



Ciberseguridad en Entornos de las Tecnologías de la Información

Módulo 5021 – Incidentes de Ciberseguridad

Ejercicio – Análisis Forense con Volatility

Pliego de Descargo

- *Los ejercicios y conocimientos contenidos en el Módulo 5021, Incidentes de Ciberseguridad, tienen un propósito exclusivamente formativo, por lo que **nunca se deberán utilizar con fines maliciosos o delictivos.***
- *Ni el Ministerio de Educación y Formación Profesional como organismo oficial, ni el CIDEAD como área integrada en el mismo, serán responsables en ningún caso de los daños directos o indirectos que pudieran derivarse del uso inadecuado de las herramientas de hacking ético utilizadas en dichos ejercicios.*





Índice de contenidos

1. Introducción
2. Instalación de Volatility
3. Instalación de Dumpit
4. Descarga de Imagen de Windows 10
5. Análisis de la Imagen con Volatility



1. Introducción

Introducción - Análisis Forense con Volatility

- En muchas ocasiones no es posible analizar un problema o detectar una amenaza estudiando los logs, los ficheros, las bases de datos, etc., pues las trazas del problema sólo están en la memoria RAM de la máquina en ciertos momentos. Esto es, son **volátiles**.
- En estas circunstancias se impone lo que denominamos **análisis forense**. Este tipo de análisis se efectúa en diferido, esto es, capturando instantáneas de la memoria RAM de la máquina, volcándolas a disco y analizándolas en detalle posteriormente.
- La herramienta *open source* más popular para efectuar este tipo de análisis es el Framework Volatility. Este será el instrumento que utilizaremos para este ejercicio de análisis forense (versión 3).
- El framework se puede ejecutar en diferentes plataformas, no obstante, por cuestiones de versatilidad y flexibilidad usaremos una máquina Linux con SO Ubuntu.



2. Instalación de Volatility

Requisitos previos a la Instalación de Volatility – Instalación de Python

- Usaremos Ubuntu como sistema operativo base.
- Volatility está escrito en Python y requiere que esté instalada la última versión del intérprete.
- Actualizaremos tanto Python como Python3

```
sudo apt-get update
sudo apt-get upgrade
sudo apt autoremove
sudo apt-get install python
sudo apt-get install python3
```

```
paco@yvonne: ~
paco@yvonne:~$ uname -a
Linux yvonne 5.4.0-53-generic #59-Ubuntu SMP Wed Oct 21 09:38:44 UTC 2020 x86_64 x86_64 x86_64 GNU/Linux
paco@yvonne:~$ sudo apt-get update
Obj:1 http://archive.ubuntu.com/ubuntu focal InRelease
Obj:2 http://ppa.launchpad.net/openjdk-r/ppa/ubuntu focal InRelease
Obj:3 http://archive.ubuntu.com/ubuntu focal-updates InRelease
Obj:4 http://archive.ubuntu.com/ubuntu focal-backports InRelease
Ign:5 http://packages.linuxmint.com ulyana InRelease
Obj:6 http://archive.canonical.com/ubuntu focal InRelease
Obj:7 http://packages.linuxmint.com ulyana Release
Obj:8 http://security.ubuntu.com/ubuntu focal-security InRelease
Leyendo lista de paquetes... Hecho
paco@yvonne:~$ sudo apt-get upgrade
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Calculando la actualización... Hecho
Los siguientes paquetes se han retenido:
  libegl-mesa0 libgbml libgll-mesa-dri libglapi-mesa libglx-mesa0 libxatracker2 linux-generic linux-headers-generic
  mesa-vdpau-drivers mesa-vulkan-drivers ubuntu-advantage-tools
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 13 no actualizados.
paco@yvonne:~$ sudo apt autoremove
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 13 no actualizados.
paco@yvonne:~$ sudo apt-get install python
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
Nota, seleccionando «python-is-python2» en lugar de «python»
python-is-python2 ya está en su versión más reciente (2.7.17-4).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 13 no actualizados.
paco@yvonne:~$ sudo apt-get install python3
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias
Leyendo la información de estado... Hecho
python3 ya está en su versión más reciente (3.8.2-0ubuntu2).
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 13 no actualizados.
paco@yvonne:~$
```

Clonación de Volatility3 desde GitHub

Repositories2K

Code?

Commits12M

Issues65K

DiscussionsBeta37

Packages5

Marketplace0

Topics43

Wikis1K

Users64

2,970 repository results

volatilityfoundation/volatility

An advanced memory forensics framework

pythonrammemorymalwarevolatility-framework

☆ 4.6k ● Python GPL-2.0 license Updated on 7 Mar

jasonstrimpel/volatility-trading

A complete set of volatility estimators based on Euan Sinclair's Volatility Trading

volatility-tradingpythonoptionstradingvolatilityoptions-trading

☆ 582 ● Python GPL-3.0 license Updated on 25 Mar

volatilityfoundation/volatility3

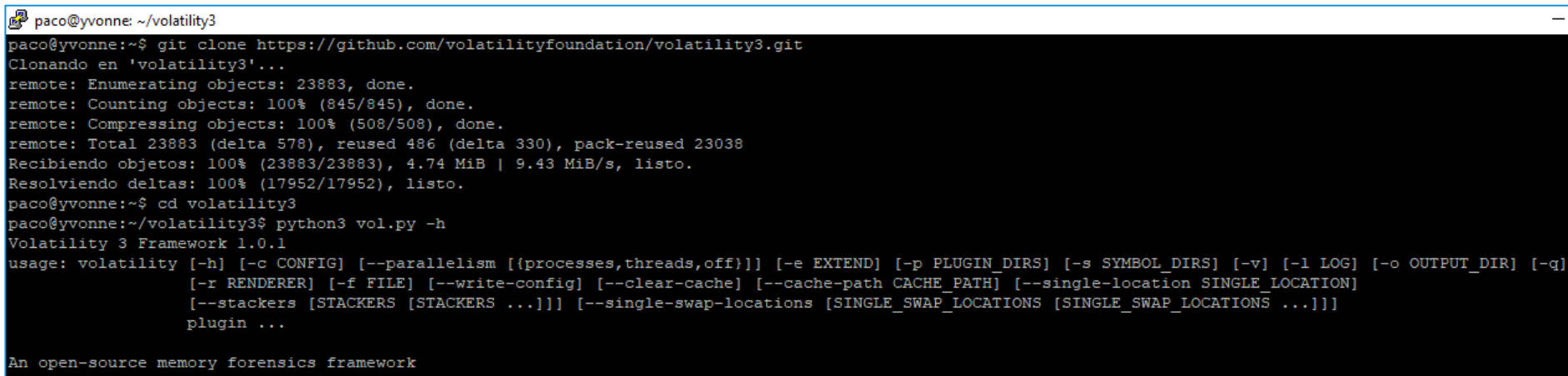
Volatility 3.0 development

☆ 511 ● Python Updated 4 days ago

Clonación de Volatility3 desde GitHub

- Clonación de Volatility 3 y comprobación de la versión (ejecutar con python3)

```
git clone https://github.com/volatilityfoundation/volatility3.git  
cd volatility3  
python3 vol.py -h
```



```
paco@yvonne: ~/volatility3  
paco@yvonne:~$ git clone https://github.com/volatilityfoundation/volatility3.git  
Clonando en 'volatility3'...  
remote: Enumerating objects: 23883, done.  
remote: Counting objects: 100% (845/845), done.  
remote: Compressing objects: 100% (508/508), done.  
remote: Total 23883 (delta 578), reused 486 (delta 330), pack-reused 23038  
Recibiendo objetos: 100% (23883/23883), 4.74 MiB | 9.43 MiB/s, listo.  
Resolviendo deltas: 100% (17952/17952), listo.  
paco@yvonne:~$ cd volatility3  
paco@yvonne:~/volatility3$ python3 vol.py -h  
Volatility 3 Framework 1.0.1  
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND] [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG] [-o OUTPUT_DIR] [-q]  
                [-r RENDERER] [-f FILE] [--write-config] [--clear-cache] [--cache-path CACHE_PATH] [--single-location SINGLE_LOCATION]  
                [--stackers [STACKERS [STACKERS ...]]] [--single-swap-locations [SINGLE_SWAP_LOCATIONS [SINGLE_SWAP_LOCATIONS ...]]]  
                plugin ...  
  
An open-source memory forensics framework
```



3. Instalación de Dumplt

Descarga de DumpIt para efectuar el Volcado de Memoria

- Dumpit es una popular y efectiva herramienta de volcado de memoria de Windows, desarrollada por Matthieu Suiche para Moonsols.
- Se puede descargar desde diferentes sitios en la red, no obstante, por simplicidad y rapidez recurriremos a GitHub:

<https://github.com/thimbleweed/All-In-USB/tree/master/utilities/DumpIt>

- Para inducir las menores alteraciones posibles de la memoria del PC a analizar, lo más recomendable es descargar la herramienta en un PC diferente, grabarla en un pendrive y ejecutarla desde ese pendrive en el PC objetivo, sin grabarla en su disco duro.



4. Descarga de Imagen de Windows 10

Volcado de Memoria RAM desde un PC con Windows

```
C:\Users\franc\Downloads\DumpIt.exe

DumpIt - v1.3.2.20110401 - One click memory memory dumper
Copyright (c) 2007 - 2011, Matthieu Suiche <http://www.msuiche.net>
Copyright (c) 2010 - 2011, MoonSols <http://www.moonsols.com>

Address space size:      10452205568 bytes (   9968 Mb)
Free space size:        45084037120 bytes (  42995 Mb)

* Destination = \\?\C:\Users\franc\Downloads\LAPTOP-DIQVST8G-20210312-172535.raw

--> Are you sure you want to continue? [y/n]
```

```
C:\Users\franc\Downloads\DumpIt.exe

DumpIt - v1.3.2.20110401 - One click memory memory dumper
Copyright (c) 2007 - 2011, Matthieu Suiche <http://www.msuiche.net>
Copyright (c) 2010 - 2011, MoonSols <http://www.moonsols.com>

Address space size:      10452205568 bytes (   9968 Mb)
Free space size:        45084037120 bytes (  42995 Mb)

* Destination = \\?\C:\Users\franc\Downloads\LAPTOP-DIQVST8G-20210312-172535.raw

--> Are you sure you want to continue? [y/n] y
+ Processing... Success.
```

Transmisión del Volcado del PC al Host con FileZilla (Cliente FTP)

- El tamaño del volcado de memoria variará en función del tamaño de la memoria física del PC a analizar, por lo que en algunos casos podremos encontrarnos con ficheros muy voluminosos a transmitir a la máquina Linux en la que efectuaremos el análisis.
- Para trasladar el fichero podremos utilizar un pendrive con formato NTFS (recomendado), que es generalmente reconocido tanto por Windows como por Linux, o bien, sftp o FileZilla (recomendado) si ambas máquinas están conectadas físicamente a la misma LAN (la transmisión inalámbrica suele ser larguísima).
- Por otra parte y como se podrá comprobar fácilmente en la práctica, cuanto mayor sea el tamaño del volcado, mayores serán también los tiempos de proceso de las ejecuciones de Volatility.

Transmisión del Volcado del PC al Host con FileZilla

FileZilla interface showing a connection to a remote server (sftp://192.168.1.76) and a local directory (C:\Users\franc\Downloads\).

Estado: Recuperando el listado del directorio "/home/pi/volatility3"...
Estado: Listing directory /home/pi/volatility3
Estado: Directorio "/home/pi/volatility3" listado correctamente

Sitio local: C:\Users\franc\Downloads\

Sitio remoto: /home/pi/volatility3

Nombre de archivo	Tamaño de archivo	Tipo de archivo	Última modificación
..			
LAPTOP-DIQV5T8G-20210312-172535.raw	10.452.205.568	Archivo RAW	12/03/2021 18:29:02
Industrial_Wireless_52001904_04.pdf	23.174.758	Adobe Acrobat Do...	28/02/2021 11:18:56
MR20 seguridad en dlms cosem.pdf	6.407.947	Adobe Acrobat Do...	22/02/2021 8:23:48
3C16980_MgmtGuide.pdf	4.160.316	Adobe Acrobat Do...	26/02/2021 23:22:22
certsi_cybersecurity_wireless_communications_indust...	3.903.382	Adobe Acrobat Do...	04/03/2021 14:08:45
tas5756m.pdf	3.780.319	Adobe Acrobat Do...	02/03/2021 20:25:05
incibe_guia_sci_003_comunicacionesalemblicas.pdf	3.597.261	Adobe Acrobat Do...	22/02/2021 11:32:57
EKM-Definitive-Guide.pdf	3.425.527	Adobe Acrobat Do...	06/03/2021 10:43:02
incibe-cert_guia_protocolos_smart_grid_2017_v2.pdf	2.690.827	Adobe Acrobat Do...	22/02/2021 8:24:01
metad_plan-director-seguridad.pdf	2.546.381	Adobe Acrobat Do...	04/03/2021 19:47:05

1 archivo seleccionado. Tamaño total: 10.452.205.568 bytes

Nombre de archivo	Tamaño d...	Tipo de archivo	Última modific...	Permisos	Propietario/...
..					
.git		Carpeta de archivos	12/03/2021 17:...	drwxr-xr-x	root root
.github		Carpeta de archivos	12/03/2021 17:...	drwxr-xr-x	root root
build		Carpeta de archivos	12/03/2021 17:...	drwxr-xr-x	root root
development		Carpeta de archivos	12/03/2021 17:...	drwxr-xr-x	root root
dist		Carpeta de archivos	12/03/2021 17:...	drwxr-xr-x	root root
doc		Carpeta de archivos	12/03/2021 17:...	drwxr-xr-x	root root
volatility3		Carpeta de archivos	12/03/2021 17:...	drwxr-xr-x	root root
volatility3.egg-info		Carpeta de archivos	12/03/2021 17:...	drwxr-xr-x	root root
.gitignore	394	Archivo GITIGNORE	12/03/2021 17:...	-rw-r--r--	root root
.readthedocs.yml	532	Archivo YML	12/03/2021 17:...	-rw-r--r--	root root
.style.yapf	7.940	Archivo YAPF	12/03/2021 17:37:37	-rw-r--r--	root root

12 archivos y 8 directorios. Tamaño total: 29.423 bytes

Archivos en cola | Transferencias fallidas (1) | Transferencias satisfactorias



5. Análisis de la Imagen con Volatility

Análisis de Imágenes con Volatility

- El análisis forense de imágenes de memoria es una ciencia extensa, variada y en constante evolución.
- Volatility es el marco de trabajo para análisis forense más avanzado del mundo y además es de código abierto.
- La documentación de Volatility es excelente y está magníficamente estructurada. Se puede consultar en el enlace siguiente:

<https://volatility3.readthedocs.io/en/latest/>

- En este apartado ejecutaremos algunos de los análisis más habituales sobre una imagen de memoria de un PC con Windows 10, que servirán para demostrar la potencia de Volatility.
- Se deberá tener en cuenta que la sintaxis de los comandos de Volatility 3 no coincide con la de las versiones anteriores.
- La ejecución de los comandos de análisis es lenta y requiere muchos recursos de memoria y CPU, máxime si la imagen es de grandes dimensiones.

Imagen Descargada de PC con Windows 10

- Dado que las imágenes de memoria son voluminosas (la del ejemplo ocupa casi 10 GB), la ejecución de Volatility suele durar entre 20 minutos y más de una hora por comando tratándose de una máquina de cierta potencia (en el ejemplo, un Minisforum 123 con un Intel Celeron), por lo que lo habitual es preparar un script con todos los comandos y opciones/plugins a testear y dejarlo corriendo en batch nocturno, almacenando los resultados del trabajo en un fichero de log.
- En cualquier caso, la recomendación habitual es dedicar a esta exploración la máquina más potente de que se disponga, si se quiere completar el trabajo de análisis en un tiempo razonable, lanzando varios de estos batches en paralelo con diferentes capturas de memoria del mismo PC.

```
paco@yvonne: ~/volatility3
paco@yvonne:~/volatility3$ ls -lh /mnt/DISCO
total 9,8G
-rwxr-xr-x 1 paco paco 9,8G may  1 09:59 descarga.raw
drwx----- 2 root root  16K may  1 09:40 lost+found
drwxrwxr-x 3 paco paco 4,0K may  1 09:49 respaldo_wordpress
paco@yvonne:~/volatility3$
```

Funcionamiento Iterativo de Volatility

- Volatility trabaja de forma iterativa, efectuando varias pasadas de análisis sobre el fichero indicado.
- Se podrá comprobar que el índice de proceso alcanza el 100% varias veces al escanear los datos repetidamente.
- Tras varias iteraciones, la aplicación tendrá toda la información necesaria para buscar la información solicitada y empezará a trabajar con ese propósito.

```
paco@yvonne: ~/volatility3
paco@yvonne:~/volatility3$ date
dom 06 jun 2021 19:59:29 CEST
paco@yvonne:~/volatility3$ python3 vol.py -f /mnt/DISCO/descarga.raw windows.handles.Handles
Volatility 3 Framework 1.0.1
Progress: 25.52           Scanning memory_layer using BytesScanner
```

Ayuda en Línea de Volatility

```
paco@yvonne: ~/volatility3
paco@yvonne:~/volatility3$ python3 vol.py --help
Volatility 3 Framework 1.0.1
usage: volatility [-h] [-c CONFIG] [--parallelism [{processes,threads,off}]] [-e EXTEND] [-p PLUGIN_DIRS] [-s SYMBOL_DIRS] [-v] [-l LOG] [-o OUTPUT_DIR] [-q]
                  [-r RENDERER] [-f FILE] [--write-config] [--clear-cache] [--cache-path CACHE_PATH] [--single-location SINGLE_LOCATION]
                  [--stackers [STACKERS [STACKERS ...]]] [--single-swap-locations [SINGLE_SWAP_LOCATIONS [SINGLE_SWAP_LOCATIONS ...]]]
                  plugin ...

An open-source memory forensics framework

optional arguments:
  -h, --help            Show this help message and exit, for specific plugin options use 'volatility <pluginname> --help'
  -c CONFIG, --config CONFIG
                        Load the configuration from a json file
  --parallelism [{processes,threads,off}]
                        Enables parallelism (defaults to off if no argument given)
  -e EXTEND, --extend EXTEND
                        Extend the configuration with a new (or changed) setting
  -p PLUGIN_DIRS, --plugin-dirs PLUGIN_DIRS
                        Semi-colon separated list of paths to find plugins
  -s SYMBOL_DIRS, --symbol-dirs SYMBOL_DIRS
                        Semi-colon separated list of paths to find symbols
  -v, --verbosity        Increase output verbosity
```

Información Básica de una Imagen de Memoria de Windows

```
paco@yvonne: ~/volatility3
paco@yvonne:~/volatility3$ python3 vol.py -f /mnt/DISCO/descarga.raw windows.info
Volatility 3 Framework 1.0.1
Progress: 100.00          PDB scanning finished
Variable      Value

Kernel Base   0xf80164600000
DTB           0xlad000
Symbols file:///home/paco/volatility3/volatility3/symbols/windows/ntkrnlmp.pdb/3FCC539FF307DD2D9C509206D352B9AA-1.json.xz
Is64Bit       True
IsPAE         False
primary 0     WindowsIntel32e
memory_layer  1 FileLayer
KdVersionBlock 0xf8016520f330
Major/Minor   15.19041
MachineType   34404
KeNumberProcessors 4
SystemTime    2021-03-12 17:26:49
NtSystemRoot  C:\WINDOWS
NtProductType NtProductWinNt
NtMajorVersion 10
NtMinorVersion 0
PE MajorOperatingSystemVersion 10
PE MinorOperatingSystemVersion 0
PE Machine     34404
PE TimeDateStamp Tue Sep 8 22:35:03 2082
paco@yvonne:~/volatility3$
```

Volatility Plugins 1/2

```
windows.bigpools.BigPools
    List big page pools.
windows.cachedump.Cachedump
    Dumps lsa secrets from memory
windows.cmdline.CmdLine
    Lists process command line arguments.
windows.dlllist.DllList
    Lists the loaded modules in a particular windows memory image.
windows.driverirp.DriverIrp
    List IRPs for drivers in a particular windows memory image.
windows.driverscan.DriverScan
    Scans for drivers present in a particular windows memory image.
windows.dumpfiles.DumpFiles
    Dumps cached file contents from Windows memory samples.
windows.envvars.Envvars
    Display process environment variables
windows.filescan.FileScan
    Scans for file objects present in a particular windows memory image.
windows.getservicesids.GetServiceSIDs
    Lists process token sids.
windows.getsids.GetSIDs
    Print the SIDs owning each process
windows.handles.Handles
    Lists process open handles.
windows.hashdump.Hashdump
    Dumps user hashes from memory
windows.info.Info
    Show OS & kernel details of the memory sample being analyzed.
windows.lsadump.Lsadump
    Dumps lsa secrets from memory
windows.malfind.Malfind
    Lists process memory ranges that potentially contain injected code.
windows.memmap.Memmap
    Prints the memory map
windows.modscan.ModScan
    Scans for modules present in a particular windows memory image.
windows.modules.Modules
    Lists the loaded kernel modules.
windows.mutantscan.MutantScan
    Scans for mutexes present in a particular windows memory image.
windows.netscan.NetScan
    Scans for network objects present in a particular windows memory image.
```

Plugins 2/2

```
windows.netstat.NetStat
    Traverses network tracking structures present in a particular windows memory image.
windows.poolscanner.PoolScanner
    A generic pool scanner plugin.
windows.privileges.Privs
    Lists process token privileges
windows.pslist.PsList
    Lists the processes present in a particular windows memory image.
windows.psscan.PsScan
    Scans for processes present in a particular windows memory image.
windows.pstree.PsTree
    Plugin for listing processes in a tree based on their parent process ID.
windows.registry.certificates.Certificates
    Lists the certificates in the registry's Certificate Store.
windows.registry.hivelist.HiveList
    Lists the registry hives present in a particular memory image.
windows.registry.hivescan.HiveScan
    Scans for registry hives present in a particular windows memory image.
windows.registry.printkey.PrintKey
    Lists the registry keys under a hive or specific key value.
windows.registry.userassist.UserAssist
    Print userassist registry keys and information.
windows.ssdt.SSDT
    Lists the system call table.
windows.statistics.Statistics
windows.strings.Strings
    Reads output from the strings command and indicates which process(es) each string belongs to.
windows.symlinkscan.SymlinkScan
    Scans for links present in a particular windows memory image.
windows.vadinfo.VadInfo
    Lists process memory ranges.
windows.virtmap.VirtMap
    Lists virtual mapped sections.
```

Visualización de los Pools en el Volcado de Memoria

```
python3 vol.py -f /mnt/DISCO/descarga.raw windows.bigpools.BigPools
```

```
windows.bigpools.BigPools
List big page pools
```

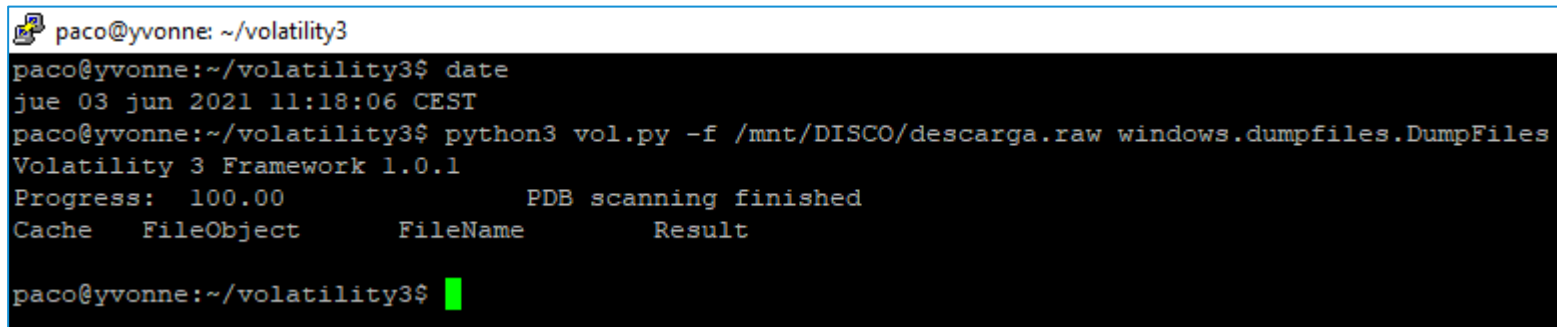
- Primer escenario: el comando va bien y se obtiene la información requerida.
- Como se puede apreciar, la ejecución del comando finaliza correctamente, mostrando todos los pools de memoria en la instantánea capturada.

```
paco@yvonne: ~/volatility3
0xfffffc886ddb3001      CcPL      NonPagedPoolNx  0x1d40
0xfffffb50afb6b9000    CM25      PagedPoolCacheAligned  0x1000
0xfffffb50b13b7b001    CM53      PagedPool       0x1000
0xfffffc886de577000    Mdl       NonPagedPoolNx  0x21b0
0xfffffc886e068a001    smWw      NonPagedPoolNx  0x1000
0xfffffc886e4429000    smNp      NonPagedPoolNx  0x1000
0xfffffb50b00be7001    MmSt      PagedPool       0x1a00
0x0                    NonPagedPool  0x0
0x0                    NonPagedPoolBaseMustSucceed  0x0
0x100000003e           NonPagedPool  0x0
0x0      2            Unknown choice 127  0x1
0x0      p            NonPagedPoolBaseMustSucceed  0x0
0x0      _            NonPagedPoolBaseMustSucceed  0x2
0x824e0000824e        A>      NonPagedPoolBaseMustSucceed  0x0
0x72b      xr          NonPagedPoolBaseMustSucceed  0x72b
0x0                    Unknown choice 127  0x0
0x0                    NonPagedPool  0x7ffc82156720
0x0      dz          NonPagedPoolBaseCacheAlignedMustS  0x0
0x0                    NonPagedPool  0x0
0x7ffc8232d3f0         NonPagedPool  0x16
0x21aa2d95f80         NonPagedPool  0x0
```


Visualización de los Ficheros Cacheados

- El propósito de este comando es descargar los ficheros cacheados en la memoria RAM en el momento del volcado.
- Segundo Escenario: como se puede apreciar, en este caso el comando finaliza bien pero no devuelve información. Esto ocurre algunas veces, ante lo cual hay que procurar descargar la memoria en diferentes momentos, hasta que se obtengan resultados satisfactorios.

```
python3 vol.py -f /mnt/DISCO/descarga.raw windows.dumpfiles.DumpFiles
```



```
paco@yvonne: ~/volatility3
paco@yvonne:~/volatility3$ date
jue 03 jun 2021 11:18:06 CEST
paco@yvonne:~/volatility3$ python3 vol.py -f /mnt/DISCO/descarga.raw windows.dumpfiles.DumpFiles
Volatility 3 Framework 1.0.1
Progress: 100.00          PDB scanning finished
Cache  FileObject      FileName      Result
paco@yvonne:~/volatility3$ █
```

Visualización de los Identificadores de Servicio en los Procesos

- Tercer escenario de ejecución típico en Volatility.
- En este caso se produce un fallo de página, esto es, la página de memoria en la que está la información requerida no está en la RAM sino en disco, en el espacio de Swap.
- En este caso, el framework indica que se podría resolver recapturando la memoria, aunque también indica que puede deberse a una protección del sistema operativo o a un bug del propio framework (en este último caso, incluso recomienda que se reporte, adjuntando la información adecuada).

```
python3 vol.py -f /mnt/DISCO/descarga.raw windows.getsids.GetSIDs
```

```
paco@yvonne: ~/volatility3
paco@yvonne:~/volatility3$ date
dom 06 jun 2021 19:24:18 CEST
paco@yvonne:~/volatility3$ python3 vol.py -f /mnt/DISCO/descarga.raw windows.getsids.GetSIDs
Volatility 3 Framework 1.0.1
Progress: 100.00          PDB scanning finished
PID      Process SID      Name

0         <Array nt_symbols!array (.ImageFileName): primary @ 0xc886b66b87a8 #15>      Token unreadable

Volatility was unable to read a requested page:
Page error 0xfffffffff8 in layer primary (Page Fault at entry 0x55cd063 in table page table)

* Memory smear during acquisition (try re-acquiring if possible)
* An intentionally invalid page lookup (operating system protection)
* A bug in the plugin/volatility3 (re-run with -vvv and file a bug)

No further results will be produced
paco@yvonne:~/volatility3$
```

Comandos con Ejecución Correcta en el Ejemplo

Tras un análisis exploratorio con el volcado de nuestro ejemplo, los siguientes comandos se han ejecutado correctamente y han arrojado resultados concretos:

- windows.bigpools.BigPools
- windows.dumpfiles.DumpFiles
- windows.getservicesids.GetServiceSIDs
- windows.getsids.GetSIDs
- windows.handles.Handles

Comandos con Ejecución Incorrecto en el Ejemplo

En los casos siguientes, los comandos no han finalizado correctamente:

- windows.cachedump.Cachedump (aborta por protección SO)
- windows.cmdline.CmdLine (aborta por protección SO)
- windows.dlllist.DllList (aborta por protección SO)
- windows.driverirp.DriverIrp (abortado manualmente, más de 45 minutos)
- windows.driverscan.DriverScan (abortado manualmente, más de 45 minutos)
- windows.envars.Envvars (aborta por protección SO)
- windows.filescan.FileScan (abortado manualmente, más de 45 minutos)
- windows.registry.certificates.Certificates (abortado manualmente, más de 45 minutos)

Resultados del Análisis

- Los resultados del análisis con Volatility no son determinísticos, pues dependen de la versión de Windows, de los parches que ésta tenga aplicados, del momento concreto de la descarga de información, de los procesos en vuelo en ese instante, de los sistemas antivirus instalados, de las protecciones del sistema operativo, etc.
- En otras palabras, no es posible asegurar que se puedan obtener datos seguros de la ejecución de comandos Volatility con opciones y plugins concretos sino que, como cualquier análisis forense, está sujeto a labores exploratorias y repetitivas sobre diferentes descargas de información, hasta dar con los datos que se necesiten.
- Así pues, Volatility no debe emplearse como única alternativa de análisis forense, pues **puede darse la circunstancia de que no se obtenga ningún resultado en algunos casos**, debido a la dificultad intrínseca del análisis de imágenes de memoria volátil.
- En cualquier caso, cuando se busca de forma sistemática un problema concreto, lo habitual es practicar múltiples descargas de la misma máquina en diferentes momentos, hasta localizar la información deseada.

Bibliografía

- <https://github.com/volatilityfoundation/volatility3>
- <https://volatility3.readthedocs.io/en/latest/>
- <https://github.com/thimbleweed/All-In-USB/tree/master/utilities/Dumplt>
- <https://www.moonsols.com>
- <http://www.msuiche.net/>
- <https://www.volatilityfoundation.org/>
- <https://volatility3.readthedocs.io/en/latest/>
- <https://digital-forensics.sans.org/media/volatility-memory-forensics-cheat-sheet.pdf>
- <https://golang.org/dl/>