

Informe Módulo 7

Digital Forensics Incident Response

Fecha: 03 de Mayo de 2024

Autor: **Azael Ramírez Pérez**

Mail: **keepcoder_test@gmail.com (ficticio)**

Empresa: **KeepCoder.inc (ficticio)**

Contenido

Ámbito y alcance.....	3
Challenges.....	5
Práctica memoria Ram.....	24
Práctica Metadatos.....	31

Ámbito y alcance

El presente trabajo está enfocado en realizar un análisis forense a una maquina Windows, para poder realizar las pruebas se utilizarán diferentes herramientas que nos permitirán recuperar y analizar datos, así mismo se documentara el proceso abordado para que otro analista pueda validar los hallazgos que se han encontrado.

Es importante mencionar que para realizar este tipo de análisis forense es necesario apoyarse de una metodología o serie de pasos a realizar para así hacer trabajo con mayor grado de profesionalismo, a continuación, se describen algunos pasos de esta metodoligia:

- **Recepción de evidencia:** Se recibe el dispositivo digital de manera legal y cadena de custodia.
- **Documentación:** Se documenta la cadena de custodia, características del dispositivo y estado inicial.
- **Acceso seguro:** Se trabaja en un entorno aislado y seguro para evitar alterar la evidencia original.
- **Adquisición de imágenes:** Se realiza una copia bit a bit (imagen forense) del dispositivo para preservar la evidencia original.
- **Análisis de la imagen:** Se examina la imagen forense con herramientas para recuperar archivos eliminados, historial de navegación, archivos temporales, etc.

- **Reporte de hallazgos:** Se documentan los hallazgos relevantes como archivos, actividad, metadatos, en un reporte pericial.
- **Presentación de evidencia:** El analista forense presenta de manera clara y precisa los hallazgos como evidencia admisible en un proceso legal.
- **Retención de evidencia:** Se mantiene la cadena de custodia y se resguarda de manera segura la evidencia digital.

Challenges

A continuación, se presenta el procedimiento que se ha seguido para poder resolver los desafíos siguientes.

Hash del fichero.

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

Procedimiento.

Para poder obtener el hash, se ha realizado lo siguiente:

1. Se ha descargado la evidencia (**Win10_PC001.vmdk**) del sitio <http://ctf.sancastell.me>.
2. En nuestro host anfitrión Windows, se ha usado el comando mostrado en la imagen 1, el cual nos permite obtener el hash de la máquina.
3. Comando ejecutado:

```
Get-FileHash Win10_PC001.vmdk -Algorithm SHA256
```

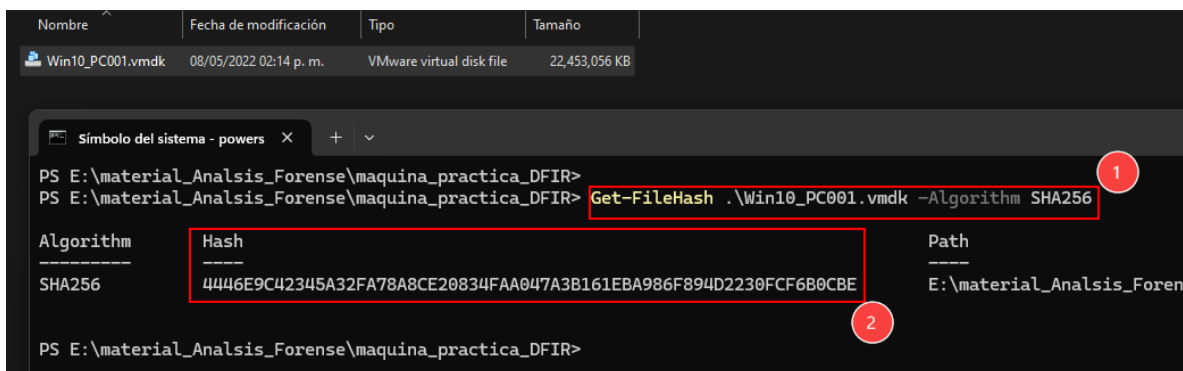


Imagen 1. Obtenemos el hash (sha256) de la maquina

Nombre de la máquina.

Indiquen el nombre de la máquina de la que se está realizando el análisis.

Procedimiento.

Para poder obtener el nombre de la máquina, se ha realizado lo siguiente:

1. Se ha usado la herramienta de **Acces Data FTK Imager** para importar la evidencia (Win10_PC001.vmdk) para así poder acceder a los registros de Windows y Logs del sistema, finalmente se ha exportado el archivo de SYSTEM, ver el punto 2 de la imagen siguiente para identificar la ruta del archivo que se extrajo.

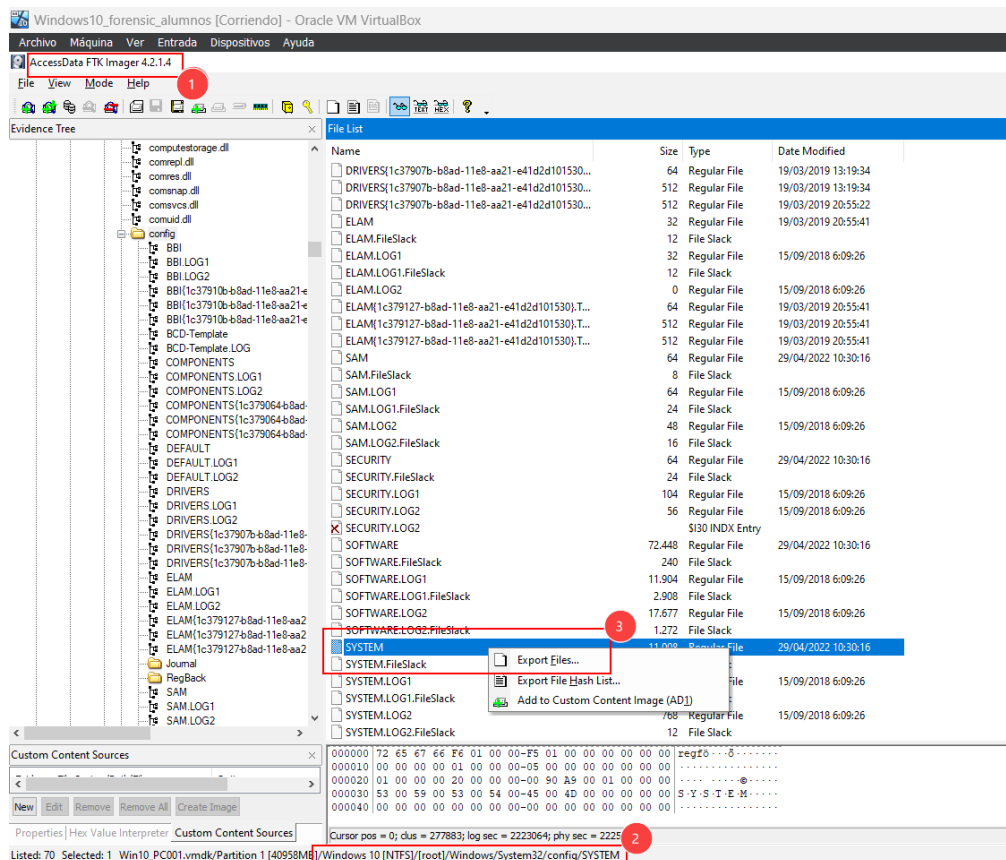


Imagen 2. Archivo **SYSTEM** exportado desde **Acces Data FTK Imager**

2. Se ha utilizado la herramienta de **AccessData Registry Viewer** para poder importar el archivo **SYSTEM** y así poder extraer el **hostname** de la evidencia (Win10_PC001.vmdk), en el punto 3 de la siguiente imagen es la ruta donde se encuentra el registro que contiene el ComputerName.

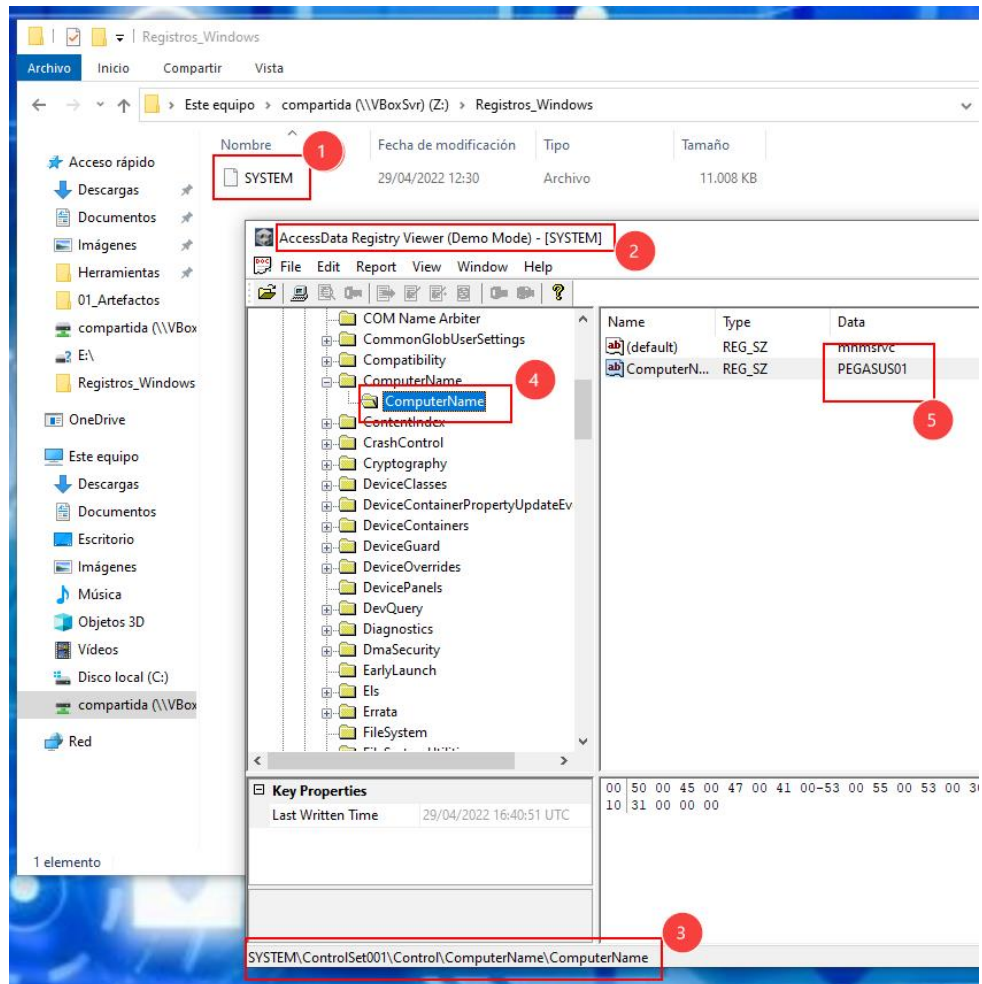
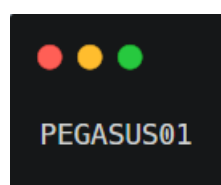


Imagen 3. Acceso al registro que contiene el ComputerName

3. El hostname de la maquina es:



Descarga fichero de control remoto.

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

Procedimiento.

Para poder encontrar el archivo que permite tener control remoto del equipo, se ha realizado el siguiente procedimiento:

1. Se ha utilizado la herramienta **Acces Data FTK Imager**, se ha importado la evidencia (Win10_PC001.vmdk).
2. Se ha localizado el directorio donde se ubican los usuarios y hemos encontrado el usuario estándar **IEUser**.
3. Se ha identificado la carpeta **Downloads** donde se ubican los archivos que este usuario ha descargado de la red, ver punto 5 de la siguiente imagen.

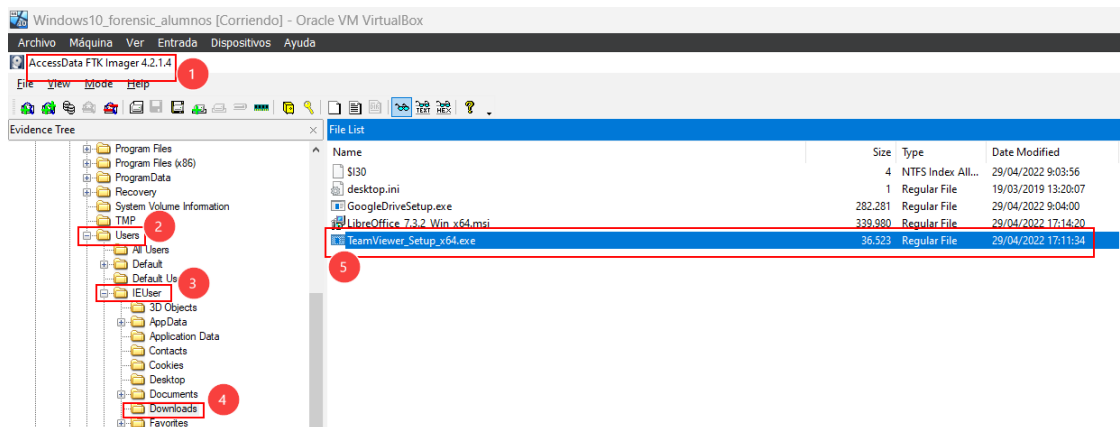
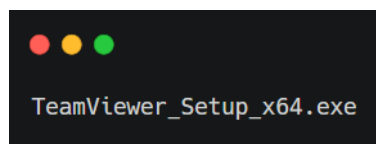


Imagen 4. Archivos descargados por el usuario

4. Finalmente se ha encontrado que ha descargado el archivo:



Ficheros eliminados.

Se sospecha que existe un fichero .zip eliminado.

Podría indicar el nombre.

Procedimiento.

Para poder encontrar los archivos eliminados de la evidencia, se ha realizado el siguiente procedimiento:

1. Inicialmente se ha extraído el archivo **\$MFT** de la evidencia (Win10_PC001.vmdk) para ello se utilizó **AccessData FTK Imager** (ver punto 4).

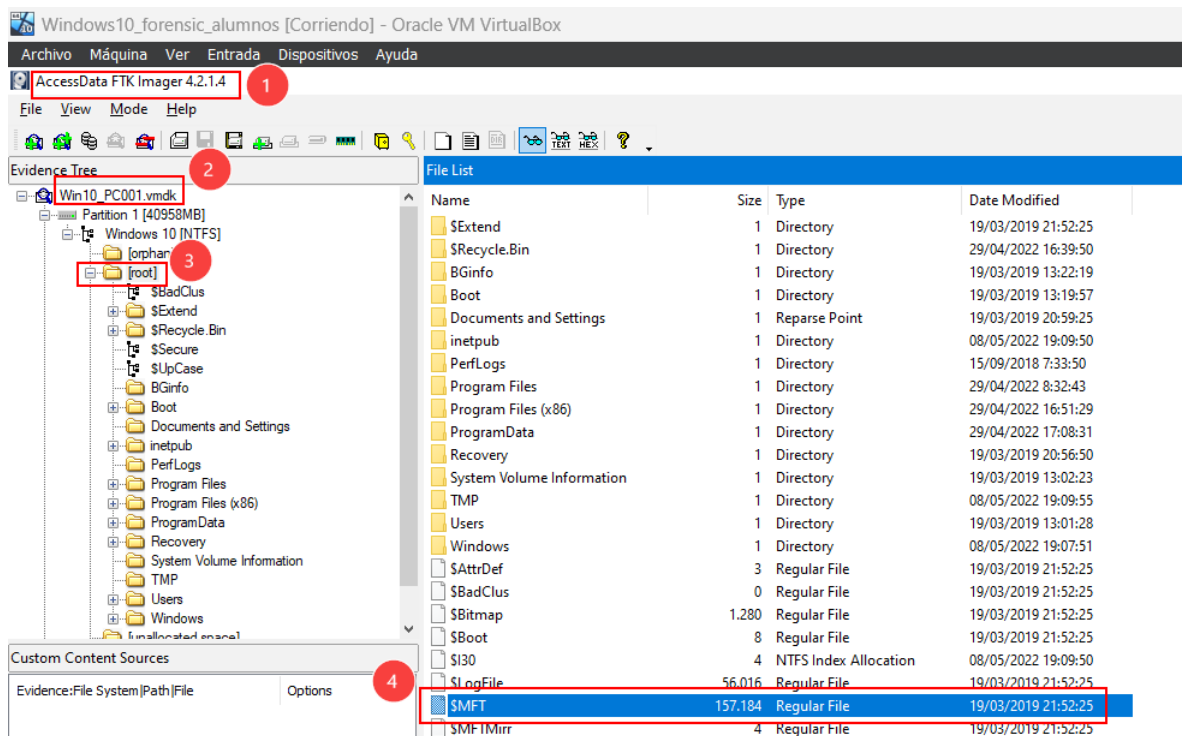


Imagen 5. Se exporta el archivo \$MFT

2. Después se usó la herramienta MFT Explorer para poder importar el archivo \$MFT, una vez importado ubicamos el directorio **\$Recycle.Bin**, después exploramos los subdirectorios que contiene y se ha logrado identificar distintos tipos de archivos que el usuario IEUser ha eliminado (.pdf, .zip y .doc), en el punto 6 se logra percibir el archivo .zip eliminado (cosas.zip).

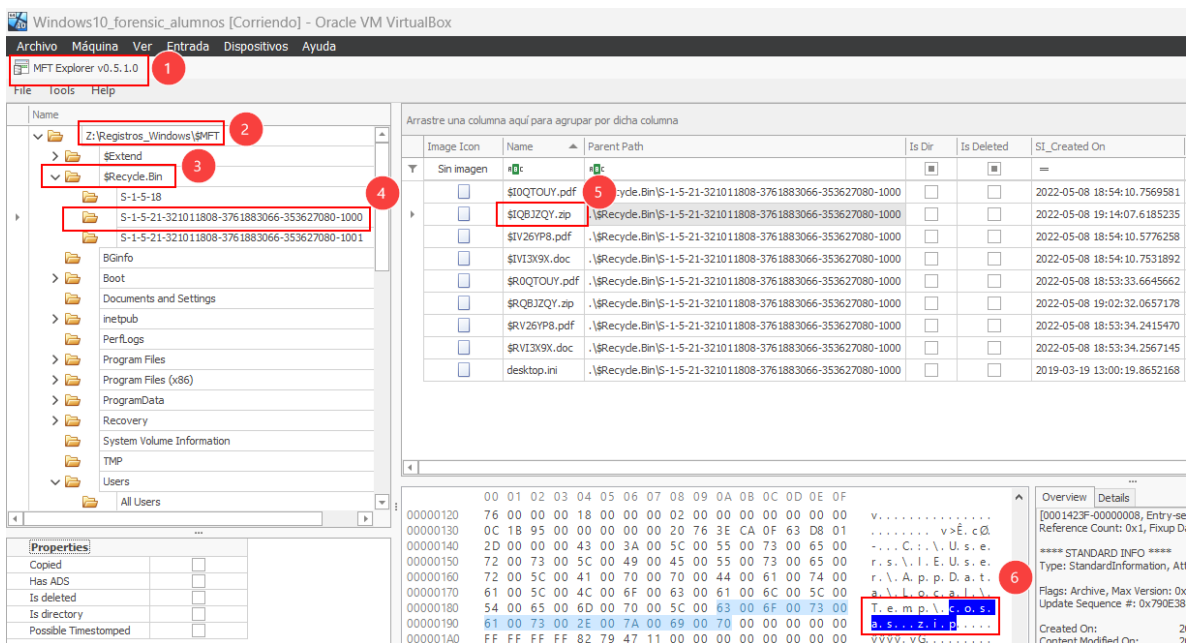


Imagen 6. Identificación de archivos eliminados a través del archivo \$MFT

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.

Procedimiento.

Para poder encontrar la fecha de ejecución del programa de control remoto, se ha realizado el siguiente procedimiento:

1. Inicialmente se ha extraído el archivo **\$MFT** de la evidencia (Win10_PC001.vmdk) para ello se utilizó **AccessData FTK Imager** (ver punto 4).

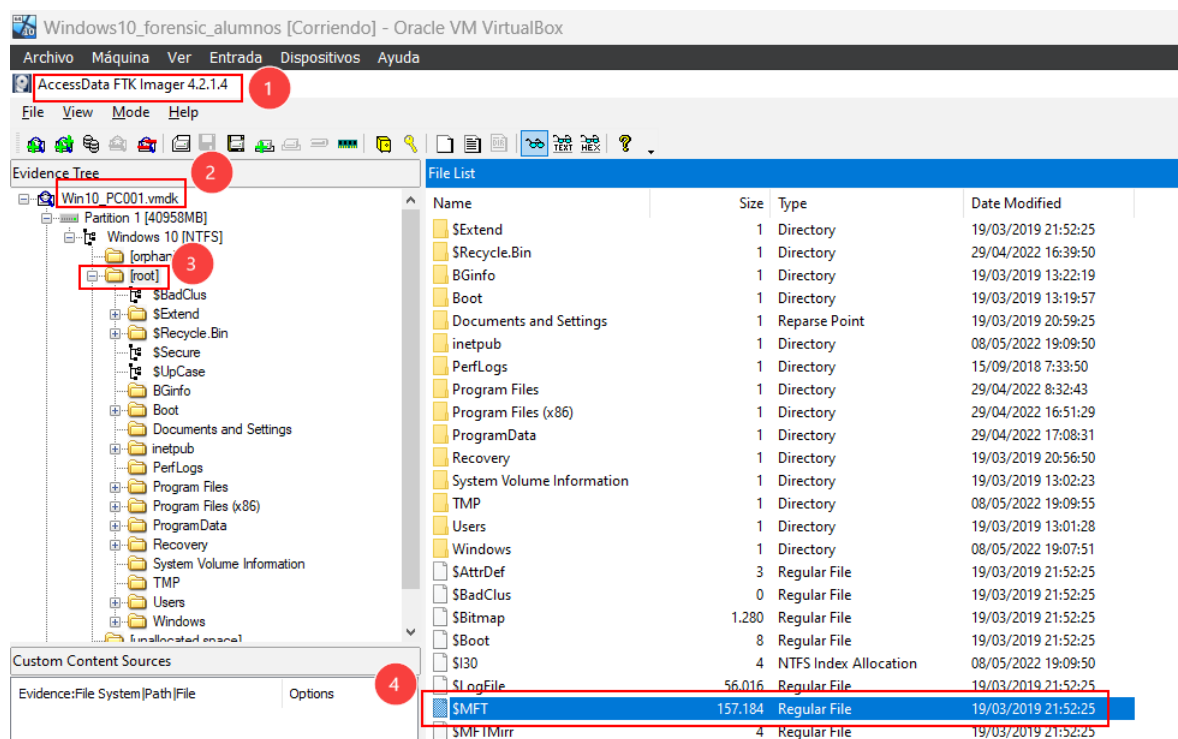


Imagen 7. Se exporta el archivo \$MFT

2. Seguidamente se utilizó la herramienta **MFT Explorer** para poder importar el archivo y de esta manera ubicar el archivo **TeamViewer_Setup_x64.exe** en la carpeta de Downloads, adicionalmente esta herramienta nos muestra en una tabla los detalles de STANDARD_INFORMATION y FILE_NAME, finalmente se ha encontrado el campo **Last Accessed ON** el cual nos indica el ultimo acceso del usuario con el archivo (ver punto 7).

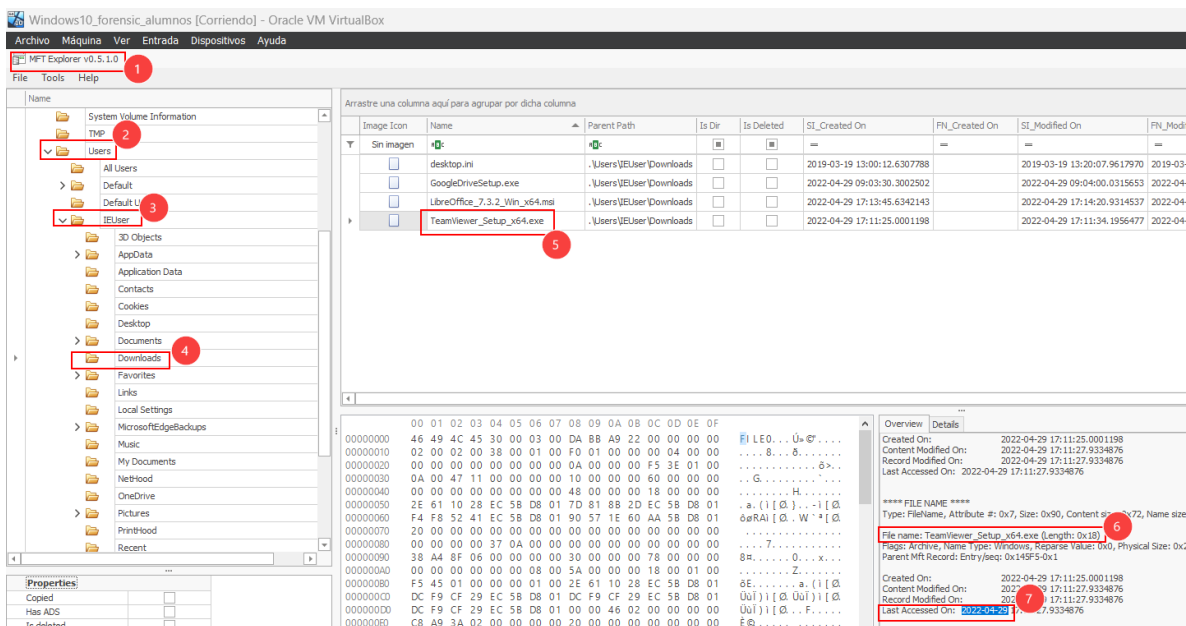


Imagen 8. Ultimo acceso al archivo TeamViewer_Setup_x64.exe

Contraseñas débiles.

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

Procedimiento.

Para poder encontrar la contraseña del usuario IEUser, se ha realizado el siguiente procedimiento:

1. Se ha utilizado la herramienta de AccessData FTK Imager para poder extraer los 2 archivos que nos ayudaran a obtener el hash del password del usuario.
 - /[root]/Windows/System32/config/SYSTEM
 - /[root]/Windows/System32/config/SAM

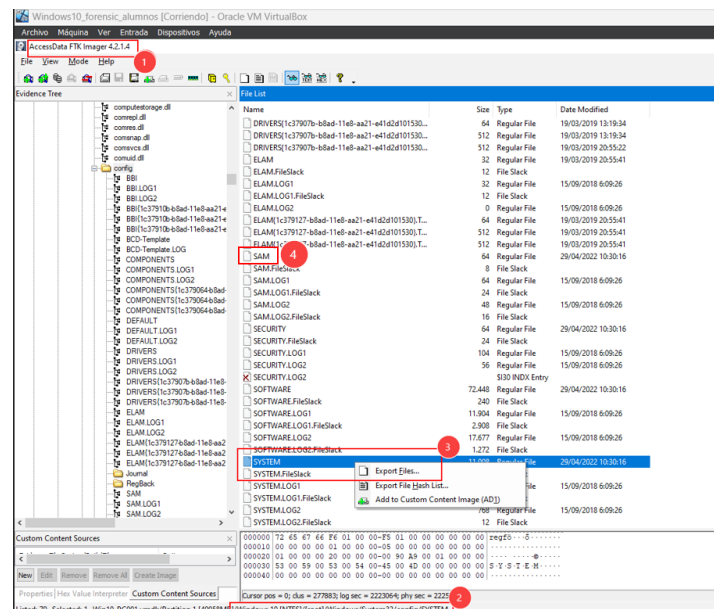


Imagen 9. Se exportan los archivos SYSTEM Y SAM de la evidencia para lograr obtener el hash del password del usuario.

2. Después se usó la herramienta **mimikatz.exe** y se ejecutó un comando (ver figura 2) para poder obtener los hash de todos los usuarios de la evidencia, a nosotros solo nos interesa obtener el hash del usuario IEUser (ver punto 4).

```
Z:\>cd Z:\mimikatz-master\mimikatz-master\Win32
Z:\mimikatz-master\mimikatz-master\Win32>
Z:\mimikatz-master\mimikatz-master\Win32>mimikatz.exe

.#####.  mimikatz 2.2.0 (x86) #18362 Feb 29 2020 11:13:10
.## ^ ##.  "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   ***/

mimikatz # lsadump::sam /system:Z:\Registros_Windows\SYSTEM /sam:Z:\Registros_Windows\SAM
Domain : PEGASUS01
SysKey : ec022a77f903a7e69e603e0c84634ff0
Local SID : S-1-5-21-321011808-3761883066-353627080

SAMKey : 939177c671faafb0fd1d1f10bc6de1190

RID : 000001f4 (500)
User : Administrator
Hash NTLM: fc525c9683e8fe067095ba2ddc971889

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

Supplemental Credentials:
* Primary:NTLM-Strong-NTOWF *
  Random Value : c6a807d33d3772144ce3407a8a73f9ef

* Primary:Kerberos-Newer-Keys *
  Default Salt : MSEDGEWIN10IEUser
  Default Iterations : 4096
  Credentials
    aes256_hmac (4096) : 72cc752f2addce7556960ad819259738c4fd86e7130cee6b06aca1137
    aes128_hmac (4096) : 7d83280d0766f4ad6510460fbd975fbc
    des_cbc_md5 (4096) : ecd9340ddff7406b
  OldCredentials
    aes256_hmac (4096) : b55700a5a2002a8a290a8f3554838fd420bcb7877b8f59ed75fd7af6b
    aes128_hmac (4096) : 64be48ded076d1592ae6df8708266f64
    des_cbc_md5 (4096) : a4ce3d75831f988c

* Packages *
  NTLM-Strong-NTOWF

* Primary:Kerberos *
  Default Salt : MSEDGEWIN10IEUser
  Credentials
    des_cbc_md5 : ecd9340ddff7406b
  OldCredentials
    des_cbc_md5 : a4ce3d75831f988c
```

Imagen 10. Ejecución de Mimikatz para obtener el hash del password

3. Una vez que obtuvimos el hash, se utilizó la herramienta **CrackStation** de la red para poder romper el hash y de esta manera lograr encontrar la contraseña en claro (ver punto 1).

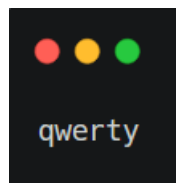


The screenshot shows the CrackStation website interface. At the top, there's a navigation bar with the CrackStation logo and links to Password Hashing Security and Defuse Security. Below this, the main heading is "Free Password Hash Cracker". A text input field contains the hash "2d20d252a479f485cdf5e171d93985bf". To the right of the input field is a reCAPTCHA widget with the text "No soy un robot" and a "Crack Hashes" button. Below the input field, a list of supported hash types is shown. At the bottom, a table displays the cracking results.

Hash	Type	Result
2d20d252a479f485cdf5e171d93985bf	NTLM	qwerty

Imagen 11. Cracking del Hash

4. La contraseña es:



Ficheros maliciosos.

En la máquina se han encontrado varios ficheros maliciosos.

En que carpeta (solamente el nombre de la carpeta) se encuentran dichos ficheros?

Procedimiento.

Para poder encontrar los archivos maliciosos, se ha realizado el siguiente procedimiento:

1. Se ha utilizado la herramienta **AccessData FTK Imager**, después se ha importado la evidencia (Win10_PC001.vmdk), después se ha revisado el directorio **TMP** y se ha observado que hay varios ficheros ejecutables (p.exe y WMIBackdoor.ps1), ver punto 5.

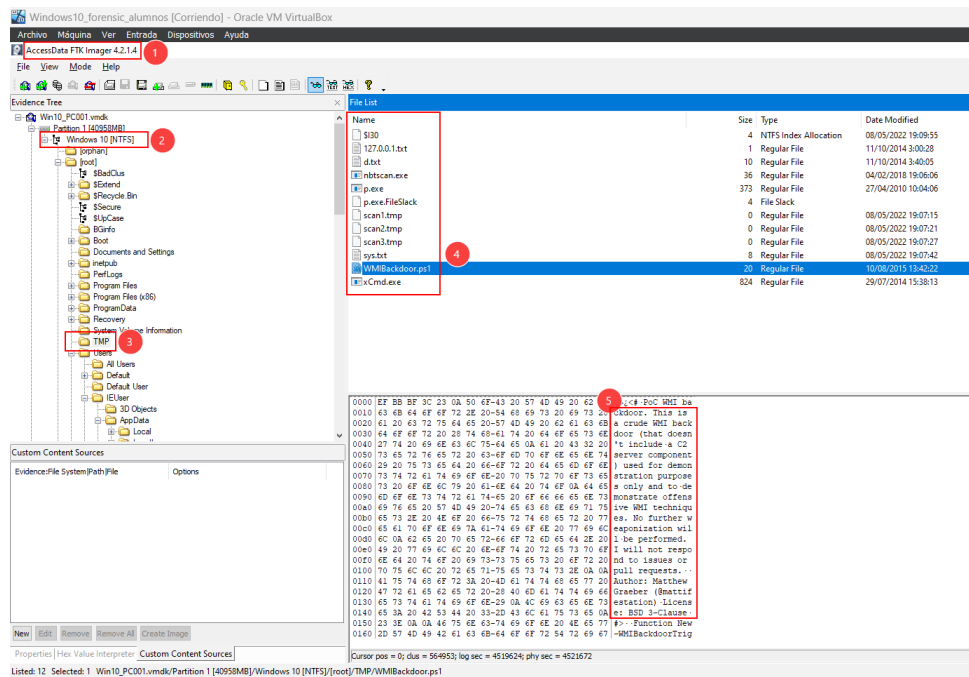


Imagen 12. Ficheros maliciosos

2. Se ha exportado y se ha analizado el directorio **Prefetch** de la evidencia, a simple vista nos percatamos que hay algunos ejecutables que podría ser maliciosos, para poder validar con mayor detalle, se ha utilizado también la herramienta **PeCMD.exe**.

Nombre	Fecha de modificación	Tipo
MUSNOTIFYICON.EXE-A201B346.pf	08/05/2022 20:58	Archivo PF
NBTSCAN.EXE-9C28C0E8.pf	08/05/2022 21:07	Archivo PF
NET.EXE-1DF3A2F6.pf	08/05/2022 21:07	Archivo PF
NET1.EXE-B8A8247B.pf	08/05/2022 21:07	Archivo PF
NGEN.EXE-8DF18334.pf	29/04/2022 10:25	Archivo PF
NGEN.EXE-E9662EB6.pf	29/04/2022 10:25	Archivo PF
NGENTASK.EXE-90AAC3ED.pf	29/04/2022 10:25	Archivo PF
NGENTASK.EXE-F262E2AB.pf	29/04/2022 10:25	Archivo PF
NOTEPAD.EXE-EB1B961A.pf	08/05/2022 21:10	Archivo PF
NSLOOKUP.EXE-0E49F32A.pf	08/05/2022 21:06	Archivo PF
ONEDRIVE.EXE-33D53679.pf	08/05/2022 20:54	Archivo PF
ONEDRIVESETUP.EXE-07609C61.pf	08/05/2022 20:53	Archivo PF
OPENWITH.EXE-2DD6FAA1.pf	08/05/2022 21:10	Archivo PF
P.EXE-7A85E64B.pf	08/05/2022 21:07	Archivo PF
PING.EXE-B29F6629.pf	08/05/2022 21:06	Archivo PF
POQEXE~1.PF	22/04/2024 21:06	Archivo PF
POWERSHELL.EXE-59FC8F3D.pf	08/05/2022 21:09	Archivo PF
PSEXESVC.EXE-51BA46F2.pf	08/05/2022 21:07	Archivo PF
READERDC64_ES_XA_CRD_SEC_INST-F62ADC5F.pf	29/04/2022 19:07	Archivo PF
REG.EXE-26976709.pf	08/05/2022 21:09	Archivo PF
REGEDIT.EXE-4748FE01.pf	29/04/2022 12:06	Archivo PF
REGSVR32.EXE-55A4EE79.pf	29/04/2022 10:33	Archivo PF
RU40BF~1.PF	22/04/2024 21:06	Archivo PF
RU43CE~1.PF	22/04/2024 21:06	Archivo PF
RUBY.EXE-4684BBC3.pf	08/05/2022 21:01	Archivo PF

Imagen 13. Revisión de los archivos que contiene el directorio Prefetch

3. **PeCMD.exe** parsea el directorio completo Prefetch y nos arroja 2 archivos CSV los cuales se han interpretado con la herramienta **TimeLine Explorer** pues es más amigable observar los detalles, se han encontrado la ejecución de P.EXE (ver punto 2), de esta manera logramos encontrar el directorio que aloja los archivos maliciosos.

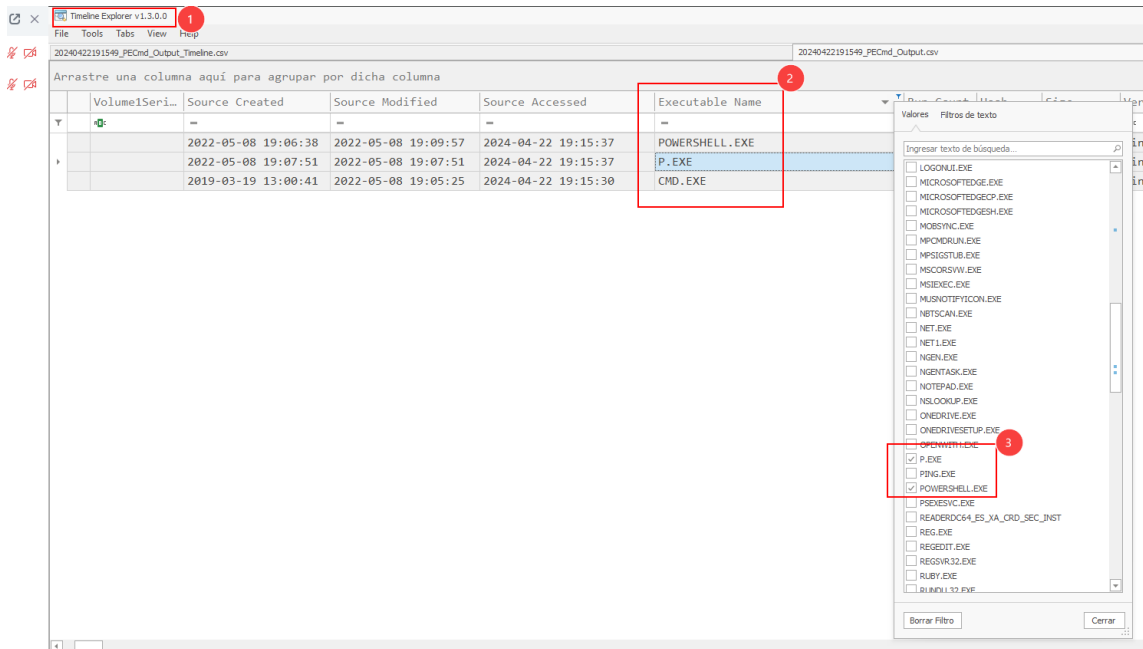
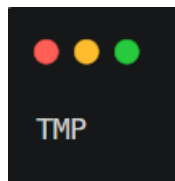


Imagen 14. Ejecutables maliciosos encontrados

4. El directorio que contiene los archivos maliciosos es:



Se ha detectado actividad sospechosa en la red, podrían indicar la IP desde la que se ha conectado a la máquina por RDP.

Procedimiento.

Para poder encontrar la IP desde la que se ha conectado, se ha realizado el siguiente procedimiento:

1. Inicialmente se ha importado la evidencia en **AccessData FTK Imager**, después se ha extraído el archivo **UsrClass.dat**

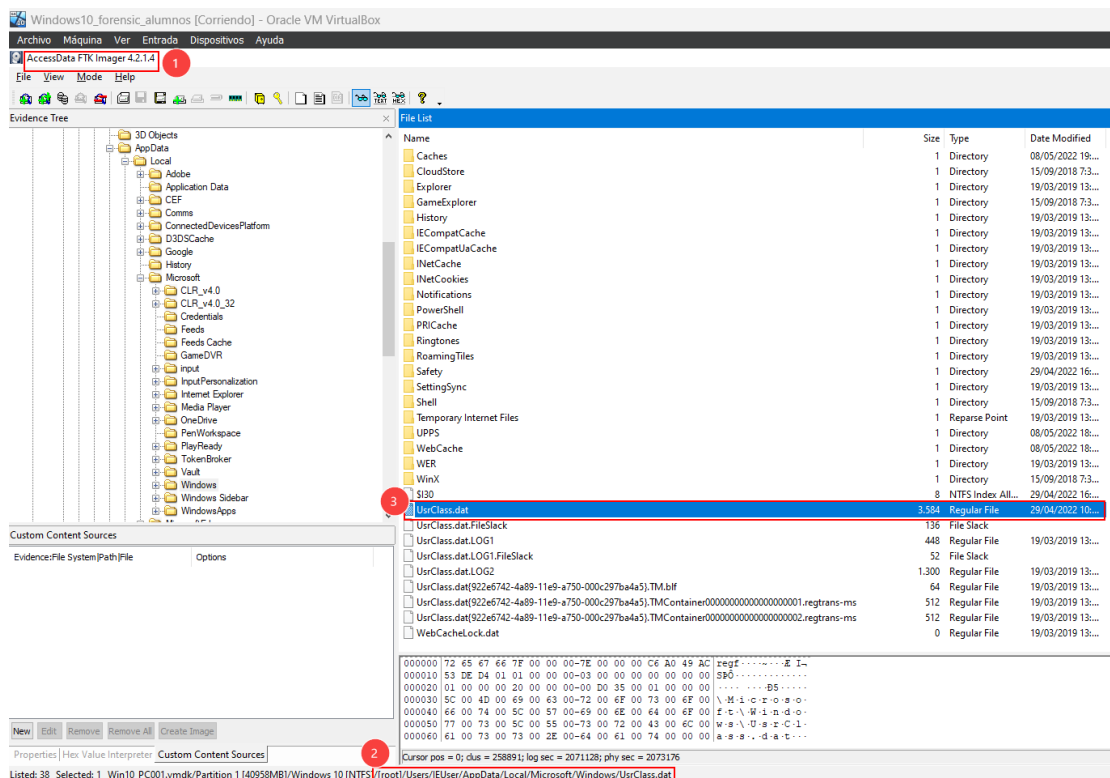


Imagen 15. Extracción archivo Usrclass.dat

2. Se ha analizado el archivo **UsrClass.dat** con la herramienta **RegRipper** y nos ha arrojado 2 archivos los cuales podemos interpretar y así mismo encontrar información de mucho interés.

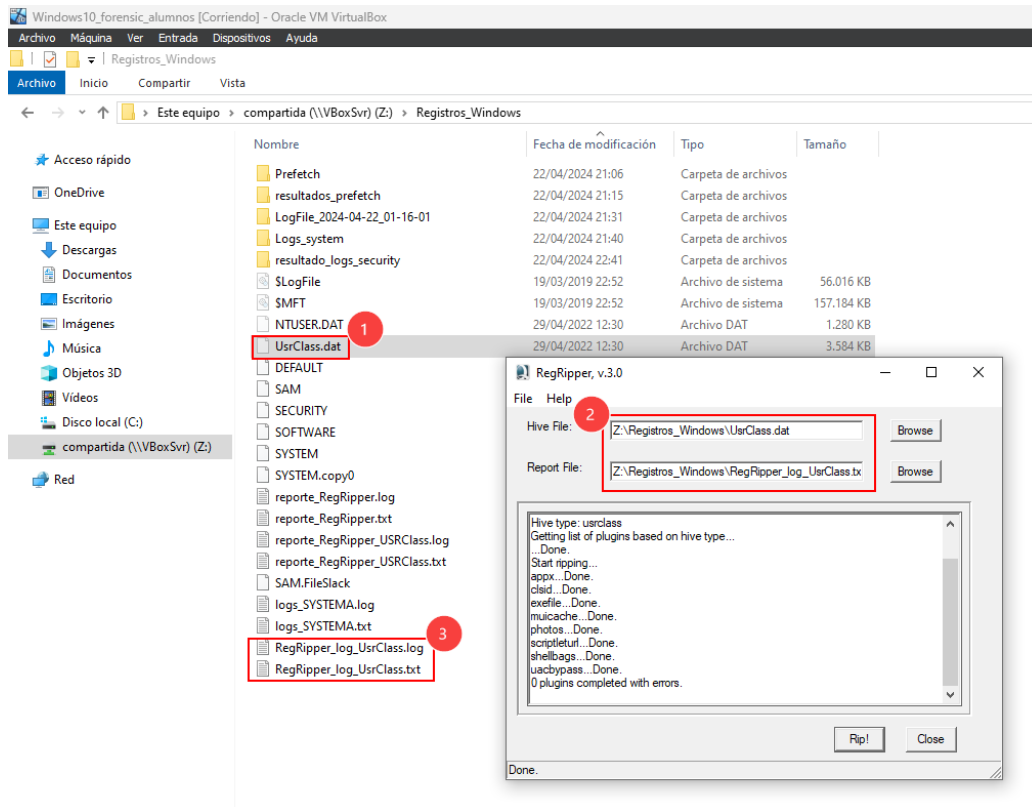


Imagen 16. Análisis del archivo UsrClass.dat con la herramienta RegRipper

3. Finalmente se ha importado el archivo **RegRipper_log_UsrClass.txt** en la herramienta de **Timeline Explorer**, esta misma ha permitido ver gran parte de la actividad que ha tenido el usuario en el equipo, de esta manera se ha identificado una conexión a través del protocolo RPD desde la IP **192.168.183.134**.

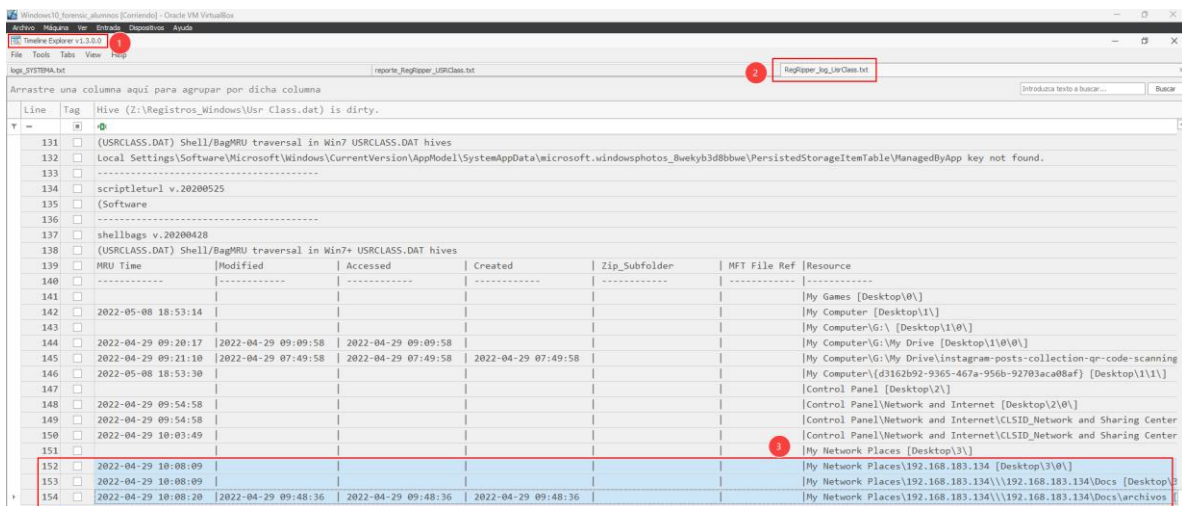
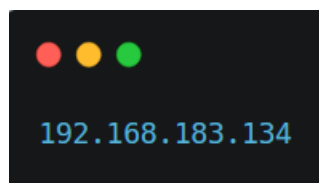


Imagen 17. Interpretación del archivo UsrClass.dat con la herramienta Timeline Explorer

4. IP de conexión remota:



5. Adicionalmente el puerto de conexión de la maquina atacante se encontró mediante los archivos de eventos del sistema, para ello primero se extrajeron los archivos.

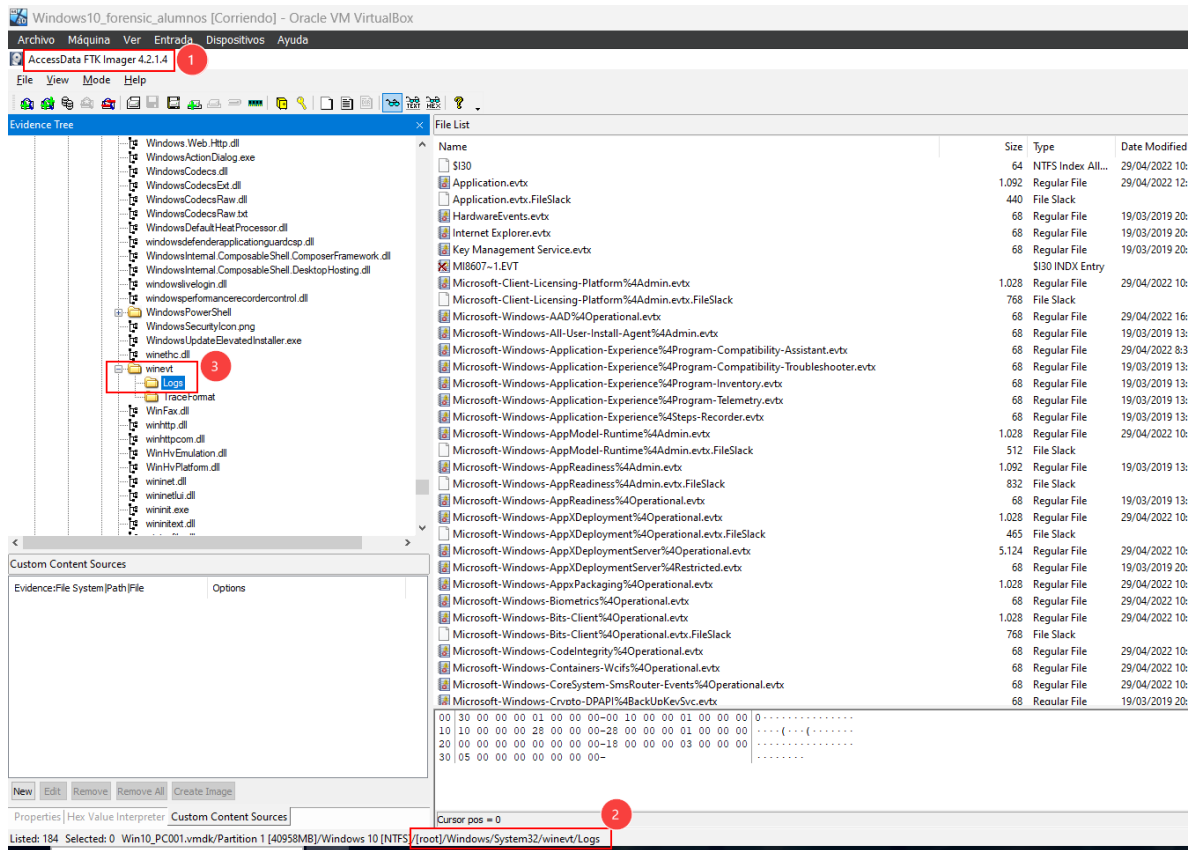


Imagen 18. Extracción de los archivos de eventos del sistema

6. Después usamos la herramienta **EvtCmd.exe** para poder parsear el archivo de **Security.evtx** y así poder obtener un archivo csv para poder interpretar de manera más amigable.

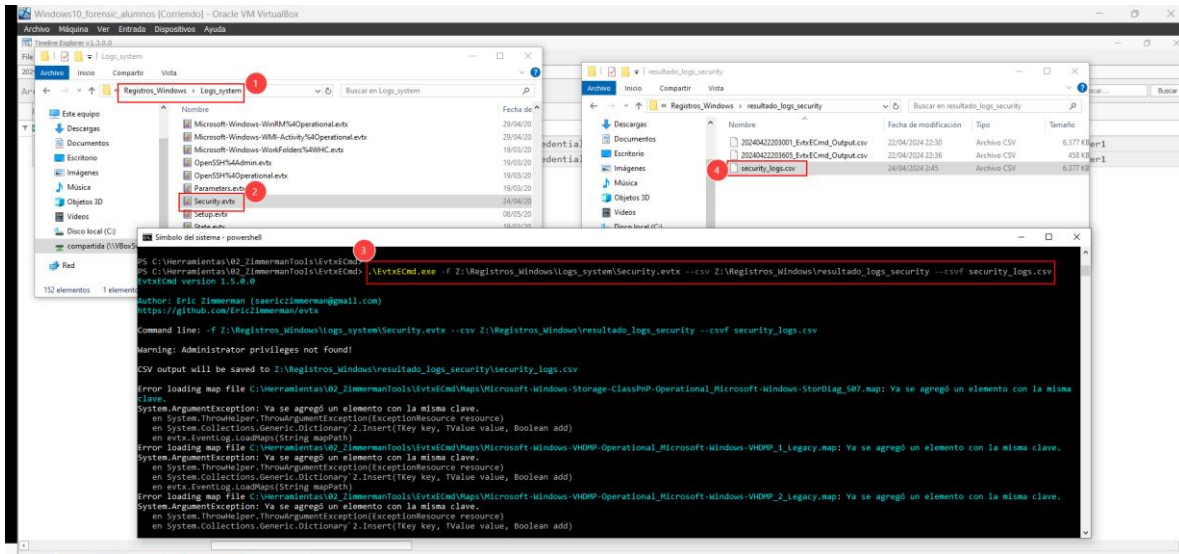


Imagen 19. Ejecución de la herramienta EvtCmd

7. Finalmente usamos la herramienta de **Timeline Explorer** en la que importamos el archivo **securty_logs.csv** y observamos que hay una columna que se llama **Remote Host** así que realizamos un filtro para ver de manera muy rápida las conexiones remotas y de esta manera hemos logrado encontrar el puerto 445 usado para la conexión.

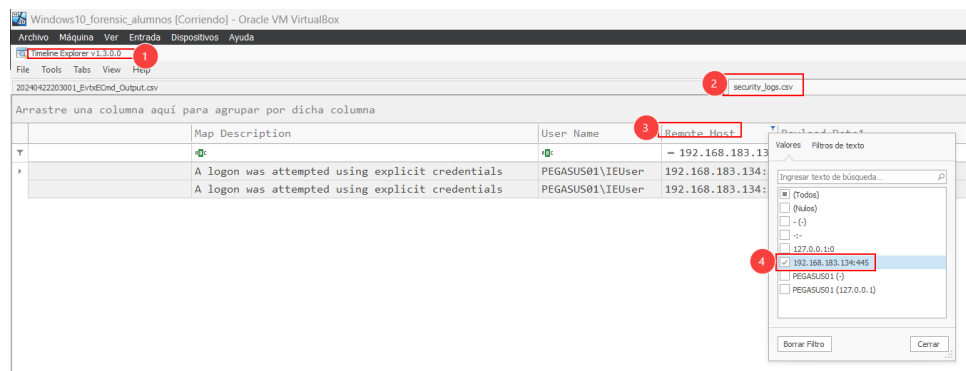


Imagen 20. Revisión de los archivos CSV

Práctica 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.

Procedimiento.

1. Para realizar el análisis de la memoria Ram, se ha configurado el siguiente escenario:
 - a. Se instalo en un PC (32 bits) la distribución de Debian 12 para una arquitectura i386, esta PC es el equipo al que se le hará el análisis de la memoria RAM.

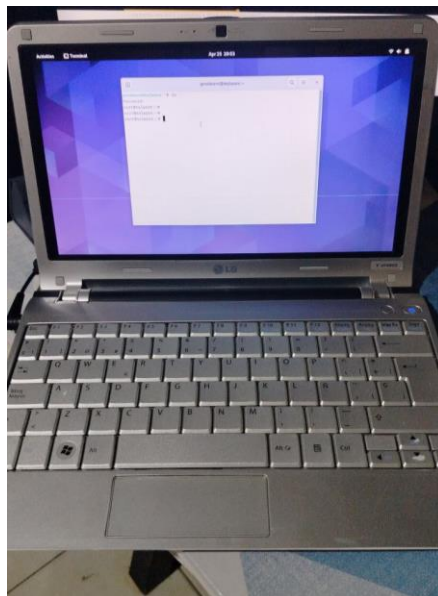
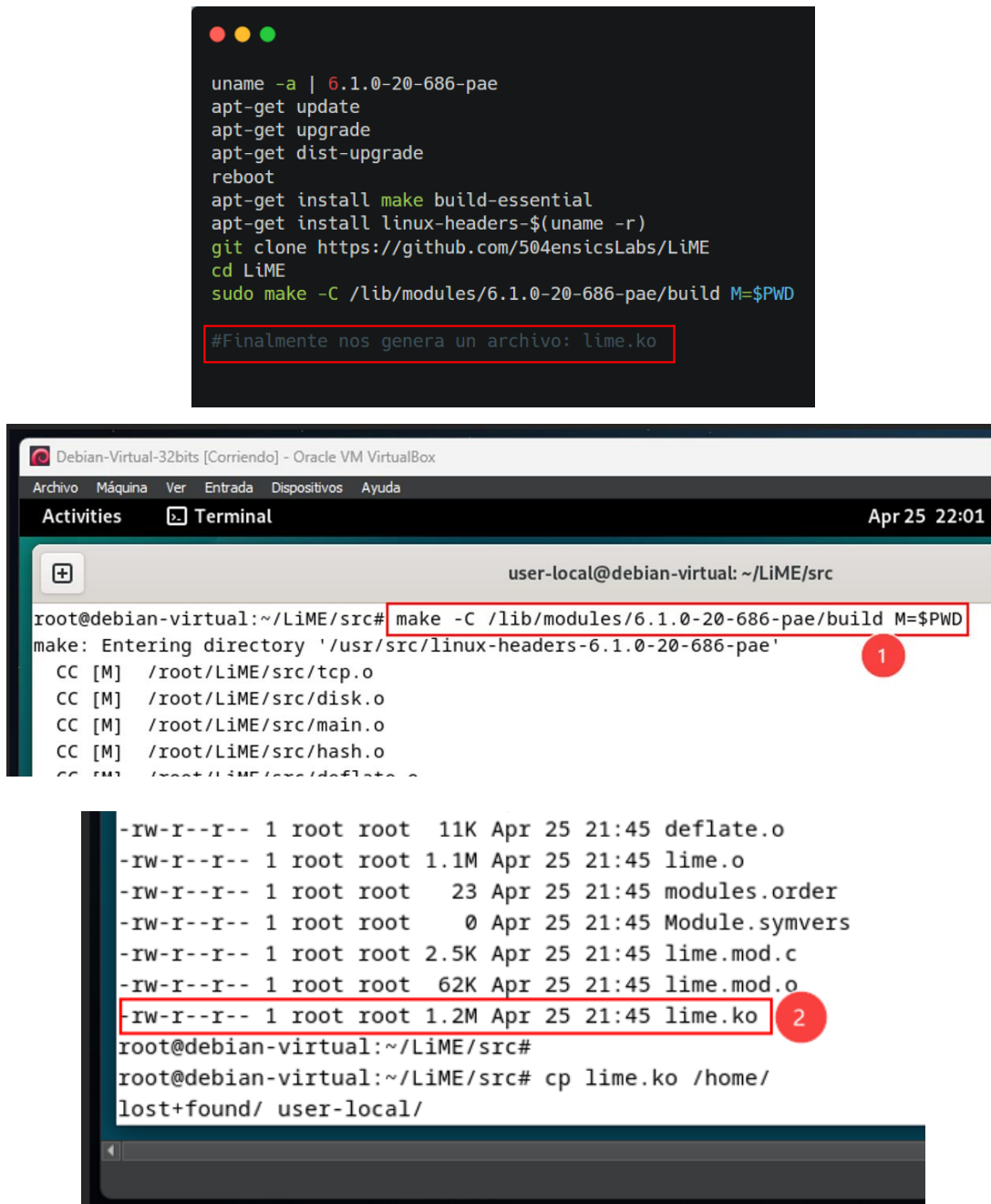


Imagen 21. PC con Debian para analizar la memoria

- b. En paralelo se configuro una Máquina Virtual usando la misma imagen ISO, después, en esta misma maquina se construyó el archivo **Kernel Object** para ello se siguieron una secuencia de pasos, ver imagen siguiente.



The image consists of two screenshots from a terminal window. The top screenshot shows a sequence of commands to set up the environment for building the kernel module. The bottom screenshot shows the execution of the 'make' command and the resulting file permissions for the generated files, with 'lime.ko' highlighted.

```
uname -a | 6.1.0-20-686-pae
apt-get update
apt-get upgrade
apt-get dist-upgrade
reboot
apt-get install make build-essential
apt-get install linux-headers-$(uname -r)
git clone https://github.com/504ensicsLabs/LiME
cd LiME
sudo make -C /lib/modules/6.1.0-20-686-pae/build M=$PWD

#Finalmente nos genera un archivo: lime.ko
```

```
user-local@debian-virtual: ~/LiME/src
root@debian-virtual:~/LiME/src# make -C /lib/modules/6.1.0-20-686-pae/build M=$PWD
make: Entering directory '/usr/src/linux-headers-6.1.0-20-686-pae'
CC [M] /root/LiME/src/tcp.o
CC [M] /root/LiME/src/disk.o
CC [M] /root/LiME/src/main.o
CC [M] /root/LiME/src/hash.o
CC [M] /root/LiME/src/deflate.o
-rw-r--r-- 1 root root 11K Apr 25 21:45 deflate.o
-rw-r--r-- 1 root root 1.1M Apr 25 21:45 lime.o
-rw-r--r-- 1 root root 23 Apr 25 21:45 modules.order
-rw-r--r-- 1 root root 0 Apr 25 21:45 Module.symvers
-rw-r--r-- 1 root root 2.5K Apr 25 21:45 lime.mod.c
-rw-r--r-- 1 root root 62K Apr 25 21:45 lime.mod.o
-rw-r--r-- 1 root root 1.2M Apr 25 21:45 lime.ko
root@debian-virtual:~/LiME/src#
root@debian-virtual:~/LiME/src# cp lime.ko /home/
lost+found/ user-local/
```

Imagen 22. Generación del archivo Kernerl Objects

c. Finalmente el archivo Kernel Object (lime.ko) se ha generado en la máquina virtual, después se ha copiado por SFTP al PC físico para que a través de la ejecución de un comando se pueda hacer la captura de la memoria RAM.

Comando ejecutado en el Host Físico para capturar la memoria RAM.

```
insmod lime.ko "path=ram_debian.lime format=lime"
```

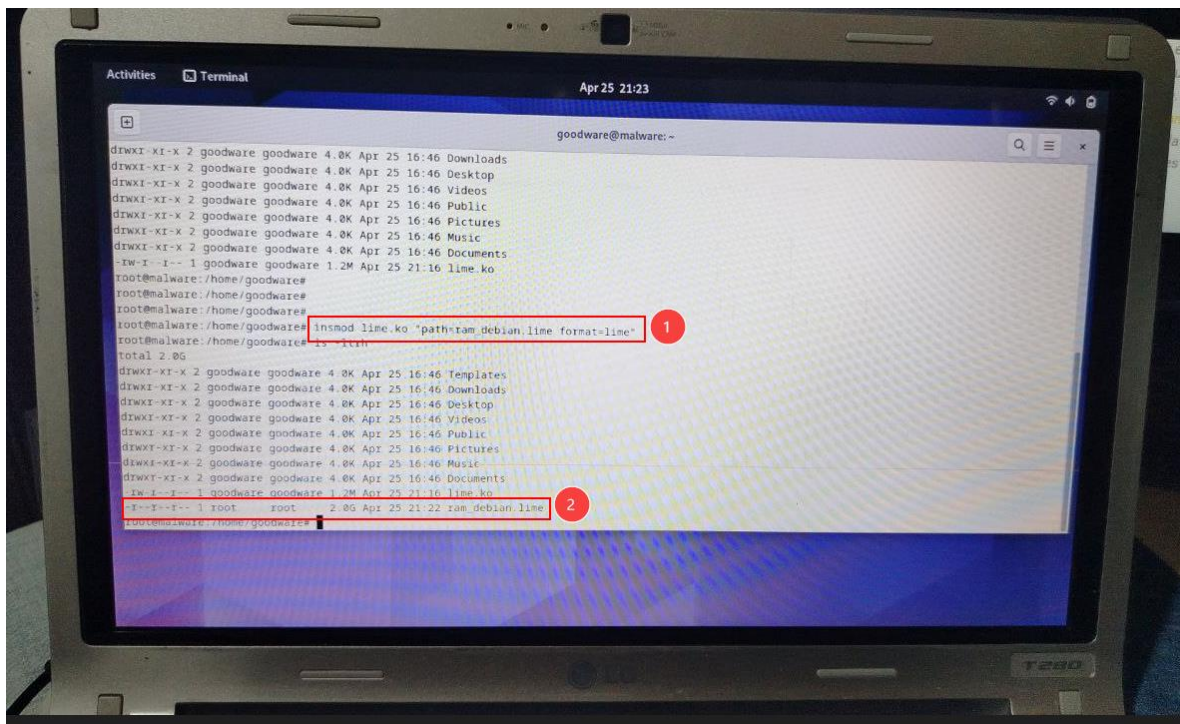
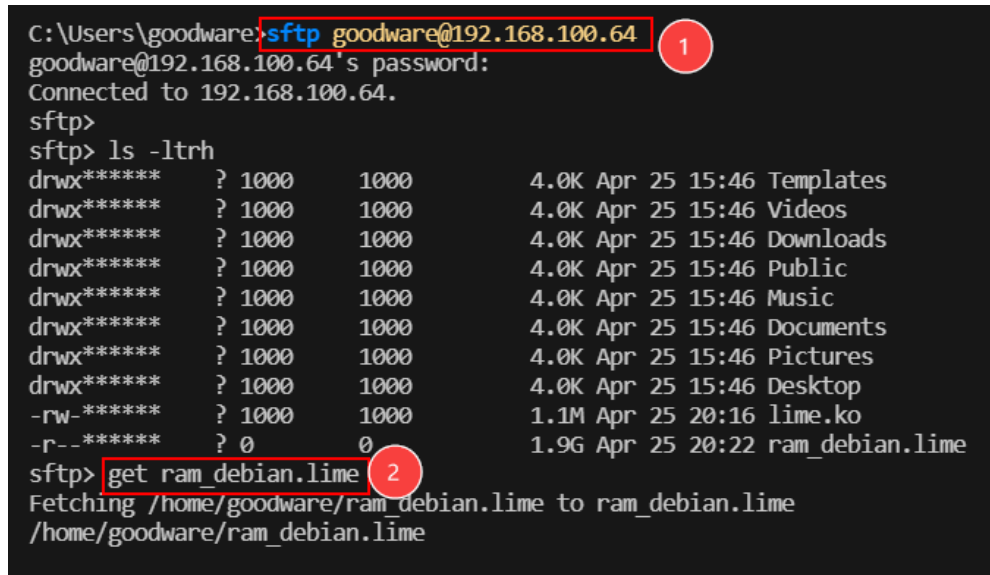


Imagen 23. Captura de la Memoria RAM del PC Físico

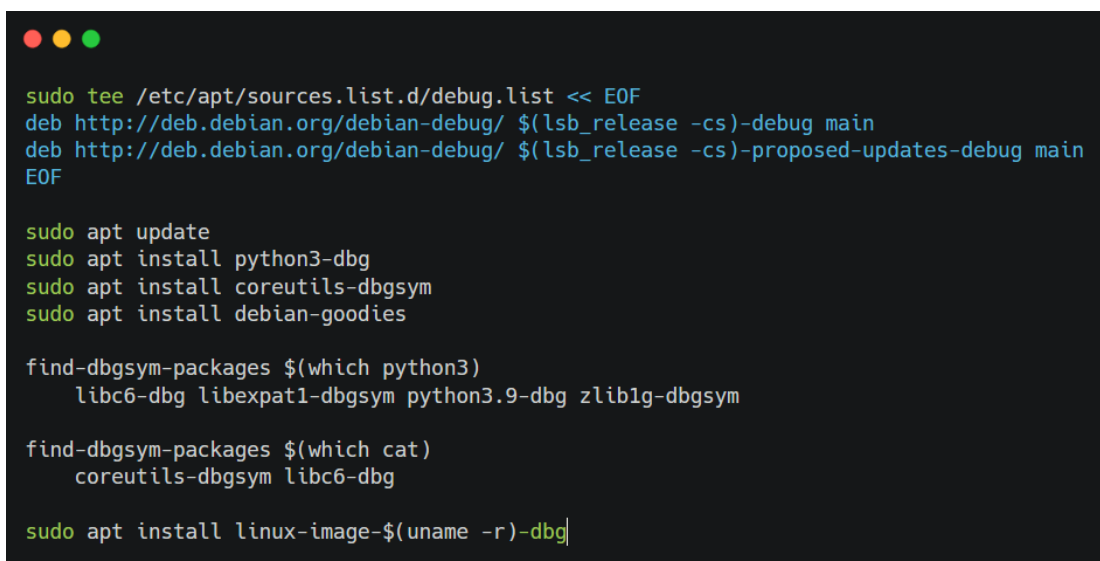
2. Finalmente se ha extraído a través de SFTP el archivo de evidencia (**ram_debian.lime**) del PC Físico para poder analizarlo y poder documentar lo que vayamos encontrando.



```
C:\Users\goodware>sftp goodware@192.168.100.64
goodware@192.168.100.64's password:
Connected to 192.168.100.64.
sftp>
sftp> ls -ltrh
drwx***** ? 1000    1000      4.0K Apr 25 15:46 Templates
drwx***** ? 1000    1000      4.0K Apr 25 15:46 Videos
drwx***** ? 1000    1000      4.0K Apr 25 15:46 Downloads
drwx***** ? 1000    1000      4.0K Apr 25 15:46 Public
drwx***** ? 1000    1000      4.0K Apr 25 15:46 Music
drwx***** ? 1000    1000      4.0K Apr 25 15:46 Documents
drwx***** ? 1000    1000      4.0K Apr 25 15:46 Pictures
drwx***** ? 1000    1000      4.0K Apr 25 15:46 Desktop
-rw-***** ? 1000    1000     1.1M Apr 25 20:16 lime.ko
-r--***** ? 0       0       1.9G Apr 25 20:22 ram_debian.lime
sftp> get ram_debian.lime
Fetching /home/goodware/ram_debian.lime to ram_debian.lime
/home/goodware/ram_debian.lime
```

Imagen 24. Extracción de la evidencia (ram_debian.lime) del equipo físico

3. Después se creó la tabla de símbolos de Debian 12 para ello se instalaron los siguientes paquetes:



```
sudo tee /etc/apt/sources.list.d/debug.list << EOF
deb http://deb.debian.org/debian-debug/ $(lsb_release -cs)-debug main
deb http://deb.debian.org/debian-debug/ $(lsb_release -cs)-proposed-updates-debug main
EOF

sudo apt update
sudo apt install python3-dbg
sudo apt install coreutils-dbgsym
sudo apt install debian-goodies


find-dbgsym-packages $(which python3)
libc6-dbg libexpat1-dbgsym python3.9-dbg zlib1g-dbgsym

find-dbgsym-packages $(which cat)
coreutils-dbgsym libc6-dbg

sudo apt install linux-image-$(uname -r)-dbg
```

Imagen 25. Instalación de paquetes y repositorios

4. Se uso dwarf2json y se generó la tabla de símbolos, se revisó la documentación oficial:
(<https://github.com/volatilityfoundation/dwarf2json>)



```
apt install golang-go
go version
git clone https://github.com/volatilityfoundation/dwarf2json.git
cd dwarf2json
go build
./dwarf2json linux --elf /boot/vmlinuz-6.1.0-20-686-pae --system-map /boot/System.map-6.1.0-20-686-pae
| xz -c > DEBIAN12x64_6.1.0-20-686-pae.json.xz

Nos genero el archivo: DEBIAN12x64_6.1.0-20-686-pae.json.xz
```

Imagen 26. Se uso dwarf2json para crear la tabla de símbolos de Debian

El archivo `DEBIAN12x64_6.1.0-20-686-pae.json.xz` se copió en la ruta `/home/user-local/volatility3/volatility3/symbols/linux` para que al ejecutar Volatility pueda hacer uso de la tabla de símbolos.

5. Finalmente se analizó la captura de la memoria RAM tomada del PC con Debian 12, y así poder ver los detalles al interior de la memoria, sin embargo, presentamos errores en la ejecución del análisis, a continuación, se muestran la evidencia del error.

```

malware@DESKTOP-T1P5T5F:~/volatility3$ python3 vol.py -vvvv -f ram.debian.lime linux.pslist.PsList
volatility3 framework 2.7.0
INFO volatility3.cli: Volatility plugins path: ['/home/malware/volatility3/volatility3/plugins', '/home/malware/volatility3/volatility3/framework/plugins']
INFO volatility3.cli: Volatility symbols path: ['/home/malware/volatility3/volatility3/symbols', '/home/malware/volatility3/volatility3/framework/symbols']
INFO volatility3.framework.automatic: Detected a linux category plugin
INFO volatility3.framework.automatic: Running automatic: ConstructionMagic
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.configuration.requirements: Symbol table requirement not yet fulfilled: plugins.PsList.kernel.symbol_table_name
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.automatic.construct_layers: Failed on requirement: plugins.PsList.kernel.symbol_table_name
DETAIL 1 volatility3.framework.automatic.construct_layers: Failed on requirement: plugins.PsList.kernel
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.automatic.construct_layers: Failed on requirement: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.automatic.construct_layers: Failed on requirement: plugins.PsList.kernel
DETAIL 1 volatility3.framework.configuration.requirements: Symbol table requirement not yet fulfilled: plugins.PsList.kernel.symbol_table_name
DETAIL 1 volatility3.framework.automatic.construct_layers: Failed on requirement: plugins.PsList.kernel
DETAIL 1 volatility3.framework.configuration.requirements: Symbol table requirement not yet fulfilled: plugins.PsList.kernel.symbol_table_name
DETAIL 1 volatility3.framework.automatic.construct_layers: Failed on requirement: plugins.PsList.kernel
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.configuration.requirements: Symbol table requirement not yet fulfilled: plugins.PsList.kernel.symbol_table_name
DETAIL 1 volatility3.framework.automatic.construct_layers: Failed on requirement: plugins.PsList
INFO volatility3.framework.automatic: Running automatic: SymbolCacheMagic
DETAIL 2 volatility3.framework.automatic.symbol_cache: Identified files: //home/malware/volatility3/volatility3/framework/symbols/linux/output.json as b'Linux version 6.1.0-20-686-pae (debian-kernel@lists.deb
ian.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld (GNU Binutils for Debian) 2.40) #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11)\n\x00'
INFO volatility3.framework.automatic: Running automatic: LayerStacker
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.configuration.requirements: Symbol table requirement not yet fulfilled: plugins.PsList.kernel.symbol_table_name
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using Linestacker
DETAIL 2 volatility3.framework.automatic.stacker: Stacked Linelayer using Linestacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using Elf64Stacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using XenCoreDumpStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using AVMLStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using QemuStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using WindowsCrashDumpStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using VmwareStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using LinuxIntelStacker
DEBUG volatility3.framework.automatic.symbol_cache: Duplicate entry for identifier b'Linux version 6.1.0-20-686-pae (debian-kernel@lists.debian.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld (GNU Binutils
for Debian) 2.40) #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11)\n\x00' : file:///home/malware/volatility3/volatility3/framework/symbols/linux/output.json and file:///home/malware/volatility3/volatility
3/symbols/linux/output.json
DEBUG volatility3.framework.automatic.linux: Identified banner: b'Linux version 6.1.0-20-686-pae (debian-kernel@lists.debian.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld (GNU Binutils
for Debian) 2.40) #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11)\n\x00'
INFO volatility3.schemas: Dependency for validation unavailable: jsonschema
DEBUG volatility3.schemas: All validations will report success, even with malformed input
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!assoc_array_ptr

```

```

DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!netns_ipvs
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!mtd_info
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!ioopf_device_param
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lcan_pkg_stats
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lcan_rcv_lists_stats
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lcan_dev_rcv_lists
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!mpls_route
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lscpt_mib
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lsmc_stats_rsn
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lsmc_stats
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!ldm_hw_stat_delta
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lgarp_port
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!mpls_dev
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lmpc_port
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!ltpc_bearer
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!ludp_tunnel_nic
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lpcpu_dstats
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lcfg80211_conn
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lcfg80211_cached_keys
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lcfg80211_cqm_config
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lphy_led_trigger
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lphylink
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lpse_control
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lsfp
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!llibipw_device
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lsmc_hashinfo
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lrfkill
DEBUG volatility3.framework.symbols: Unresolved reference: LintelStacker!lcfg80211_internal_bss
DEBUG volatility3.framework.automatic.linux: Linux ASLR shift values determined: physical c000000 virtual c000000
DEBUG volatility3.framework.automatic.linux: DTB was found at: 0xde80000
DETAIL 2 volatility3.framework.automatic.stacker: Stacked IntelLayer using LinuxIntelStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using Elf64Stacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using XenCoreDumpStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using AVMLStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using QemuStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using WindowsCrashDumpStacker
DETAIL 2 volatility3.framework.automatic.stacker: Attempting to stack using VmwareStacker
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.configuration.requirements: Symbol table requirement not yet fulfilled: plugins.PsList.kernel.symbol_table_name
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.configuration.requirements: Symbol table requirement not yet fulfilled: plugins.PsList.kernel.symbol_table_name
DETAIL 1 volatility3.framework.automatic.construct_layers: Failed on requirement: plugins.PsList.kernel
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name
DETAIL 1 volatility3.framework.configuration.requirements: IndexError - No configuration provided: plugins.PsList.kernel.layer_name.memory_layer

```



```

DEBUG volatility3.framework.automatic.stacker: Stacked layers: ['Intellayer', 'LimeLayer', 'FileLayer']
INFO volatility3.framework.automatic: Running automatic: SymbolFinder
INFO volatility3.framework.automatic: Running automatic: LinuxSymbolFinder
DETAIL 1 volatility3.framework.configuration.requirements: Symbol table requirement not yet fulfilled: plugins.Pslist.kernel.symbol_table.name
DEBUG volatility3.framework.automatic.symbol_cache: Duplicate entry for identifier b'linux version 6.1.0-20-686-pae (debian-kernel@lists.debian.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld (GNU Binutils for Debian) 2.40) #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11)\n\x00': file:///home/malware/volatility3/volatility3/framework/symbols/linux/output.json and file:///home/malware/volatility3/volatility3/symbols/linux/output.json
DEBUG volatility3.framework.automatic.symbol_finder: Identified banner: b'linux version 6.1.0-20-686-pae (debian-kernel@lists.debian.org) (gcc-12 (Debian 12.2.0-14) 12.2.0, GNU ld (GNU Binutils for Debian) 2.40) #1 SMP PREEMPT_DYNAMIC Debian 6.1.85-1 (2024-04-11)\n\x00'
DEBUG volatility3.framework.automatic.symbol_finder: Using symbol library: file:///home/malware/volatility3/volatility3/framework/symbols/linux/output.json
INFO volatility3.schemas: Dependency for validation unavailable: jsonschema
DEBUG volatility3.schemas: All validations will report success, even with malformed input
INFO volatility3.framework.automatic: Running automatic: KernelModule

OFFSET (V)  PID  TID  PPID  COMM  File output
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1lassoc_array_ptr
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1netns_ipv6
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1mtid_info
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1iopf_device_param
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1can_rcv_stats
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1can_rcv_lists_stats
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1can_dev_rcv_lists
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1mpis_route
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1scip_mib
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1lsm_stats_rsn
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1lsm_stats
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1dm_hw_stat_delta
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1garp_port
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1mpis_dev
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1mrp_port
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1tipc_bearer
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1udp_tunnel_nic
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1pcpu_dstats
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1cf80211_com
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1cf80211_cached_keys
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1cf80211_cqm_config
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1phy_led_trigger
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1phylink
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1pse_control
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1sfp
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1libipw_device
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1lsm_hashinfo
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1rkill
DEBUG volatility3.framework.symbols: Unresolved reference: symbol_table.name1cf80211_internal_bss
malware@DESKTOP-T1F5T5F: ~/volatility3

```

```

python3 vol.py -f ram_debian.lime linux.pslist.PsList

```

```

malware@DESKTOP-T1F5T5F:~/volatility3$ python3 vol.py -f ram_debian.lime linux.pslist.PsList
Volatility 3 Framework 2.7.0
Progress: 100.00      Stacking attempts finished
OFFSET (V)  PID  TID  PPID  COMM  File output

```

Imagen 27. Ejecución de Volatility

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 comparais

Yo os he dado 3 ejemplos, si se os ocurre otro mecanismo en el que podáis probar, usado, se valorará positivamente.

Procedimiento.

Inicialmente se ha seleccionado la imagen con la que se pretende hacer la extracción de metadatos.

Se han visualizado los detalles de la fotografía desde el apartado de detalles de nuestro Sistema Operativo Windows y hemos encontrado algunos metadatos interesantes como:

- La fecha de captura de la fotografía
- Las dimensiones de la fotografía
- El fabricante y modelo de la cámara

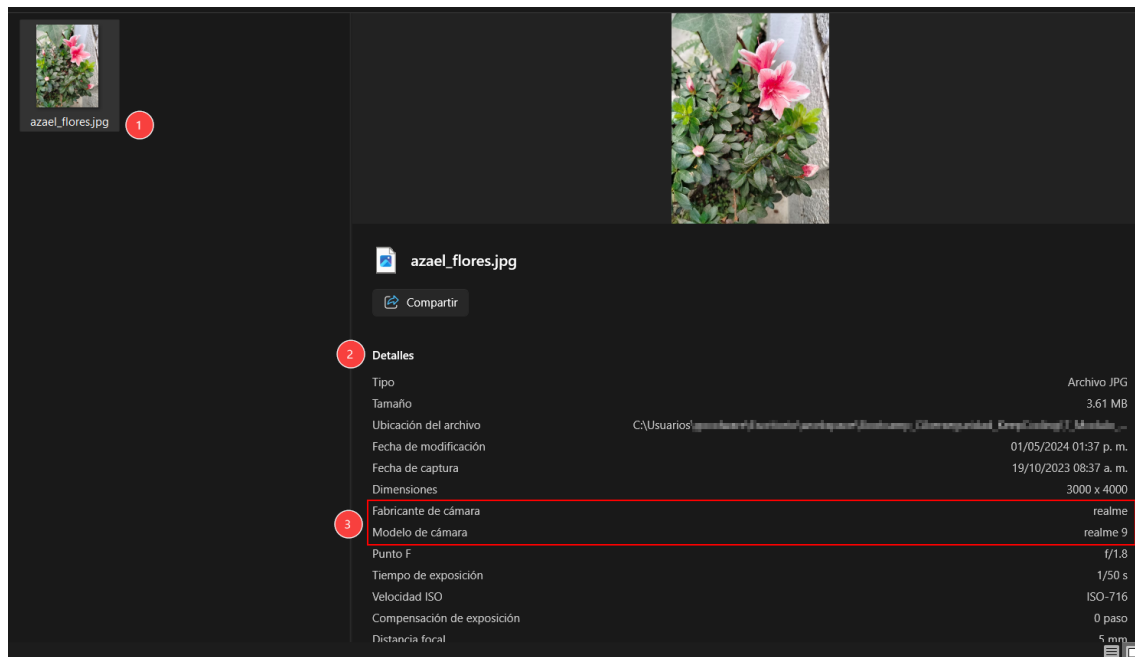


Imagen 28. Metadatos de la imagen.

Adicionalmente se ha usado la herramienta **Exif Tool** del autor **David Bombal** para poder extraer los metadatos de la fotografía, para ello se ejecuto un script de Python, enseguida se muestran los resultados.

```
hon-scripts> python.exe .\exif.py
C:\Users\goodware\Desktop\workspace\Bootcamp_Ciberseguridad_KeepCoding\7_scripts\exif.py:43: SyntaxWarning: invalid escape sequence '\ '
print("""

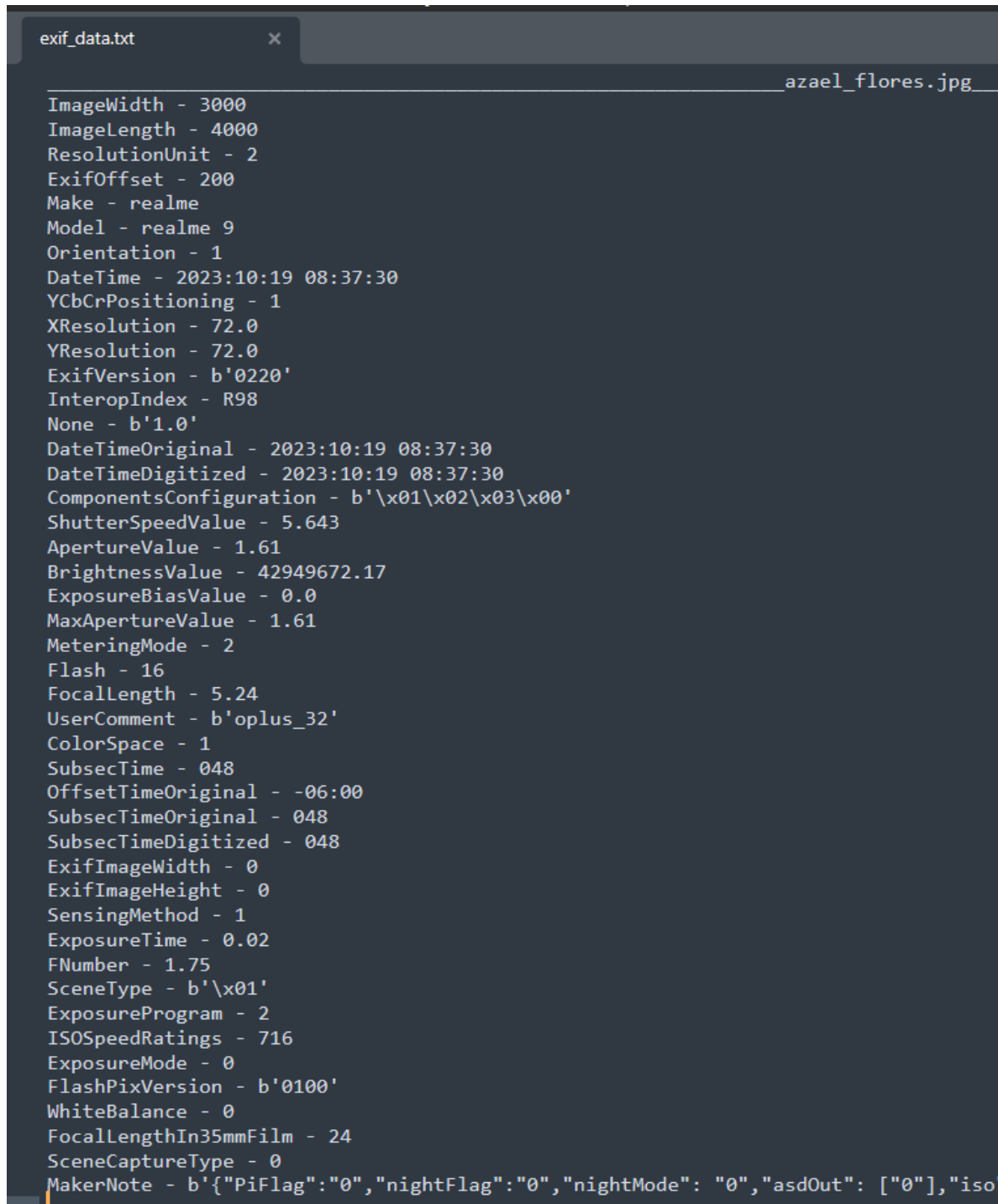
David Bombal

EXIF TOOL

How do you want to receive the output:
1 - File
2 - Terminal
Enter choice here: 1
PS C:\Users\goodware\Desktop\workspace\Bootcamp_Ciberseguridad_KeepCoding\7_scripts>
```

Imagen 29. Ejecución de Exif Tool

La herramienta **Exif Tool** creo un archivo **exif_data.txt** con los detalles de los metadatos que contiene la fotografía, a continuación, se muestra la información que extrajo el script.



```
exif_data.txt x
_azael_flores.jpg
ImageWidth - 3000
ImageLength - 4000
ResolutionUnit - 2
ExifOffset - 200
Make - realme
Model - realme 9
Orientation - 1
DateTime - 2023:10:19 08:37:30
YCbCrPositioning - 1
XResolution - 72.0
YResolution - 72.0
ExifVersion - b'0220'
InteropIndex - R98
None - b'1.0'
DateTimeOriginal - 2023:10:19 08:37:30
DateTimeDigitized - 2023:10:19 08:37:30
ComponentsConfiguration - b'\x01\x02\x03\x00'
ShutterSpeedValue - 5.643
ApertureValue - 1.61
BrightnessValue - 42949672.17
ExposureBiasValue - 0.0
MaxApertureValue - 1.61
MeteringMode - 2
Flash - 16
FocalLength - 5.24
UserComment - b'oplus_32'
ColorSpace - 1
SubsecTime - 048
OffsetTimeOriginal - -06:00
SubsecTimeOriginal - 048
SubsecTimeDigitized - 048
ExifImageWidth - 0
ExifImageHeight - 0
SensingMethod - 1
ExposureTime - 0.02
FNumber - 1.75
SceneType - b'\x01'
ExposureProgram - 2
ISOSpeedRatings - 716
ExposureMode - 0
FlashPixVersion - b'0100'
WhiteBalance - 0
FocalLengthIn35mmFilm - 24
SceneCaptureType - 0
MakerNote - b'{"PiFlag":"0","nightFlag":"0","nightMode": "0","asdOut": ["0"],"iso
```

Imagen 30. Metadatos encontrados en la imagen

Extracción de Metadatos después de haber enviado la imagen por WhatsApp.

La imagen original se envió por whatsapp, después se descargó a nuestro PC y se ejecuto la herramienta de Exif Tool sobre esta misma, se observa que esta nueva versión de la imagen no contiene metadatos, ver imagen siguiente.

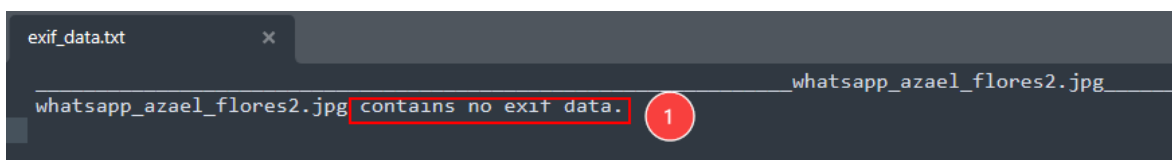


Imagen 31. Imagen enviada a WhatsApp

Extracción de Metadatos después de haber enviado la imagen por Telegram.

La imagen original se envió por whatsapp, después se descargó a nuestro PC y se ejecuto la herramienta de Exif Tool sobre esta misma, se observa que esta nueva versión de la imagen no contiene metadatos, ver imagen siguiente.

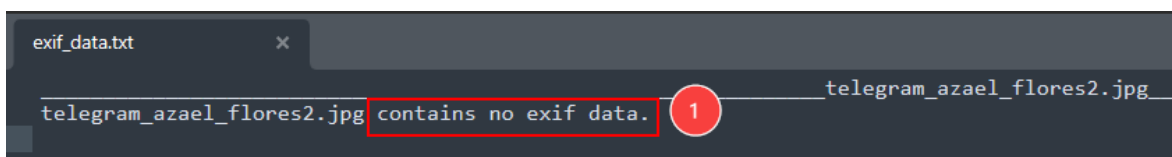
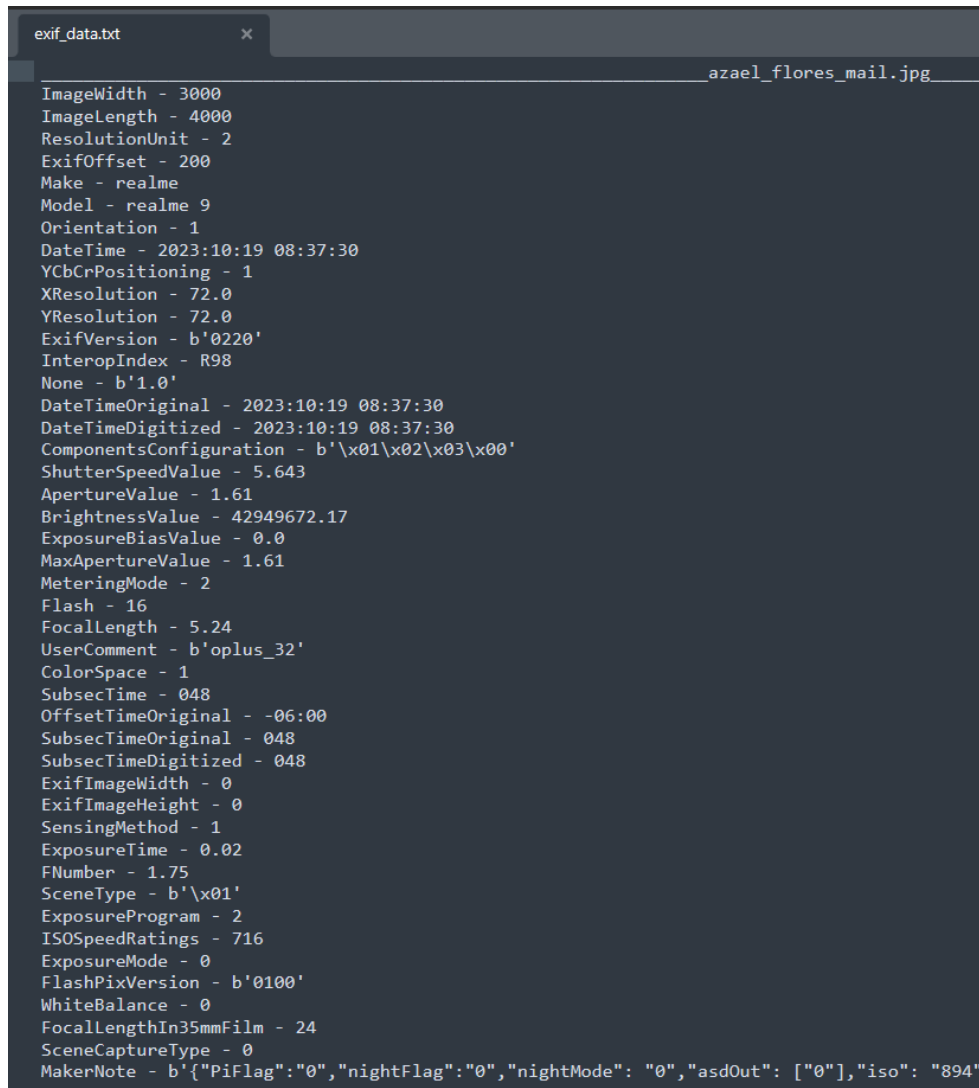


Imagen 32. Imagen enviada a Telegram

Extracción de Metadatos después de haber enviado la imagen por Mail.

La imagen original se envió por **G-mail**, después se descargó a nuestro PC y se ejecutó la herramienta de **Exif Tool** sobre esta misma, se observa que esta nueva versión de la imagen (la que se descargó de G-mail) mantiene los metadatos de la imagen original, debido a ello se puede concluir que el proveedor de correo no extrae ni usa los metadatos de los archivos digitales.



```
exif_data.txt x azael_flores_mail.jpg
ImageWidth - 3000
ImageLength - 4000
ResolutionUnit - 2
ExifOffset - 200
Make - realme
Model - realme 9
Orientation - 1
DateTime - 2023:10:19 08:37:30
YCbCrPositioning - 1
XResolution - 72.0
YResolution - 72.0
ExifVersion - b'0220'
InteropIndex - R98
None - b'1.0'
DateTimeOriginal - 2023:10:19 08:37:30
DateTimeDigitized - 2023:10:19 08:37:30
ComponentsConfiguration - b'\x01\x02\x03\x00'
ShutterSpeedValue - 5.643
ApertureValue - 1.61
BrightnessValue - 42949672.17
ExposureBiasValue - 0.0
MaxApertureValue - 1.61
MeteringMode - 2
Flash - 16
FocalLength - 5.24
UserComment - b'oplus_32'
ColorSpace - 1
SubsecTime - 048
OffsetTimeOriginal - -06:00
SubsecTimeOriginal - 048
SubsecTimeDigitized - 048
ExifImageWidth - 0
ExifImageHeight - 0
SensingMethod - 1
ExposureTime - 0.02
FNumber - 1.75
SceneType - b'\x01'
ExposureProgram - 2
ISOSpeedRatings - 716
ExposureMode - 0
FlashPixVersion - b'0100'
WhiteBalance - 0
FocalLengthIn35mmFilm - 24
SceneCaptureType - 0
MakerNote - b'{"PiFlag":"0","nightFlag":"0","nightMode": "0","asdOut": ["0"],"iso": "894"'
```

Imagen 33. Imagen enviada por Mail, mantiene los metadatos