IES Zaidín-Vergeles

Análisis Forense en Windows. La volatilidad.

25 de febrero de 2021

Tabla de contenidos

1. **Introducción** 1

2. **Análisis con herramientas del sistema** 3

3. **Análisis forense de memoria RAM** 9

4. **Volatility** 15

5. **Más Volatilidad Comandos y Herramientas** 19

6. **Conclusiones** 21

CAPÍTULO 1

Introducción

La volatilidad es la capacidad de persistencia en el tiempo de los datos. A la hora de realizar una adquisición de evidencias en el análisis forense, es importante tener en cuenta el orden de adquisición y siempre se extraerán las evidencias volátiles en primer lugar.

Orden de volatilidad: - Registros y contenidos de la memoria caché del equipo - Tablas de enrutamiento de redes, caché ARP, tabla de procesos, estadísticas del kernel y memoria - Información temporal del sistema - Datos contenidos en disco - Logs del sistema - Configuración física y topología de la red donde se encuentra el equipo. - Documentos

La información volátil que se debe estudiar primero es la hora y la fecha del sistema, se deben revisar los procesos que están en ejecución, las conexiones de red (tanto TCP como UDP), los puertos abiertos y las aplicaciones asociadas a ellos, fundamental los usuarios que se encuentren logueados en el sistema, y por supuesto, los contenidos de la memoria y ficheros swap (hiberfile.sys y pagefile.sys que pueden aportar datos sobre direcciones web, passwords, comandos ejecutados por consola, etc). Por último, sería fundamental buscar elementos ocultos como rootkits para comprobar si el equipo está comprometido.

Es decir, los datos volátiles nos darán una información valiosisima como son los usuarios, las conexiones establecidas, todos los datos que nos pueden aportar la RAM y los servicios. Para extraer esta información necesitamos ciertas herramientas, no nativas o incluso nativas del propio sistema operativo.

En este capítulo se mostrarán tanto herramientas nativasy comandos de Windows como herramientas especiales para la obtención del tipo de datos del que se está hablando como la Suite de Sysinternals (https://docs.microsoft.com/es-es/sysinternals/downloads/sysinternals-suite), Nirsoft, etcétera.

CAPÍTULO 2

Análisis con herramientas del sistema

Como ya se ha comentado, además de la memoria RAM, se deben considerar otros elementos muy importantes también como pueden ser los registros de la caché, la caché ARP, las tablas de enrutamiento, los procesos en ejecución, etc. Partiendo de este punto, antes de hacer un volcado de la memoria RAM se debe realizar la recopilación de estos datos volátiles, que, aunque podemos obtener parte con el volcado de RAM es importante conocer las herramientas que nos pueden aportar esa información.

Todas las herramientas que se muestran a continuación se pueden consultar en la página oficial de Microsoft, dentro de las herramientas específicas de la Suite Sysinternals (https://docs.microsoft.com/es-es/sysinternals/) o bien en la página de Nirsoft.

2.1 Información de Procesos activos y Servicios en ejecución

Con estas herramientas podemos ver los servicios en ejecución. Esto es muy útil, entre otras cosas, para ver las posibles conexiones maliciosas. Estos datos son fundamentales analizarlos antes de apagar el equipo.

1. Eventos del sistem a. (Sysinternals)

```
PsLoglist.exe >> Eventos_del_sistema.txt
```

2. Procesos en ejecución en memoria. Herramienta utilizada PSList. (Sysinternals)

```
pslist.exe /accepteula >> Procesos.txt
```

3. Especificación de procesos en ejecución de procesos y consumo de recursos. (Windows)

```
tasklist.exe >> Procesosenuso.txt
```

4. Procesos en ejecución e información de cada proceso. Herramienta utilizada CProcess (Nirsoft)

```
cprocess.exe /stext Procesosdeusuarios.txt
```

5. Árbol jerárquico de los procesos en ejecución. (Sysinternals).

```
pslisst.exe -t /accepteula >> procesosarbol.txt
```

6. Detalle de todos los procesos en ejecución y listado de librerías DLL asociados a cada proceso. Herramienta utilizada: (Sysinternals)

```
listdlls.exe /accepteula >> Procesosdependencias.txt
```

7. Listado agrupado de procesos. Herramienta utilizada: (DiamondCS)

```
openports.exe -path >> Mapa_agrupado_puertos_procesos.txt
```

8. Información sobre los ficheros y directorios que un programa tiene abiertos. Herramienta utilizada Handle. (Sysinternals)

```
Handle.exe /accepteula >> Procesosmanejadores.txt
```

9. Los servicios en ejecución

```
PsService.exe >> Servicios_en_ejecucion.txt
```

2.2 Información de Red y conexiones

Herramientas para la volcar información de las conexiones abiertas del sistema, una vez se apague el equipo esta información se pierde.

1. Ipconfig muestra datos de las interfaces de red
    a. Configuración de las interfaces de red. (Windows)

    ```
    ipconfig /all >> Configuracion_red.txt
    ```

    b. Listado de las conexiones de DNS que se han realizado

    ```
    ipconfig /displaydns >> DNS_consultas.txt
    ```

2. Adaptadores de red en modo promiscuo. Herramienta utilizada: (ntsecurity.nu)

```
promiscdetect.exe >> Adaptadorespromiscuos.txt
```

3. Nbtstat devuelve información sobre las conexiones NetBIOS.

    También veremos que se puede utilizar con opciones específicas para usuarios. NBT (NetBIOS sobre TCP/IP)

    a. Muestra las estadísticas del protocolo y las conexiones actuales de TCP/IP usando NBT (Netbios sobre TCP/IP)

```
nbtstat -s >> Sesionnetbios.txt
```

b. Información de la caché NetBios

```
nbtstat -c >> Cachenetbios.txt
```

4. Netstat devuelve información sobre puertos y conexiones abiertas

   Por ejemplo, la opción -a devuelve puertos y conexiones abiertos. -b devuelve el tráfico del proceso, -es devuelve estadísticas de las conexiones, -r las conexiones de NetBIOS continuación ejemplos habituales:

   a. Listado de conexiones activas

```
netstat -an findstr /i "estado listening established
↪" >> Conexionesactivas.txt
```

   b. Relación de aplicaciones con puertos abiertos

```
netstat -anob > AplicacionesPuertosAbiertos.txt
```

   c. Tabla de enrutamiento: tabla de rutas de las redes accedidas, la máscara de red yla puerta de enlace

```
netstat -r >> Tablarutas.txt
```

   d. Conexión es activas, se especifica el protocolo, direcciones IP remotas y los puertos.

```
netstat -ano >> Conexionesactivas.txt
```

5. Otros comandos:

   a. Fichero hosts

```
type c:\windows\system32\drivers\etc\hosts >> Hosts.
↪txt
```

   b. Transferencia de archivos sobre NetBIOS

```
net file > transferencia-ficheros-sobre-netbios.txt
```

   c. Caché ARP

```
arp-a>> arp-cache.txt
```

   d. Route devuelve la configuración de la red

```
Route PRINT
```

   e. Listado de todos los protocolos de red (FTP, Telnet, mailto...) que están instalados en el sistema. (Nirsoft)

```
urlprotocolview.exe /stext RedProtocolos.txt
```

2.3 Información de Ficheros

1. Listado de las unidades de red «mapeadas»

```
net use > UnidadesMapeadas.txt
```

2. Carpetas compartidas: listado de recursos compartidos

```
net share> CarpetasCompartidas.txt
```

3. Listado de ficheros abiertos. (Nirsoft) START/WAIT

```
openedfilesview.exe /stext Ficherosabiertos.txt
```

4. Ficheros remotos abiertos. Herramienta utilizada PSFile. (Sysinternals)

```
psfile.exe /accepteula >> Ficheros_remotos_abiertos.txt
```

2.4 Información de usuarios

Se muestran ahora herramientas para obtener información sobre los usuarios activos. Se pueden usar herramientas nativas para ver los usuarios y herramientas externas que pueden extraer el hash de la contraseña o la misma en texto plano.

1. Herramientas y comandos nativos:
    a. Usuarios de recursos compartidos

    ```
    Net use >> recursos compartidos.txt
    ```

    b. Usuarios de netbios

    ```
    nbtstat -n >> usuarios-recurso-compartidos.txt
    ```

    c. Usuarios de recursos compartidos

    ```
    net USERS >> Usuarios-locales-y-remotos.txt
    ```

    d. Usuarios remotos que han iniciado sesión

    ```
    net sessions >> Usuarios-remotos-ip.txt
    ```

2. Herramientas externas:
    a. Muestra las sesiones activas en el sistema. Herramienta utilizada. (Sysinternals)

```
logonsessions.exe /accepteula >> Sesiones-activas.
↪txt
```

b. Listado de usuarios que han iniciado sesión localmente en el equipo. Herramienta utilizada. Muestra usuarios locales y remotos (Sysinternals)

```
psloggedon.exe /accepteula >> Usuarios inicio-
↪sesion.txt
```

c. Listado de usuarios que han iniciado sesión localmente en el equipo. Muestra el SID del usuario (Sysinternals)

```
psGetsid.exe >> Usuarios-sid.txt
```

2.5 Información útil del sistema

1. Tiempo de actividad del sistema: período desde que el equipo se encuentra Herramienta utilizada. (Microsoft)

```
uptime.exe >> Tiempoencendido.txt
```

2. Contenido del portapapeles. Herramienta utilizada (Sourceforge)

```
pclip.exe>>Contenidoportapapeles.txt
```

3. O bien con la herramienta. (Nirsoft)

```
InsideClipboard. exe /stext Informacion_portapapeles.txt
```

4. Histórico de la consola de comandos

```
doskey /history >> HistoricoCMD.txt
```

5. Listado de servicios en ejecución

```
SC query >> servicios-ejecucion.txt
```

CAPÍTULO 3

Análisis forense de memoria RAM

En todo análisis forense, se debe atender a un orden de volatilidad. Los datos contenidos en la memoria RAM y el archivo de paginación están en la lista de evidencias volátiles a obtener pero al ser frágil, debido a que se libera y se reasigna de forma dinámica, es determinante como se obtiene. Por esto mismo, es por lo que muchas empresas que se dedican al forense informático, en muchas de sus variantes, deciden prescindir de investigar lo que hay en RAM, para dedicarse a otra información menos volátil, como por ejemplo el contenido del disco duro. El problema viene cuando sólo tenemos como pruebas un volcado de memoria (DUMP) o un archivo de paginación (pagefile.sys).

En la memoria se almacenan evidencias fundamentales como las conexiones establecidas, procesos en ejecución, las contraseñas de volúmenes cifrados, etc. Se debe contar con dos tipos de procesos, la física (la real del sistema) y la virtual (fichero pagefile.sys y swapfile.sys). La memoria optimiza el uso de la RAM ya que el sistema operativo envía ahí temporalmente la información que no sea necesaria en ese momento para los procesos en ejecución y posteriormente cuando alguno de los procesos la solicite. En Windows 10 (en Windows 8 también) aparece un nuevo archivo de memoria virtual que se llama swapfile.sys. Está guardado en el disco dc junto con pagefile.sys y hiberfil.sys. Esto se debe a que Windows intercambia algunos tipos de dato que no están siendo usados en el archivo de intercambio. Actualmente, este archivo es esas nuevas aplicaciones universales, también conocidas anteriormente como aplicaciones Metro o Modern UI.

3.1 Aplicaciones Metro y modelo de ejecución

Las aplicaciones metro presentan una característica común, lo más importante es el contenido. Por eso, a partir de Windows 8 las aplicaciones de este tipo presentan pocos controles en la interfaz, el objetivo es mostrarnos texto, video, imágenes o lo que sea. Aun así, se siguen necesitando los controles, es por esto que, las aplicaciones Metro tienen ciertos elementos comunes de interfaz que ayudan a hacer ciertas tareas: el principal es la App Bar o la barra de herramientas.

Esta barra contiene todos los comandos que podemos usar en cada una de las pantallas de la aplicación, y lo más relevante es que es contextual, se adapta a lo que estamos haciendo. Esta App Bar está oculta cuando estamos usando la aplicación y no aparecerá hasta que no deslicemos con el dedo desde la parte inferior de la pantalla. Existen dos características a destacar de una aplicación Metro, no tiene botón de cerrar y cuando la aplicación no está en pantalla se queda congelada/ bloqueada.

Estas diferencias son las que definen tres estados posibles de una aplicación Metro: en ejecución, suspendida y parada (Not Running). Cuando se ejecuta por primera vez la aplicación, pasa al estado de ejecución en el que es posible interactuar con ella. Si se cambia a otra aplicación, el estado pasa a ser suspendido: Windows guarda el estado de la aplicación en memoria, pero pausa todos los procesos que tenga en ejecución.

Este es un concepto fundamental: mientras una aplicación normal se sigue ejecutando aun estando minimizada, una aplicación Metro no. Esto tiene varias ventajas, principalmente un menor consumo de CPU; pero también inconvenientes: no se puede dejar que un proceso se ejecute en segundo plano y la aplicación no avisa para pedir el foco.

Mientras la aplicación esté suspendida y haya memoria, WindowS seguirá guardando su estado se retorna a ella cambiando de aplicaciones o pulsando de nuevo en su icono se reactivará y recuperará su estado anterior. Si por el contrario no hay suficiente memoria RAM, Windows cerrara o por completo, Al volverla a ejecutar, no recuperará su estado automáticamente y se ejecutará desde el principio, salvo que el desarrollador la haya programado para guardar datos de recuperación al desactivarse. No es necesario que el usuario mate las aplicaciones Metro tengan demasiadas aplicaciones en el ordenador, lo hace el sistema automáticamente.

Tampoco es necesario cerrar una aplicación cuando se acaba de usar ya que una aplicación suspendida no acapara recursos del sistema, ni siquiera podemos hacerlo: no hay ningún tipo de opción para salir. Además de esto, uno de los inconvenientes más destacados es cómo se distribuyen las aplicaciones. Tienen que estar contenidas totalmente en el paquete de aplicación, no pueden descargar componentes ejecutables adicionales para poder funcionar. Esto quiere decir que no posible usar frameworks como Java, y que aplicaciones con varios componentes binarios (por ejemplo, una distribución LaTeX) deben estar en un único paquete, sin descargar nada al espacio de usuario

También existen restricciones más técnicas a la hora de acceder a APIS de bajo nivel del sistema. Por ejemplo, no se pueden usar los Sockets lo que rompe la compatibilidad con muchas librerías existentes, y que además impide crear aplicaciones más complejas que transmitan datos a través de la red.

Metro también refuerza el hecho de que las aplicaciones estén aisladas entre sí. Esto impide que se puedan crear lanzadores de aplicaciones, que no se puedan modificar las características de las aplicaciones Metro, que no se puedan comunicar entre ellas de forma que no sea compartir archivos... Cierra mucho las posibilidades con respecto a lo que tenemos en el escritorio.

Y todo esto unido a las restricciones que aplique Microsoft en el Windows Store: contenidos que puedan resultar ofensivos a algunos, aplicaciones de seguridad que puedan ser detectadas como malware... Si en el proceso de revisión se encuentra algo que incumple las reglas, se rechazará la aplicación y no llegará a los usuarios hasta que no se corrijan los fallos.

3.2 Swapfile.sys, Pagefile.sys e Hiberfile.sys

Como pagefile.sys y hiberfile.sys, el archivo de swap es guardado en el directorio raíz C:\ por defecto. Son visibles si se tiene habilitada la opción de «Ver archivos y carpetas ocultas» y se tiene desactivada la opción de «Esconder archivos protegidos del sistema operativo». Hiberfil.sys es el fichero usado por el sistema operativo Windows para guardar todo el contenido de la RAM durante el proceso de hibernación. También ayuda a habilitar la función de inicio rápido en Windows 8 y Windows 10. Pagefile.sys es el fichero donde el sistema operativo página la memoria cuando no hay más espacio disponible en la memoria RAM y el sistema necesita más.

En resumen, el archivo de intercambio swapfile.sys está actualmente en uso para el nuevo modelo de aplicaciones de intercambio de Microsoft Metro Apps.

Esencialmente, el archivo de paginación es usado para operaciones normales de Windows, mientras que el nuevo marco de Microsoft utiliza un archivo diferente para realizar las operaciones de intercambio.

3.3 Tipos de volcados

3.3.1 1. Adquisición local desde un dispositivo extraíble

En este caso se hace un volcado de memoria a un dispositivo de almacenamiento externo conectado al sistema del que deseamos obtener la evidencia. Nunca se debe hacer el volcado a un disco duro local del sistema porque se podría sobrescribir datos que podrían ser importantes. Otro factor a tener en cuenta es el tamaño de las RAM actuales por lo que la unidad de destino debe estar formateado en NTFS o en otro sistema de ficheros que admita estos tamaños.

Es necesario ser cuidadoso a la hora de utilizar las unidades externas por si se infectan con malware en el sistema comprometido, es por esto que no se debe usar el mismo dispositivo externo en distintos volcados.

Se debe hacer un borrado seguro sobre la unidad externa utilizada antes de volver a utilizarla.

3.3.2 2. Adquisición remota

En un escenario de adquisición remota, se pueden utilizar distintas técnicas para poder obtener evidencias.

a. Lanzar herramientas a través de la red a la máquina de la que se desean extraer las evidencias a través de PsExec (https://docs.microsoft.com/es-es/sysinternals/downloads/psexec). Esta herramienta de la suite SysInternals es similar a un telnet que permite ejecutar procesos en otros sistemas. Los usos más potentes de PsExec incluyen el lanzamiento de comandos interactivos en sistemas remotos y herramientas de habilitación remota como IpConfig que de otro modo no tienen la capacidad de mostrar información sobre sistemas remotos.

> Algunos analizadores de antivirus informan que una o más de las herramientas están infectadas con un virus de «administración remota». Ninguno de los PsTools (las Process Utilities de la Suite) contiene virus, pero han sido utilizados por virus, por lo que activan notificaciones de virus.

b. Utilizar las carpetas que Windows comparte para realizar tareas administrativas. Se podría copiar las evidencias en C$ o ADMIN$ para ser compartidas con Server Message Block (SMB. el protocolo de red que permite compartir archivos, impresoras, etc entre nodos de red de equipos que utilizan Windows.). A continuación, se programa una tarea o se instala un servicio en el sistema de destino que ejecuta la(s) herramienta(s) y envía el contenido de la memoria física a través de ncat u otro programa/protocolo de conexión.

Los principales problemas con este método son la exposición de las credenciales de administrador y el contenido de la RAM del sistema objetivo que se envía en texto plano a través de la red. La memoria principal del ordenador contiene una gran cantidad de información confidencial que puede revelarse cuando se adquiere en texto plano a través de una red abierta. En entornos de dominio o empresariales, las credenciales de administrador de dominio proporcionan una forma de acceder al sistema de destino. Sin embargo, si el equipo de destino ya está comprometido, los atacantes pueden recuperar los tokens de autenticación generados a partir de la memoria para su uso en ataque de «Pass the hash».

c. Una mejor solución es crear una cuenta de administrador temporal que sólo permita el acceso al sistema de destino. A continuación, se deshabilita la cuenta de administrador temporal una vez finalizada la adquisición y se auditen los intentos subsiguientes de utilizar las credenciales. También se pueden buscar conexiones de bloqueo desde la máquina de destino en el firewall o en el enrutador (excepto a/ desde los sistemas involucrados en la adquisición remota). Esto evita que cualquier malware o atacante utilice las credenciales robadas para infiltrarse aún más en la red. La adquisición de un recurso compartido de red debe utilizarse sólo como un tipo de ordenadores, pero puede ser necesario en algunos casos limitados. A partir de SMB 3.0 (Windows Server 2012), se admite la encriptación de extremo a extremo.

Además, algunas herramientas (por ejemplo, CryptCat, KNTDD, F-ResponseEnterprise) soportan la adquisición de evidencia a través de la red usando SSL / TLS. También puede considerar la compresión de las pruebas antes de transferirlo a través de la red para reducir el tiempo requerido y el ancho de banda.

Es fundamental que se calculen los hashes de integridad antes y después de la transferencia, para asegurarse de que la evidencia no ha cambiado durante la transmisión.

3.4 Métodos de adquisición

Uno de los problemas para las imágenes de memoria es la verificación de que refleja el contenido actual de la memoria en el momento de su creación. El registro no estructurado del contenido de la memoria en un momento puntual se denomina core dump o memory dump.

El análisis de la memoria puede revelar si el contenido de la imagen es consistente con la disposición conocida y la estructura de un sistema operativo determinado, así como de responder a otras preguntas, pero no puede responder a la pregunta de si la imagen refleja con precisión el sistema de la que fue tomada en el momento en que fue tomada.

Para el análisis tenemos distintos tipos de técnicas y utilidades que permiten el volcado y dependen del SO en el que estamos trabajando. Para Windows:

Volcados por fallo configurando el SO para crear un volcado de memoria completa de Windows (conocido como pantallazo azul o BSOD). Volcado LiveKD (SysInternals) nombre heredado por la herramienta. Utilización de ficheros de paginación o hiberfil.sys. Este archivo se puede analizar y descomprimidos para obtener la imagen de memoria. Dumplt de Comae. Se podría decir que es la más destacable por su sencillez y compatibilidad con las distintas versiones de Windows. Tiene una versión gratuita. Es una fusión de win32dd y win64dd en un ejecutable, no hay opciones, simplemente se le pide al usuario un doble clic sobre el ejecutable y esto es suficiente para generar una copia de la memoria física en el directorio actual. Dumplt es la utilidad perfecta para desplegar en llave USB para una rápida operación de respuesta a incidentes. Rapido, pequeño y portátil.

Es suficiente ejecutar la aplicación desde el intérprete de comandos. Se realiza el volcado de memoria en formato RAW en el mismo directorio desde donde se ejecute el programa por eso lo memo más recomendable siempre es hacerlo desde un dispositivo externo.

3.5 Análisis de procesos en RAM

La memoria RAM es de vital importancia en una investigación, ya que, entre otras cosas, podremos encontrarnos con lo siguiente:

- Procesos en ejecución
- Procesos en fase de terminación
- Conexiones activas (TCP-UDP)
- Ficheros mapeados (Drivers, Ejecutables, Ficheros)
- Objetos en caché (HTML, JavaScript, Passwords)
- Elementos ocultos (Rootkits)

Como en toda investigación, la información que se puede recopilar depende de muchos factores, tales como el sistema operativo, el TimeLine de la máquina y lógicamente, el tamaño de la memoria RAM. Si se estudia como ejemplo a Windows Vista y Windows XP se puede observar que Windows Vista maneja de forma diferente los datos en memoria RAM. Utiliza mucho el acceso a RAM, para no cargar tanto al disco duro, ya que el acceso

a memoria RAM es mucho más rápido que el acceso a disco. Windows XP no carga tanto la RAM, y en cambio utiliza mucho más el archivo de paginación.

La memoria la podemos estructurar en lo que se denomina KernelLand que, tiene permiso para acceder a todo el espacio de memoria y puede comunicarse con el hardware, y el UserLand que es un nivel con menor privilegio (tiene acceso a memoria virtual y las Apis podrán comunicarse con el Kernel).

Dependiendo del sistema operativo tendremos más o menos herramientas para realizar búsquedas. Como ejemplo, Memparser es una herramienta creada a raíz de un reto forense. Antes de desarrollar la herramienta en cuestión, hubo que debuggear un kernel de un Windows 2000 SP4 para buscar similitudes, identificar estructuras, y analizar el código resultante del debugging. Cada proceso en Windows es representado por un bloque ejecutivo de proceso (Executive Process).

Cada bloque representa a un proceso y contiene estructuras y datos relacionados con ese objeto. En la memoria también residen los hilos que crean estos procesos (ETHREAD), tokens de acceso, Kernel Process (KPROCESS), el cual tiene información sobre el tanto por ciento del kernel utilizado por cada proceso, etc…

Cuando se estudia la memoria se busca lo siguiente:

- EPROCESS: la estructura que contiene datos relativos a un pr0ceso
- Objetos de Kernel
- Drivers
- Contenido de memoria

Aunque Memparser es una herramienta que ya tiene unos años (lanzada en 2005) es útil para equipos con windows 2000 y windows 2000 server. No se debe descartar porque como ya se ha comentado existen empresas muy desactualizadas y puede ser necesario utilizarla. Permite, desde listar los procesos que hay en un volcado de memoria, hasta por ejemplo sacar el contenido de la memoria de un ejecutable, y con el que sería posible buscar restos de malware u otras evidencias.

CAPÍTULO 4

Volatility

Es una colección de herramientas, implementado en Python bajo la GNU (Open Source GPLV2), para la extracción de recursos digitales de la memoria volátil (RAM). Este framework para el análisis de volcado de memoria está orientado a extraer de una imagen los datos volátiles. Su API es ampliable y permite utilización de scripts por lo que su potencial es inmenso. Las técnicas de extracción se realizan de forma completamente independiente del sistema que está siendo investigado, pero ofrecen visibilidad del estado de ejecución del sistema, es decir, podemos utilizar Windows y/o Linux. Nos permite trabajar con:

1. Detalles de la imagen (fecha, hora, número de CPU)
2. Los procesos en ejecución
3. Proceso de SID y variables de entorno
4. Conexiones de red
5. DLLS cargados para cada proceso
6. Los objetos del kernel / (archivos, claves, exclusiones mutuas)
7. Los módulos del kernel
8. Volcado de cualquier proceso, DLL o módulo en el disco
9. Mapeo físico a las direcciones virtuales
10. Memoria direccionable para cada proceso
11. Mapas de memoria para cada proceso
12. Extraer muestras ejecutables
13. Historias de comandos (cmd.exe) y datos de consola de entrada /salida
14. PE información de versión
15. Las tablas del sistema de llamada

16. Secciones del Registro
17. Volcado LM/ NTLM hashes y secretos LSA
18. Ayudar al usuario y exploración shimcache
19. Analizar en busca de patrones de bytes, expresiones regulares o cadenas en la memoria
20. Analizar tiempos del núcleo y funciones de devolución de llamada
21. Informe sobre los servicios de Windows

La distribución de este framework está disponible en: https://github.com/volatilityfoundation/volatility

La comunidad es seria y consta de grandes profesionales e investigadores del análisis forense y | respuesta ante incidentes o malware.

Como ejemplo:

La gran ventaja de utilizar este framework es que se tiene un marco único y coherente para analizar volcados de memoria Ram, tanto de 32 como de 64 bits de Windows, Linux, OSX y Android, es decir. En su estructura hay profiles y plugins que se utilizarán según las necesidades.

Vista de los procesos en ejecución

```
volatility pslist -f xp-ejemplo-2021-02-20-1120.img
```

Mostrar las relaciones entre los procesos en forma de árbol

```
volatility pstree -f xp-ejemplo-2021-02-20-1120.img
```

Lista de las conexiones, solo en XP y Windows 2003

```
volatility connections -f xp-ejemplo-2021-02-20-1120.img
```

Lista de las conexiones tanto cerradas como abiertas, solo en XP y Windows 2003

```
volatility connscan -f xp-ejemplo-2021-02-20-1120.img
```

Muestra de sockets a la escucha, solo XP y Windows 2003

```
volatility sockets -f xp-ejemplo-2021-02-20-1120.img
```

Lista de las dlls cargadas por procesos

```
volatility dllist -f xp-ejemplo-2021-02-20-1120.img
```

Lista de los handles abiertos para cada proceso.

```
volatility handles-f xp-ejemplo-2021-02-20-1120.img
```

Encuentra y lista las Hives en uso

```
volatility hivelist -f xp-ejemplo-2021-02-20-1120.img
```

Obtiene la clave de registro que se especifique, las subclaves y sus valores correspondientes

```
volatility printkey -f xp-ejemplo-2021-02-20-1120.img -K
↪"Microsoft\Windows NT\CurrentVersion\windows"
```

Extrae las credenciales de dominio almacenadas en memoria

```
volatility hashdump-f xp-ejemplo-2021-02-20-1120.img
```

Muestra los drivers del kernel cargados

```
volatility modules-f xp-ejemplo-2021-02-20-1120.img
```

Extracción de ejecutables

```
volatility procdump -f xp-ejemplo-2021-02-20-1120.img -p 3256 --
↪dump-dir=c:/temp
```

Timeline

```
volatility timeliner -f xp-ejemplo-2021-02-20-1120.img >␣
↪lineaTiempo.txt
```

Recupera los comandos escritos en consola

```
volatility consoles-f xp-ejemplo-2021-02-20-1120.img
```

Busca servicios de windows

```
volatility svcscan -f xp-ejemplo-2021-02-20-1120.img
```

Muestra que páginas están residentes en memoria

```
volatility memmap -f xp-ejemplo-2021-02-20-1120.img
```

Busca los drivers en la memoria física utilizando pool tag scanning

```
volatility driverscan-f xp-ejemplo-2021-02-20-1120.img
```

Encontrar FILE_OBJECTS en memoria física usando pool tag scanning

```
volatility filescan -f xp-ejemplo-2021-02-20-1120.img
```

CAPÍTULO 5

Más Volatilidad Comandos y Herramientas

Existen una serie de comandos útiles para obtener información del sistema analizado, son las siguientes:

Tabla 1: **Comandos útiles**

COMANDO	FUNCIÓN
Date /t & time /t	Fecha y hora
Ipconfig /all	Información tcp/ip
Netstat -aon	Conexiones abiertas y puertos en espera, con su PID asociado
psinfo -shd	Información del sistema (hardware, software, hotfixes, versiones, etc.)
pslist -t	Lista de procesos
at	Lista de tareas programadas (%windir %tasksfolder)
psloggedon	Usuarios logueados y hora de logon
psloglist	Volcado de log de eventos
psservice	Información de servicios de sistema
netuse, netaccounts, netsession, netshare, netuser	Conexiones netbios/smb
listdlls	Lista de DLLS cargadas en el sistema
sigcheck -u e c:windows	Lista de ficheros (.exe,.dll) no firmados
streams -s c:	Lista de ficheros con alternate data streams (ads)
logonsessions -p	Sesiones actuales y procesos por sesión
arp -a	Muestra la tabla de caché ARP
ntlast	Muestra eventos de logon correctos y fallidos
routeprint	Muestra tabla de enrutado IP
autorunsc	Muestra elementos de autoejecución
hfind c:	Ficheros ocultos
promiscdetect	Detecta interfaces de red en modo promiscuo

continué en la próxima página

Tabla 1 – proviene de la página anterior

COMANDO	FUNCIÓN
volume_dump	Muestra información sobre volúmenes, puntos de montaje, sistema de ficheros,etc.
pwdump2	Muestra hashes (nthash / Imhash) de cuentas locales
lsadump2	Muestra LSA secrets (necesita SeDebugPrivilege)
strings	Busca cadenas ASCII / Unicode en ficheros

5.1 Herramientas gráficas

Se presentan diversas herramientas gráficas de Sysinternals para trabajar con datos volátiles:

- **Rootkit revealer:** detecta rootkits (tanto usermode como kernelmode), se puede descargar de https://technet.microsoft.com/en-us/sysinternals/rootkitrevealer.aspx
- **Process explorer (procexpy procmon):** devuelve información útil sobre procesos, librerías que usan, recursos accedidos, conexiones de red, etc. Se puede descargar de https://technet.microsoft.com/en-us/sysinternals/processexplorer.aspx
- **Tcpview:** muestra conexiones de red y aplicaciones asociadas, se puede descargar de https://technet.microsoft.com/en-us/sysinternals/tcpview.aspx.

CAPÍTULO 6

Conclusiones

Es muy interesante la información que se puede encontrar en sistemas hibernados. En alguna ocasión, al comprobar los datos del fichero «hiberfil.sys» en algunos PCs donde los trabajadores de la empresa tenían la costumbre de no apagar nunca los equipos, aparecieron las claves de acceso a ficheros cifrados que arrojaron gran luz sobre la competencia desleal que se estaba produciendo por parte del responsable de Marketing y varios responsables más de distintos departamentos.