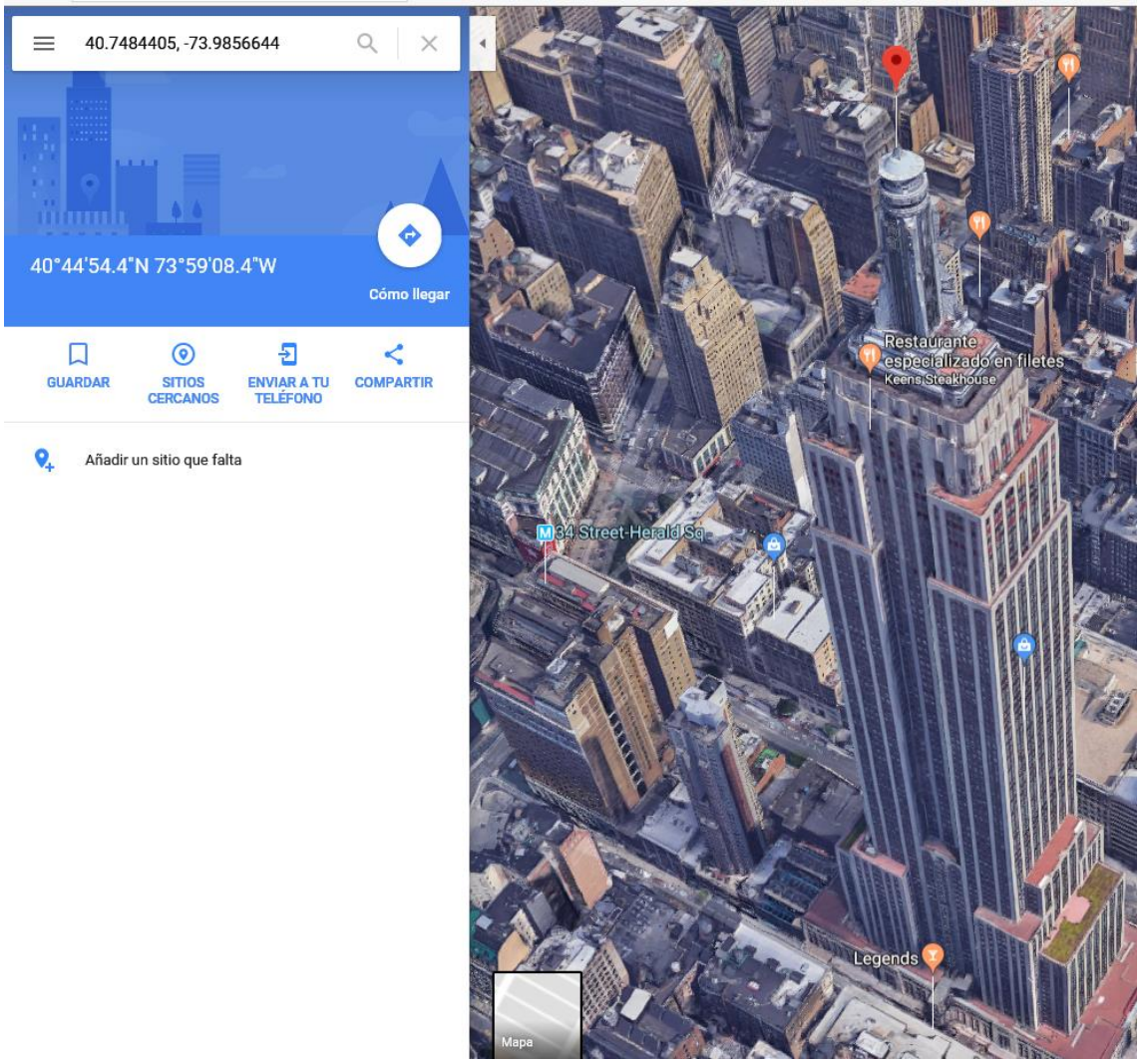
```
>volatility -f morfeo.dmp --profile=Win7SP1x64 mftparser >> mfttable.txt
```

Ahora sí, analizando el fichero resultante y buscando la cadena UAM obtenemos:

```
Users\anubis\AppData\Roaming\Microsoft\Windows\Recent\uam.lnk
Users\anubis\AppData\Roaming\Microsoft\Windows\Recent\uam (2).lnk
Users\anubis\AppData\Local\Temp\VirtualBox Dropped Files\2018-03-12T20_33_51.765201500Z\UAM(2)~1.JPG
$Recycle.Bin\S-1-5-~1\$RGKWFUM.jpg

$FILE_NAME
Creation                      Modified                      MFT Altered                   Access Date
Name/Path
------------------------------ ------------------------------ ------------------
2018-03-12 20:33:51 UTC+0000 2018-03-12 20:33:51 UTC+0000   2018-03-12 20:33:51 UTC+0000   2018-03-12
20:33:51 UTC+0000   Users\anubis\Desktop\uam.jpg
$OBJECT_ID
Object ID: dec70d8f-3426-e811-9f5d-08002736a682
Birth Volume ID: 80000000-a000-0000-0000-180000000100
Birth Object ID: 81000000-1800-0000-3c68-746d6c3e0d0a
Birth Domain ID: 093c6865-6164-3e0d-0a09-093c7469746c
$DATA
0000000000: 3c 68 74 6d 6c 3e 0d 0a 09 3c 68 65 61 64 3e 0d   <html>...<head>.
0000000010: 0a 09 09 3c 74 69 74 6c 65 3e 43 6f 6f 72 64 65   ...<title>Coorde
0000000020: 6e 61 64 61 73 20 64 65 20 4d 6f 72 66 65 6f 3c   nadas.de.Morfeo<
0000000030: 2f 74 69 74 6c 65 3e 0d 0a 09 3c 2f 68 65 61 64   /title>...</head
0000000040: 3e 0d 0a 09 3c 62 6f 64 79 3e 0d 0a 09 09 3c 68   >...<body>....<h
0000000050: 31 3e 34 30 2e 37 34 38 34 34 30 35 2c 20 2d 37   1>40.7484405,.-7
0000000060: 33 2e 39 38 35 36 36 34 34 3c 2f 68 31 3e 0d 0a   3.9856644</h1>..
0000000070: 09 3c 2f 62 6f 64 79 3e 0d 0a 3c 2f 68 74 6d 6c   .</body>..</html

$FILE_NAME
Creation                      Modified                      MFT Altered                   Access Date
Name/Path
------------------------------ ------------------------------ ------------------
2018-03-11 14:38:29 UTC+0000 2018-03-11 14:38:29 UTC+0000   2018-03-12 17:50:40 UTC+0000   2018-03-11
14:38:29 UTC+0000   $Recycle.Bin\S-1-5-~1\$RGKWFUM.jpg
$OBJECT_ID
Object ID: 6f07a6c8-1d26-e811-bfd2-08002736a682
Birth Volume ID: 80000000-8800-0000-0000-180000000100
Birth Object ID: 6e000000-1800-0000-3c68-746d6c3e0d0a
Birth Domain ID: 093c6865-6164-3e0d-0a09-093c7469746c
$DATA
0000000000: 3c 68 74 6d 6c 3e 0d 0a 09 3c 68 65 61 64 3e 0d   <html>...<head>.
0000000010: 0a 09 09 3c 74 69 74 6c 65 3e 55 41 4d 20 46 4c   ...<title>UAM.FL
0000000020: 41 47 3c 2f 74 69 74 6c 65 3e 0d 0a 09 3c 2f 68   AG</title>...</h
0000000030: 65 61 64 3e 0d 0a 09 3c 62 6f 64 79 3e 0d 0a 09   ead>...<body>...
0000000040: 09 3c 68 31 3e 55 41 4d 7b 4e 33 30 5f 69 35 5f   .<h1>UAM{N30_i5_
0000000050: 34 5f 47 30 44 7d 3c 2f 68 31 3e 0d 0a 09 3c 2f   4_G0D}</h1>...</
0000000060: 62 6f 64 79 3e 0d 0a 3c 2f 68 74 6d 6c 3e         body>..</html>
```

En el fichero '*$RGKWFUM.jpg*' de la papelera de reciclaje vemos que contiene una flag del tipo **UAM{N30_i5_4_G0D}** pero puesto que buscamos una localización revisamos los datos del otro fichero '*uam.jpg*' que como título tiene "Coordenadas de Morfeo" y nos da las coordenadas **40.7484405, -73.9856644**.

Asumiendo que esas coordenadas son longitud y latitud, las ponemos en un servicio de coordenadas (por ejemplo, Google Maps) y nos mostrará su ubicación física:



Así que Morfeo está en el Empire State Building, y la flag será:

UAM{Empire_State_Building}

REFERENCIAS

http://resources.infosecinstitute.com/memory-forensics-and-analysis-using-volatility/

https://w00tsec.blogspot.com.es/2015/02/extracting-raw-pictures-from-memory.html

https://github.com/fireeye/Volatility-Plugins/tree/master/shimcachemem

https://github.com/volatilityfoundation/volatility/wiki/Volatility-Usage

https://backtrackacademy.com/articulo/forensic-analysis-extracting-and-reconstructing-images-from-memory-dumps

https://github.com/volatilityfoundation/volatility/wiki/Command-Reference-Gui

https://volatility-labs.blogspot.com.es/2012/09/movp-34-recovering-tagclipdata-whats-in.html

https://downloads.volatilityfoundation.org/releases/2.4/CheatSheet_v2.4.pdf

http://www.pentestingexperts.com/windows-gui-memory-forensics-clipboard-windows-atoms-message-and-event-hooks/

http://www.pentestingexperts.com/windows-gui-forensics-session-objects-window-stations-and-desktop/

https://cqureacademy.com/blog/forensics/memory-dump-analysis

https://steemit.com/security/@nybble/forensic-extracting-files-from-mft-table-with-volatility-part-2-en

https://www.evild3ad.com/956/volatility-memory-forensics-basic-usage-for-malware-analysis/