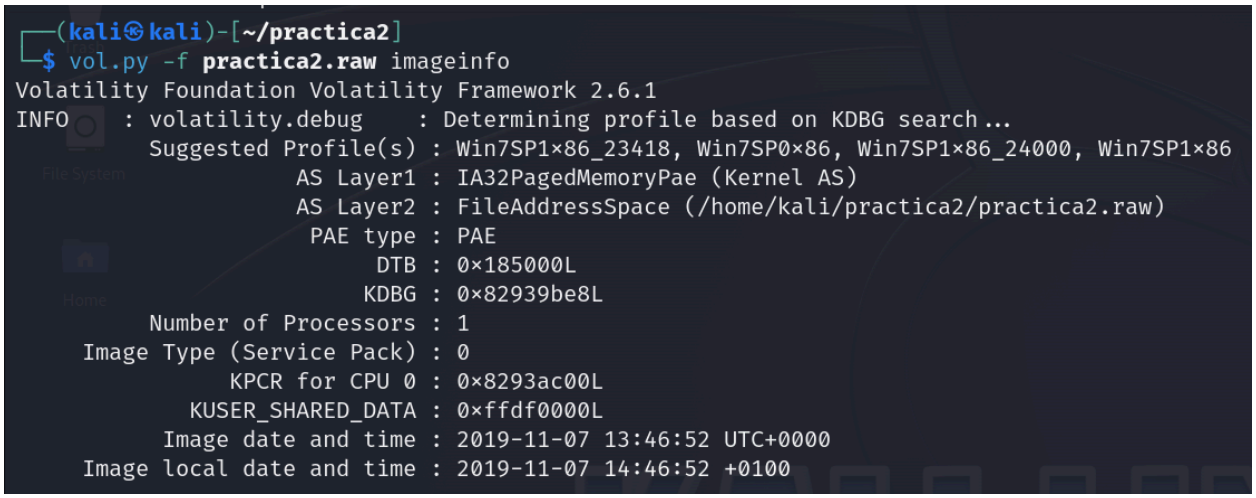
Practica 2

Analizando la RAM 2

Nos enfrentamos a una situación forense en la que la policía ha arrestado a un individuo, y como prueba, contamos con su ordenador encendido. Se ha llevado a cabo la captura de la memoria RAM y un análisis de la memoria no volátil. Durante este proceso, se ha identificado un archivo desconcertante del cual no se tiene información sobre su contenido o propósito.

Contamos con dos archivos: '**practica2.raw**', que constituye la adquisición de la memoria RAM, y '**magic_file**', el cual debemos investigar para determinar su contenido. Nuestro primer paso consistirá en analizar la adquisición de la memoria RAM utilizando Volatility, utilizando el comando **'imageinfo'** para obtener el perfil correspondiente.



A continuación, una vez que conocemos que el perfil es '**Win7SP1=86_23418**', procederemos a analizar los procesos en búsqueda de alguno que nos parezca sospechoso o que llame nuestra atención

```
┌──(kali㉿kali)-[~/practica2]
└─$ vol.py -f practica2.raw --profile=Win7SP1×86_23418 pstree
Volatility Foundation Volatility Framework 2.6.1
Name                                                  Pid   PPid   Thds   Hnds Time
-------------------------------------------------- ------ ------ ------ ------ ----
 0×863cad40:wininit.exe                               396    316      3     75 2019-11-07 13:39:49 UTC+0000
. 0×8641c7b8:services.exe                             492    396      8    216 2019-11-07 13:39:49 UTC+0000
.. 0×86536910:svchost.exe                            1152    492     14    369 2019-11-07 13:39:51 UTC+0000
.. 0×8649da88:taskhost.exe                           1664    492      9    197 2019-11-07 13:39:52 UTC+0000
.. 0×92c58030:WmiApSrv.exe                            256    492      5    109 2019-11-07 13:42:29 UTC+0000
.. 0×864f1210:svchost.exe                            1044    492     17    697 2019-11-07 13:39:51 UTC+0000
.. 0×87ef8030:wmpnetwk.exe                           2204    492     15    450 2019-11-07 13:39:59 UTC+0000
.. 0×86464c08:vmacthlp.exe                            676    492      3     53 2019-11-07 13:39:50 UTC+0000
.. 0×86567ae0:svchost.exe                            1412    492     18    303 2019-11-07 13:39:51 UTC+0000
.. 0×87eb1b60:SearchIndexer.                         2092    492     11    644 2019-11-07 13:39:59 UTC+0000
.. 0×94f0e840:svchost.exe                            2676    492      9    347 2019-11-07 13:40:02 UTC+0000
.. 0×92c55368:svchost.exe                            3092    492      9    299 2019-11-07 13:41:54 UTC+0000
.. 0×87f22c28:vmtoolsd.exe                           1728    492      9    295 2019-11-07 13:39:52 UTC+0000
.. 0×864b1ad0:svchost.exe                             848    492     26    486 2019-11-07 13:39:50 UTC+0000
... 0×863d0868:dwm.exe                               1336    848      3     69 2019-11-07 13:39:51 UTC+0000
.. 0×87fd9740:svchost.exe                            2520    492     20    310 2019-11-07 13:40:01 UTC+0000
.. 0×8649c348:svchost.exe                             804    492     23    571 2019-11-07 13:39:50 UTC+0000
... 0×87f9d378:audiodg.exe                            932    804      6    129 2019-11-07 13:46:43 UTC+0000
.. 0×87e22778:sppsvc.exe                             3076    492      4    143 2019-11-07 13:41:54 UTC+0000
.. 0×865e8030:VGAuthService.                         1580    492      3     86 2019-11-07 13:39:52 UTC+0000
.. 0×8655a030:spoolsv.exe                            1376    492     13    304 2019-11-07 13:39:51 UTC+0000
.. 0×86473030:svchost.exe                             720    492      8    295 2019-11-07 13:39:50 UTC+0000
.. 0×86585b20:msdtc.exe                              2044    492     14    152 2019-11-07 13:39:56 UTC+0000
.. 0×864b7710:svchost.exe                             876    492     35    993 2019-11-07 13:39:50 UTC+0000
.. 0×86514b00:dllhost.exe                            1572    492     15    202 2019-11-07 13:39:54 UTC+0000
.. 0×86483030:svchost.exe                             616    492     10    353 2019-11-07 13:39:50 UTC+0000
... 0×87ecf2c8:WmiPrvSE.exe                          2948    616     10    290 2019-11-07 13:40:04 UTC+0000
... 0×87e77d40:WmiPrvSE.exe                           404    616      9    197 2019-11-07 13:39:54 UTC+0000
. 0×8642a7f8:lsass.exe                                500    396      8    731 2019-11-07 13:39:49 UTC+0000
. 0×8642d030:lsm.exe                                  508    396     10    139 2019-11-07 13:39:49 UTC+0000
 0×861bdd40:csrss.exe                                 336    316      9    491 2019-11-07 13:39:49 UTC+0000
 0×863c9030:csrss.exe                                 388    380     10    203 2019-11-07 13:39:49 UTC+0000
 0×863dbd40:winlogon.exe                              428    380      3    110 2019-11-07 13:39:49 UTC+0000
 0×84f4a8e8:System                                      4      0     95    522 2019-11-07 13:39:46 UTC+0000
. 0×85aacc48:smss.exe                                 248      4      2     29 2019-11-07 13:39:46 UTC+0000
 0×8655a900:explorer.exe                              1384   1328     32    884 2019-11-07 13:39:51 UTC+0000
. 0×92c10030:TrueCrypt.exe                           3612   1384      6    249 2019-11-07 13:40:26 UTC+0000
. 0×94f71640:MagnetRAMCaptu                          3120   1384      6    269 2019-11-07 13:40:06 UTC+0000
. 0×87f4c030:vmtoolsd.exe                            1764   1384      9    201 2019-11-07 13:39:52 UTC+0000
```

El proceso '**TrueCrypt.exe**' ha captado mi atención; ahora vamos a profundizar en nuestra investigación sobre él.

```
 0×8655a900:explorer.exe                              1384   1328     32    884 2019-11-07 13:39:51 UTC+0000
. 0×92c10030:TrueCrypt.exe                           3612   1384      6    249 2019-11-07 13:40:26 UTC+0000
. 0×94f71640:MagnetRAMCaptu                          3120   1384      6    269 2019-11-07 13:40:06 UTC+0000
. 0×87f4c030:vmtoolsd.exe                            1764   1384      9    201 2019-11-07 13:39:52 UTC+0000
```

Para analizar más detalladamente este proceso, procederemos a realizar un volcado en nuestro sistema utilizando el comando **'memdump'**. Utilizaremos la bandera '**-p**' seguida del número de identificación del proceso, en este caso, **3612**, y '**--dump-dir**' para especificar el directorio de volcado, que en este caso será el directorio actual.

```
┌──(kali㉿kali)-[~/practica2]
└─$ vol.py -f practica2.raw --profile=Win7SP1x86_23418 memdump -p 3612 --dump-dir .
Volatility Foundation Volatility Framework 2.6.1
************************************************************************
Writing TrueCrypt.exe [  3612] to 3612.dmp
```

Después de realizar el volcado del proceso, observamos que se ha generado un archivo llamado '**3612.dmp**'. Vamos a emplear el comando **'strings'** para mostrar las secuencias de caracteres imprimibles dentro de este archivo, y también utilizaremos **'less'** para examinar el contenido de manera más cómoda.

```
┌──(kali㉿kali)-[~/practica2]
└─$ ls
3612.dmp  magic_file  practica2.raw

┌──(kali㉿kali)-[~/practica2]
└─$ strings 3612.dmp | less
```

Al examinar el fichero con strings, hemos encontrado el fichero '**magic_file'**, en Tras analizar el archivo con el comando **'strings'**, hemos identificado la presencia del archivo '**magic_file**' en la ruta '**C:/Users/Pepe/Documents/magic_file**'. Con esta información, podemos establecer la asociación entre dicho archivo y el proceso **TrueCrypt**.

```
iw0fz
\??\C:\Users\Pepe\Documents\magic_file
.sjw
4jw@
_1jw@
#4u4
C:\Usersl
bow_Q
iw0fz
C:\Users\Pepe
Crypt\Post-Install Task - Release Notes
uHk0
```

Ahora que hemos establecido la relación entre el archivo '**magic_file**' y TrueCrypt, surge la posibilidad de que sea un volumen cifrado con TrueCrypt. Para explorar las opciones con Volatility, hemos identificado el plugin 'truecryptsummary', el cual nos proporciona información valiosa. Entre los datos revelados, destaca la contraseña utilizada para cifrar el volumen, que en este caso es '**P4ss_H4rd!**'.

```
┌──(kali㉿kali)-[~/practica2]
└─$ vol.py -f practica2.raw --profile=Win7SP1x86 truecryptsummary
Volatility Foundation Volatility Framework 2.6.1
Registry Version      TrueCrypt Version 7.1a
Password              P4ss_H4rd! at offset 0x8b4cc064
Process               TrueCrypt.exe at 0x92c10030 pid 3612
Service               truecrypt state SERVICE_RUNNING
Kernel Module         truecrypt.sys at 0x8b498000 - 0x8b4cf000
Symbolic Link         E: → \Device\TrueCryptVolumeE mounted 2019-11-07 13:46:40 UTC+0000
Symbolic Link         Volume{dc2e13dc-0161-11ea-85ac-000c29021919} → \Device\TrueCryptVolumeE mounted 2019-11-07 13:41:12 UTC+0000
Symbolic Link         E: → \Device\TrueCryptVolumeE mounted 2019-11-07 13:46:40 UTC+0000
File Object           \Device\TrueCryptVolumeE\ at 0x86c230
File Object           \Device\TrueCryptVolumeE\ at 0xf2a0688
Driver                \Driver\truecrypt at 0xf293c80 range 0x8b498000 - 0x8b4ceb80
Device                TrueCryptVolumeE at 0x87f721c8 type FILE_DEVICE_DISK
Container             Path: \??\C:\Users\Pepe\Documents\magic_file
Device                TrueCrypt at 0x85a93b50 type FILE_DEVICE_UNKNOWN
```

Como una verificación adicional, podemos revisar nuevamente el archivo del proceso utilizando el comando **'strings'**, focalizándose en la búsqueda de la contraseña. Al hacerlo, podemos confirmar que la contraseña '**P4ss_H4rd!**' está presente en el archivo.

```
┌──(kali㉿kali)-[~/practica2]
└─$ strings 3612.dmp | grep -Fi P4ss_H4rd! -C 10
7ATkA
llDPNbXb
49##
_3←3
MKe\e
UZH~Z
**68z
FFrhy
f10<$$yy_L
ppP@
P4ss_H4rd!
truecrypt.sys
```

Descargamos TrueCrypt para poder desencriptar y montar el volumen en nuestro sistema.
Usando la password que hemos encontrado.

```
┌──(kali㉿kali)-[~/practica2]
└─$ mkdir mount

┌──(kali㉿kali)-[~/practica2]
└─$ truecrypt magic_file mount/

WARNING: Using TrueCrypt is not secure (see help for more information).

Enter password for /home/kali/practica2/magic_file:
Enter keyfile [none]:
Protect hidden volume (if any)? (y=Yes/n=No) [No]:
```

Ahora tenemos acceso al volumen, confirmando que el archivo '**magic_file**' era, de hecho, un volumen encriptado con TrueCrypt. Al explorar su contenido, descubrimos un archivo de texto simple con la flag '**TrueCrypt_1s_Easy_T0_Br34k**'. Con esto, hemos concluido la práctica.