



ANÁLISIS FORENSE

Unidad 1. Actividad 6



28 DE NOVIEMBRE DE 2023
CARLOS DÍAZ MONTES
ESPECIALIZACIÓN DE CIBERSEGURIDAD

Índice

Ejercicio 1.	¡Error! Marcador no definido.
Ejercicio 2.	¡Error! Marcador no definido.
Ejercicio 3.	¡Error! Marcador no definido.
Ejercicio 4.	¡Error! Marcador no definido.

Cuestiones a resolver

En el Json podemos ver esto:

```
29 },
30 "osVersion": {
31   "buildNumber": 7601,
32   "majorVersion": 6,
33   "minorVersion": 1,
34   "productType": 1,
35   "servicePackMajor": 1,
36   "servicePackMinor": 0,
37   "suiteMask": 256
38 },
39 "serviceInfo": {
```

Buscando por internet podemos ver que es un windows 7.

1. Verifica que los agentes corruptos del FBI no han encontrado y modificado la imagen de memoria para poder invalidarla en un juicio.

Aquí vemos que esta modificado con sha356:

```
14 },
15 "fileInfo": {
16   "fileSize": 2147024896,
17   "sha256": "a518111a8f288d94fb4fb0069e36a884e1483f72b51b876303b6c7cfc945715"
18 },
```

Y para ver si ha sido modificado hay que hacer un hash:

```
(kali@kali) ~[~/volatility]
$ sha256sum /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp
a518111a8f288d94fb4fb0069e36a884e1483f72b51b876303b6c7cfc945715 /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp
```

Como vemos la imagen es igual.

2. Comprueba en el Registro de Windows que la imagen de memoria pertenece al ordenador de Tanner, llamado "DRIVERGAME".

Primero tenemos que usar el hivelist para encontrar el disco virtual de los usuarios:

```
(kali@kali) ~[~/volatility]
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 hivelist
Volatility Foundation Volatility Framework 2.6.1
Virtual Physical Name
-----
0xfffffa802df7010 0+000000005c4d010 \?VC:\System Volume Information\System32\Config\DEFAULT
0xfffffa802e12010 0+0000000040501010 \SystemRoot\System32\Config\DEFAULT
0xfffffa800000010 0+000000007d03f010 [no name]
0xfffffa80000002010 0+00000000524010 \REGISTRY\MACHINE\SYSTEM
0xfffffa80000003010 0+000000002d0e010 \REGISTRY\MACHINE\HARDWARE
0xfffffa800110010 0+0000000044ae010 \SystemRoot\System32\Config\SOFTWARE
0xfffffa800120410 0+0000000067fc010 \Device\HarddiskVolume1\Boot\BCD
0xfffffa80000002010 0+000000002070f30 \REGISTRY\MACHINE\SECURITY
0xfffffa80011a510 0+0000000076f010 \SystemRoot\System32\Config\SAM
0xfffffa8000f0b010 0+0000000026d5010 \?VC:\Windows\ServiceProfiles\NetworkService\NTUSER.DAT
0xfffffa80010b010 0+0000000026d7010 \?VC:\Windows\ServiceProfiles\LocalService\NTUSER.DAT
0xfffffa80013c3010 0+000000001ae5a010 \?VC:\Users\Tanner\AppData\Local\Microsoft\Windows\UsrClass.dat
0xfffffa800160010 0+000000001b043010 \?VC:\Users\Tanner\ntuser.dat
```

Ahora usamos hivedump para encontrar donde registra los nombres de usuarios:

```
(kali@kali) ~[~/volatility]
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 hivedump --hive-offset=0xfffffa8000024010 | grep "ComputerName"
Volatility Foundation Volatility Framework 2.6.1
2022-02-11 17:59:35 UTC+0000 \CMI-Creation[2A7B991-78BE-4F9D-891E-7CB51D4737F5]\ControlSet001\Control\ComputerName
2022-02-11 16:52:11 UTC+0000 \CMI-Creation[2A7B991-78BE-4F9D-891E-7CB51D4737F5]\ControlSet001\Control\ComputerName\ComputerName
2022-02-11 17:59:35 UTC+0000 \CMI-Creation[2A7B991-78BE-4F9D-891E-7CB51D4737F5]\ControlSet001\Control\ComputerName\ComputerName
2022-02-11 17:59:35 UTC+0000 \CMI-Creation[2A7B991-78BE-4F9D-891E-7CB51D4737F5]\ControlSet001\Control\ComputerName\ActiveComputerName
```

Ahora que sabemos la clave usamos el comando print:

```
(kali@kali) ~/volatility
└─$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 printkey -K "ControlSet001\Control\ComputerName\ComputerName"
Volatility Foundation Volatility Framework 2.6.1
Legend: (S) = Stable (V) = Volatile

Registry: \REGISTRY\MACHINE\SYSTEM
Key name: ComputerName (S)
Last updated: 2022-02-11 16:52:11 UTC+0000

Subkeys:
Values:
REG_SZ ComputerName : (S) mmsspy
REG_SZ ComputerName : (S) DRIVERGAME
```

Y ya podemos comprobar que es DRIVERGAME

3. [1,75] Para garantizar que nadie había intervenido el ordenador, comprueba que Tanner es el único usuario que hay en el equipo (aparte de los típicos usuarios por defecto de Windows, tales como “Administrador”, etc.), y obtén su contraseña. Para éste último, debería bastar con usar uno de esos sitios web que invierten hashes. El comisario ya se ocupará del resto.

Primero buscamos el hash Lm:

```
REG_SZ ComputerName : (S) DRIVERGAME
(kali@kali) ~/volatility
└─$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 hashdump
Volatility Foundation Volatility Framework 2.6.1
Administrador:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Invitado:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Tanner:1001:aad3b435b51404eeaad3b435b51404ee:3ec585243c919f4217175e1918e07780:::
HomeGroupUser$:1002:aad3b435b51404eeaad3b435b51404ee:2e94c93e09e3a6ceb22117c9b4afea42:::
```

Ahora podemos ver que el único usuario es tanner (más los que crea por defecto windows)

Ahora para poner la contraseña hay que descifrar el hash:

✓ Found:

3ec585243c919f4217175e1918e07780:abc123.

✗ Left:

? Hash Identifier

aad3b435b51404eeaad3b435b51404ee:3ec585243c919f4217175e1918e07780

4. Encuentra pruebas de quién es el objetivo del próximo golpe, y de que quien da la orden es el Sr. Castaldi (2 pt por averiguar esto, 1 pt por sacar el resto de la conversación).

Con el plugin consoles podemos ver las consolas que estaban abiertas.

```
(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf__home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 console
Volatility Foundation Volatility Framework 2.6.1
*****
ConsoleProcess: conhost.exe Pid: 3412
Console: 0xffff8a6200 CommandHistorySize: 50
HistoryBufferCount: 4 HistoryBufferMax: 4
OriginalTitle: %SystemRoot%\system32\cmd.exe
Title: .\chatApp2.bat
AttachedProcess: java.exe Pid: 4856 Handle: 0x88
AttachedProcess: java.exe Pid: 4276 Handle: 0xe0
AttachedProcess: cmd.exe Pid: 2956 Handle: 0x5c
---
CommandHistory: 0x19e670 Application: java.exe Flags: Allocated, Reset
CommandCount: 2 LastAdded: -1 LastDisplayed: 1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x88
Cmd #0 at 0x192050: Perdono, se?or Castaldi, pero ?Qui?n es el blanco?
Cmd #1 at 0x190570: Tengo que ...
---
CommandHistory: 0x19e490 Application: java.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0xe0
---
```

Después del comando ponemos pstree al final y miramos el 3412, 4856, 4276 y 2956.

```
(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf__home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 pstree
Volatility Foundation Volatility Framework 2.6.1
Name Pid PPid Thds Hnds Time
0xfffffa8001c05060:firefox.exe 2012 1944 77 1618 2022-02-11 18:30:55 UTC+0000
0xfffffa8001fc6b30:firefox.exe 2928 2012 18 284 2022-02-11 18:31:40 UTC+0000
0xfffffa80020e4060:firefox.exe 3960 2012 18 289 2022-02-11 18:31:24 UTC+0000
0xfffffa8001b60060:firefox.exe 2328 2012 18 285 2022-02-11 18:31:40 UTC+0000
0xfffffa8001f75b30:firefox.exe 3884 2012 18 284 2022-02-11 18:30:57 UTC+0000
0xfffffa8002118b30:firefox.exe 2128 2012 18 284 2022-02-11 18:31:38 UTC+0000
0xfffffa800216f340:firefox.exe 1480 2012 6 156 2022-02-11 18:34:31 UTC+0000
0xfffffa80023b9060:firefox.exe 2168 2012 14 270 2022-02-12 13:37:20 UTC+0000
0xfffffa8001fb9b30:firefox.exe 1604 2012 6 165 2022-02-11 18:30:57 UTC+0000
0xfffffa8001d0cb30:firefox.exe 2892 2012 41 432 2022-02-11 18:30:56 UTC+0000
0xfffffa8001eb4b30:firefox.exe 4728 2012 20 301 2022-02-11 18:33:48 UTC+0000
0xfffffa8001d81570:firefox.exe 1956 2012 18 287 2022-02-11 18:30:58 UTC+0000
0xfffffa8002143b30:firefox.exe 3296 2012 18 284 2022-02-11 18:31:25 UTC+0000
0xfffffa8001f9a060:firefox.exe 996 2012 18 284 2022-02-11 18:31:38 UTC+0000
0xfffffa8002161b30:firefox.exe 1724 2012 18 284 2022-02-11 18:31:25 UTC+0000
0xfffffa8001f2a060:firefox.exe 3304 2012 14 264 2022-02-12 13:37:46 UTC+0000
0xfffffa8001b40060:firefox.exe 4880 2012 18 285 2022-02-11 18:33:50 UTC+0000
0xfffffa80025c0b30:firefox.exe 4556 2012 14 264 2022-02-11 20:26:16 UTC+0000
0xfffffa8002580b30:firefox.exe 2000 2012 17 280 2022-02-11 20:22:48 UTC+0000
```

Usamos el comando memdump para descargar los procesos:

```
(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf__home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 memdump -p 3412 -D /media/sf__home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
*****
Writing conhost.exe [ 3412] to 3412.dmp

(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf__home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 memdump -p 4856 -D /media/sf__home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
*****
Writing java.exe [ 4856] to 4856.dmp

(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf__home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 memdump -p 4276 -D /media/sf__home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
*****
Writing java.exe [ 4276] to 4276.dmp

(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf__home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 memdump -p 2956 -D /media/sf__home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
*****
Writing cmd.exe [ 2956] to 2956.dmp
```

Ahora comprobamos cada uno con el comando strings. Primero el 4276:

```
(kali@kali)-[/media/sf__home_kali_Desktop_Unidad1_Actividad_6]
$ strings 4276.dmp | grep "Castaldi"
Castaldi
Castaldi2
Castaldi
Castaldi
Castaldi(
Volatility Foundation Volatility Framework 2.6.1
```

Ahora el 4856:

```
(kali@kali)-[/media/sf_home_kali_Desktop_Unidad1_Actividad_6]
$ strings 4856.dmp | grep "Castaldi"
Castaldi says:
Castaldi+NL
Castaldi
Castaldi says:
Castaldi, pero
Castaldi
Castaldi
Castaldi
or Castaldi, pero
or Castaldi, pero
or Castaldi, pero
or Castaldi, pero
or Castaldi, pero
Castaldi
Castaldi
Castaldi says: Hemos perdido mucho tiempo con el asunto de Maddox, pero ahora vamos a recuperarlo.
As
Castaldi
Castaldi
```

Como vemos con este nos sirve para verlo.

Ahora para ver el contenido completo:

```
(kali@kali)-[/media/sf_home_kali_Desktop_Unidad1_Actividad_6]
$ strings 4856.dmp | grep -A 10 -B 10 "Castaldi"
+-k8
aQ#-
aQ#-
Tengo que...
n es el blanco?
Castaldi says:
No has o
do lo que he dicho? Nos vamos ya, chaval. Ser
mejor que comiences a actuar como un poli si es que quieres conducir el coche para llevar a cabo el plan...
pasar porA
el que conduzca el Cadillac. Todo lo que tiene que hacer es pasar por donde est
Jean Paul y
Pumba! Ya tenemos un nuevo martir y una nueva noticia de primera p
gina. Voy a colocar cuatro coches de refuerzo por si surgiera algo imprevisto. Uno en cada cruce.
T w
QT w
```

```
Castaldi says:
No has o
do lo q
e dicho? Nos vamos ya, chava
mejor que comiences a actuar como un poli si es que quieres conducir el coche para llevaH
cabo
planXew
```

```
Castaldi says:
No has o
do lo que he dicho? Nos vamos ya, chaval. Ser
mejor que comiences a actuar como un poli si es que quieres conducir el coche para llevar a cabo el plan...
pasar porA
el que conduzca el Cadillac. Todo lo que tiene que hacer es pasar por donde est
Jean Paul y
Pumba! Ya tenemos un nuevo martir y una nueva noticia de primera p
gina. Voy a colocar cuatro coches de refuerzo por si surgiera algo imprevisto. Uno en cada cruce.
T w
```

5. A Tanner le enviaron un archivo con una foto (driver_you_are_the_wheelman.jpg) que tenía un mensaje escondido con instrucciones. Estamos interesados en saber si las ha recibido y leído. Para ello, encuentra pruebas de que en el directorio de Descargas de Tanner estaba dicho archivo (1,25 pt), y de que en algún momento lo abrió con el Visualizador de Fotos de Windows, también conocido como Windows Photo Viewer (1,5 pt). Nota: no te mates en buscar el mensaje secreto en la imagen, ya que no existe, haz sólo lo que se te pide.

Usando el comando filescan podemos buscarlo, después usamos grep para simplificar la búsqueda:

```
(kali@kali)~/volatility
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 files
can | grep "Downloads"
Volatility Foundation Volatility Framework 2.6.1
0x000000007f72dd0 2 1 R--rd \Device\HarddiskVolume2\Users\Tanner\Downloads
0x000000007f72f20 2 1 R--rd \Device\HarddiskVolume2\Users\Tanner\Downloads
0x000000007fb05f20 15 0 R--rd \Device\HarddiskVolume2\Users\Tanner\Downloads\desktop.ini
0x000000007fc1d850 2 0 -W-rw- \Device\HarddiskVolume2\Users\Tanner\Downloads\driver_you_are_the_wheelman.jpg
0x000000007fceae0 15 0 R--rd \Device\HarddiskVolume2\Users\Tanner\Downloads\Firefox Installer.exe
```

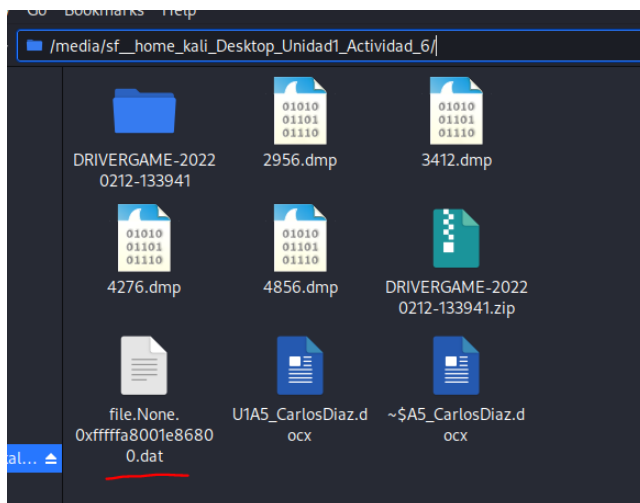
Otra forma de hacerlo es con mftparser

```
(kali@kali)~/volatility
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 mftpa
rser | grep "Downloads"
Volatility Foundation Volatility Framework 2.6.1
2022-02-11 16:53:37 UTC+0000 2022-02-11 16:53:37 UTC+0000 2022-02-11 16:53:37 UTC+0000 2022-02-11 16:53:37 UTC+0000 Users\Tanner\Links\Downloads.lnk
2022-02-11 16:47:27 UTC+0000 2022-02-11 16:47:27 UTC+0000 2022-02-11 16:47:27 UTC+0000 2022-02-11 16:47:27 UTC+0000 Users\Public\Downloads
2022-02-11 16:47:27 UTC+0000 2022-02-11 16:47:27 UTC+0000 2022-02-11 16:47:27 UTC+0000 2022-02-11 16:47:27 UTC+0000 Users\Default\Downloads
2022-02-11 16:53:27 UTC+0000 2022-02-11 16:53:27 UTC+0000 2022-02-11 16:53:27 UTC+0000 2022-02-11 16:53:27 UTC+0000 Users\Tanner\Downloads
2022-02-11 19:25:13 UTC+0000 2022-02-11 19:25:13 UTC+0000 2022-02-11 19:25:13 UTC+0000 2022-02-11 19:25:13 UTC+0000 Users\Tanner\Downloads\APACHE-1.4-B
\APACHE-1.4\lib\JAVAAXA-1.JAR
```

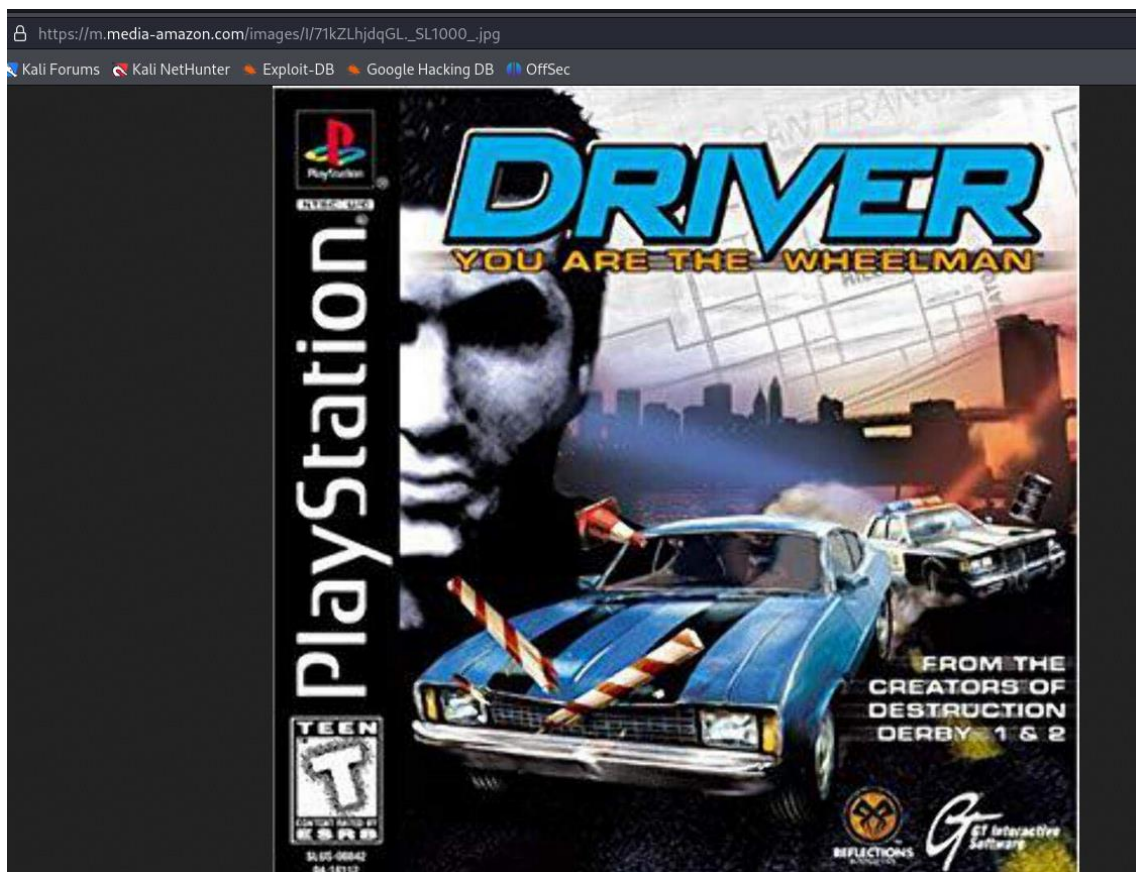
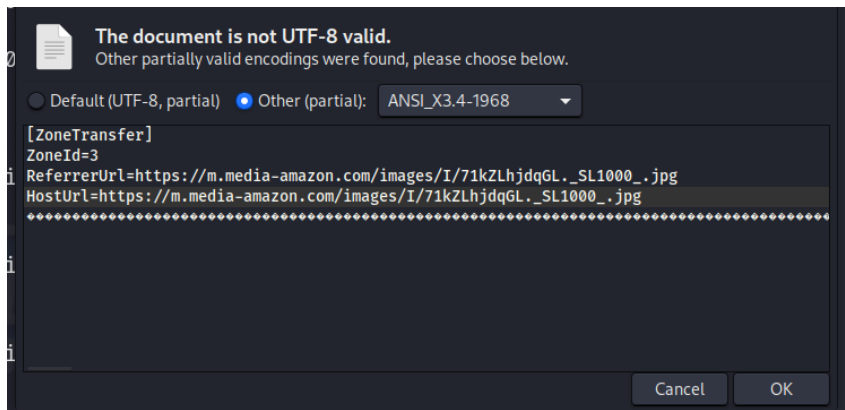
Ahora usamos dumpfiles para extraer los ficheros:

```
(kali@kali)~/volatility
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 dumpf
iles -Q 0x000000007fc1d850 -D /media/sf_home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x7fc1d850 None \Device\HarddiskVolume2\Users\Tanner\Downloads\driver_you_are_the_wheelman.jpg
```

Como vemos en nuestra carpeta compartida, tenemos ahora un .dat:



Ahora al abrirlo nos saldrá un enlace y podremos ver la imagen:



En nuestro caso, es el proceso con PID 2612. Podemos ver que dicho proceso tiene varios ficheros abiertos, y uno de ellos es \Device\HarddiskVolume2\Program Files\Windows Photo Viewer\es-ES\PhotoViewer.dll.mui, pero desgraciadamente ninguno es la imagen que buscamos:

```
(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 handle
es -p 2612 -t FILE
Volatility Foundation Volatility Framework 2.6.1
```

Offset(V)	Pid	Handle	Access	Type	Details
0xfffffa8001ab7610	2612	0xc	0*100020	File	\Device\HarddiskVolume2\Windows\System32
0xfffffa8001d3ff20	2612	0*5c	0*100001	File	\Device\KsecDD
0xfffffa8001fe5320	2612	0*12c	0*100020	File	\Device\HarddiskVolume2\Windows\winsxs\amd64_microsoft.windows.gdiplus_6595b64144ccf1df_1.1.7601.17514_none_2b24536c71ed437a
64144ccf1df_1.1.7601.17514_none_2b24536c71ed437a	2612	0*168	0*100020	File	\Device\HarddiskVolume2\Windows\winsxs\amd64_microsoft.windows.common-contro
ls_6595b64144ccf1df_6.0.7601.17514_none_fa396087175ac9ac	2612	0*1bc	0*120089	File	\Device\HarddiskVolume2\Program Files\Windows Photo Viewer\es-ES\PhotoViewer
0xfffffa800205f920	2612	0*240	0*120089	File	\Device\HarddiskVolume2\Windows\Fonts\StaticCache.dat
0xfffffa80021c8e20	2612	0*250	0*100020	File	\Device\HarddiskVolume2\Windows\winsxs\amd64_microsoft.windows.common-contro
ls_6595b64144ccf1df_6.0.7601.17514_none_fa396087175ac9ac	2612				

Volcamos el espacio de memoria:

```
(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 memdump -p 2612 -D /media/sf_home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
*****
Writing dllhost.exe [ 2612] to 2612.dmp
```

Para comprobar si ha sido ejecutado, usamos strings sobre ese proceso:

```
(kali@kali)-[~/media/sf_home_kali_Desktop_Unidad1_Actividad_6]
$ strings /media/sf_home_kali_Desktop_Unidad1_Actividad_6/2612.dmp | grep "driver_you_are_the_wheelman.jpg" -A 1 -B 1
Acti
driver_you_are_the_wheelman.jpg
MRUListEx
driver_you_are_the_wheelman.jpg
OpenWithList
~
~{~n~a~T
file:///C:/Users/Tanner/Downloads/driver_you_are_the_wheelman.jpg
{"state":1,"endTime":1644604442144,"fileSize":9130223}
{"state":1,"endTime":1644673102685,"fileSize":134121}
file:///C:/Users/Tanner/Downloads/driver_you_are_the_wheelman.jpg
{"state":1,"endTime":1644604442144,"fileSize":9130223}
```

Miscelanea

¿Es posible obtener el contenido del fichero chatApp2.bat que se menciona al emplear el plugin consoles?

```
(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 filescan -p 2612 -D /media/sf_home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
0x000000007f22b660 16 0 RW \Device\HarddiskVolume2\DoubleRatchet\target\classes\chatApp2.bat
```

Ahora podemos descargar con dumpfiles y ver su contenido:

```
(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf_home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 dumpfiles -Q 0x000000007f22b660 -D /media/sf_home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x7f22b660 None \Device\HarddiskVolume2\Users\Tanner\Downloads\driver_you_are_the_wheelman.jpg
```

Como se puede apreciar, la aplicación recibe como parámetro un fichero llamado party2.properties. Podemos repetir el proceso e intentar descargarlo:

```
(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf__home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 filescan | grep "party2.properties"
Volatility Foundation Volatility Framework 2.6.1
0x000000007f749620 16 0 RW-- \Device\HarddiskVolume2\DoubleRatchet\target\classes\party2.properties

(kali@kali)-[~/volatility]
$ python2.7 vol.py -f /media/sf__home_kali_Desktop_Unidad1_Actividad_6/DRIVERGAME-20220212-133941/DRIVERGAME-20220212-133941.dmp --profile=Win7SP1x64 dumpfiles -Q 0x000000007f749620 -D /media/sf__home_kali_Desktop_Unidad1_Actividad_6/
Volatility Foundation Volatility Framework 2.6.1
DataSectionObject 0x7f749620 None \Device\HarddiskVolume2\DoubleRatchet\target\classes\party2.properties
```

Ahora vemos el contenido del archivo descargado:

```
(kali@kali)-[/media/sf__home_kali_Desktop_Unidad1_Actividad_6]
$ cat file.None.0xfffffa8002052140.dat
# General configuration
# -----
party/name=Tanner
party/actsAsParty1=false
#party/verbose=true
party/verbose=false
#party/mode=mqttv3
#party/mode=mqttv5
party/mode=udpsocket
#party/mode=jms

# Configuration for the UDP socket mode
# -----
localSendingSocket/ipAddress=169.254.56.145
localSendingSocket/port=13342
localReceivingSocket/ipAddress=169.254.56.145
localReceivingSocket/port=13344
remoteReceivingSocket/ipAddress=169.254.170.248
remoteReceivingSocket/port=13343

# Configuration for the JMS mode
# -----
java.naming.provider.url=rmi://127.0.0.1:1099
java.naming.factory.initial=org.exolab.jms.jndi.InitialContextFactory
sendingQueue/name=queue1
receivingQueue/name=queue2

# Configuration for the MQTT mode
# -----
#mqtt/Server=tcp://broker.mqttdashboard.com:1883
mqtt/Server=tcp://52.210.173.185:1883
#mqtt/Server=tcp://mastropiero.det.uvigo.es:1883
#mqtt/Server=tcp://localhost:1883
mqtt/sendingTopic=batman.out
mqtt/receivingTopic=batman.in
#mqtt/sendingTopic=robinSentMessages3
#mqtt/receivingTopic=batmanSentMessages3
mqtt/userName=sinf
mqtt/password=snif20
#mqtt/userName=
#mqtt/password=
```