

---

# INFORME DFIR

---

## METADATOS

Vamos a tomar una foto con la cámara de mi notebook en Windows y analizar los metadatos con la herramienta exiftool.

*Probaremos la primera fase de la foto original con sus metadatos*

Comando

"exiftool(-k).exe" metadatos.jpg

```
Microsoft Windows [Versión 10.0.22621.2861]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\sebb\Desktop\KEEP CODING\CIBERSEGURIDAD\FORENSE\exiftool-12.72>"exiftool(-k).exe" metadatos.jpg
ExifTool Version Number      : 12.72
File Name                    : metadatos.jpg
Directory                    : .
File Size                     : 121 kB
Zone Identifier               : Exists
File Modification Date/Time   : 2023:12:31 15:36:23-03:00
File Access Date/Time        : 2024:01:01 14:49:35-03:00
File Creation Date/Time      : 2024:01:01 14:49:34-03:00
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                  : 96
Y Resolution                  : 96
Exif Byte Order               : Big-endian (Motorola, MM)
Software                      : Windows 11
Date/Time Original            : 2023:12:31 15:36:19
Sub Sec Time Original         : 170
GPS Latitude Ref              : South
GPS Longitude Ref             : West
Padding                       : (Binary data 4108 bytes, use -b option to extract)
Image Width                   : 1280
Image Height                  : 720
Encoding Process              : Baseline 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
Date/Time Original            : 2023:12:31 15:36:19.170
GPS Latitude                   : 38 deg 57' 38.69" S
GPS Longitude                  : 68 deg 4' 2.96" W
GPS Position                   : 38 deg 57' 38.69" S, 68 deg 4' 2.96" W
-- press ENTER --
```

Segundo paso enviamos la foto por whats app y volvemos a descargar la foto enviada para analizar que metadatos cambiaron

## COMANDO

"exiftool(-k).exe" metadatos\_wpp.jpg

```
Microsoft Windows [Versión 10.0.22621.2861]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\seba\Desktop\KEEP CODING\CIBERSEGURIDAD\FORENSE\exiftool-12.72>"exiftool(-k).exe" metadatos_wpp.jpg
ExifTool Version Number      : 12.72
File Name                    : metadatos_wpp.jpg
Directory                    : .
File Size                     : 114 kB
Zone Identifier               : Exists
File Modification Date/Time   : 2024:01:01 17:13:41-03:00
File Access Date/Time        : 2024:01:01 17:15:08-03:00
File Creation Date/Time      : 2024:01:01 17:14:10-03:00
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                  : 96
Y Resolution                  : 96
Image Width                   : 1280
Image Height                  : 720
Encoding Process               : Baseline 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
-- press ENTER --
```

## ANALIZANDO LOS CAMBIOS EN LOS METADATOS PODEMOS OBSERVAR:

- 1) El peso de la foto disminuyo
- 2) Las fechas de creación, modificación y acceso cambiaron a la hora que se subió a whats app y fue abierta.
- 3) Desaparecieron los siguientes metadatos:
  - La forma en que se almacenan los bytes en el archivo
  - El software que se utilizo
  - La latitud, longitud y posicion del gps del lugar donde se tomo la foto
  - El padding original de la foto
  - La fecha y hora original de creación del archivo

Ahora enviaremos la foto a través de telegram y analizaremos que cambios surgen en los metadatos luego de descargarla.

## COMANDO

"exiftool(-k).exe" metadatos\_telergam.jpg

```
C:\Users\sebb\Desktop\KEEP CODING\CIBERSEGURIDAD\FORENSE\exiftool-12.72>"exiftool(-k).exe" metadatos_telergam.jpg
ExifTool Version Number      : 12.72
File Name                    : metadatos_telergam.jpg
Directory                    : .
File Size                    : 116 kB
File Modification Date/Time   : 2024:01:01 18:16:02-03:00
File Access Date/Time        : 2024:01:01 18:16:03-03:00
File Creation Date/Time       : 2024:01:01 18:16:02-03:00
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                  : 96
Y Resolution                  : 96
Exif Byte Order               : Big-endian (Motorola, MM)
Software                      : Windows 11
Date/Time Original            : 2023:12:31 15:36:19
Sub Sec Time Original         : 170
GPS Latitude Ref              : South
GPS Longitude Ref             : West
Padding                      : (Binary data 4108 bytes, use -b option to extract)
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
Date/Time Original            : 2023:12:31 15:36:19.170
GPS Latitude                  : 38 deg 57' 38.69" S
GPS Longitude                 : 68 deg 4' 2.96" W
GPS Position                  : 38 deg 57' 38.69" S, 68 deg 4' 2.96" W
-- press ENTER --
```

*Comparando los resultados de metadatos otorgados por exiftool podemos observar que también hay cambios:*

- 1) Los horarios de creación, acceso y modificación cambiaron al horario que se subió el archivo a telegram y fue abierto. Pero sigue apareciendo la fecha y hora de creación del archivo original
- 2) Desapareció la “zone identifier” que es la zona desde donde la cual se descargo el archivo, aunque no nos daba información relevante de la fuente de aparición original.
- 3) El proceso de codificación o conversión de datos cambio

## Ahora analizaremos los metadatos de la imagen enviada por Gmail

### COMANDO

"exiftool(-k).exe" metadatos\_gmail.jpg

```
C:\Users\sebba\Desktop\KEEP CODING\CIBERSEGURIDAD\FORENSE\exiftool-12.72>"exiftool(-k).exe" metadatos_gmail.jpg
ExifTool Version Number      : 12.72
File Name                    : metadatos_gmail.jpg
Directory                    : .
File Size                    : 96 kB
Zone Identifier               : Exists
File Modification Date/Time   : 2024:01:01 21:28:32-03:00
File Access Date/Time        : 2024:01:01 21:28:33-03:00
File Creation Date/Time      : 2024:01:01 21:28:32-03:00
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                  : 96
Y Resolution                  : 96
Image Width                   : 1199
Image Height                  : 675
Encoding Process               : Baseline DCT, Huffman coding
Bits Per Sample                : 8
Color Components               : 3
Y Cb Cr Sub Sampling          : YCbCr4:2:0 (2 2)
Image Size                    : 1199x675
Megapixels                    : 0.809
-- press ENTER --
```

*Comparando los resultados de metadatos otorgados por exiftool podemos observar que Gmail es la plataforma de mensajería con mayor pérdida de metadatos de las 3 analizadas:*

- 1) El tamaño del archivo disminuyó considerablemente
- 2) La fecha de creación, modificación y acceso cambiaron al horario que se envió el email
- 3) El tamaño de la imagen en píxeles cambió de 1280x720 a 1199x675
- 4) Los megapíxeles disminuyeron de 0.922 a 0.809
- 5) Desaparecieron los datos:
  - Exif byte order
  - Software del sistema que creó la imagen
  - La fecha de la creación del archivo original
  - Sub sec time original
  - La latitud, longitud y posición del GPS para encontrar la ubicación donde se creó el archivo
  - Las referencias de latitud y longitud del gps
  - El padding

---

# CHALLENGE CTF

---

En el siguiente ejercicio resolveremos un Capture The Flag de análisis forense

## FLAG HASH DEL FICHERO

Para sacar el hash de la evidencia utilice la herramienta Get-FileHash de powershell de mi Windows host.

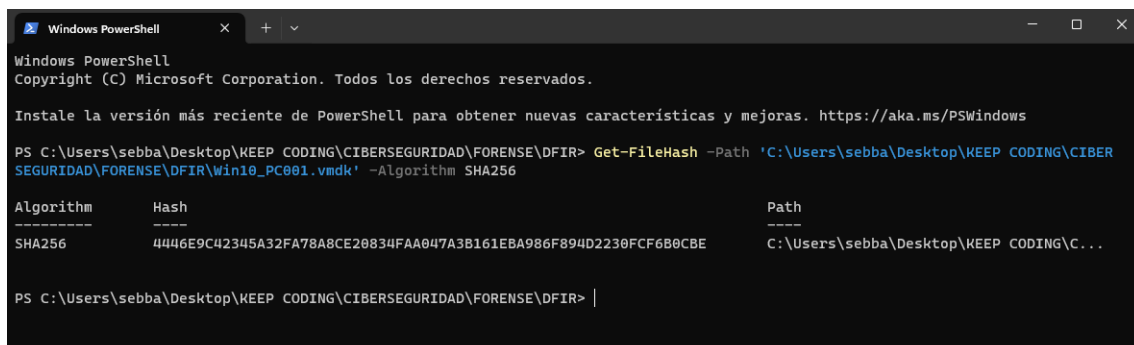
Get-FileHash es un cmdlet de PowerShell que se utiliza para calcular el hash (resumen criptográfico) de un archivo.

Antes de realizar el calculo del hash, debemos corroborar que la evidencia no fue abierta ni alterada por ningún software, de lo contrario el calculo del hash cambiara.

Para conseguir el hash simplemente abri una powershell dentro de la carpeta donde se encuentra la evidencia y ejecute el siguiente comando para calcular su hash sha 256

Comando:

```
Get-FileHash -Path 'C:\Users\sebb\Desktop\KEEP CODING\CIBERSEGURIDAD\Forensic\DFIR> Get-FileHash -Path 'C:\Users\sebb\Desktop\KEEP CODING\CIBERSEGURIDAD\Forensic\DFIR\Win10_PC001.vmdk' -Algorithm SHA256
```



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\sebb\Desktop\KEEP CODING\CIBERSEGURIDAD\Forensic\DFIR> Get-FileHash -Path 'C:\Users\sebb\Desktop\KEEP CODING\CIBERSEGURIDAD\Forensic\DFIR\Win10_PC001.vmdk' -Algorithm SHA256

Algorithm      Hash
-----
SHA256         4446E9C42345A32FA78A8CE20834FAA047A3B161EBA986F894D2230FCF6B0CBE
Path
-----
C:\Users\sebb\Desktop\KEEP CODING\CIBERSEGURIDAD\Forensic\DFIR\Win10_PC001.vmdk

PS C:\Users\sebb\Desktop\KEEP CODING\CIBERSEGURIDAD\Forensic\DFIR>
```

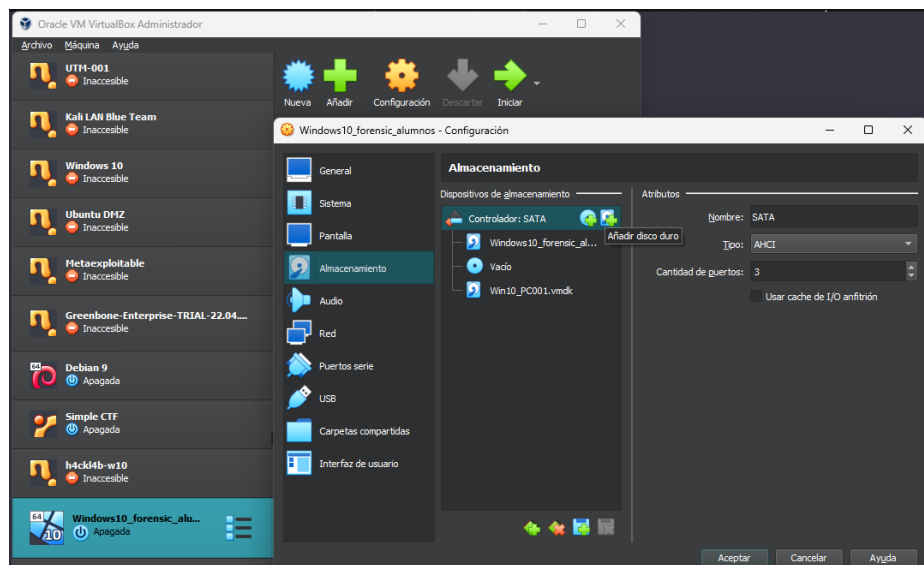
Obtuve el hash:

4446E9C42345A32FA78A8CE20834FAA047A3B161EBA986F894D223  
0FCF6B0CBE

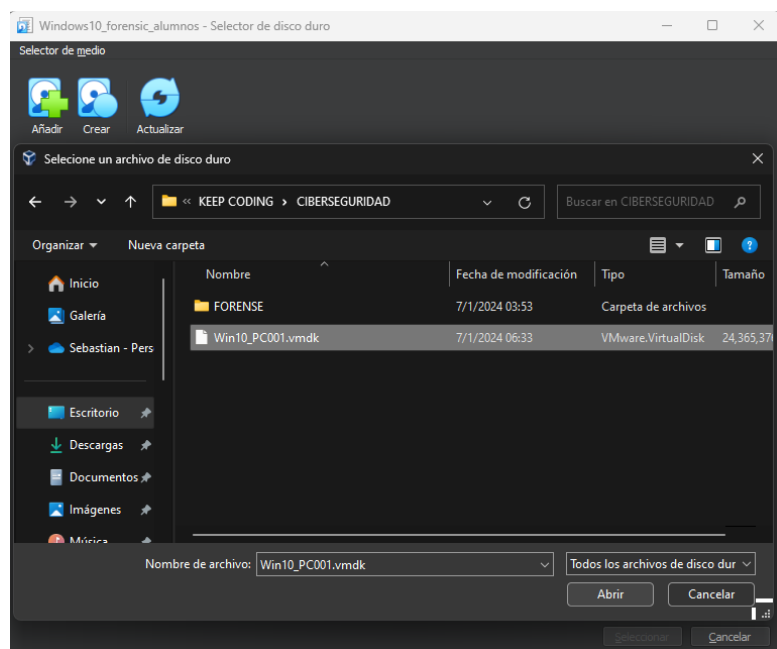
# FLAG NOMBRE DE LA MAQUINA

Lo primero que debemos hacer para obtener el nombre de la maquina es montar la evidencia dentro de nuestra maquina virtual utilizada para analizar, para esto debemos tener apagado Windows Forense

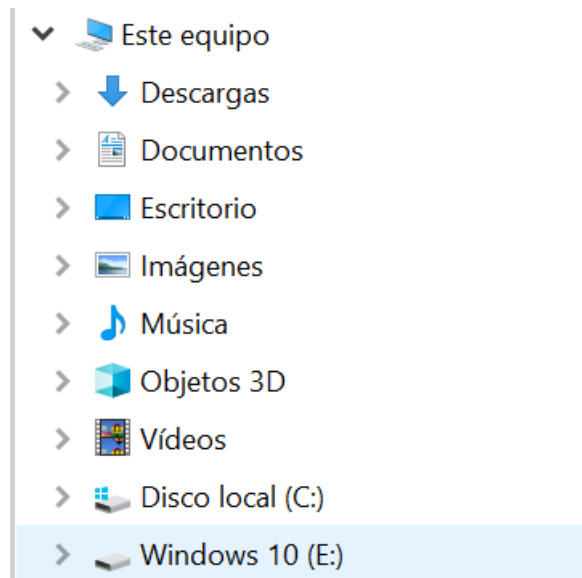
En virtual box dentro de configuración, en la pestana de almacenamiento debemos agregar un disco duro



Luego Anadimos nuestro archivo de evidencia como disco duro y guardamos

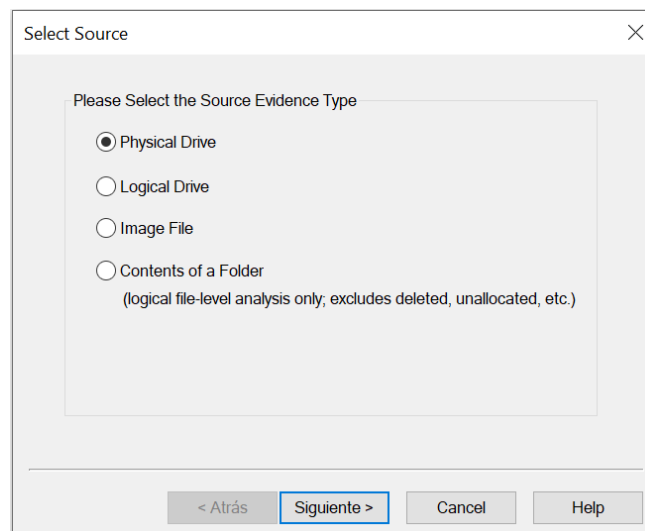


Ahora cuando prendamos nuestra Windows Forense podremos ver la evidencia cargada como disco duro lista para ser utilizada por software de analisis.



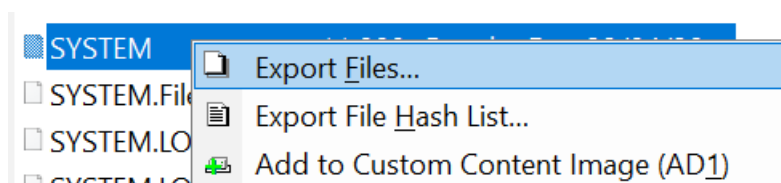
Utilice las herramientas FTK y RegRipper para encontrar el nombre de la maquina.

Para esto cargue el disco de la evidencia en FTK como disco físico

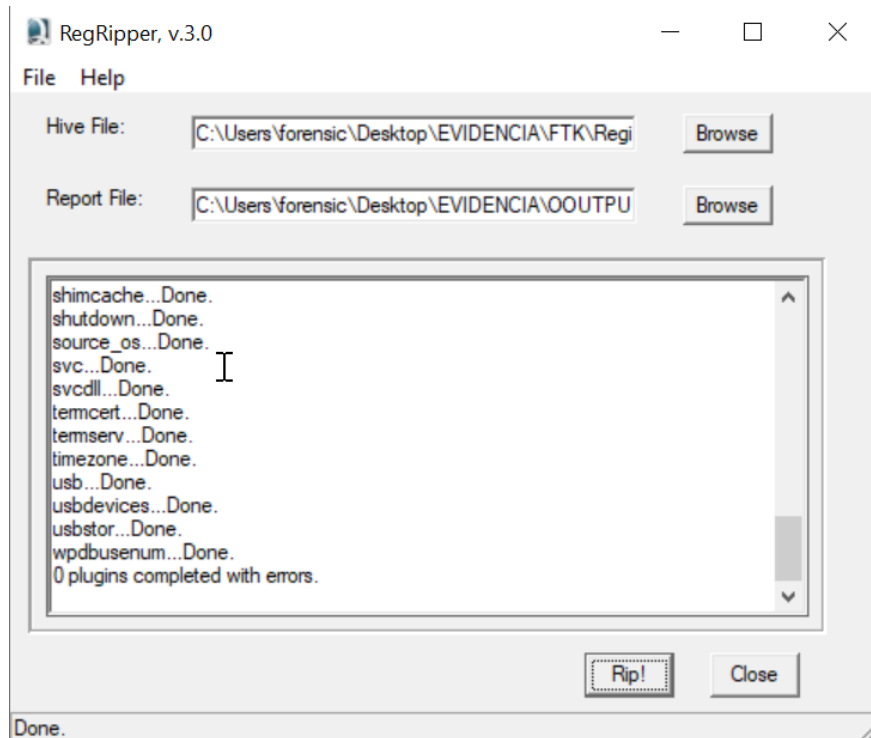


Luego extraje el fichero de Registros de Windows SYSTEM dentro de la ruta

C:/Windows/system32/Config de la evidencia.



Parsie el fichero SYSTEM con RegRipper para hacerlo legible



Lo que me dio un fichero txt donde se encuentran los datos del sistema

```
-----  
compname v.20090727  
(System) Gets ComputerName and Hostname values from System hive  
  
ComputerName      = PEGASUS01  
TCP/IP Hostname   = PEGASUS01  
-----
```

Dentro de la sección compname del output txt conseguido a través de RegRipper obtuvimos el  
nombre de la maquina

Resultado:

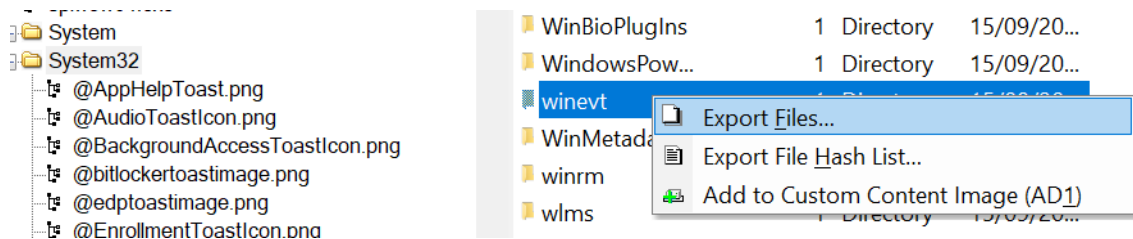
PEGASUS01



# FLAG Ficheros maliciosos y Powershell Maliciosa

El objetivo de esta bandera es descubrir en que carpeta hay un script de comandos powershell malicioso.

Para encontrarlo extraje con FTK la carpeta Logs en el directorio Windows/System32/winevt

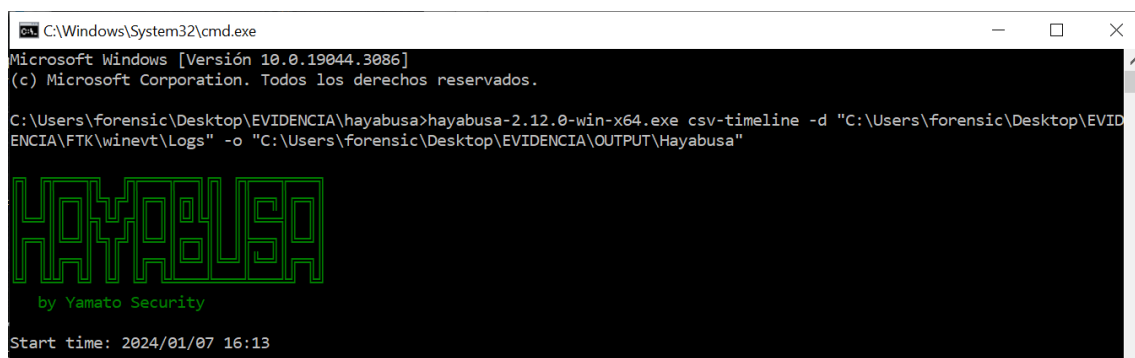


Luego ejecute la herramienta Hayabusa en Logs para realizar mi análisis de malware forense

Ejecutamos un cmd en la carpeta del ejecutable de Hayabusa

## COMANDO:

```
hayabusa-2.12.0-win-x64.exe csv-timeline -d  
"C:\Users\forensic\Desktop\EVIDENCIA\FTK\winevt\Logs" -o  
"C:\Users\forensic\Desktop\EVIDENCIA\OUTPUT\Hayabusa"
```







Abrimos el output de hayabusa con el TimeLine explorer e iniciamos el análisis para encontrar el fichero y la carpeta.

Encontre excesivos movimientos sospechosos y alertas de seguridad, por lo que decidí investigar cronológicamente desde la primer alerta registrada hasta la mas reciente utilizando la columna de Timestamp.

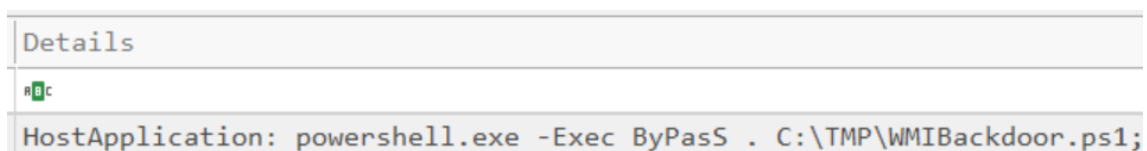
Con esta columna podemos ordenar los eventos en lapso temporal e iniciar en el primer o ultimo suceso.



Analizando los primeros movimiento sospechoso desde que la maquina se vulnero descubri un fichero de PowerShell conocido como un backdoor con fines demostrativos para ilustrar técnicas ofensivas.

Line	Tag	Timestamp	Computer	Channel
=				
1879	<input type="checkbox"/>	2022-05-08 21:09:55.971 +02:00	PEGASUS01	PwShClassic
1878	<input type="checkbox"/>	2022-05-08 21:09:54.550 +02:00	PEGASUS01	Sec
1877	<input type="checkbox"/>	2022-05-08 21:09:53.965 +02:00	PEGASUS01	Sec

En la columna de detalles estaba el nombre del archivo powershell y la carpeta donde se alojo



Definitivamente este fichero inicio los daños en el sistema

C:\TMP\WMIBackdoor.ps1


FICHERO MALICIOSO : TMP

POWERSHELL MALICIOSA: WMIBackdoor.ps1

# FLAG Descarga de Fichero de Control Remoto

Para descubrir el fichero de control remoto utilice el mismo csv de hayabusa ordenando cronológicamente la fecha de inicio y agregando el filtro “remote”.

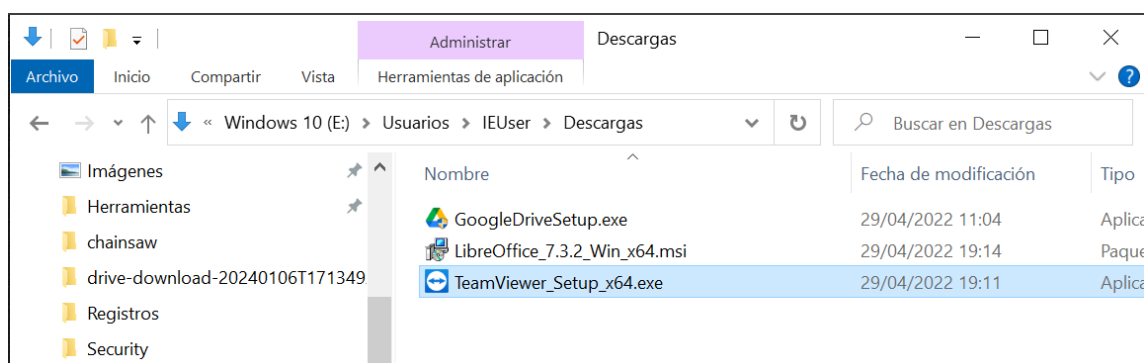
El segundo evento del filtro tenia en detalles la palabra Team Viewer que es un programa para Escritorios Remotos

Rule Title
 nmap
Potential CVE-2023-23397 Exploitation Attempt - SMB
Remote Access Tool Services Have Been Installed - System

En la columna detalles podíamos observar un archivo llamado “TeamViewer\_Service.exe”, aunque este no era el fichero de descarga que la flag nos solicita.

Svc: TeamViewer | Path: "C:\Program Files\TeamViewer\TeamViewer\_Service.exe"

Por lo que decidí ir a la carpeta de descargas de IEUser y encontré el fichero de descarga de Teamviewer

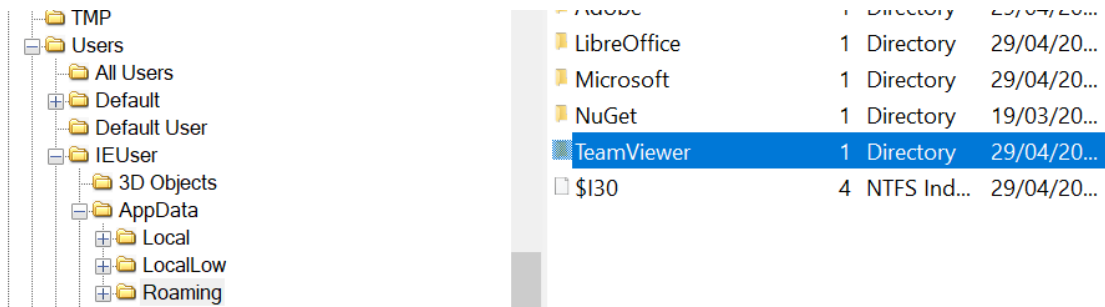


FICHERO DESCARGADO DE CONTROL REMOTO:

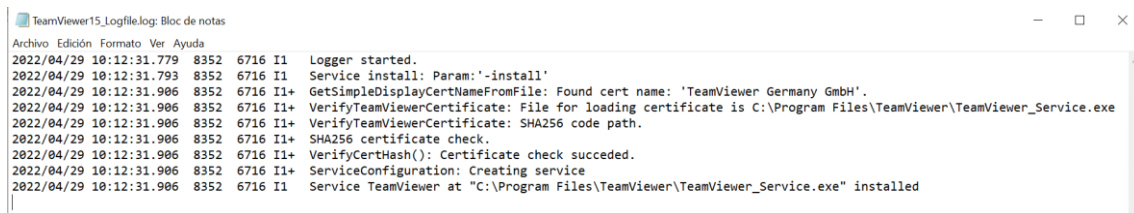
TeamViewer.Setup\_x64.exe

# FLAG Fecha Descarga Software Control Remoto

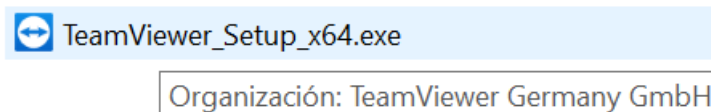
Continuando decidí investigar los datos del fichero Roaming sobre Team Viewer ya que en el directorio Users\IEUser\AppData\Roaming las aplicaciones pueden almacenar archivos de configuración y datos personalizados para un usuario. Por ejemplo el historial de uso.



Dentro de Roaming extrajimos un fichero específico con el historial de su descarga



El nombre del archivo descargable es correlativo con la lista obtenida de Roaming, por lo que nos provee el día que se ejecuto el archivo descargable



2022/04/29 'TeamViewer Germany GmbH'

2022/04/29 10:12:31.906 8352 6716 I1+ GetSimpleDisplayCertNameFromFile: Found cert name: 'TeamViewer Germany GmbH'.

El enunciado acepta la fecha unicamente con el formato aaaa-mm-dd

La fecha de descarga Team Viewer: 2022/04/29

## FLAG Fecha de Ejecución Programa de Control Remoto

En el fichero de salida csv obtenido por Hayabusa habíamos encontrado el evento de Team Viewer de cuando se ejecuto el servicio .exe del programa de control remoto.

Filtramos la búsqueda por “TeamViewer” y obtenemos la fecha y hora exacta.

```
Svc: TeamViewer | Path: "C:\Program Files\TeamViewer\TeamViewer_Service.exe"
```

2022-04-29

La fecha de ejecución del programa de control remoto: 2022/04/29

## FLAG Ip y Puerto de Conexión Maquina Atacante

Para obtener el puerto por el que ingreso el atacante y la IP de su maquina utilice otra vez el csv del directorio Logs extraído de la evidencia en la dirección Windows/System32/winevt obtenido por Hayabusa.

Analizando la terminología que utiliza Hayabusa en el Timeline Explorer deduje que cuando se especifica sobre el Protocolo de Internet (IP) utiliza el termino “SrcIP”

Con esta deducción filtre por la palabra SrcIP.

Como resultado reduje todos los eventos a 2 opciones de IP:

A) 127.0.0.1 (LOCAL O LOCALHOST)

B) 192.168.183.134

Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
TgtUser: user1	SrcUser: IEUser	SrcIP: 192.168.183.134	Proc:	TgtSvr: dev
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
TgtUser: user1	SrcUser: IEUser	SrcIP: 192.168.183.134	Proc:	TgtSvr: dev
Type: 3 - NETWORK	TgtUser: Guest	SrcComp: PEGASUS01	SrcIP: -	AuthPkg: Negotiate   Proc: C:\Windows\explorer.exe   SubStatus: 0xc0000072
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7
Type: 5 - SERVICE	TgtUser: SYSTEM	SrcComp: -	SrcIP: -	LID: 0x3e7

Filtramos los eventos con direcciones de localhost que es una direccion de loopback.

Buscamos unicamente las que tienen la direccion restante.

rastrea una columna aquí para agrupar por dicha columna		192.168.183.134	x	Buscar
Extra Field Info				
<div> <div></div> <div> <div>AddressLength: 16   ConnectionType: 1   InstanceName: \Device\LanmanRedirector   InstanceNameLength: 24   LocalAddress: 00000000000000000000000000000000   Reason: 4   RemoteAddress: 020001BDC0A8B7860000000000000000   AddressLength: 16   ServerName: 192.168.183.134   ServerNameLength: 16   SessionId: 2837665060   Status: 0   TreeId: 0   IpPort: 445   LogonGuid: 00000000-0000-0000-0000-000000000000   ProcessId: 0x4   SubjectDomainName: PEGASUS01   SubjectLogonId: 0x707c1   SubjectUserSid: S-1-5-21-321011808-37611   Address: 020001BDC0A8B7860000000000000000   AddressLength: 16   ConnectionType: 1   Reason: 7   ServerName: 192.168.183.134   ServerNameLength: 16   Status: 3221225996   IpPort: 445   LogonGuid: 00000000-0000-0000-0000-000000000000   ProcessId: 0x4   SubjectDomainName: PEGASUS01   SubjectLogonId: 0x707c1   SubjectUserSid: S-1-5-21-321011808-37611</div> </div> </div>				

Hemos conseguido los dos resultados en la línea 1278

IP del atacante

TgtUser: user1	SrcUser: IEUser	SrcIP: 192.168.183.134	Proc:	TgtSvr: dev
----------------	-----------------	------------------------	-------	-------------

El puerto donde se conecto es el puerto SMB (Server Message Block) que es un protocolo de red que facilita el intercambio de recursos y servicios entre computadoras en una red local o en internet

IpPort: 445
-------------

IP ATACANTE: 192.168.183.134

PUERTO DE CONEXIÓN: 445

# CONTRASEÑAS DEBILES

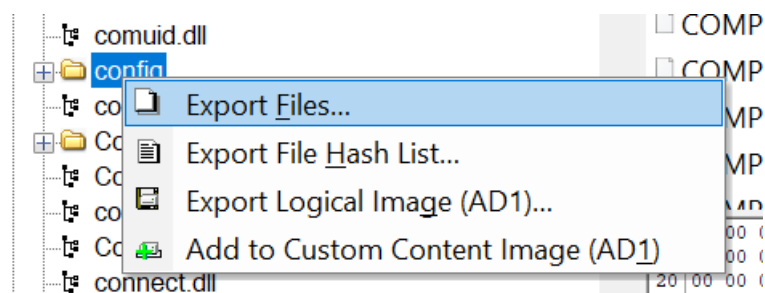
*Para obtener la contraseña emplearemos Mimikatz que es una herramienta para recuperar credenciales almacenadas en sistemas Windows utilizando técnicas de inyección en memoria para extraer información directamente de la memoria del sistema, lo que le permite evadir algunas medidas de seguridad.*

Aplicaremos su función de dumpeo en los archivos SAM y SYSTEM.

SAM contiene información sobre cuentas de usuario y sus contraseñas en forma de Hashes.

SYSTEM contiene información relacionada con la configuración del sistema de seguridad, incluidos los hashes y contraseñas de los usuarios

El primer paso es extraer de FTK los archivos SAM y SYSTEM encontrados en la dirección C:/Windows/system32/Config



Abrimos un cmd en la carpeta X64 donde se encuentra alojado Mimikatz y dumpeamos el fichero SAM con SYSTEM para recopilar sus contraseñas guardadas en memoria.

*NOTA EXTRA -> llevaremos los ficheros SAM y SYSTEM a la carpeta C: para facilitar el camino*

Para ejecutar Mimikatz debemos usar el comando :

Mimikatz.exe

Una vez dentro estamos listos para dumpear con el comando:

```
lsadump::sam /SYSTEM:C:\SYSTEM /sam:C:\SAM
```

```

mimikatz 2.2.0 x64 (oe.eo)

OldCredentials
  des_cbc_md5      : a4ce3d75831f988c

RID : 000003ea (1002)
User : sshd
Hash NTLM: 42760776cade85fd98103a0f44437800

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
  Random Value : 59027b35c620e96f83d319ebd31577be

* Primary:Kerberos-Newer-Keys *
  Default Salt : MSEDGEWIN10sshd
  Default Iterations : 4096
  Credentials
    aes256_hmac      (4096) : 9c6818e8b29d2a66b5b66321b95faedfd793908ae666cc254aacaae8d9cdd0c3
    aes128_hmac      (4096) : 8e4a19ecfa0cfff16aadf1491aa848d3
    des_cbc_md5      (4096) : 64d51f51efad018a

* Packages *
  NTLM-Strong-NTOWF

* Primary:Kerberos *
  Default Salt : MSEDGEWIN10sshd
  Credentials
    des_cbc_md5      : 64d51f51efad018a

mimikatz #
```

El output que nos importa es el hash de la contraseña de usuario que nos entrega en el apartado Hash NTLM .

```

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

Este Hash podemos descifrarlo en <https://crackstation.net/>

2d20d252a479f485cdf5e171d93985bf

No soy un robot

reCAPTCHA

Privacidad - Términos

Crack Hashes

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin), QubesV3.1BackupDefaults

Hash	Type	Result
2d20d252a479f485cdf5e171d93985bf	NTLM	qwerty

Hemos conseguido la contraseña del usuario IEUser

CONTRASEÑA DEBIL: **qwerty**



---

# MEMORIA RAM

---

Para este apartado de la práctica, debéis 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.

Haremos la adquisición de una memoria ram en un sistema operativo Windows 10, utilizando las herramientas:

- WinPmem
- Volatility
- Python
- Bloc de Notas

## EXTRACCION DE MEMORIA RAM

DESCARGAMOS <https://github.com/Velocidex/WinPmem/releases>

Abrimos cmd con permisos de administrador donde esta el descargado y utilizamos el siguiente comando:

```
winpmem_mini_x64_rc2.exe windows_ram.mem
```

```
C:\Users\forensic\Desktop\RAM>winpmem_mini_x64_rc2.exe windows_ram.mem
WinPmem64
Extracting driver to C:\Users\forensic\AppData\Local\Temp\pme6014.tmp
Driver Unloaded.
Loaded Driver C:\Users\forensic\AppData\Local\Temp\pme6014.tmp.
Deleting C:\Users\forensic\AppData\Local\Temp\pme6014.tmp
The system time is: 01:54:27
Will generate a RAW image
- buffer_size_: 0x1000
CR3: 0x00001AA000
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
pad
- length: 0x1000

00% 0x00000000 .
copy_memory
- start: 0x1000
- end: 0x9f000

00% 0x00001000 .
Padding from 0x0009F000 to 0x00100000
pad
```

WinPmem Comenzara a extraer nuestra adquisicion de memoria Ram en el Windows 10

## PREPARACION DEL ENTORNO DE ANALISIS

Descargaremos la herramienta Volatility que posteriormente utilizaremos para analizar la extracción de memoria que adquirimos y extraemos su contenido

<https://www.volatilityfoundation.org/releases-vol3>

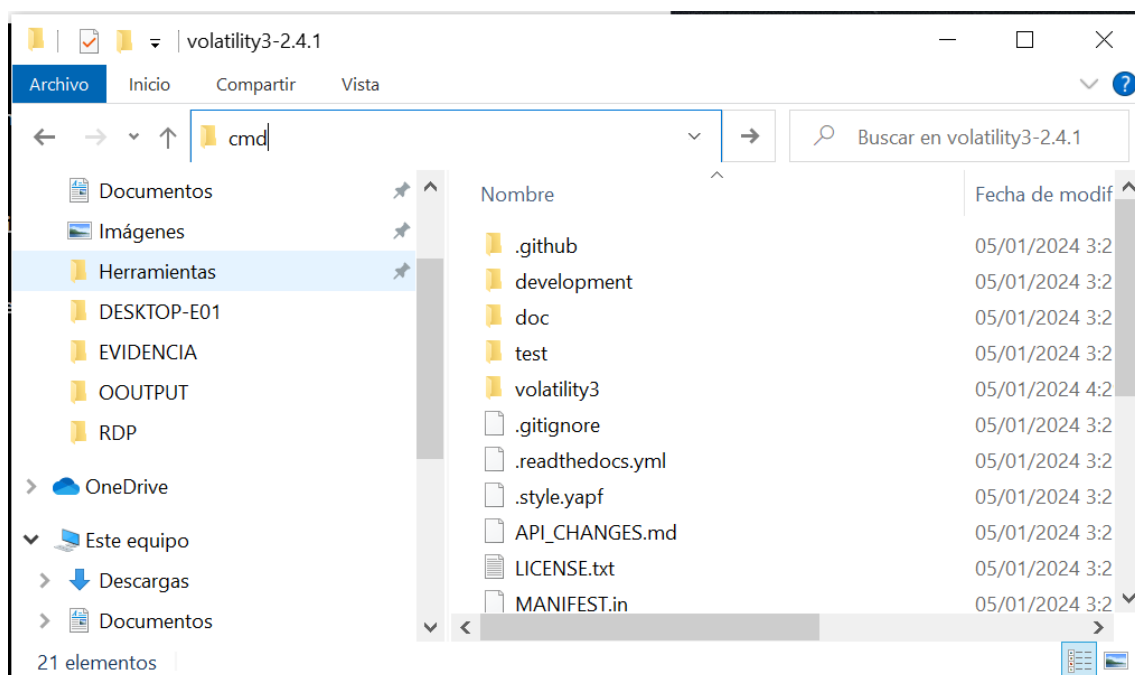
Volatility es una herramienta de código abierto utilizada en informática forense y seguridad informática para analizar la memoria volátil de sistemas informáticos. Permite extraer y examinar

información clave de la memoria RAM, como procesos en ejecución, conexiones de red, registros de eventos y más. Su funcionalidad incluye el análisis de procesos, identificación de malware, revisión de conexiones de red y la capacidad de buscar indicadores de compromiso en la memoria.

Esta escrita en el lenguaje de programación Python, por lo que debemos instalar Python para utilizarla. Podemos obtenerlo en el siguiente enlace

<https://www.python.org/downloads/release/python-3121/>

Para hacer un análisis de la memoria ram con volatility debemos abrir un cmd (símbolo de sistema) dentro de la carpeta extraída de volatility



Luego con el siguiente comando podemos ver la lista de análisis que volatility puede ejecutar en la memoria extraída

```
python vol.py -f
```

# ANALISIS DE MEMORIA

Vamos a hacer dos análisis de la memoria extraída, uno sobre los procesos y otro de para ver las bibliotecas dinámicas enlazadas a cada proceso.

## PSLIST

Este comando se utiliza para listar los procesos en ejecución en la memoria del sistema. Proporciona información detallada sobre cada proceso, como el nombre del proceso, el identificador de proceso (PID), el tiempo de creación, el tiempo de usuario y el tiempo del kernel. Analizar la salida de pslist es útil para identificar procesos maliciosos, sospechosos o inusuales en el sistema.

### COMANDO

```
python vol.py -f C:\Users\forensic\Desktop\RAM\windows_ram.mem windows.pslist
```

Microsoft Windows [Versión 10.0.19044.3886]  
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\forensic\Desktop\EVIDENCIA\Volatility\volatility3-2.4.1>python vol.py -f C:\Users\forensic\Desktop\RAM\windows\_ram.mem windows.pslist

Volatility 3 Framework 2.4.1

Progress: 100.00 PDB scanning finished

PID	PPID	ImageFileName	Offset(V)	Threads	Handles	SessionId	Wow64	CreateTime	ExitTime
File output									
4	0	System	0xc00a97094880	126	-	N/A	False	2024-01-05 19:15:49.000000	N/A
92	4	Registry	0xc00a971b8040	4	-	N/A	False	2024-01-05 19:15:36.000000	N/A
-	Disabled								
396	4	smss.exe	0xc00a97cc0040	2	-	N/A	False	2024-01-05 19:15:49.000000	N/A
-	Disabled								
500	488	csrss.exe	0xc00a97d86140	10	-	0	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
568	488	wininit.exe	0xc00a9e4f3080	1	-	0	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
584	560	csrss.exe	0xc00a9e4fa140	12	-	1	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
668	560	winlogon.exe	0xc00a9e524080	5	-	1	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
704	568	services.exe	0xc00a9dd30180	7	-	0	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
712	568	lsass.exe	0xc00a9e529080	9	-	0	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
816	668	fontdrvhost.exe	0xc00a9e5ce240	5	-	1	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
832	568	fontdrvhost.exe	0xc00a9e5cf080	5	-	0	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
840	704	svchost.exe	0xc00a9e5d1340	18	-	0	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
956	704	svchost.exe	0xc00a9ecc9240	12	-	0	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								
1004	704	svchost.exe	0xc00a9ed10340	5	-	0	False	2024-01-05 19:15:59.000000	N/A
-	Disabled								

# DLLLIST

Este comando se utiliza para listar las DLL (Dynamic Link Libraries) cargadas en el espacio de memoria de cada proceso. Proporciona información sobre las bibliotecas dinámicas enlazadas a cada proceso, lo que puede ser crucial para identificar componentes maliciosos o detectar actividades sospechosas relacionadas con cargas dinámicas.

## COMANDO

```
python vol.py -f C:\Users\forensic\Desktop\RAM\windows_ram.mem dlllist
```

Microsoft Windows [Versión 10.0.19044.3886]  
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\forensic\Desktop\EVIDENCIA\Volatility\volatility3-2.4.1>python vol.py -f C:\Users\forensic\Desktop\RAM\windows\_ram.mem dlllist

Volatility 3 Framework 2.4.1

Progress: 100.00 PDB scanning finished

PID	Process	Base	Size	Name	Path	LoadTime	File	output
396	smss.exe	0x7FF7CDD50000	0x28000	smss.exe	\SystemRoot\System32\smss.exe	2024-01-05 19:15:49.000000	Disabled	
396	smss.exe	0x7FF7A56D0000	0x1F8000	ntdll.dll	C:\WINDOWS\SYSTEM32\ntdll.dll	2024-01-05 19:15:49.000000	Disabled	
580	csrss.exe	0x7FF6A8A40000	0x70000	csrss.exe	C:\WINDOWS\system32\csrss.exe	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A56D0000	0x1F8000	ntdll.dll	C:\WINDOWS\SYSTEM32\ntdll.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A2D90000	0x18000	CSRSRV.dll	C:\WINDOWS\SYSTEM32\CSRSRV.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A2D90000	0x16000	basesrv.DLL	C:\WINDOWS\system32\basesrv.DLL	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A2D50000	0x15000	winsrv.DLL	C:\WINDOWS\system32\winsrv.DLL	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A31D0000	0x2F6000	kernelbase.dll	C:\WINDOWS\SYSTEM32\kernelbase.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A45D0000	0xBF000	kernel32.dll	C:\WINDOWS\SYSTEM32\kernel32.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A2D20000	0x23000	winsrvext.dll	C:\WINDOWS\SYSTEM32\winsrvext.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A3F70000	0x19D000	USER32.dll	C:\WINDOWS\system32\USER32.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A4960000	0x354000	combase.dll	C:\WINDOWS\SYSTEM32\combase.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A2F00000	0x4E000	cfgmgr32.dll	C:\WINDOWS\SYSTEM32\cfgmgr32.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A4920000	0x2C000	GDI32.dll	C:\WINDOWS\system32\GDI32.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A3080000	0x22000	win32u.dll	C:\WINDOWS\system32\win32u.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A34D0000	0x100000	ucrtbase.dll	C:\WINDOWS\SYSTEM32\ucrtbase.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A4C00000	0x126000	RPCRT4.dll	C:\WINDOWS\SYSTEM32\RPCRT4.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A30B0000	0x115000	gdi32full.dll	C:\WINDOWS\SYSTEM32\gdi32full.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A3690000	0x9D000	msvcp_win.dll	C:\WINDOWS\system32\msvcp_win.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A2D10000	0xD0000	sxsrv.DLL	C:\WINDOWS\system32\sxsrv.DLL	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A2B50000	0xA2000	sxs.dll	C:\WINDOWS\system32\sxs.dll	2024-01-05 19:15:59.000000	Disabled	
580	csrss.exe	0x7FF7A35D0000	0xA2000	bcryptPrimitives.dll	C:\WINDOWS\SYSTEM32\bcryptPrimitives.dll	2024-01-05 19:15:59.000000	Disabled	
584	csrss.exe	0x7FF6A8A40000	0x70000	csrss.exe	C:\WINDOWS\system32\csrss.exe	2024-01-05 19:15:59.000000	Disabled	
584	csrss.exe	0x7FF7A56D0000	0x1F8000	ntdll.dll	C:\WINDOWS\SYSTEM32\ntdll.dll	2024-01-05 19:15:59.000000	Disabled	
584	csrss.exe	0x7FF7A2D90000	0x18000	CSRSRV.dll	C:\WINDOWS\SYSTEM32\CSRSRV.dll	2024-01-05 19:15:59.000000	Disabled	
584	csrss.exe	0x7FF7A2D90000	0x16000	basesrv.DLL	C:\WINDOWS\system32\basesrv.DLL	2024-01-05 19:15:59.000000	Disabled	
584	csrss.exe	0x7FF7A2D50000	0x15000	winsrv.DLL	C:\WINDOWS\system32\winsrv.DLL	2024-01-05 19:15:59.000000	Disabled	
584	csrss.exe	0x7FF7A31D0000	0x2F6000	-	-	2024-01-05 19:15:59.000000	Disabled	