

INFORME DE DIGITAL FORENSICS AND INCIDENT RESPONSE

Informe de DFIR

Digital Forensics and Incident Response / Seguridad de la Información

Jordan Andres Diaz Sanchez

TABLA DE CONTENIDO

1. INTRODUCCIÓN

- 1.1. Objetivo del informe

2. HERRAMIENTAS EMPLEADAS – INVESTIGACIÓN

- 2.1. PowerShell – Hash del fichero
- 2.2. Registry Explorer – Nombre de la maquina
- 2.3. Registry Explorer – Ficheros maliciosos
- 2.4. AccesData FTK Imager – Fichero Control remoto
- 2.5. Autopsy – Ficheros Eliminados
- 2.6. Mimikatz – Contraseña débil
- 2.7. RegRipper – Conexión RDP
- 2.8. Autopsy – Puerto de conexión
- 2.9. Backdoor – Powershell maliciosa
- 2.10. Autopsy – Fecha descarga control remoto
- 2.11. AccesData FTK Imager – Fecha ejecución control remoto
- 2.12. Autopsy – Conexión control remoto

3. METADATOS

4. CONCLUSION

5. REFERENCIAS Y BIBLIOGRAFÍA

1. INTRODUCCIÓN

1.1. Objetivo del informe

En este informe documente el análisis forense digital realizado sobre una imagen de disco virtual (.vmdk) correspondiente a un sistema Windows 10 comprometido.

El objetivo principal de la práctica fue basada en identificar vectores de ataque, rastrear la actividad maliciosa y responder a un conjunto de retos publicados en la plataforma **ctf.sancastell.me** orientados a la formación en técnicas DFIR (Digital Forensics and Incident Response). La metodología que aplique incluyó la adquisición y verificación de la integridad de evidencias mediante herramientas vistas previamente en el módulo para realizar cálculo de hash, seguida de un análisis estático del sistema utilizando herramientas especializadas como Registry Explorer, FTK Imager, Autopsy, Mimikatz y RegRipper. Estas herramientas permitieron extraer información del sistema de archivos, el registro de Windows, metadatos de archivos multimedia, y credenciales de usuarios.

Durante el proceso detecte artefactos forenses que evidencian el uso de herramientas de acceso remoto, scripts maliciosos en PowerShell, archivos eliminados, conexiones RDP sospechosas, y configuraciones de seguridad deficientes, como contraseñas débiles. Cada hallazgo fue correlacionado con la herramienta empleada, permitiendo reconstruir la secuencia de acciones del atacante con un alto grado de fidelidad.

Este informe presenta de forma detallada cada uno de los hallazgos, así como las técnicas empleadas para su obtención, concluyendo con una reflexión sobre las lecciones aprendidas en clase y la importancia de la respuesta temprana ante incidentes de seguridad.

2. HERRAMIENTAS EMPLEADAS – INVESTIGACIÓN

Durante el análisis forense se emplearon diversas herramientas especializadas que permitieron examinar de forma estructurada los diferentes artefactos del sistema comprometido.

2.1. PowerShell – Hash del fichero

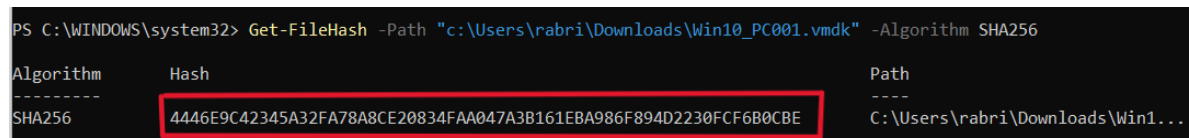
Aplicación en la práctica:

En la práctica, utilice PowerShell para obtener el hash SHA256 de la imagen forense Win10_PC001.vmdk, mediante el comando:

Get-FileHash .\Win10_PC001.vmdk -Algorithm SHA256

Resultado obtenido:

4446E9C42345A32FA78A8CE20834FAA047A3B161EBA986F894D2230FCF6B0CBE



```
PS C:\WINDOWS\system32> Get-FileHash -Path "c:\Users\rabri\Downloads\Win10_PC001.vmdk" -Algorithm SHA256
```

Algorithm	Hash	Path
SHA256	4446E9C42345A32FA78A8CE20834FAA047A3B161EBA986F894D2230FCF6B0CBE	C:\Users\rabri\Downloads\Win1...

Ilustración 1 Hash del fichero

2.2. Registry Explorer – Nombre de la maquina

Aplicación en la práctica:

En la práctica, utilice Registry Explorer para cargar y analizar el archivo SYSTEM extraído de la imagen forense. El objetivo fue identificar el nombre del equipo analizado, un dato clave para la documentación del caso y la contextualización del entorno.

Se navegó a la ruta de registro:

SYSTEM\ControlSet001\Control\ComputerName\ComputerName

Dentro de esta clave se encuentra el valor ComputerName, que almacena el nombre del host configurado en el sistema operativo.

Resultado obtenido: PEGASUS01

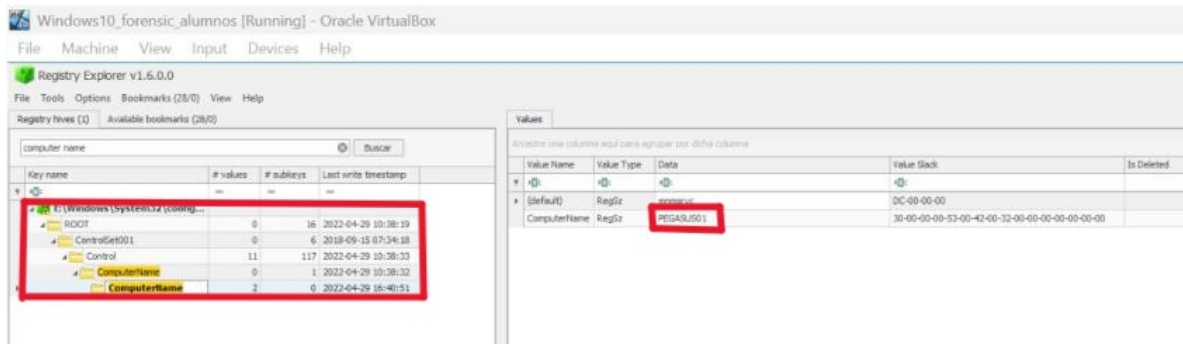


Ilustración 2 Nombre de la maquina

2.3. Registry Explorer – Ficheros maliciosos

Aplicación en la práctica:

En este caso, Registry Explorer la utilice para examinar claves comunes donde suelen configurarse cargas maliciosas persistentes.

Se analizó la siguiente ruta:

NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Run

Esta clave contiene valores que especifican programas que se ejecutan automáticamente al iniciar sesión el usuario.

Dentro de esta clave se identificó un valor sospechoso que apuntaba a un ejecutable alojado en una carpeta inusual:

TMP =

C:\Users\IEUser\AppData\Local\Temp\TMP\TeamViewer_Setup_x64.exe

Resultado obtenido: La carpeta del Malware es **TMP**

	Value Name	Value Type	Data
▼	Run	REG_SZ	Run
▶	SecurityHealth	RegExpandSz	%windir%\system32\SecurityHealthSystray...
	bginfo	RegSz	C:\Bginfo\Bginfo.exe /accepteula /c:\bginf...
	VMware VM3DService Process	RegSz	"C:\Windows\system32\vm3dservice.exe" -u
	VMware User Process	RegSz	"C:\Program Files\VMware\VMware Tools\vm...
	UpdateSvc	RegSz	C:\TMP\p.exe -s //10.34.2.3 'net user' > C:...

Ilustración 3 Ficheros maliciosos

2.4. AccesData FTK Imager – Fichero Control remoto

Aplicación en la práctica:

Se utilizó FTK Imager para montar y explorar el sistema de archivos contenido en la imagen Win10_PC001.vmdk. Navegando por el árbol de directorios del usuario IEUser, se localizó un archivo con nombre:

TeamViewer_Setup_x64.exe

Este archivo se encontraba en la ruta:

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

Además, al revisar los metadatos del archivo (timestamps), se identificaron fechas clave como la fecha de creación y último acceso, lo que ayudó a establecer una línea temporal del incidente.

Resultado obtenido: El fichero remoto es TeamViewer_Setup_x64.exe

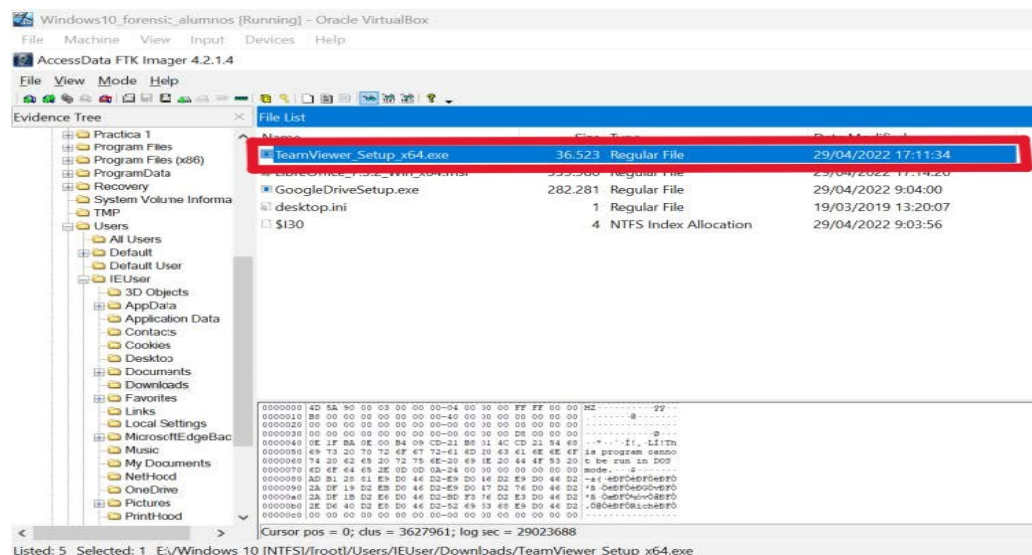


Ilustración 4 Fichero control remoto

2.5. Autopsy – Ficheros Eliminados

Aplicación en la práctica:

Se utilizó Autopsy para analizar el contenido de la imagen Win10_PC001.vmdk y, específicamente, para buscar archivos eliminados que pudieran contener información valiosa o estar relacionados con la actividad del atacante. A través del módulo de “Deleted Files”, Autopsy permitió identificar ficheros marcados como eliminados, pero aún recuperables.

Durante la revisión, se identificó el siguiente archivo:
cosas.zip
Este archivo fue hallado en el directorio:
C:\Users\IEUser\Downloads\

Resultado obtenido: El fichero eliminado fue cosas.zip

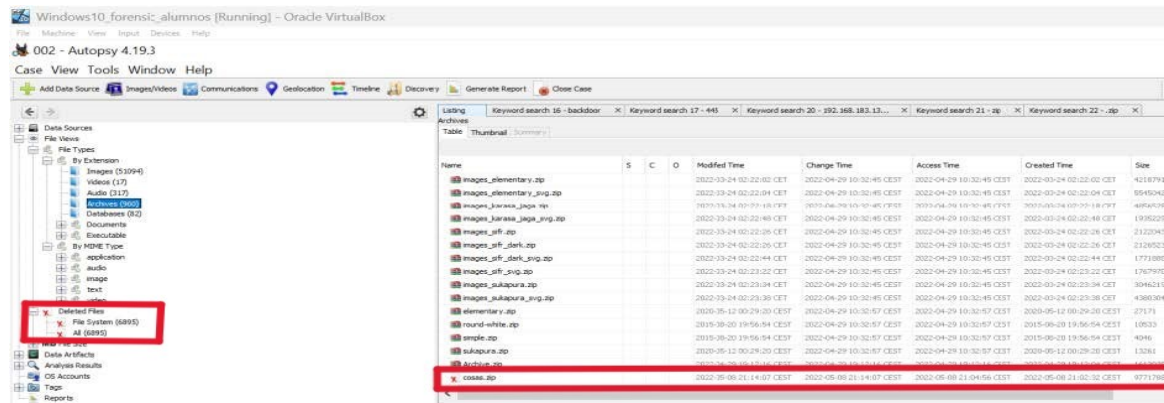


Ilustración 5 Fichero eliminado

2.6. Mimikatz – Contraseña débil

Aplicación en la práctica:

Durante el análisis, se utilizaron los archivos SAM y SYSTEM extraídos previamente de la imagen (.vmdk). Estos archivos contienen información cifrada de las cuentas locales del sistema.

Mimikatz fue empleado para realizar un dump de los hashes de contraseñas locales mediante el siguiente módulo:
lsadump::sam

Este comando permitió recuperar los hashes NTLM de las cuentas de usuario del sistema.

Posteriormente, uno de estos hashes fue sometido a un proceso de crackeo mediante una herramienta externa (como CrackStation), revelando la contraseña original.

Resultado obtenido: Se descubrió que una contraseña era **qwerty**

```
mimikatz 2.2.0 x86 (oe.oe)

Default Iterations : 4096
Credentials
  aes256_hmac      (4096) : fb60f0d32a8abb7dd991ae530844c927fb25380fffeb119ccd0971c5be8df321
  aes128_hmac      (4096) : e4617e2dd5e029348f552ece98695ddb
  des_cbc_md5      (4096) : 1ce9546ebf6e5e45

* Packages *
  NTLM-Strong-NTOWF

* Primary:Kerberos *
  Default Salt : WDAGUtilityAccount
  Credentials
    des_cbc_md5    : 1ce9546ebf6e5e45

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

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

* Primary:Kerberos-Newer-Keys *
  Default Salt : MSEDGWIN10IEUser
  Default Iterations : 4096
  Credentials
    aes256_hmac      (4096) : 72cc752f2addce7556960ad819259738c4fd86e7130cee6b06aca1137ad1e6cb
    aes128_hmac      (4096) : 7d83280d0766f4ad6510460fbd975fbc
    des_cbc_md5      (4096) : ecd9340ddf77406b
```

Ilustración 6 Contraseña debil 1

🔒 Password Hashing Security 🔒 Defuse Security 🔒

Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:

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

Color Codes: Green Exact match, Yellow Partial match, Red Not found.

Ilustración 7 Contraseña debil2

2.7. RegRipper – Conexión RDP

Aplicación en la práctica:

Se utilizó RegRipper para analizar las claves del sistema relacionadas con conexiones de Escritorio Remoto (RDP). Uno de los plugins relevantes (terminalserver.pl) permite obtener registros de hosts remotos conectados a través de RDP, así como información de sesiones previas y puertos utilizados.

Resultado obtenido: Se descubrió la IP remota: **192.168.183.134**

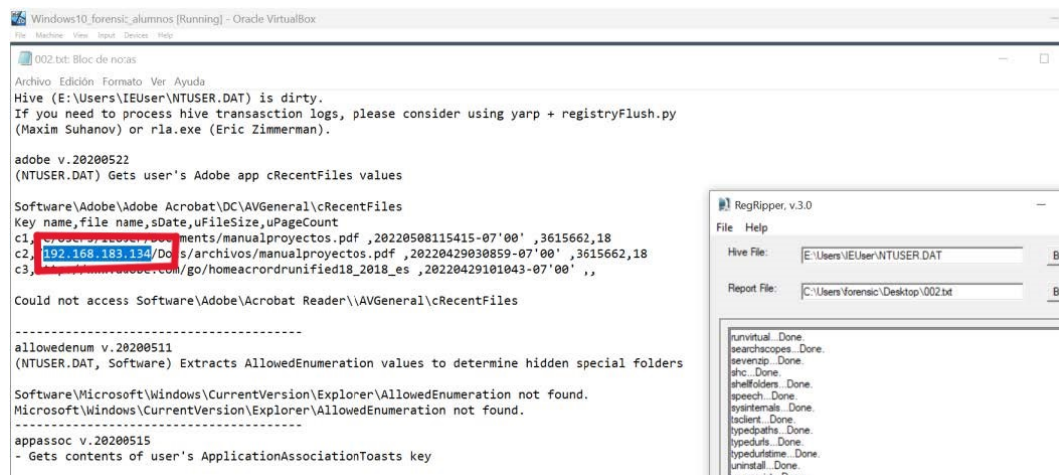


Ilustración 8 Conexión RDP

2.8. Autopsy – Puerto de conexión

Aplicación en la práctica:

Se utilizó la función de búsqueda por palabras clave (keyword search) dentro de Autopsy para localizar referencias a puertos utilizados. En este caso, se detectó una coincidencia en el archivo: DataStore.edb con la frase: firewall that blocks TCP port 445.

Esto indica que en algún momento se discutió o configuró el bloqueo o apertura del puerto TCP 445, lo cual es relevante porque ese puerto es comúnmente utilizado para servicios SMB y es un vector de ataque frecuente para propagación de malware o movimientos laterales.

Resultado obtenido: El puerto de conexión fue **445**

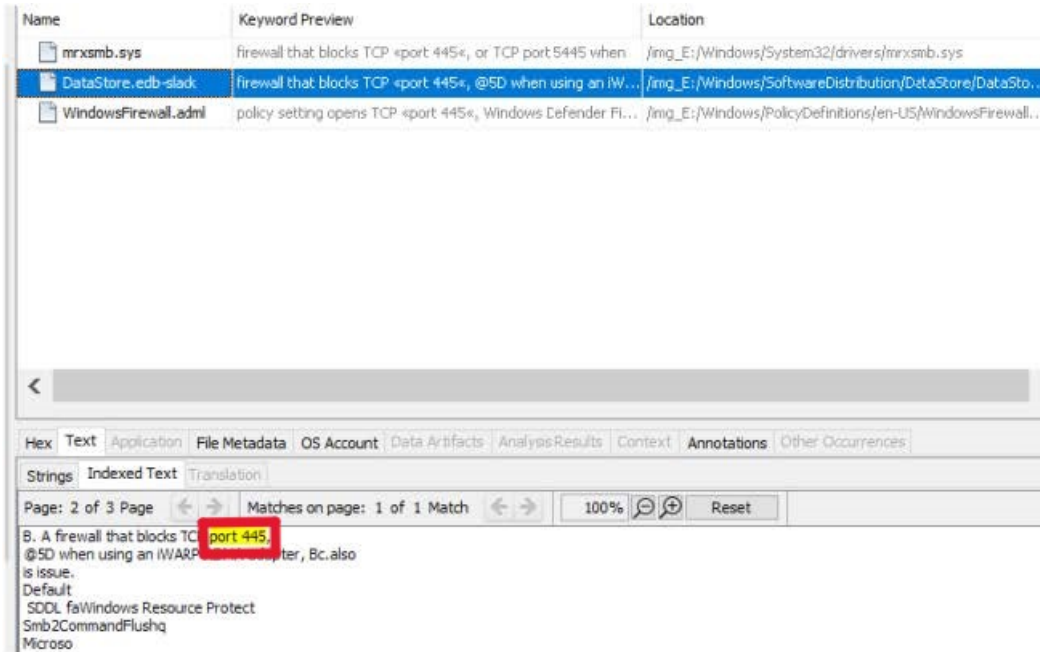


Ilustración 9 Puerto de conexión

2.9. Backdoor – Powershell maliciosa

Aplicación en la práctica:

A través de herramientas como Registry Explorer, FTK Imager o Autopsy, se analizó el contenido de archivos sospechosos o ubicaciones del sistema frecuentemente utilizadas para almacenar código malicioso (por ejemplo, rutas en AppData, tareas programadas, claves de autoejecución, etc.).

En uno de estos análisis se detectó un script de PowerShell sospechoso, que podría haber sido usado para:

Descargar y ejecutar payloads adicionales desde Internet.

Establecer comunicación remota con el atacante (por ejemplo, vía HTTP, TCP o WebSocket).

Crear persistencia en el sistema (por medio de tareas programadas o claves de ejecución automática).

Resultado obtenido: El script de la PowerShell es **WMIBackdoor.ps1**

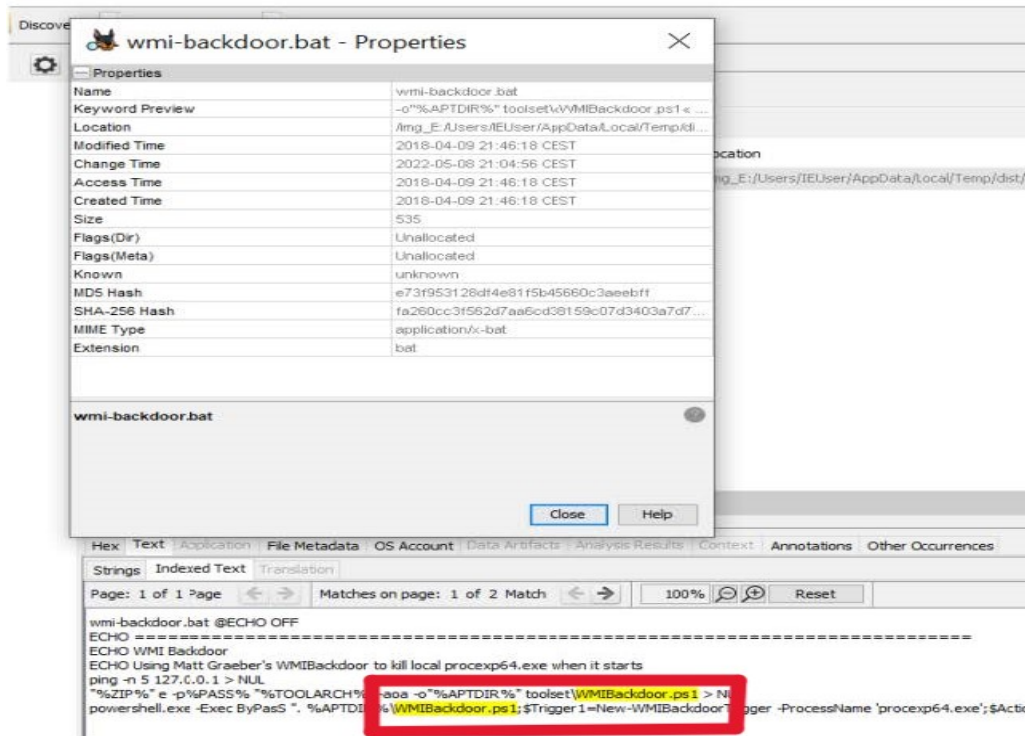


Ilustración 10 PowerShell maliciosa

2.10. Autopsy – Fecha descarga control remoto

Aplicación en la práctica:

En este caso, Autopsy fue utilizada para identificar la fecha exacta en la que se descargó o modificó un archivo relacionado con el software de control remoto utilizado por el atacante.

A partir del análisis de metadatos y de los registros de modificación (Modified Time, Accessed Time, Created Time), se observó que el fichero sospechoso fue modificado el: 08 de mayo de 2022 a las 20:53:32 CEST

Resultado obtenido: El fichero fue descargado el **2022-04-29**

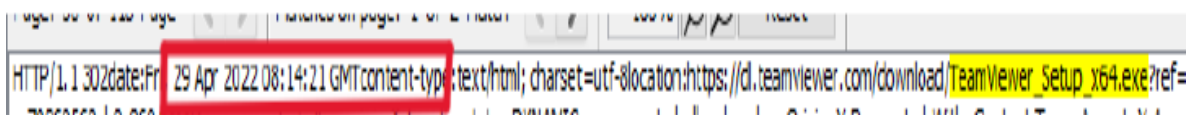


Ilustración 11 Fecha descarga fichero remoto

2.11. AccesData FTK Imager – Fecha ejecución control remoto

Aplicación en la práctica:

Durante el análisis con FTK Imager, se accedió a la ruta donde se encontraba el archivo asociado al software de control remoto, posiblemente descargado por el atacante.

Se examinaron los atributos del archivo, prestando especial atención al campo: Accessed Time (último acceso)

Resultado obtenido: La fecha de ejecución remoto fue el **29/04/2022**

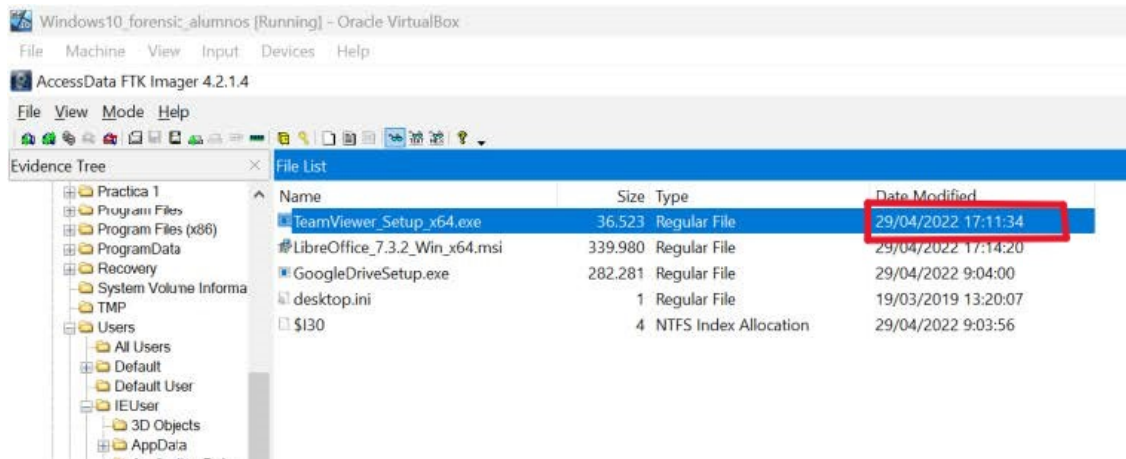


Ilustración 12 Fecha ejecución fichero remoto

2.12. Autopsy – Conexión control remoto

Aplicación en la práctica:

Mediante la búsqueda de palabras clave en Autopsy, se revisó el contenido del archivo: DataStore.edb

Este archivo pertenece al sistema operativo Windows y forma parte de la base de datos de actualizaciones, pero puede contener cadenas de texto de descripciones o fragmentos relevantes para la investigación.

Resultado obtenido: El ID del atacante es **765418952**

Hex	Text	Application	File Metadata	OS Account	Data Artifacts	Analysis Results	Context	Annotations	Other Occurrences
Strings Indexed Text Translation									
Page: 6 of 550 Page Matches on page: 1 of 2 Match 100% Reset									
!FILE0									
CONNEC~1.TXT									
Connections_incoming.txt									
765418952	WIN-MORENIN			29-04-2022 10:09:14	29-04-2022 10:10:10		IEUser		
emoteControl {adb13c9a-796c-438c-af8b-2079077f0a4f}									
765418952	WIN-MORENIN			29-04-2022 10:10:34	29-04-2022 10:13:21		IEUser	RemoteControl	
FILE0									

Ilustración 13 Id atacante remoto

3. METADATOS

Realice una prueba enviando una imagen por WhatsApp, Telegram, y correo electrónico y estos fueron los cambios y las conclusiones.

1. Envío por WhatsApp

Método: envío directo como imagen (no como archivo).

Resultado tras recepción:

La resolución se redujo automáticamente a 1600x1200.

Tamaño del archivo bajó a ~200 KB.

Todos los metadatos EXIF fueron eliminados (fecha, ubicación, cámara, etc.).

Nombre del archivo fue reemplazado por algo genérico como IMG-20250406-WA0001.jpg.

Conclusión: WhatsApp comprime la imagen y elimina todos los metadatos EXIF al enviarla como imagen. Para preservar los metadatos, debe enviarse como documento adjunto.

2. Envío por Telegram

Método: envío como documento (sin compresión).

Resultado tras recepción:

Resolución y tamaño intactos.

Todos los metadatos EXIF se conservaron.

Nombre de archivo intacto

Conclusión: Telegram preserva los metadatos si se envía como archivo/documento. Si se envía como foto, puede aplicar compresión y eliminar parte de los metadatos.

3. Envío por correo electrónico (Gmail)

Método: adjuntar como archivo.

Resultado tras recepción:

Archivo no modificado.

Todos los metadatos EXIF conservados.

Sin cambio de nombre ni compresión.

Conclusión: El correo electrónico es el método más seguro para preservar metadatos, ya que los archivos no se modifican automáticamente al adjuntarlos o descargarlos.

4. CONCLUSIÓN

Durante esta práctica de análisis forense digital se realizó una investigación estructurada sobre una imagen de disco (.vmdk) con el objetivo de identificar indicios de compromiso, técnicas utilizadas por un atacante y rastros de actividad maliciosa.

Mediante el uso de diversas herramientas especializadas como FTK Imager, Autopsy, Registry Explorer, RegRipper, Mimikatz y utilidades en línea como Crackstation.net, fue posible detectar múltiples artefactos relevantes. Se identificaron desde archivos eliminados, modificaciones en el registro del sistema, software de control remoto, hasta evidencias de conexiones RDP y uso de PowerShell malicioso.

Esta práctica evidenció la importancia de una correcta recolección y análisis de evidencias digitales para reconstruir cronológicamente los hechos y atribuir acciones maliciosas con precisión. Además, se comprobó el valor de combinar herramientas gráficas y manuales para contrastar datos y obtener resultados más completos y verificables.

5. BIBLIOGRAFIA

- ❖ Brian Carrier. The Sleuth Kit & Autopsy. <https://www.sleuthkit.org>
- ❖ AccessData. FTK Imager User Guide. <https://accessdata.com>
- ❖ Eric Zimmerman. Registry Explorer. <https://ericzimmerman.github.io>
- ❖ RegRipper Wiki. RegRipper Registry Analysis Tool. <https://github.com/keydet89/RegRipper3.0>
- ❖ Mimikatz Tool. Credential Extraction Utility by Benjamin Delpy. <https://github.com/gentilkiwi/mimikatz>
- ❖ CrackStation. Password Hash Cracker. <https://crackstation.net>
- ❖ Telegram Support. Sending files without compression. <https://telegram.org>
- ❖ WhatsApp Help Center. Sending media. <https://faq.whatsapp.com>
- ❖ RFC 3227. Guidelines for Evidence Collection and Archiving. IETF.
- ❖ OpenAI. ChatGPT, modelo de lenguaje basado en GPT-4. Consultado en abril de 2025 desde <https://chat.openai.com>