

DFIR

Digital Forensics & Incident Response

Caso práctico

Gerard Díaz Hoyos

KeepCoding
Bootcamp de Ciberseguridad
5a edición

Entorno de pruebas

Para el desarrollo de las siguientes tareas, usaremos entornos y herramientas enfocadas claramente a la rama de la informática forense. Éstas serán:

- **Virtual Box**: para el despliegue de máquinas virtuales
- Máquina virtual **Tsurugi Linux**: distribución de Linux basada en *Ubuntu* que incluye un gran número de herramientas forenses
- Máquina virtual de **Windows 10** enfocada también a la rama de DFIR, con la que tendremos ya preinstaladas algunas de las herramientas de forense digital más utilizadas y que se usarán más adelante (*FTK Imager*, *Arsenal Image Mounter*, *Kape*, varias herramientas de Erick Zimmerman, etc.)

Práctica Windows

La primera parte de la práctica consiste en hacer un análisis de dicha máquina utilizando las herramientas proporcionadas a lo largo del curso. Además es muy importante indicar las herramientas utilizadas y cómo se ha llegado a obtener el resultado.

Tenemos la sospecha de que el usuario del equipo está sacando información de la compañía de su equipo y además el equipo de monitorización ha levantado una alerta indicando comportamientos extraños en el equipo, por ello nos han llamado para que analicemos el equipo y podamos determinar qué indicios y evidencias existen sobre las sospechas fundadas. Además deberéis de registraros en la página ctf.sancastell.me utilizando el código de registro Keepcoding2023/*. Dentro encontrareis una serie de retos que responder. Es necesario al finalizar los retos entregar una memoria indicando cómo se han hallado y las herramientas utilizadas.

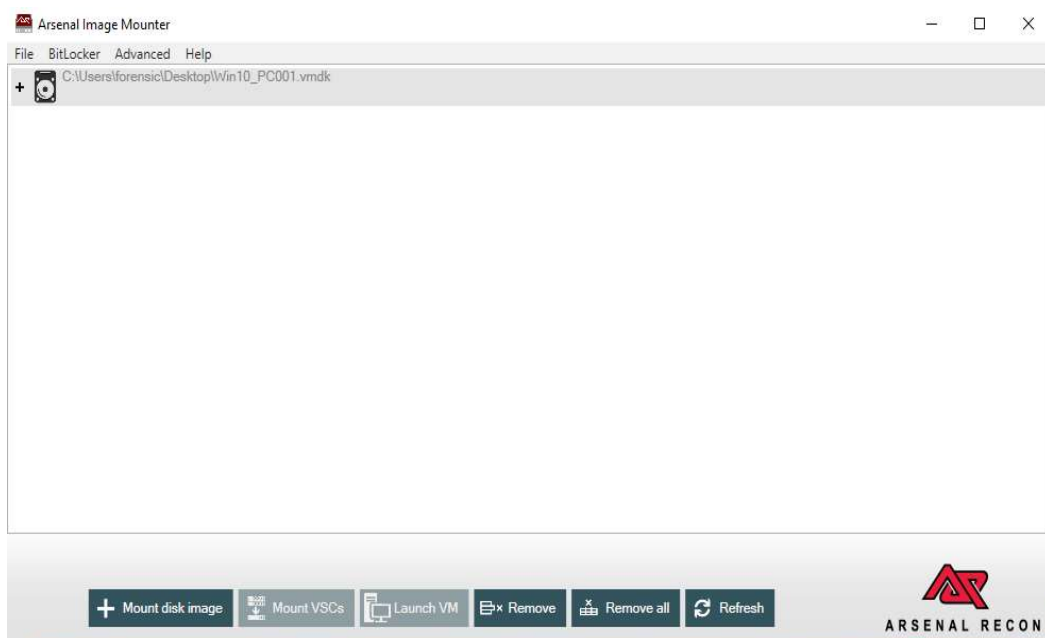
Preparación:

Se nos ha proporcionado la evidencia con el siguiente formato:



Es una imagen de un Windows 10 en formato **.vmdk**.

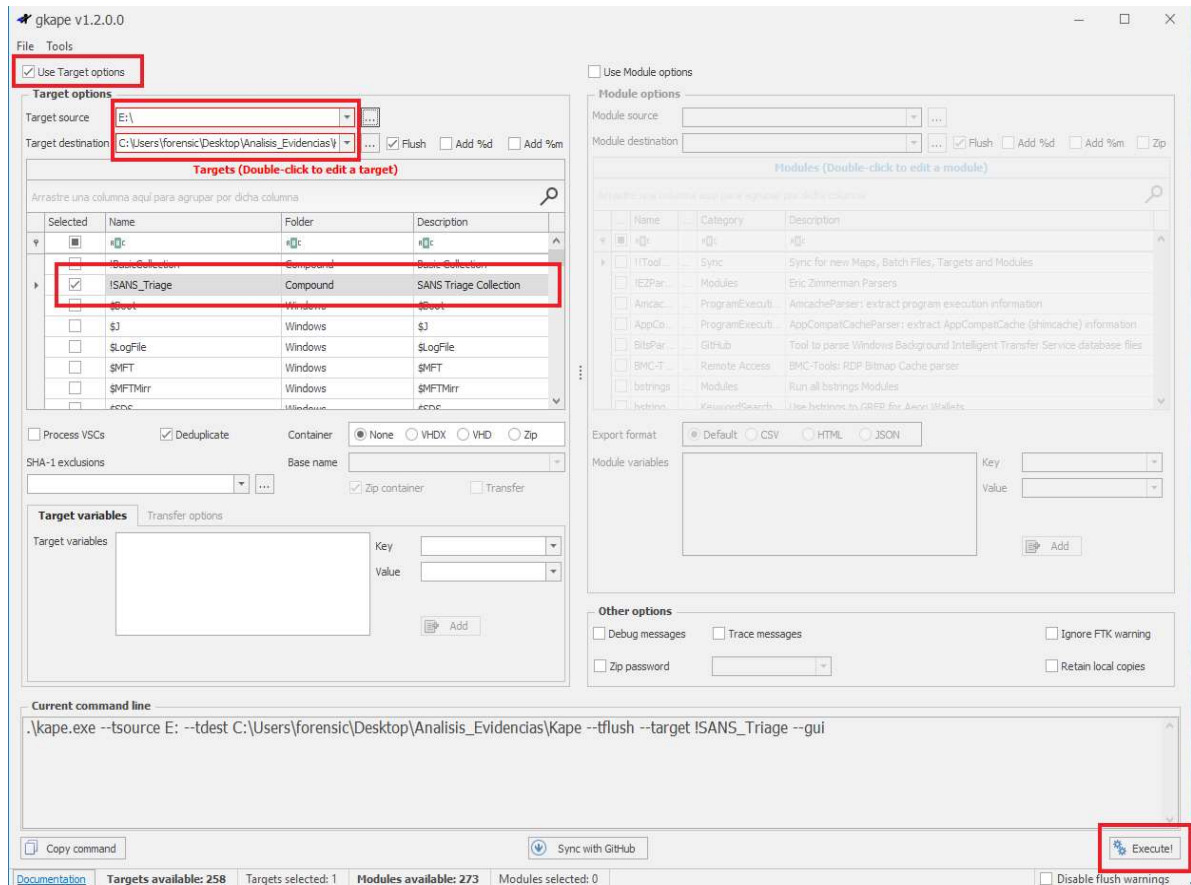
- 1) **Arsenal Image Mounter**: Herramienta que nos permitirá montar la imagen de la evidencia como un nuevo volumen en nuestra propia máquina. Sólo deberemos proporcionarle el archivo de la evidencia y pulsar el botón “*Mount Disk Image*” para que se nos añada la unidad E en nuestro sistema de archivos. Nos interesa dejar realizado este paso ya que la siguiente herramienta que vamos a usar, Kape, funciona con volúmenes montados.



Dispositivos y unidades (3)



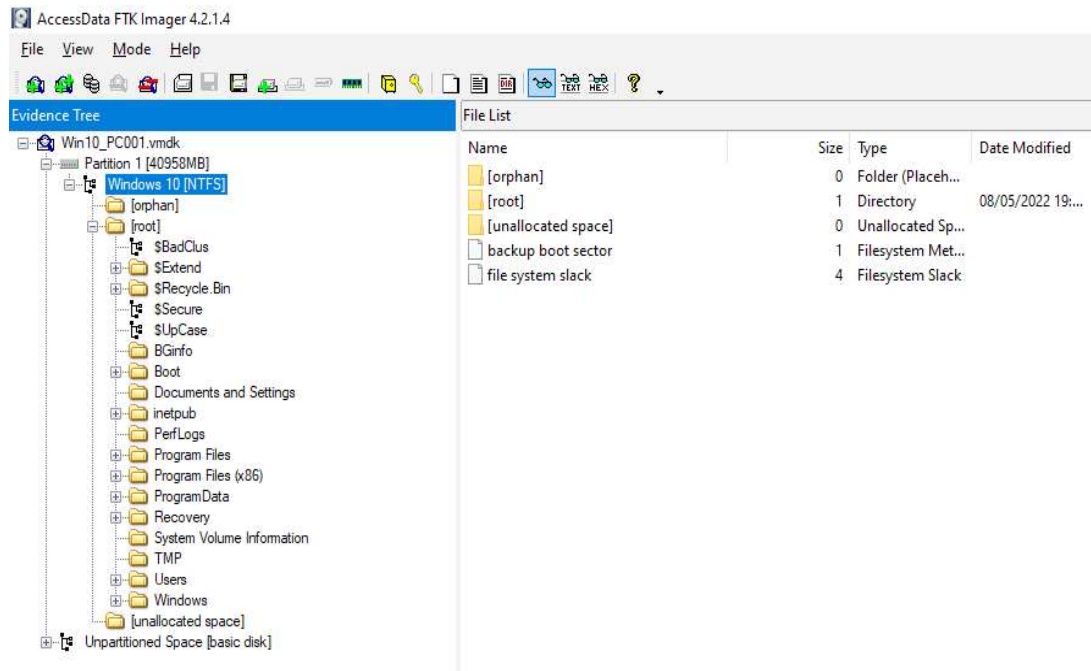
- 2) **Kape:** herramienta muy utilizada en el mundo de la informática forense que nos permite la extracción y análisis de los artefactos de un volumen de disco.
- Usaremos la interfaz gráfica y nos centraremos en la parte izquierda, que es la propia de la extracción de artefactos. Seleccionaremos el volumen a analizar, la ruta en la que nos copiará la extracción, un “compound” suficiente que agrupe diversos artefactos de interés (para no tener que buscar y seleccionar uno por uno) y procederemos a darle al botón “Execute”.



Obtendremos un *output* tal que así:

Este equipo > Escritorio > Analisis_Evidencias > E >				
Nombre	Fecha de modificación	Tipo	Tamaño	
SExtend	29/05/2023 20:20	Carpeta de archivos		
SRecycle.Bin	29/05/2023 20:19	Carpeta de archivos		
Program Files	29/05/2023 20:20	Carpeta de archivos		
ProgramData	29/05/2023 20:20	Carpeta de archivos		
Users	29/05/2023 20:19	Carpeta de archivos		
Windows	29/05/2023 20:20	Carpeta de archivos		
\$Boot	29/05/2023 20:20	Archivo	8 KB	
\$LogFile	29/05/2023 20:20	Archivo	56.016 KB	
\$MFT	19/03/2019 22:52	Archivo	157.184 KB	
\$Secure_\$SDS	19/03/2019 22:52	Archivo	2.316 KB	

- 3) **AccessData FTK Imager**: herramienta que nos muestra la estructura del sistema de archivos de la evidencia y nos permite también consultar información al respecto y analizar artefactos de utilidad. Es interesante combinar esta herramienta con **Kape**, por ejemplo. Es muy sencilla de usar y tan sólo deberemos darle al botón del menú “*Add Evidence Item*” y seleccionar la imagen de la evidencia.



- 4) Editores de texto compatibles con archivos con extensión **.vcs**: dependiendo de la situación, podremos usar **Excel**, **Notepad++** o **Timeline Explorer**, entre otras muchas alternativas.

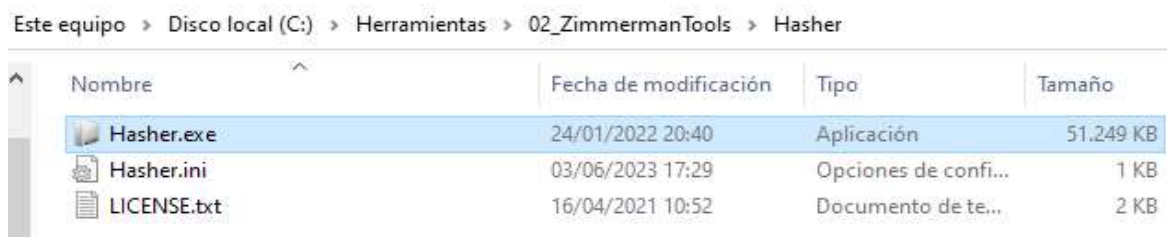
Reto 1: Hash del fichero

Como analistas de la máquina, lo primero que debemos obtener es el hash **sha-256** de la evidencia.

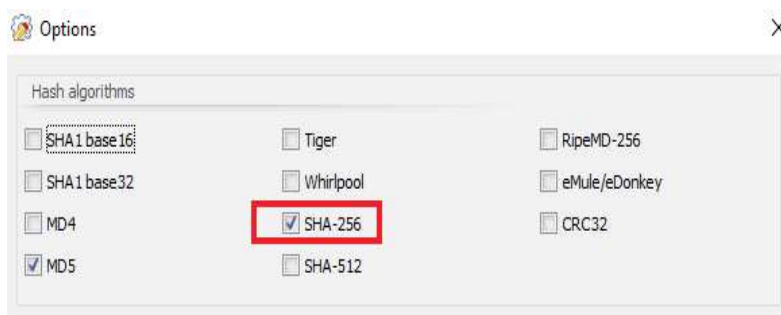
Para sacar el hash **sha-256** de la evidencia o de cualquier otro fichero, tenemos un amplio abanico de herramientas disponibles (*Hasher*, *HashMyFiles*, *OpenHashTab*, etc.) o scripts que se pueden ejecutar a través de la consola de comandos.

Aprovechando que tenemos una máquina virtual de Window 10 con herramientas forenses ya incluidas, utilizaremos *Hasher*.

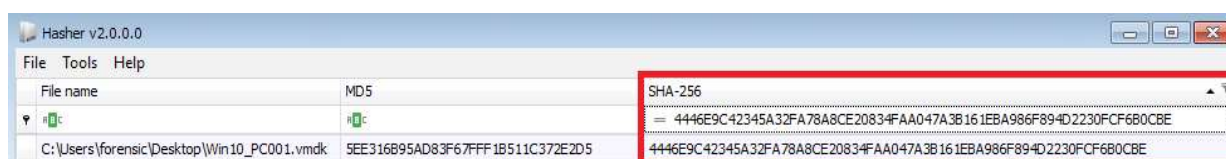
Abriremos el programa indicado:



En opciones, seleccionaremos el tipo de algoritmo que queremos aplicar para el *hash*; en este caso será el **sha-256**:



Seleccionaremos la evidencia y esperaremos el tiempo necesario para que genere el hash correspondiente:



Reto 2: Nombre de la máquina

Una forma rápida y sencilla de saber el nombre de la máquina es a través de los ficheros de eventos de Windows. Para observar el contenido de estos archivos, no necesitamos usar un editor de texto de terceros, puesto que haciendo doble *click* en ellos, Windows abrirá directamente el suyo propio: Visor de Eventos. Al analizar estos ficheros encontraremos información variada, entre la que estará el nombre de la máquina a la que pertenecen.

A modo de ejemplo, vamos a buscar y abrir el archivo **security.evtx** (logs de seguridad).

Para ello, nos iremos a la siguiente ruta (puesto que en los sistemas Windows 10, siempre se alojan aquí):

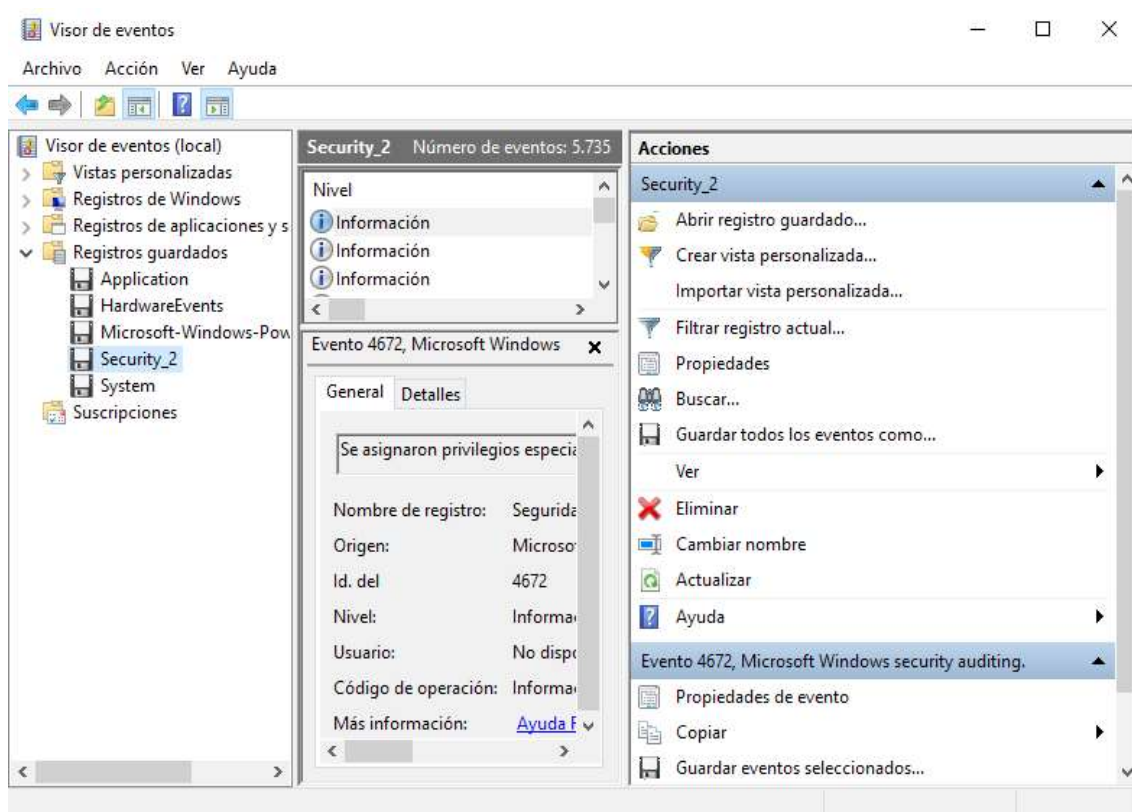
Este equipo > Escritorio > Analisis_Evidencias > E > Windows > System32 > winevt > logs

Donde E es el volumen montado mediante la imagen de la evidencia que estamos analizando.

Veremos que en dicha carpeta nos aparecerán todos los registros de eventos:

Microsoft-Windows-WindowsSystemAss...	29/04/2022 10:34	Registro de eventos	68 KB
Microsoft-Windows-WindowsUpdateClie...	29/04/2022 12:50	Registro de eventos	1.028 KB
Microsoft-Windows-Winlogon%4Operati...	29/04/2022 12:39	Registro de eventos	1.028 KB
Microsoft-Windows-WinRM%4Operatio...	29/04/2022 12:39	Registro de eventos	68 KB
Microsoft-Windows-WMI-Activity%4Op...	29/04/2022 12:39	Registro de eventos	1.028 KB
OpenSSH%4Operational.evtx	19/03/2019 14:29	Registro de eventos	68 KB
Security.evtx	30/05/2023 0:01	Registro de eventos	5.188 KB
Setup.evtx	08/05/2022 20:55	Registro de eventos	68 KB
System.evtx	04/06/2023 20:27	Registro de eventos	1.092 KB
ThinPrint Diagnostics.evtx	29/04/2022 18:32	Registro de eventos	68 KB
Windows PowerShell.evtx	08/05/2022 21:07	Registro de eventos	1.092 KB

Hacemos doble click izquierdo en cualquier de ellos y se nos abrirá el Visor de Eventos:



Y dentro de la información general, podremos encontrar el nombre de la máquina:

Nombre de registro:	Seguridad		
Origen:	Microsoft Windows security	Registrado:	08/05/2022 21:11:05
Id. del	4672	Categoría de tarea:	Special Logon
Nivel:	Información	Palabras clave:	Auditoría correcta
Usuario:	No disponible	Equipo:	PEGASUS01
Código de operación:	Información		
Más información:	Ayuda Registro de eventos		

Reto 3: Ficheros maliciosos

En la máquina se han encontrado varios ficheros maliciosos. ¿En qué carpeta se encuentran dichos ficheros?

Para empezar a rastrear posibles archivos maliciosos, podríamos lanzar la herramienta **Loki** en la unidad E montada. **Loki** es una herramienta forense que busca indicadores de compromiso para intentar localizar malware en el sistema. Automatiza el proceso de rastrear todos los archivos sospechosos, los registros, los archivos comprimidos, comparar metadatos, etc. Mediante el uso de reglas específicas e internas, nos otorga información valiosa de las posibles amenazas que podrían haber dentro del sistema.

Para ejecutarlo, usaremos la consola de comandos *cmd* y el siguiente *script*:

```
C:\Herramientas\04_loki_0.44.2\loki>loki.exe -p E:\ --noprocscan --logfolder C:\Users\forensic\Desktop\Análisis_Evidencias\Loki
```

Iniciará un largo escaneo de toda la unidad y nos sacará finalmente un informe que repasaremos a continuación (lo visualizamos con *Timeline Explorer*):

```
LOKI: Info: MODULE: Init MESSAGE: Setting LOKI process with PID: 8904 to priority IDLE
LOKI: Info: MODULE: FileScan MESSAGE: Scanning Path E:\ ...
LOKI: Alert: MODULE: FileScan MESSAGE: FILE: E:\inetpub\wwwroot\tests.jsp SCORE: 140 TYPE: JSP SIZE: 657 FIRST_BYTES: 3c2
LOKI: Warning: MODULE: FileScan MESSAGE: FILE: E:\TMP\nbtscan.exe SCORE: 60 TYPE: UNKNOWN SIZE: 36864 FIRST_BYTES: / <fi
LOKI: Alert: MODULE: FileScan MESSAGE: FILE: E:\TMP\p.exe SCORE: 105 TYPE: EXE SIZE: 381816 FIRST_BYTES: 4d5a900003000000
LOKI: Alert: MODULE: FileScan MESSAGE: FILE: E:\Users\Public\procdump64.exe SCORE: 105 TYPE: EXE SIZE: 341672 FIRST_BYTES
LOKI: Alert: MODULE: FileScan MESSAGE: FILE: E:\Users\Public\svchost.exe SCORE: 220 TYPE: EXE SIZE: 8192 FIRST_BYTES: 4d5
LOKI: Notice: MODULE: Results MESSAGE: Results: 4 alerts
LOKI: Result: MODULE: Results MESSAGE: Indicators detected!
```

Nos encontramos 5 registros a tener en cuenta (vienen asignados con las etiquetas “Alert” y “Warning”; vemos que 4 de dichos registros son ficheros .exe: dos de ellos se encuentran en la carpeta **TMP** y los otros dos en la de **Users\public**)

Visualizamos la carpeta **TMP**

Encontramos 4 archivos que podrían ser, al menos, sospechosos de contener malware:

File List			
Name	Size	Type	Date Modified
SI30	4	NTFS Index All...	08/05/2022 19:...
127.0.0.1.txt	1	Regular File	11/10/2014 3:0...
d.txt	10	Regular File	11/10/2014 3:4...
nbtscan.exe	36	Regular File	04/02/2018 19:...
p.exe	373	Regular File	27/04/2010 10:...
p.exe.FileSlack	4	File Slack	
scan1.tmp	0	Regular File	08/05/2022 19:...
scan2.tmp	0	Regular File	08/05/2022 19:...
scan3.tmp	0	Regular File	08/05/2022 19:...
sys.txt	8	Regular File	08/05/2022 19:...
WMIBackdoor.ps1	20	Regular File	10/08/2015 13:...
xCmd.exe	824	Regular File	29/07/2014 15:...

Tal y como nos indica el propio **Loki**, podemos hacer una comprobación rutinaria mediante la página web de [virustotal](https://www.virustotal.com).

```
(ES) Loki recommends checking the elements on virustotal.com or Google and triage with a professional tool like THOR https://nexttron-systems.com/thor in corporate networks.
```

Y, efectivamente, *virustotal* nos alerta de que **nbtscan.exe** y **xCmd.exe** son troyanos, que **p.exe** no representa peligrosidad según los reportes en Internet y que **WMIBackdoor.ps1** podría tratarse de una herramienta de *hacking* o troyano.

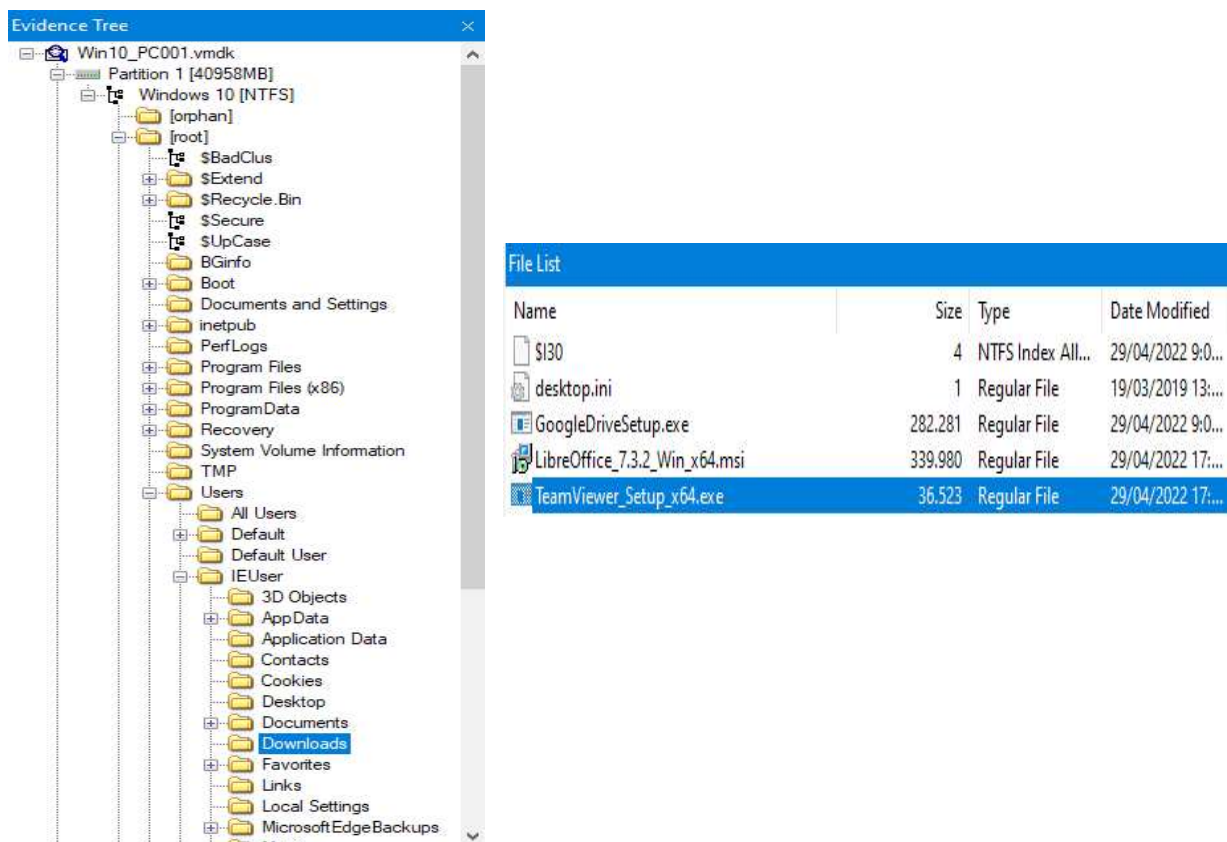
Por otro lado, los 2 archivos de los que hemos sido alertados procedentes de la carpeta **Users\public** no representarían ningún peligro, a priori.

Podemos concluir que, en este caso, la carpeta que aloja archivos maliciosos es la **TMP**, aunque no es la única que los alberga en todos los casos, ya que podríamos también encontrarlos en la carpetas de descargas del usuario, en directorios de programas, archivos de sistema, adjuntos de correos electrónicos, etc.

Reto 4: Descarga fichero de control remoto

Escriba el nombre del fichero .exe de un programa de control remoto que se ha descargado el usuario.

Siguiente un poco la lógica, deberíamos empezar buscando en la carpeta de descargas del usuario.



Name	Size	Type	Date Modified
\$I30	4	NTFS Index All...	29/04/2022 9:0...
desktop.ini	1	Regular File	19/03/2019 13:...
GoogleDriveSetup.exe	282.281	Regular File	29/04/2022 9:0...
LibreOffice_7.3.2_Win_x64.msi	339.980	Regular File	29/04/2022 17:...
TeamViewer_Setup_x64.exe	36.523	Regular File	29/04/2022 17:...

Se ha utilizado el software **FTK Imager** para poder observar el contenido de la carpeta **downloads** del usuario **IEUser**.

Como se puede apreciar, se encuentran 2 archivos .exe, siendo el **Team Viewer** el que pertenece a un programa de control remoto.

Si no se encontrara aquí, deberíamos buscar registros de descargas de los navegadores usados, registros del sistema que nos indiquen instalaciones y ejecuciones de software, comparar metadatos o hacer análisis de firmas y *hashes* en algunos casos.

Reto 5: Ficheros eliminados

Se sospecha que existe un fichero .zip eliminado; encontrar su nombre.

Para obtener un resultado rápido y directo, podemos utilizar una herramienta incluida en la máquina de Windows 10 Forense que se llama **RBCmd.exe** que nos permitirá analizar e interpretar los logs generados por los archivos y movimientos producidos en la papelera de reciclaje.

RBCmd.exe	05/08/2022 13:05	Aplicación	3.607 KB
-----------	------------------	------------	----------

Es una herramienta que funciona por consola de comandos. Los *scripts* que deberemos usar para *parsear* el volumen-evidencia serán los siguientes:

```
C:\Herramientas\02_ZimmermanTools>RBCmd.exe -h
```

Y lo ejecutaremos como tal indicando la ruta objetivo o de origen, la ruta de la salida y el nombre del fichero seleccionado, respectivamente:

```
C:\Herramientas\02_ZimmermanTools>RBCmd.exe -d C:\Users\forensic\Desktop\Análisis_Evidencias\E --csv C:\Users\forensic\Desktop\Análisis_Evidencias\Papelera --csvf papelera.csv
```

De este modo obtendremos un fichero con extensión *.csv* con la información que buscamos; lo abriremos con Timeline Explorer y podremos hallar un archivo comprimido *.zip*, que es el que resuelve el presente reto:

Line	Tag	Deleted On	Source Name	File Type	File Name	File Size
=	<input type="checkbox"/>	=				=
1	<input type="checkbox"/>	2022-05-08 18:54:10	C:\Users\forensic\Desktop...	\$I	C:\Users\IEUser\Documents\02_AnejoII_EstrucyContTFG_a.pdf	219389
2	<input type="checkbox"/>	2022-05-08 19:14:07	C:\Users\forensic\Desktop...	\$I	C:\Users\IEUser\AppData\Local\Temp\cosas.zip	9771788
3	<input type="checkbox"/>	2022-05-08 18:54:10	C:\Users\forensic\Desktop...	\$I	C:\Users\IEUser\Documents\CONFIDENTIAL document list.pdf	50673
4	<input type="checkbox"/>	2022-05-08 18:54:10	C:\Users\forensic\Desktop...	\$I	C:\Users\IEUser\Documents\Documento Seguridad HipoSEMG.doc	0

Si nos fijamos, los tipos de archivo que nos ha interpretado la herramienta usada son ***Index Records (\$I30 o \$I)***. Estos archivos se utilizan en conjunto con la estructura de la MFT, para mantener un registro de los archivos y directorios presentes en el sistema. Gracias a ellos, podemos saber si un archivo estuvo en una ruta determinada, como es el caso de la solución de este reto.

Reto 6: Fecha de ejecución programa de control remoto

Sabemos que se ha ejecutado el programa *Team Viewer* en el equipo; ¿Podrían indicar la fecha en la que se ejecutó? Formato: *dd/mm/yyyy*

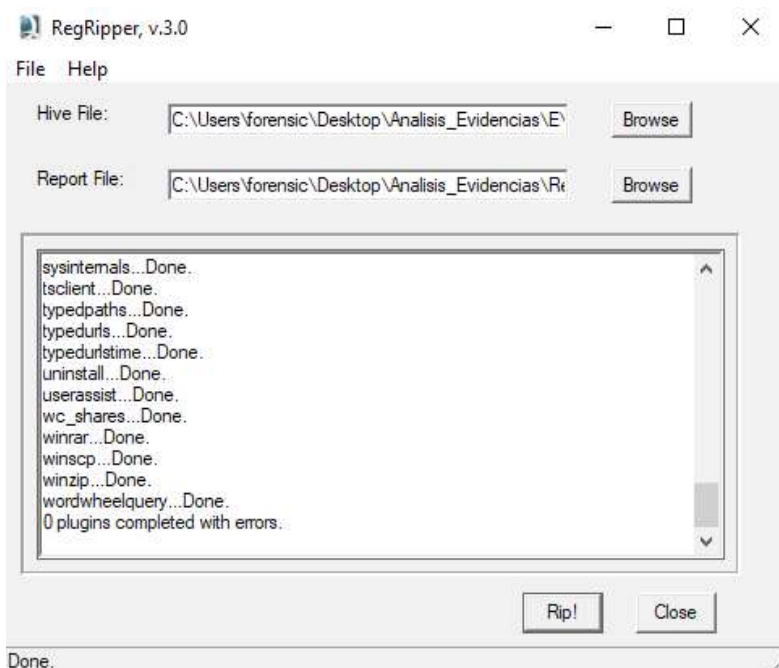
Para encontrar la última fecha de ejecución de *Team Viewer*, en un primer momento he pensado que podría visualizar los archivos de registro de Windows, específicamente, los de usuario: ***ntuser.dat*** y ***usrclass.dat***. Conocemos la ruta en la que se encuentran dichos archivos:

ntuser.dat → C:\Users\{nombre de usuario}\NTUSER.DAT

usrclass.dat → C:\Users\{nombre de usuario}\AppData\Local\Microsoft\Windows\USRCLASS.DAT

Para interpretar dichos registros he usado ***RegRipper*** (herramienta también incluida por defecto en la máquina virtual de Windows 10 Forense con la que estamos trabajando).

Dicha herramienta tan solo nos pide que seleccionemos un *input* y un *output*:



Pensaba que con el registro *usrclass.dat* tendría bastante, puesto que estos ficheros contienen información en cuanto a la ejecución de programas, pero aquí no encontré ninguna fecha de ejecución como tal:

```
C:\Program Files\TeamViewer\TeamViewer.exe.FriendlyAppName (TeamViewer)
C:\Program Files\TeamViewer\TeamViewer.exe.ApplicationCompany (TeamViewer Germany GmbH)
```

Al menos, se halla la evidencia de que sí se ejecutó en algún momento el programa *Team Viewer* por el usuario **IEUser**.

Procedo, entonces, a observar también el *ntuser.dat* con *Timeline Explorer* y encuentro más información al respecto. Si filtramos el registro por la palabra “*Viewer*”, obtenemos la siguiente información:

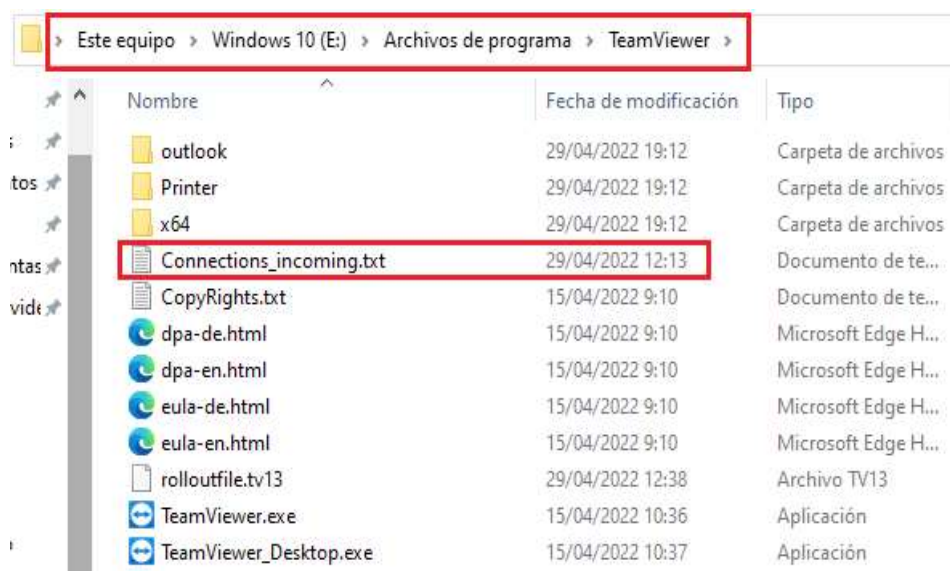
Line	Tag	Hive (C:\Users\forensic\Desktop\Análisis_Evidencias\E\Users\IE User\NTUSER.DAT) is dirty.
=	<input type="checkbox"/>	
333	<input type="checkbox"/>	2018-09-15 07:28:42Z - C:\Users\IEUser\Downloads\TeamViewer_Setup_x64.exe
337	<input type="checkbox"/>	2018-09-15 07:28:42Z - C:\Program Files\TeamViewer\TeamViewer.exe
512	<input type="checkbox"/>	TeamViewer.lnk
544	<input checked="" type="checkbox"/>	2022-04-29 10:30:10Z TeamViewer
869	<input type="checkbox"/>	{6D809377-6AF0-444B-8957-A3773F02200E}\TeamViewer\TeamViewer.exe (1)
896	<input type="checkbox"/>	C:\Users\IEUser\AppData\Local\Temp\TeamViewer\TeamViewer_.exe
912	<input type="checkbox"/>	{0139D44E-6AFE-49F2-8690-3DAFCAE6FFB8}\TeamViewer.lnk (1)

Se pueden observar 2 fechas: 15/09/2018 y 29/04/2022; por lógica, se podría suponer que la más antigua haría referencia a la fecha de instalación o creación (además de que se trata de un archivo **Setup_x64.exe*) y que la más actual haría referencia a la fecha de última ejecución.

Para comprobarlo, me dispongo a ver la carpeta donde está alojado dicho *software*:

C:\Users\IEUser\AppData\Local\Temp\TeamViewer\

Aquí se puede confirmar que la fecha de ejecución es, efectivamente, la última que nos aparecía en el segundo registro que hemos analizado.



765418952	WIN-MORENIN	29-04-2022 10:09:14	29-04-2022 10:10:10	IEUser	RemoteControl	{adb13c9a-796c-438c-af8b-2079077f0a4f}
765418952	WIN-MORENIN	29-04-2022 10:10:34	29-04-2022 10:13:21	IEUser	RemoteControl	{301db382-67ac-4b5a-9bec-d1a4aa0ad30}

Reto 7: Powershell maliciosa

En el equipo hay un script de powershell malicioso con extensión .ps1. ¿Podrían indicar cuál es?

El script de Powershell lo encontramos al usar la herramienta **Loki** para encontrar ficheros maliciosos. Gracias al resultado que nos dio **Loki**, observamos que en la carpeta **TMP** se alojaba *Malware*. Entre los archivos que allí habían, se encontraba el **WMIBackdoor.ps1**, que completaba el presente reto.

Este equipo > Windows 10 (E:) > TMP

Nombre	Fecha de modificación	Tipo	Tamaño
127.0.0.1.txt	11/10/2014 5:00	Documento de te...	1 KB
d.txt	11/10/2014 5:40	Documento de te...	10 KB
nbtscan.exe	04/02/2018 20:06	Aplicación	36 KB
p.exe	27/04/2010 12:04	Aplicación	373 KB
scan1.tmp	08/05/2022 21:07	Archivo TMP	0 KB
scan2.tmp	08/05/2022 21:07	Archivo TMP	0 KB
scan3.tmp	08/05/2022 21:07	Archivo TMP	0 KB
sys.txt	08/05/2022 21:07	Documento de te...	8 KB
WMIBackdoor.ps1	10/08/2015 15:42	Script de Window...	20 KB
xCmd.exe	29/07/2014 17:38	Aplicación	824 KB

Reto 8: Contraseñas débiles

Existen sospechas que la contraseña del usuario IEUser es una contraseña débil, lo que ha permitido al atacante acceder a ella. Indicar la contraseña del usuario.

En ocasiones nos puede interesar saber si el usuario de una determinada máquina hacía uso de una contraseña débil con la que algún atacante haya podido sacar provecho.

La herramienta **Mimikatz** nos puede ayudar a obtener el listado de usuarios y los *hashes* de sus contraseñas de una evidencia determinada.

La máquina Virtual de Windows 10 que se está usando para la realización de estos retos también la tiene ya instalada de base, aunque la reconoce como peligrosa o maliciosa, bloqueando su acceso y poniendo en cuarentena algunos de sus archivos directamente. Por eso, se recomienda desactivar la seguridad de Windows, al menos temporalmente, mientras vayamos a utilizar la herramienta.

Mimikatz funciona por consola de comandos:

```
C:\Herramientas\01_Artefactos\mimikatz-master\x64>mimikatz.exe

.#####.  mimikatz 2.2.0 (x64) #18362 Feb 29 2020 11:13:36
.## ^ ##.  "A La Vie, A L'Amour" - (oe.eo)
## / \ ##  /*** Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
## \ / ##   > http://blog.gentilkiwi.com/mimikatz
'## v #'    Vincent LE TOUX ( vincent.letoux@gmail.com )
'#####'    > http://pingcastle.com / http://mysmartlogon.com   ***/
```

Usaremos el siguiente comando para que analice los registros **SAM** y **SYSTEM** (de los que deberemos indicar la ruta en la que se encuentran):

```
mimikatz # lsadump::sam /system:C:\Users\forensic\Desktop\Análisis_Evidencias\Kape\E\Windows\System32\
config\SYSTEM /sam:C:\Users\forensic\Desktop\Análisis_Evidencias\Kape\E\Windows\System32\config\SAM
Domain : PEGASUS01
SysKey : ec022a77f903a7e69e603e0c84634ff0
Local SID : S-1-5-21-321011808-3761883066-353627080
```

(también nos facilita el nombre de la máquina)

De este modo se nos mostrará un listado de los usuarios con información diversa de cada uno. A nosotros el que nos interesa será el **hash NTLM** del usuario **IEUser** (las contraseñas de los usuario de Windows siempre se guardan con un **hash NTLM**):

```
RID : 000003e8 (1000)
User : IEUser
Hash NTLM: 2d20d252a479f485cdf5e171d93985bf
```

Y con este **hash** en nuestro poder, podemos utilizar algún software o servicio web que nos permita descifrarla mediante diccionarios. Una página web conocida que nos permite realizar esta tarea es [crackstation](https://crackstation.net/). Copiaremos y pegaremos el **hash** mostrado en la web y obtendremos el resultado del reto:

2d20d252a479f485cdf5e171d93985bf	NTLM	qwerty
----------------------------------	------	--------

Efectivamente, el usuario estaba empleando una contraseña muy débil, fácil de romper mediante diccionarios y fuerza bruta.

Reto 9: Conexión programa control remoto

Existe la sospecha de que se hayan conectado al equipo desde un programa de control remoto. ¿Podrían decir cuál es el ID desde el se conecta el atacante?

Utilizando el **FTK Imager** para ir navegando por el árbol de archivos de la imagen de la evidencia, podemos irnos a la carpeta del software de control remoto, que en este caso es **Team Viewer**, y observar si encontramos algo de interés:

Evidence Tree

- Win10_PC001.vmdk
 - Partition 1 [40958MB]
 - Windows 10 [NTFS]
 - [orphan]
 - [root]
 - \$BadClus
 - \$Extend
 - \$Recycle.Bin
 - \$Secure
 - \$UpCase
 - BGinfo
 - Boot
 - Documents and Settings
 - inetpub
 - PerfLogs
 - Program Files
 - Adobe
 - Common Files
 - Google
 - internet explorer
 - LibreOffice
 - Microsoft Silverlight
 - Puppet Labs
 - TeamViewer**
 - outlook
 - Printer
 - x64
 - Uninstall Information
 - UNP
 - VMware

File List

Name	Size	Type	Date Modified
outlook	1	Directory	29/04/2022 17:...
Printer	1	Directory	29/04/2022 17:...
x64	1	Directory	29/04/2022 17:...
\$I30	16	NTFS Index All...	29/04/2022 10:...
Connections_incomin...	1	Regular File	29/04/2022 10:...
Copyrights.txt	1.816	Regular File	15/04/2022 7:1...
dpa-de.html	15	Regular File	15/04/2022 7:1...
dpa-de.html.FileSlack	2	File Slack	
dpa-en.html	15	Regular File	15/04/2022 7:1...
dpa-en.html.FileSlack	2	File Slack	
eula-de.html	112	Regular File	15/04/2022 7:1...
eula-de.html.FileSlack	1	File Slack	
eula-en.html	103	Regular File	15/04/2022 7:1...
eula-en.html.FileSlack	2	File Slack	
collatfile.txt	1	Regular File	20/04/2022 10:...

En el archivo *Connections_incoming.txt* nos da la ID que estamos buscando en este reto:

765418952	WIN-MORENIN	29-04-2022 10:09:14	29-04-2022 10:10:10	IEUser	RemoteControl	{adb13c9a-796c-438c-af8b-2079077f0a4f}
765418952	WIN-MORENIN	29-04-2022 10:10:34	29-04-2022 10:13:21	IEUser	RemoteControl	{301db382-67ac-4b5a-9bec-d1a44aa0ad30}

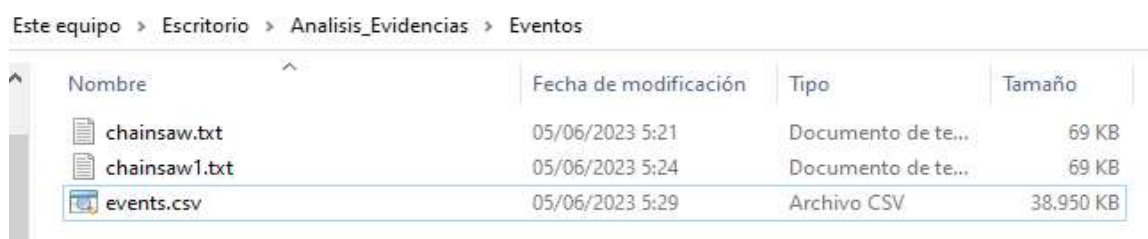
Reto 10: Conexión RDP y Reto 11: Puerto de conexión máquina atacante

Se ha detectado actividad sospechosa en la red. ¿Podrían indicar la IP desde la que se ha conectado la máquina por RDP?

Los registros de eventos de Windows nos pueden ofrecer información bastante precisa de si se han intentado conectar a nuestra máquina por algún canal determinado y mediante alguna aplicación determinada. Normalmente, los eventos de “*Security*” serán los que nos permitan visualizar información acerca de este tipo de intentos de acceso.

Podemos “*parsear*” los eventos de Windows mediante varias herramientas: **Chainsaw**, **Hayabusa**, **EvtxECmd**, etc. La cuestión será pasarle a una de estas herramientas el conjunto de eventos que queremos interpretar y abrirlo con un editor de texto que nos permita filtrar según la información que queremos localizar.

De este modo, mediante *Timeline Explorer*, abrimos el archivo con extensión .csv obtenido mediante **chainsaw**:



y dentro del editor, podríamos filtrar por “*channel: Security*” y por “*Remote Host*” para ver el listado de Ips que nos aparezcan.

A logon was attempted using explicit credentials	WORKGROUP\PEGASUS01\$	127.0.0.1:0	Target: PEGASUS01\IEUser
A logon was attempted using explicit credentials	PEGASUS01\IEUser	192.168.183.134:445	Target: PEGASUS01\user1
A logon was attempted using explicit credentials	PEGASUS01\IEUser	192.168.183.134:445	Target: PEGASUS01\user1
Successful logon	WORKGROUP\MSEdgeWIN10\$	MSEdgeWIN10 (127.0.0.1)	Target: MSEdgeWIN10\default
Successful logon	WORKGROUP\MSEdgeWIN10\$	MSEdgeWIN10 (127.0.0.1)	Target: MSEdgeWIN10\IEUser

Y obtenemos que el usuario **IEUser** se conectó mediante la IP **192.168.183.134** y el puerto 445.

Retos superados:

Challenges			
Forensic			
Hash del fichero ✓ 10	Nombre de la máquina ✓ 10	Ficheros maliciosos ✓ 15	Descarga fichero de control remoto ✓ 15
Ficheros eliminados ✓ 15	Puerto de conexión máquina atacante ✓ 15	Fecha de ejecución programa ✓ 20	Powershell maliciosa ✓ 25
Contraseñas débiles ✓ 25	Conexión programa control remoto ✓ 30	Conexión RDP ✓ 30	

Práctica Memoria RAM

Para este apartado de la práctica, se deberá de hacer una adquisición de memoria RAM sobre el sistema operativo a vuestra elección. Se deberán indicar los pasos seguidos para la realización de la adquisición, así como la ejecución de mínimo dos comandos con *volatility*.

Sistema Operativo: Windows 10 (la máquina virtual con herramientas forenses que llevamos usando en el desarrollo de los retos).

Adquisición:


La adquisición la haremos con la herramienta *WinPmem*, lanzando la consola de comandos como administrador.

Ejecutaremos el comando siguiente:

```
C:\Herramientas\01_Artefactos\RAM>winpmem_mini_x64_rc2.exe evidencia_ram_win.mem
```

Y obtendremos la adquisición:

```
4 memory ranges:
Start 0x00001000 - Length 0x0009E000
Start 0x00100000 - Length 0x00002000
Start 0x00103000 - Length 0xDFEED000
Start 0x100000000 - Length 0x1CE00000
max_physical_memory_ 0x11ce00000
Acquisition mode PTE Remapping
Padding from 0x00000000 to 0x00001000
```

 evidencia_ram_win.mem

09/06/2023 21:29

Archivo MEM

4.667.392 KB

Interpretación:

Para el *parseo* de la memoria RAM, utilizaremos la herramienta *Volatility*.

Por algún motivo que desconozco, tras hacer varias pruebas y usar varios comandos de *volatility*, no obtenía ningún tipo de resultado; por pantalla no aparece nada directamente, como si no hubiera ningún registro que analizar en el archivo *.mem* obtenido y los archivos que generaba como *output* pesaban 0kb y estaban vacíos. Por otro lado, el tamaño de la adquisición pareciera ser el correcto, pues es de 4Gb, que es la RAM que tiene la máquina virtual de Windows 10 que se está usando.

2 ejemplos al respecto:

Comando para sacar los procesos registrados en la RAM:

```
C:\Herramientas\01_Artefactos\RAM\volatility3-1.0.0\volatility3-1.0.0>vol.py
-f "C:\Users\forensic\Desktop\Analisis_Evidencias\RAM\adquisicion_ram_win.mem"
windows.psscan
```

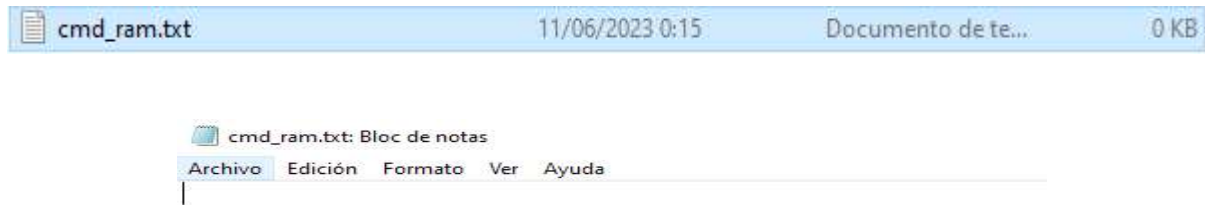
Resultado:

```
Volatility 3 Framework 1.0.0
Progress: 100.00 PDB scanning finished
PID PPID ImageFileName Offset Threads Handles SessionId Wow64 CreateTime ExitTime File output
_
```

O utilizando el comando para sacar los comandos utilizados por el usuario del sistema:

```
C:\Herramientas\01_Artefactos\RAM\volatility3-1.0.0\volatility3-1.0.0>vol.py -f "C:\Users\forensic\Desktop\Análisis_Evidencias\RAM\adquisicion_ram_win.mem" windows.cmdline.Cmdline >> C:\Users\forensic\Desktop\Análisis_Evidencias\RAM\cmd_ram.txt
```

Resultado obtenido:



(Si supieras la razón de ello, encantado de conocerla y poderlo probar de nuevo)

Práctica Metadatos

La idea de este ejercicio es examinar cómo las plataformas de mensajería quitan una serie de metadatos cuando las enviamos entre unas y otras. Necesito que hagáis una prueba con una foto vuestra:

1. Miréis los metadatos que tiene inicialmente
2. La envíen por whatsapp y los volváis a mirar
3. La envíen por telegram y lo volváis a comparar
4. La enviéis por email y la comparéis

La imagen que vamos a analizar va a ser la siguiente:



Y la herramienta para analizar los metadatos de la misma será **Exiftool**. La ejecutaremos desde Linux, previamente habiéndola clonado desde *Github*.

El resultado que obtenemos de los metadatos en su estado original es el siguiente:

```
ExifTool Version Number      : 12.57
File Name                    : playa.jpg
Directory                   : .
File Size                    : 3.2 MB
File Modification Date/Time  : 2023:06:10 21:43:38-04:00
File Access Date/Time       : 2023:06:10 21:43:38-04:00
File Inode Change Date/Time  : 2023:06:10 21:43:38-04:00
File Permissions             : -rw-rw-rw-
File Type                   : JPEG
File Type Extension         : jpg
MIME Type                   : image/jpeg
Exif Byte Order              : Little-endian (Intel, II)
Make                        : samsung
Camera Model Name           : SM-G988B
Orientation                 : Horizontal (normal)
X Resolution                 : 72
Y Resolution                 : 72
Resolution Unit             : inches
Software                    : G988BXXSCDUK1
Modify Date                 : 2022:01:17 17:10:48
Y Cb Cr Positioning         : Centered
Exposure Time               : 1/2344
F Number                    : 2.2
Exposure Program            : Program AE
ISO                          : 64
Exif Version                : 0220
Date/Time Original          : 2022:01:17 17:10:48
Create Date                 : 2022:01:17 17:10:48
Offset Time                 : +01:00
Offset Time Original        : +01:00
Shutter Speed Value         : 1
Aperture Value              : 2.2
Brightness Value            : 18.5
Exposure Compensation       : 0
Max Aperture Value          : 2.2
Metering Mode               : Center-weighted average
Flash                       : No Flash
Focal Length                : 2.2 mm
Color Space                 : sRGB
Exif Image Width            : 4000
Exif Image Height           : 2252
Exposure Mode               : Auto
White Balance               : Auto
Digital Zoom Ratio          : 1
Focal Length In 35mm Format : 13 mm
Scene Capture Type          : Standard
Image Unique ID             : 512LLMF01MM
Compression                 : JPEG (old-style)
Thumbnail Offset            : 814
Thumbnail Length            : 50310
Image Width                 : 4000
Image Height                : 2252
Encoding Process            : Baseline DCT, Huffman coding
Bits Per Sample             : 8
Color Components            : 3
Y Cb Cr Sub Sampling        : YCbCr4:2:0 (2 2)
Time Stamp                  : 2022:01:17 11:10:49.032-05:00
MCC Data                    : Spain
Aperture                    : 2.2
Image Size                  : 4000x2252
Megapixels                  : 9.0
Scale Factor To 35 mm Equivalent: 5.9
Shutter Speed               : 1/2344
Date/Time Original          : 2022:01:17 17:10:48+01:00
Modify Date                 : 2022:01:17 17:10:48+01:00
Thumbnail Image             : (Binary data 50310 bytes, use -b option to
extract)
Circle Of Confusion         : 0.005 mm
Field Of View               : 108.3 deg
Focal Length                : 2.2 mm (35 mm equivalent: 13.0 mm)
Hyperfocal Distance         : 0.44 m
Light Value                 : 14.1
```


Nos la pasamos por **mail** (Google Mail) y la descargamos para comprobar si hay alguna alteración remarcable:

```
ExifTool Version Number      : 12.57
File Name                    : playa_mail.jpg
Directory                    : .
File Size                     : 3.2 MB
File Modification Date/Time   : 2023:06:10 21:53:16-04:00
File Access Date/Time        : 2023:06:10 21:53:17-04:00
File Inode Change Date/Time   : 2023:06:10 21:53:16-04:00
File Permissions              : -rw-rw-rw-
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
Exif Byte Order               : Little-endian (Intel, II)
Make                         : samsung
Camera Model Name             : SM-G988B
Orientation                   : Horizontal (normal)
X Resolution                  : 72
Y Resolution                  : 72
Resolution Unit               : inches
Software                     : G988BXXSCDUK1
Modify Date                   : 2022:01:17 17:10:48
Y Cb Cr Positioning           : Centered
Exposure Time                 : 1/2344
F Number                      : 2.2
Exposure Program              : Program AE
ISO                           : 64
Exif Version                  : 0220
Date/Time Original            : 2022:01:17 17:10:48
Create Date                   : 2022:01:17 17:10:48
Offset Time                   : +01:00
Offset Time Original          : +01:00
Shutter Speed Value           : 1
Aperture Value                : 2.2
Brightness Value              : 18.5
Exposure Compensation         : 0
Max Aperture Value            : 2.2
Metering Mode                 : Center-weighted average
Flash                        : No Flash
Focal Length                  : 2.2 mm
Color Space                   : sRGB
Exif Image Width              : 4000
Exif Image Height             : 2252
Exposure Mode                 : Auto
White Balance                  : Auto
Digital Zoom Ratio            : 1
Focal Length In 35mm Format   : 13 mm
Scene Capture Type            : Standard
Image Unique ID               : 512LLMF01MM
Compression                   : JPEG (old-style)
Thumbnail Offset              : 814
Thumbnail Length              : 50310
Image Width                   : 4000
Image Height                  : 2252
Encoding Process               : Baseline DCT, Huffman coding
Bits Per Sample               : 8
Color Components              : 3
Y Cb Cr Sub Sampling          : YCbCr4:2:0 (2 2)
Time Stamp                    : 2022:01:17 11:10:49.032-05:00
MCC Data                      : Spain
Aperture                      : 2.2
Image Size                    : 4000x2252
Megapixels                   : 9.0
Scale Factor To 35 mm Equivalent: 5.9
Shutter Speed                 : 1/2344
Date/Time Original            : 2022:01:17 17:10:48+01:00
Modify Date                   : 2022:01:17 17:10:48+01:00
Thumbnail Image               : (Binary data 50310 bytes, use -b option to extract)
Circle Of Confusion           : 0.005 mm
Field Of View                 : 108.3 deg
Focal Length                  : 2.2 mm (35 mm equivalent: 13.0 mm)
Hyperfocal Distance           : 0.44 m
Light Value                   : 14.1
```

En este caso contendrían exactamente los mismo metadatos.

Ahora se hace la comprobación descargándola desde **Whatsapp**:

```
Exiftool Version Number      : 12.57
File Name                    : playa_whatsapp.jpeg
Directory                    : .
File Size                    : 287 kB
File Modification Date/Time   : 2023:06:10 22:04:54-04:00
File Access Date/Time        : 2023:06:10 22:04:54-04:00
File Inode Change Date/Time   : 2023:06:10 22:04:54-04:00
File Permissions              : -rw-rw-rw-
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
JFIF Version                 : 1.01
Resolution Unit               : None
X Resolution                  : 1
Y Resolution                  : 1
Image Width                  : 1600
Image Height                  : 900
Encoding Process              : Progressive DCT, Huffman coding
Bits Per Sample               : 8
Color Components              : 3
Y Cb Cr Sub Sampling         : YCbCr4:2:0 (2 2)
Image Size                   : 1600x900
Megapixels                   : 1.4
```

El resultado es notablemente diferente. Se ha reducido mucho la cantidad de información obtenida, el peso del archivo se ha reducido drásticamente gracias a la famosa compresión que aplica **whatsapp** cuando se envían imágenes, se ha redimensionado el tamaño, se ha reducido la resolución (de 9.0 Mp a 1.4), no obtenemos información sobre el día en el que se hizo la fotografía, etc.

Por último, haremos lo mismo utilizando la aplicación de **Telegram**:

```
ExifTool Version Number      : 12.57
File Name                    : playa_telegram.jpeg
Directory                    : .
File Size                    : 313 kB
File Modification Date/Time   : 2023:06:10 22:19:16-04:00
File Access Date/Time        : 2023:06:10 22:19:17-04:00
File Inode Change Date/Time   : 2023:06:10 22:19:16-04:00
File Permissions              : -rw-rw-rw-
File Type                    : JPEG
File Type Extension          : jpg
MIME Type                    : image/jpeg
JFIF Version                 : 1.01
Resolution Unit               : inches
X Resolution                  : 72
Y Resolution                  : 72
Profile CMM Type              : 
Profile Version               : 4.3.0
Profile Class                 : Display Device Profile
Color Space Data              : RGB
Profile Connection Space      : XYZ
Profile Date Time             : 0000:00:00 00:00:00
Profile File Signature        : acsp
Primary Platform              : Unknown ( )
CMM Flags                     : Not Embedded, Independent
Device Manufacturer          : 
Device Model                  : 
Device Attributes             : Reflective, Glossy, Positive, Color
Rendering Intent              : Media-Relative Colorimetric
Connection Space Illuminant   : 0.9642 1 0.82491
Profile Creator               : 
Profile ID                    : 0
Profile Description           : sRGB
Red Matrix Column             : 0.43607 0.22249 0.01392
Green Matrix Column           : 0.38515 0.71687 0.09708
Blue Matrix Column            : 0.14307 0.06061 0.7141
Red Tone Reproduction Curve   : (Binary data 40 bytes, use -b option to extract)
Green Tone Reproduction Curve : (Binary data 40 bytes, use -b option to extract)
Blue Tone Reproduction Curve  : (Binary data 40 bytes, use -b option to extract)
Media White Point             : 0.9642 1 0.82491
Profile Copyright             : Google Inc. 2016
Image Width                   : 1280
Image Height                  : 720
Encoding Process              : Progressive DCT, Huffman coding
Bits Per Sample               : 8
Color Components              : 3
Y Cb Cr Sub Sampling         : YCbCr4:2:0 (2 2)
Image Size                   : 1280x720
Megapixels                   : 0.922
```

En **Telegram** apreciamos que se conservan más metadatos que en **whatsapp**, la comprensión es prácticamente la misma (algo inferior en **Telegram**) y la resolución también es ligeramente inferior; también se ha redimensionado en este caso. Seguimos sin tener información sobre la fecha de creación de la fotografía, etc.

Aunque debemos tener en cuenta que **whatsapp** y **Telegram** nos permiten enviar archivos como documentos, siendo ésta la forma de respetar al máximo posible el archivo original que queremos transmitir. A modo de ejemplo, vamos a verlo con Whatsapp:

```
ExifTool Version Number      : 12.57
File Name                    : playa_whatts.jpg
Directory                    : .
File Size                     : 3.2 MB
File Modification Date/Time   : 2023:06:10 22:27:58-04:00
File Access Date/Time        : 2023:06:10 22:27:59-04:00
File Inode Change Date/Time   : 2023:06:10 22:27:58-04:00
File Permissions              : -rw-rw-rw-
File Type                    : JPEG
File Type Extension           : jpg
MIME Type                     : image/jpeg
Exif Byte Order               : Little-endian (Intel, II)
Make                          : samsung
Camera Model Name             : SM-G988B
Orientation                   : Horizontal (normal)
X Resolution                   : 72
Y Resolution                   : 72
Resolution Unit                : inches
Software                      : G988BXXSCDUK1
Modify Date                   : 2022:01:17 17:10:48
Y Cb Cr Positioning           : Centered
Exposure Time                 : 1/2344
F Number                      : 2.2
Exposure Program              : Program AE
ISO                            : 64
Exif Version                  : 0220
Date/Time Original            : 2022:01:17 17:10:48
Create Date                   : 2022:01:17 17:10:48
Offset Time                   : +01:00
Offset Time Original          : +01:00
Shutter Speed Value           : 1
Aperture Value                : 2.2
Brightness Value              : 18.5
Exposure Compensation         : 0
Max Aperture Value            : 2.2
Metering Mode                 : Center-weighted average
Flash                         : No Flash
Focal Length                  : 2.2 mm
Color Space                   : sRGB
Exif Image Width              : 4000
Exif Image Height             : 2252
Exposure Mode                 : Auto
White Balance                 : Auto
Digital Zoom Ratio            : 1
Focal Length In 35mm Format   : 13 mm
Scene Capture Type            : Standard
Image Unique ID               : 512LLMF01MM
Compression                   : JPEG (old-style)
Thumbnail Offset              : 814
Thumbnail Length              : 50310
Image Width                   : 4000
Image Height                  : 2252
Encoding Process               : Baseline DCT, Huffman coding
Bits Per Sample               : 8
Color Components              : 3
Y Cb Cr Sub Sampling          : YCbCr4:2:0 (2 2)
Time Stamp                    : 2022:01:17 11:10:49.032-05:00
MCC Data                      : Spain
Aperture                      : 2.2
Image Size                    : 4000x2252
Megapixels                    : 9.0
Scale Factor To 35 mm Equivalent: 5.9
Shutter Speed                 : 1/2344
Date/Time Original            : 2022:01:17 17:10:48+01:00
Modify Date                   : 2022:01:17 17:10:48+01:00
Thumbnail Image               : (Binary data 50310 bytes, use -b option to extract)
Circle Of Confusion           : 0.005 mm
Field Of View                  : 108.3 deg
Focal Length                  : 2.2 mm (35 mm equivalent: 13.0 mm)
Hyperfocal Distance          : 0.44 m
Light Value                   : 14.1
```

El resultado es exactamente igual que la original, puesto que se está limitando a transmitir el archivo desde su plataforma, sin más.

