Clonación de la Memoria de un Pendrive

CIBERSEGURIDAD EN ENTORNOS DE LAS TECNOLOGÍAS DE LA INFORMACIÓN

MODULO ANALISIS FORENSE

ALUMNO: XINWEI WU

RESUMEN EJECUTIVO

Este informe documenta el proceso de clonación forense de un dispositivo USB (pendrive) en dos entornos: Linux (SIFT Workstation) y Windows 10 Professional. El objetivo principal fue garantizar la integridad de la evidencia digital mediante:

- Prevención del montaje automático para evitar modificaciones no deseadas.
- Clonación bit-a-bit utilizando herramientas estándar (dc3dd, dd, FTK Imager).
- Validación con hashes criptográficos (SHA-256 y SHA-512) para verificar la autenticidad de las imágenes.

Hallazgos clave:

- En Linux, las clonaciones con dc3dd y dd produjeron hashes idénticos, confirmando una réplica exacta.
- En Windows, la imagen generada con FTK Imager mostró hashes diferentes, sugiriendo posibles alteraciones por montaje automático o manejo de sectores.
- Los métodos para evitar el montaje automático fueron más efectivos en Linux (udisks2, reglas udev) que en Windows (diskpart, directivas de grupo).

1. Introducción

La preservación de la integridad de la evidencia digital constituye un pilar fundamental en el ámbito de la informática forense. En este contexto, la clonación forense de dispositivos de almacenamiento representa un procedimiento crítico cuyo objetivo primordial es garantizar la exactitud y autenticidad de los datos adquiridos, asegurando que no se produzcan modificaciones no autorizadas durante el proceso de adquisición.

El presente informe documenta de manera exhaustiva el proceso de clonación forense de un dispositivo USB (pendrive) en dos entornos operativos distintos: Linux (SIFT Workstation) y Windows 10 Professional. La elección de estos sistemas responde a la necesidad de evaluar las metodologías y herramientas disponibles en cada plataforma, así como su eficacia en la prevención del montaje automático, un fenómeno que puede comprometer la validez de la evidencia al permitir alteraciones no controladas en los metadatos o el contenido del dispositivo.

1.1. Marco Teórico y Relevancia Forense

En el ámbito de la informática forense, la clonación bit-a-bit (también conocida como imagen forense) se erige como el estándar de oro para la adquisición de evidencia digital. Este proceso implica la copia exacta y sector por sector del dispositivo de origen, incluyendo tanto los espacios asignados como los no asignados, lo que permite preservar integralmente la estructura del sistema de archivos y los datos potencialmente ocultos o eliminados.

La validación de la integridad de la imagen forense se realiza mediante funciones hash criptográficas (SHA-256, SHA-512), las cuales generan un identificador único para el conjunto de datos. Cualquier mínima alteración en el contenido del dispositivo o en la imagen resultante producirá un valor hash radicalmente distinto, lo que convierte a estos algoritmos en herramientas indispensables para garantizar la cadena de custodia.

No obstante, un desafío recurrente en este proceso es el montaje automático del dispositivo por parte del sistema operativo. Cuando un pendrive se monta automáticamente, el sistema puede escribir metadatos (timestamps, registros de acceso) o incluso alterar estructuras del sistema de archivos (como el journaling en NTFS o ext4), lo que invalida el principio de no alteración de la evidencia. Por ello, resulta imperativo implementar mecanismos que prevengan este comportamiento antes de realizar la clonación.

2. Metodología

2.1. Entorno Utilizado

2.1.1. Hardware

- Pendrive analizado:
    - Marca/Modelo: Verbatim STORE N GO
    - Capacidad: 16 GB (14.46 GiB útiles).
    - Particiones:
        - /dev/sdb1: 2 GB (NTFS).
        - /dev/sdb2: 12.5 GB (NTFS).
    - Número de serie: 072117102605FE61 (obtenido con dmesg).

2.1.2. Software

1. Linux (SIFT Workstation)

- **Distribución:** Ubuntu-based (SIFT v3.0).
- **Kernel:** Linux 5.4.0-xx-generic.
- **Herramientas forenses utilizadas**:
    - **Detección/Montaje:**
        - **dmesg:** Monitoreo de conexión USB.
        - **fdisk -l:** Listado de particiones.
        - **mount:** Verificación de puntos de montaje.
    - **Clonación:**
        - **dc3dd:** Versión 7.2.646 (con soporte para hashes integrados).
        - **dd**: Versión coreutils 8.30 (opción status=progress para monitoreo).
    - **Configuración del sistema:**
        - **udev:** Versión 245.4-4ubuntu3.
        - **udisks2:** Versión 2.8.4-1ubuntu2.

2. Windows 10 Professional

- **Versión del SO:** Windows 10 Pro 21H2 (Build 19044.2006).
- **Herramientas forenses utilizadas:**
    - **FTK Imager:** Versión 4.7.3 (formato RAW .dd).
    - **PowerShell:** Comando Get-FileHash para validación (SHA-256/SHA-512).
    - **Configuración del sistema:**
        - **diskpart:** Versión 10.0.19041.3636.
        - **Editor de directivas de grupo:** gpedit.msc.

2.2. Pasos Seguidos

2.2.1. En Linux

2.2.1.1. Verificación de montaje automático con dmesg, fdisk -l, y mount.

Comandos:

- “dmesg | tail -20” - Detecta dispositivos USB recién conectados.
- “sudo fdisk -l” - Lista particiones (ver /dev/sdb).
- “mount | grep sd” - Verifica si está montado

Comando “dmesg | tail -20”:

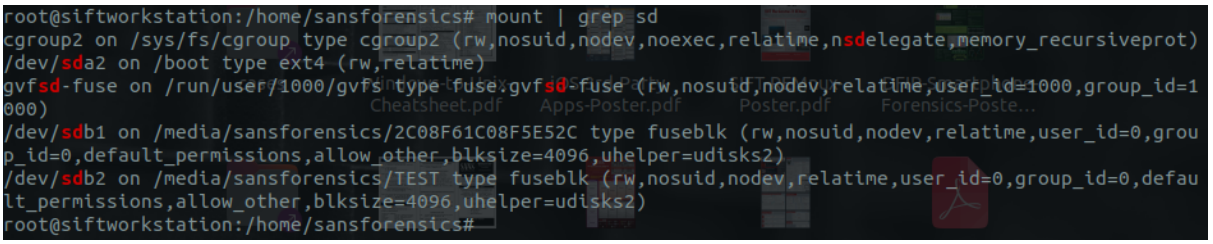
```
root@siftworkstation:/home/sansforensics# dmesg | tail -20
[   69.153595] usb-storage 2-1:1.0: USB Mass Storage device detected
[   69.154624] scsi host3: usb-storage 2-1:1.0
[   69.154827] usbcore: registered new interface driver usb-storage
[   69.165004] usbcore: registered new interface driver uas
[   69.717237] cups-proxyd[3754]: segfault at 18 ip 000055b5af91dd75 sp 00007fff37f2e850 error 4 in cups
-proxyd[55b5af91a000+7000]
[   69.717270] Code: 83 3d ee b2 00 00 00 41 54 55 48 89 fd 53 0f 85 f4 00 00 00 48 8d 1d 69 3d 00 00 48
 63 45 1c 48 89 df 48 c1 e0 05 48 03 45 08 <48> 8b 50 18 8b 70 14 e8 0f d0 ff ff 44 8b 65 18 48 89 c7 45
 85 e4
[   70.747197] scsi 3:0:0:0: Direct-Access     Verbatim STORE N GO        PMAP PQ: 0 ANSI: 4
[   70.748277] sd 3:0:0:0: Attached scsi generic sg2 type 0
[   70.763321] sd 3:0:0:0: [sdb] 30322688 512-byte logical blocks: (15.5 GB/14.5 GiB)
[   70.766709] sd 3:0:0:0: [sdb] Write Protect is off
[   70.766711] sd 3:0:0:0: [sdb] Mode Sense: 23 00 00 00
[   70.770125] sd 3:0:0:0: [sdb] No Caching mode page found
[   70.770134] sd 3:0:0:0: [sdb] Assuming drive cache: write through
[   70.794470]  sdb: sdb1 sdb2
[   70.807909] sd 3:0:0:0: [sdb] Attached SCSI removable disk
[   71.710847] ntfs3: Max link count 4000
[   71.710851] ntfs3: Enabled Linux POSIX ACLs support
[   71.710852] ntfs3: Read-only LZX/Xpress compression included
[   71.711727] ntfs3: Unknown parameter 'windows_names'
[   71.712465] ntfs3: Unknown parameter 'windows_names'
root@siftworkstation:/home/sansforensics#
```

Comando: “sudo fdisk -l”:

```
Disk /dev/sdb: 14.46 GiB, 15525216256 bytes, 30322688 sectors
Disk model: STORE N GO
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xcecf16d8

Device     Boot   Start      End  Sectors  Size Id Type
/dev/sdb1          2048  4196351  4194304    2G  7 HPFS/NTFS/exFAT
/dev/sdb2       4196352 30320639 26124288 12.5G  7 HPFS/NTFS/exFAT
root@siftworkstation:/home/sansforensics#
```

Comando: “mount | grep sd”:



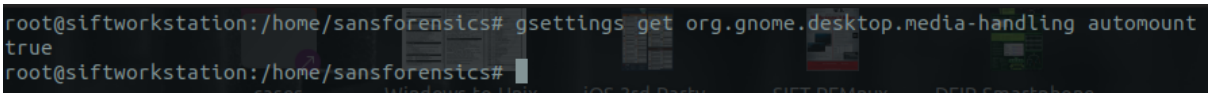
Está montado según los comandos anteriores.

2.2.1.2. Evitar el montaje automático

Método A: Desactivar automontaje con gsettings

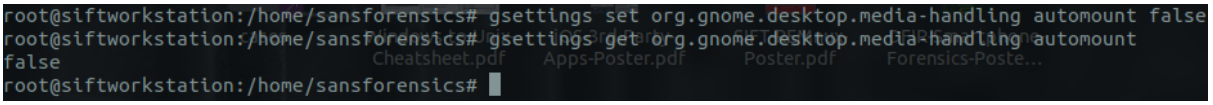
Identificamos el entorno con:

“gsettings get org.gnome.desktop.media-handling automount”



Se observa en la salida, que sale true el automontaje.

Lo desactivamos con:
gsettings set org.gnome.desktop.media-handling automount false



Y verificamos con el primer comando y vemos que nos sale false el automontaje. Pero aun así sigue montado.

A pesar de haber desactivado el automontaje mediante gsettings, el sistema operativo continuó montando el dispositivo USB. Esto se debe a que gsettings solo afecta al entorno gráfico (GNOME/MATE), mientras que el montaje automático también está controlado por servicios de bajo nivel como udisks2 y reglas udev

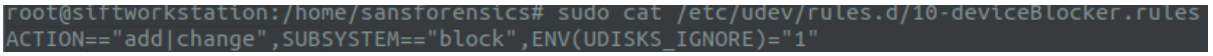
Método B: Configurar reglas udev

1. **Crear o edita el archivo:**

sudo nano /etc/udev/rules.d/10-deviceBlocker.rules

2. **Añadir las siguientes lineas:**

ACTION=="add|change",SUBSYSTEM=="block",ENV(UDISKS_IGNORE)="1"
SUBSYSTEM=="usb",ENV(UDISKS_AUTO)="0"



Funcionamiento de la regla ACTION=="add|change", SUBSYSTEM=="block", ENV{UDISKS_IGNORE}="1"

- ACTION=="add|change": Se activa cuando:

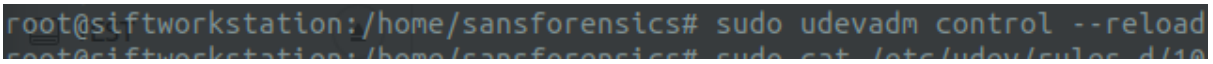
    - Un dispositivo es conectado ("add")
    - O cuando su estado cambia ("change")
- SUBSYSTEM=="block": Aplica solo a dispositivos de bloque (discos, particiones)
- ENV{UDISKS_IGNORE}="1": Establece una variable de entorno que indica al servicio udisks2 que debe ignorar este dispositivo

Efecto esperado:
- El sistema detecta el dispositivo pero evita:
    - Montaje automático
    - Interacción mediante interfaces gráficas
    - Creación de iconos en el escritorio

Funcionamiento de la regla:
- **SUBSYSTEM=="usb", ENV{UDISKS_AUTO}="0"**
    - SUBSYSTEM=="usb": Aplica específicamente a dispositivos USB
    - ENV{UDISKS_AUTO}="0": Desactiva las funciones automáticas de udisks2 para:
        - Montaje automático
        - Sondear sistemas de archivos
        - Notificar a otras aplicaciones

3. **Recargar las reglas con:**

sudo udevadm control --reload

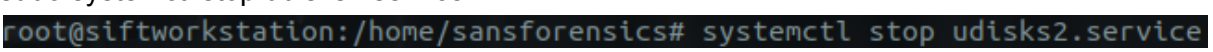


A pesar de esta configuración, se observó que el sistema continuaba montando automáticamente el dispositivo, lo que sugiere que:
- El servicio udisks2 podría estar ignorando estas variables en esta versión/distribución
- Existen mecanismos alternativos de montaje no controlados por estas reglas
- Otras reglas del sistema (en /lib/udev/rules.d/) tienen mayor prioridad"

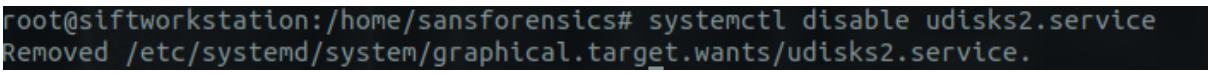
Y tras verificar que las reglas UDEV estaban correctamente configuradas pero no surtían efecto, se determinó - en línea con las advertencias del documento - que este método no era efectivo en esta configuración particular del sistema, procediendo a utilizar alternativas más robustas

Método C: mediante la desactivación del servicio udisks2

Detener el servicio udisk2:
sudo systemctl stop udisks2.service



Desactivarlo para que no se reinicie:
sudo systemctl disable udisks2.service



Verificar que el pendrive no se monte automáticamente con el comando “sudo fdisk -l”se ve que el sistema lo detecta.

```
Disk /dev/sdb: 14.46 GiB, 15525216256 bytes, 30322688 sectors
Disk model: STORE N GO
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x8c63319a
```

Y con el comando mount | grep sd, se ve que no está montado porque no aparece el /dev/sdb.

```
root@siftworkstation:/home/sansforensics# mount | grep sd
cgroup2 on /sys/fs/cgroup type cgroup2 (rw,nosuid,nodev,noexec,relatime,nsdelegate,memory_
recursiveprot)
/dev/sda2 on /boot type ext4 (rw,relatime)
gvfsd-fuse on /run/user/1000/gvfs type fuse.gvfsd-fuse (rw,nosuid,nodev,relatime,user_id=1
000,group_id=1000)
```

Nota:
Si aparece en mount, está montado. Si solo aparece en fdisk -l, está detectado pero no montado.

2.2.1.3. Obtener el número de serie del pendrive

Se obtiene mediante el siguiente comando:

dmesg | tail -20

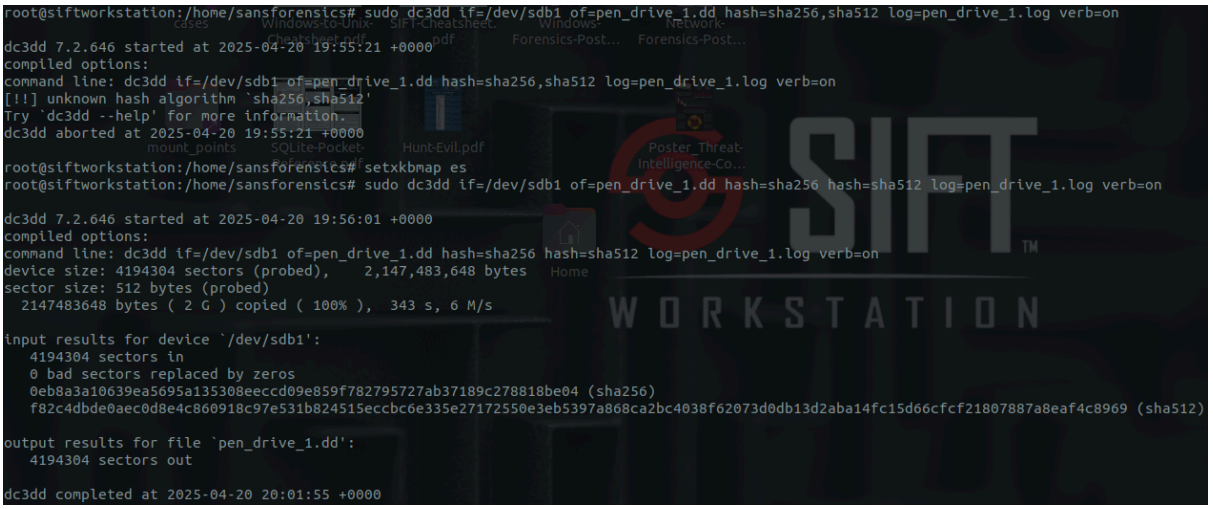
```
root@siftworkstation:/home/sansforensics# dmesg | tail -20
[   71.711727] ntfs3: Unknown parameter 'windows_names'
[   71.712465] ntfs3: Unknown parameter 'windows_names'
[  440.963288] usb 2-1: USB disconnect, device number 2
[  445.821425] usb 2-1: new high-speed USB device number 3 using ehci-pci
[  446.173317] usb 2-1: New USB device found, idVendor=18a5, idProduct=0302, bcdDevice= 1.00
[  446.173334] usb 2-1: New USB device strings: Mfr=1, Product=2, SerialNumber=3
[  446.173339] usb 2-1: Product: STORE N GO
[  446.173344] usb 2-1: Manufacturer: Verbatim
[  446.173346] usb 2-1: SerialNumber: 0721171D2605FE61
[  446.178503] usb-storage 2-1:1.0: USB Mass Storage device detected
[  446.178681] scsi host3: usb-storage 2-1:1.0
[  450.774479] scsi 3:0:0:0: Direct-Access     Verbatim STORE N GO       PMAP PQ: 0 ANSI: 4
[  450.775830] sd 3:0:0:0: Attached scsi generic sg2 type 0
[  450.791621] sd 3:0:0:0: [sdb] 30322688 512-byte logical blocks: (15.5 GB/14.5 GiB)
[  450.795572] sd 3:0:0:0: [sdb] Write Protect is off
[  450.795575] sd 3:0:0:0: [sdb] Mode Sense: 23 00 00 00
[  450.799044] sd 3:0:0:0: [sdb] No Caching mode page found
[  450.799047] sd 3:0:0:0: [sdb] Assuming drive cache: write through
[  450.822766]  sdb: sdb1 sdb2
[  450.840919] sd 3:0:0:0: [sdb] Attached SCSI removable disk
root@siftworkstation:/home/sansforensics#
```

2.2.1.4. Clonar el pendrive con dc3dd

Con el siguiente comando:
sudo dc3dd if=/dev/sdb1 of=pen_drive_1.dd hash=sha256 hash=sha512 log=pen_drive_1.log verb=on

Explicación de parámetros:

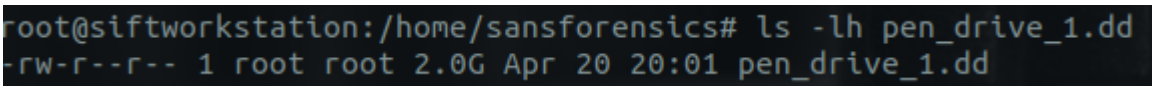
- if=/dev/sdb: Dispositivo de entrada (tu pendrive)
- of=pen_drive_1.dd: Archivo de salida (imagen forense)
- hash=sha256 hash=sha512: Cálculo de hashes de integridad
- log=pen_drive_1.log: Registro detallado del proceso
- verb=on: Modo verbose (muestra progreso)



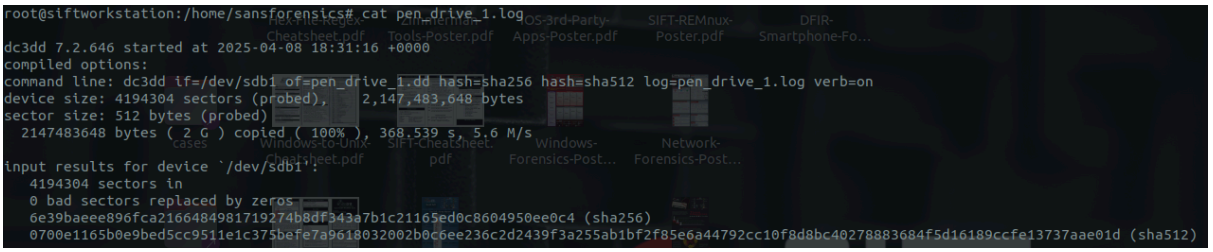
Hashes:
(sha256):0eb8a3a10639ea5695a135308eeccd09e859f782795727ab37189c278818be04

(sha512):
f82c4dbde0aec0d8e4c860918c97e531b824515eccbc6e335e27172550e3eb5397a868ca2bc4038f62073d0db13d2aba14fc15d66cfcf21807887a8eaf4c8969

Verificamos post clonación:
Verifica tamaño de la imagen:
ls -lh pen_drive_1.dd



Revisar nuevamente por si hiciera falta los hashes y errores:
cat pen_drive_1.log



2.2.1.5. Clonar el pendrive con dd

Clonar la partición sdb1 con dd:
sudo dd if=/dev/sdb1 of=pen_drive_2.dd conv=noerror status=progress

```
root@siftworkstation:/home/sansforensics# sudo dd if=/dev/sdb1 of=pen_drive_2.dd conv=noerror status=progress
2144977408 bytes (2.1 GB, 2.0 GiB) copied, 307 s, 7.0 MB/s
4194304+0 records in
4194304+0 records out
2147483648 bytes (2.1 GB, 2.0 GiB) copied, 307.32 s, 7.0 MB/s
```

Calcular hashes manualmente:

- **pen_drive_2(sha256):**
  0eb8a3a10639ea5695a135308eeccd09e859f782795727ab37189c278818be04

```
root@siftworkstation:/home/sansforensics# sha256sum pen_drive_2.dd
0eb8a3a10639ea5695a135308eeccd09e859f782795727ab37189c278818be04  pen_drive_2.dd
```

- **pen_drive_2(sha512)**
  f82c4dbde0aec0d8e4c860918c97e531b824515eccbc6e335e27172550e3eb5397a8
  8ca2bc4038f62073d0db13d2aba14fc15d66cfcf21807887a8eaf4c8969

```
root@siftworkstation:/home/sansforensics# sha512sum pen_drive_2.dd
f82c4dbde0aec0d8e4c860918c97e531b824515eccbc6e335e27172550e3eb5397a868ca2bc4038f62073d0db13d2aba14fc15d66cfcf21807887a8eaf4c8969  pen_drive_2.dd
```

2.2.1.6. Análisis de Resultados

1. **Coincidencia de hashes:**
   Los valores hash SHA-256 y SHA-512 son idénticos para ambas imágenes (pen_drive_1.dd y pen_drive_2.dd), lo que demuestra que:
   - La clonación fue exacta y bit-a-bit perfecta
   - No hubo alteraciones durante el proceso
   - Ambas herramientas (dc3dd y dd) produjeron resultados forensemente válidos
2. **Implicaciones forenses:**
   Esta coincidencia permite afirmar que:
   - Las imágenes son réplicas forensemente válidas del original
   - Pueden ser admitidas como evidencia digital
   - Cumplen con el principio de integridad en informática forense
3. **Conclusión técnica:**
   El hecho de que dos herramientas diferentes (dc3dd y dd) produzcan imágenes con hashes idénticos confirma que:
   - El dispositivo fuente no tuvo cambios entre clonaciones
   - No hubo errores de lectura
   - Los metadatos y contenido fueron preservados completamente

2.2.2. Clonación en Windows 10 Professional

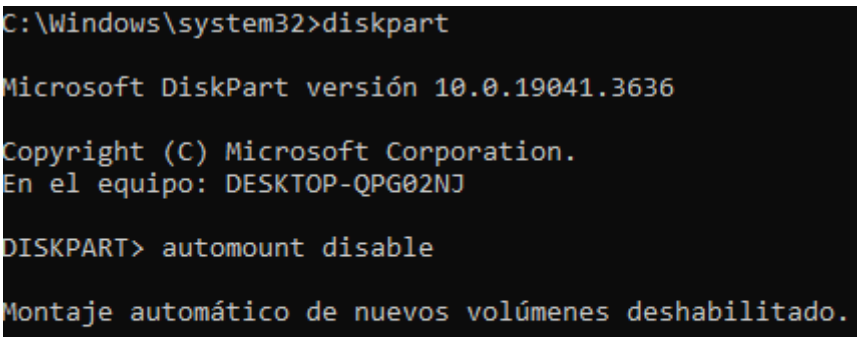
2.2.2.1. Métodos para evitar montaje automático

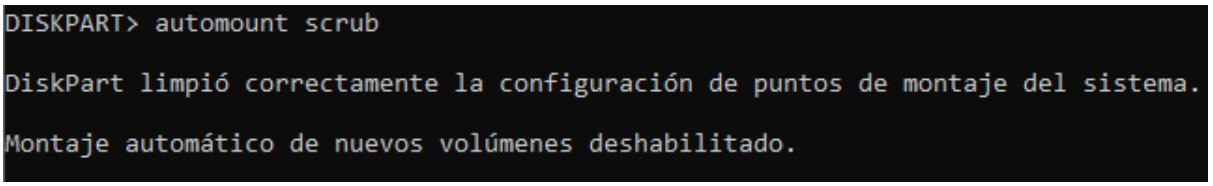
2.2.2.1.1. Usando diskpart:

Procedimiento realizado:

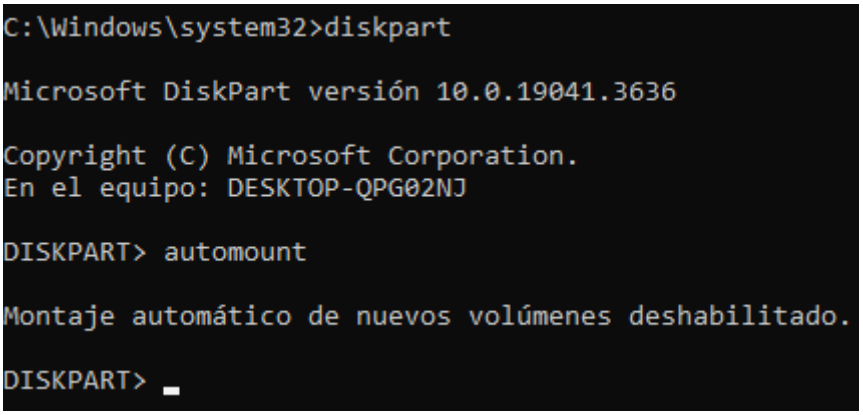
1. Se ejecuta CMD como administrador

2. Se inició DiskPart mediante el comando diskpart.
3. Se introdujo el comando automount disable.
    - **automount disable:** Evita que Windows asigne letras de unidad automáticamente.
4. Se introdujo el comando automount scrub
    - **automount scrub:** Elimina las letras de unidad persistentes de dispositivos anteriores.



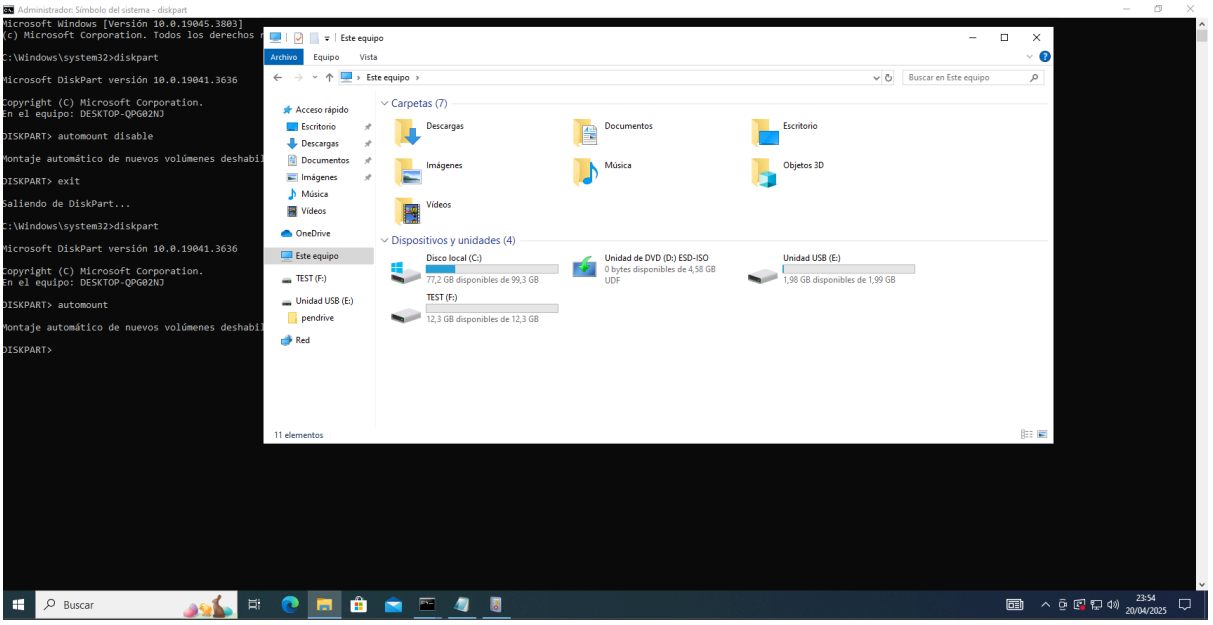


5. Se verificó el estado con automount, mostrando "Montaje automático de nuevos volúmenes deshabilitado"



Resultado observado:

- A pesar de la configuración aplicada, el dispositivo USB seguía montándose automáticamente al conectarlo
- El sistema asignaba automáticamente una letra de unidad y mostraba la interfaz de autoejecución



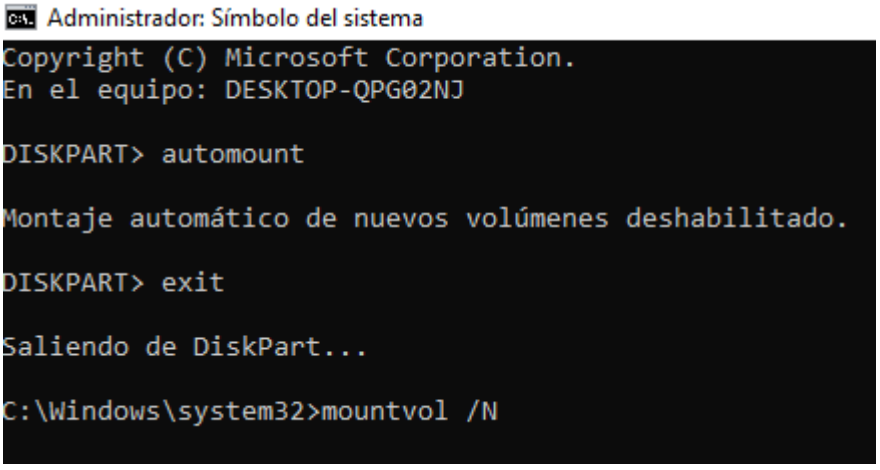
Análisis técnico:

El comportamiento observado indica que:

1. La directiva automount disable en Windows 10 no previene completamente el montaje de dispositivos USB
2. Existen múltiples capas en el subsistema de almacenamiento de Windows que pueden anular esta configuración:
    - Servicio "Shell Hardware Detection" (responsable de diálogos de autoejecución)
    - Configuraciones de directiva de grupo adicionales
    - Drivers específicos del fabricante del dispositivo

2.2.2.1.2 Usando mountvol:

Deshabilita el montaje automático de volúmenes y elimina letras de unidad asignadas previamente. Mediante el comando “mountvol /N”

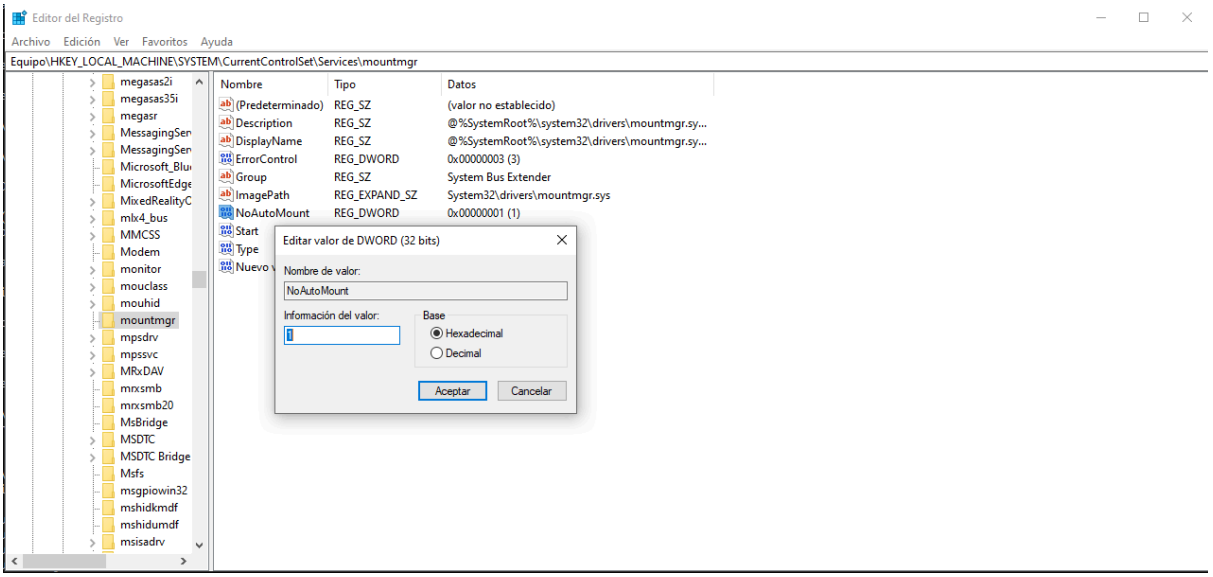


Durante la prueba, se observó que el comando mountvol /N no previno el montaje automático del pendrive, a pesar de ejecutarse con privilegios de administrador y reiniciar el sistema.

2.2.2.1.3. Registro de Windows:

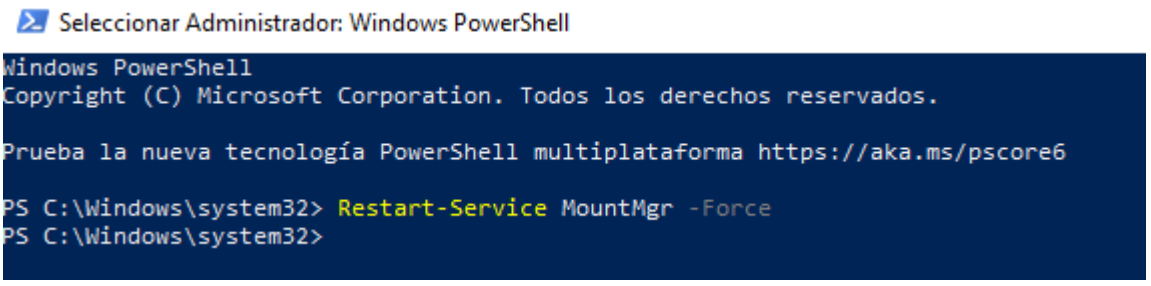
Modificar la clave:

1. Abrir el editor de registros de windows
2. Navega a: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MountMgr
3. Crea un valor DWORD llamado "NoAutoMount" con valor 1



Se modificó la clave del registro NoAutoMount siguiendo la documentación oficial, pero el sistema continuó montando automáticamente el dispositivo USB. Tras investigar, se identificó que este comportamiento puede deberse a:

- La presencia de políticas de grupo que anulan la configuración del registro.
- Un reinicio insuficiente del servicio MountMgr.



Aun reiniciando, el problema persiste, y puede que Windows esté ignorando esta configuración debido a políticas superiores

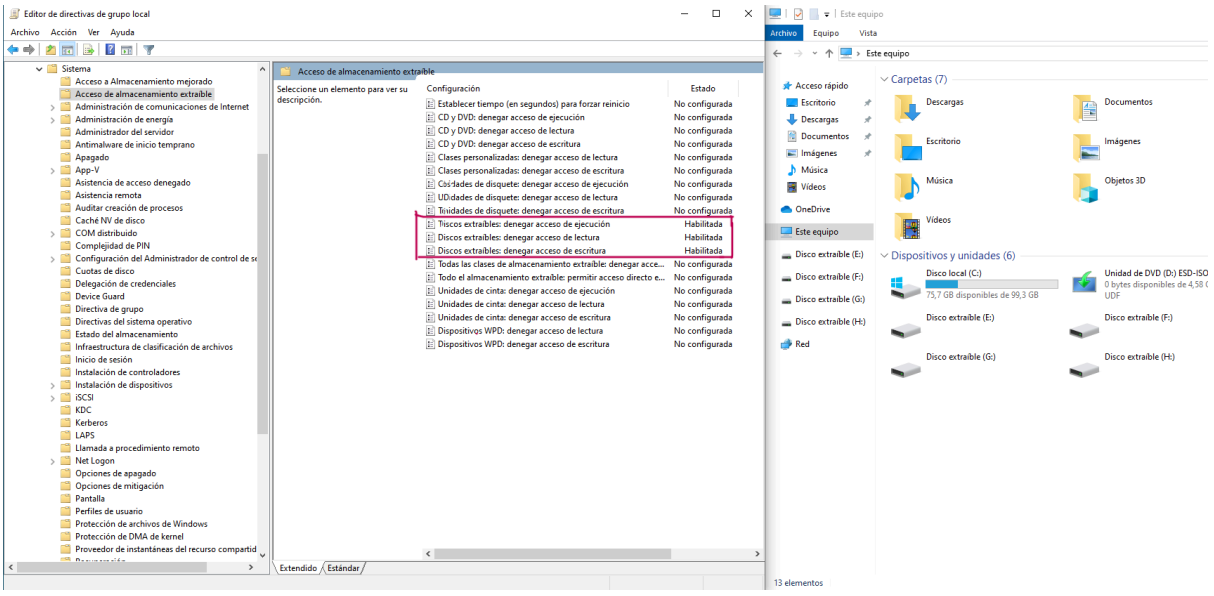
2.2.2.1.4. Directivas de grupo:

Configura políticas para bloquear acceso a dispositivos extraíbles.
Pasos:

1. Abrir el Editor de directivas de grupo local.
2. **Navegar a:**
   Configuración del Equipo > Plantillas Administrativas > Sistema > Acceso de almacenamiento extraíble
3. **Habilitar las tres políticas de "Almacenamiento extraíble" que son:**
   - **Discos extraíbles: denegar acceso de ejecución**. Impide que se ejecuten archivos directamente desde el pendrive.
   - **Discos extraíbles: denegar acceso de lectura.** Bloquea la lectura de datos en el pendrive, evitando que Windows lo monte automáticamente para explorar su contenido.
   - **Discos extraíbles: denigrar acceso de escritura.** Evita que se escriban datos en el pendrive, lo que ayuda a preservar la evidencia forense.
4. Para que sufra efecto usar el comando en el cmd como admin para que surja efecto: **gpupdate /force**

```
C:\Windows\system32>gpupdate /force
Actualizando directiva...

La actualización de la directiva de equipo se completó correctamente.
Se completó correctamente la Actualización de directiva de usuario.
```



Políticas clave:

Política	Efecto	Comando de aplicación
Denegar acceso de ejecución	Bloquea archivos .exe	gpupdate /force
Denegar acceso de lectura	Previene montaje para lectura	gpupdate /force
Denegar acceso de escritura	Protege contra modificaciones	gpupdate /force

Notas importantes:
Estas políticas no desactivan la detección del pendrive (aparecerá en el Administrador de dispositivos), pero evitarán que Windows lo monte o interactúe con él.

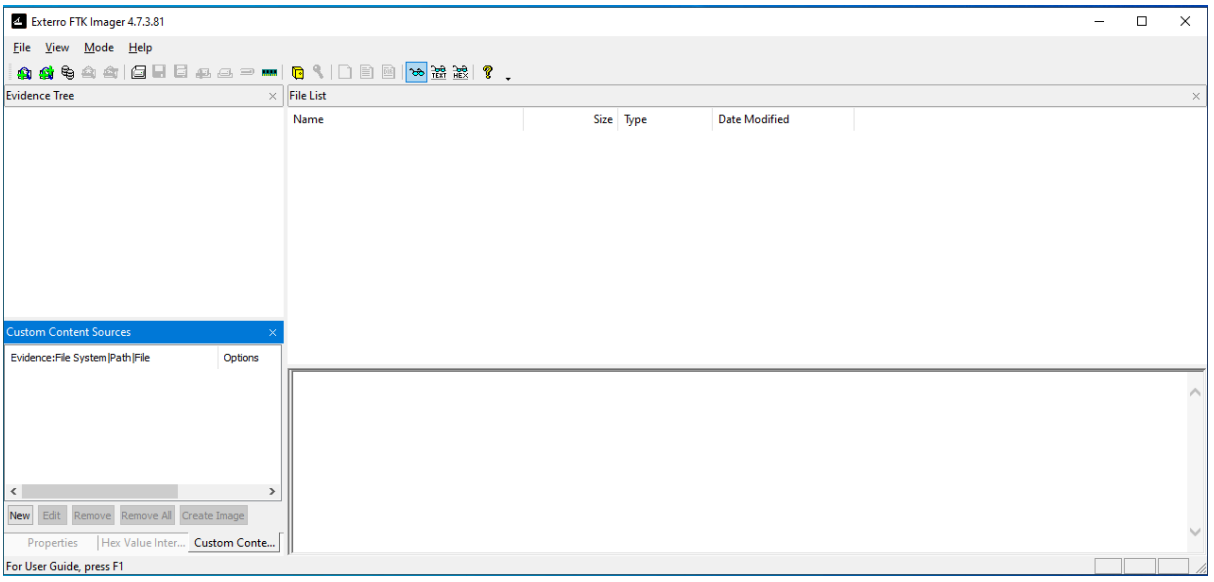
2.2.2.2. Clonación con FTK Imager

FTK Imager es una herramienta profesional de adquisición forense desarrollada por Exterro, que permite:

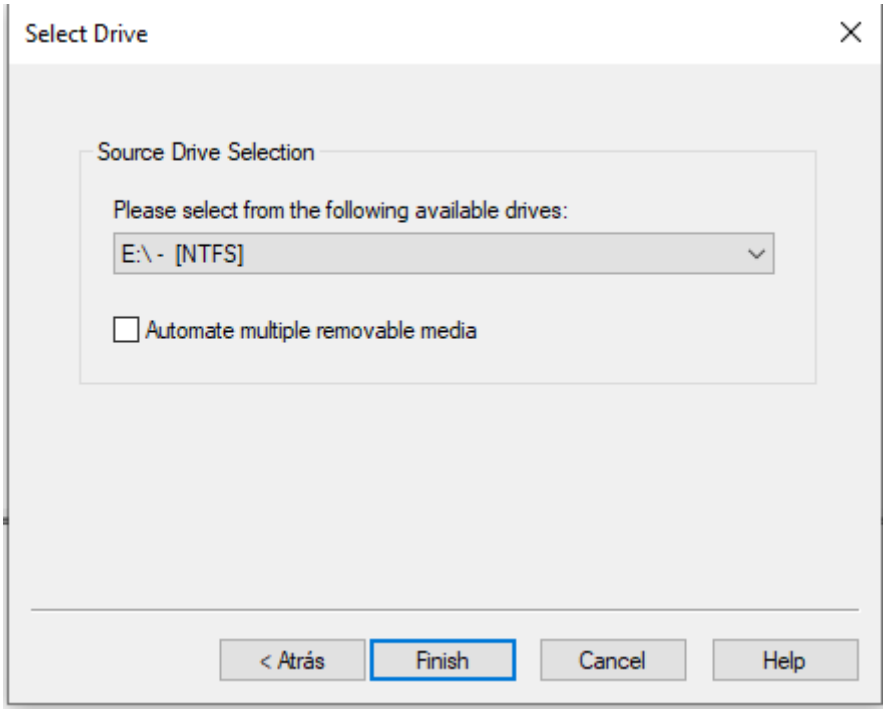
- Crear imágenes bit-a-bit de dispositivos de almacenamiento.
- Generar hashes criptográficos para verificación de integridad.
- Analizar evidencias sin alterar los datos originales

Pasos de la clonación con FTK Imager

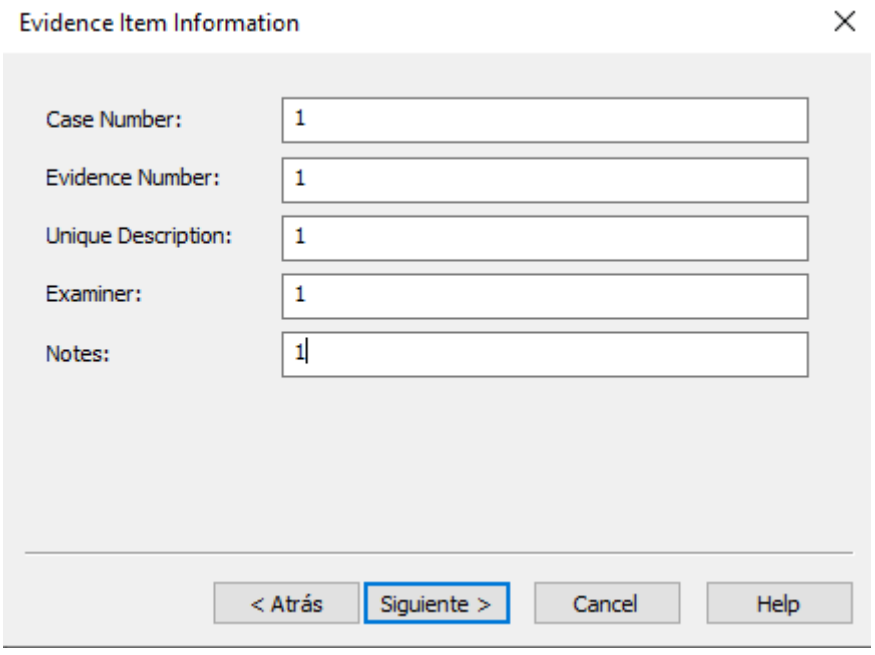
1. Descarga e instala FTK Imager v4.7.3.81
2. Preparación:
    - Ejecutar FTK Imager como administrador.
    - Conectar el dispositivo USB



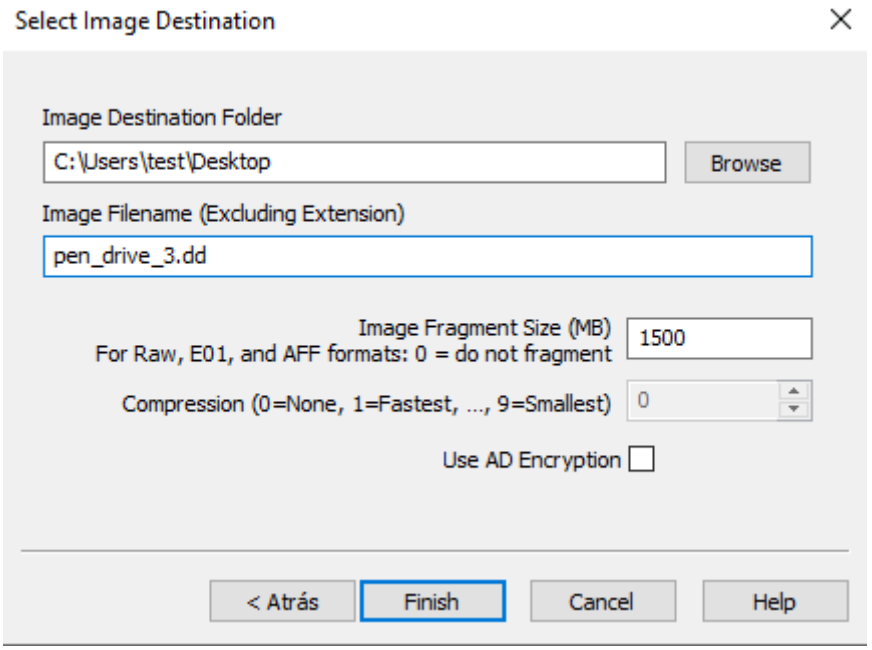
3. Ve a File > Create Disk Image > Logical Drive.
4. Seleccionamos la partición, en mi caso es la E y finalizamos:



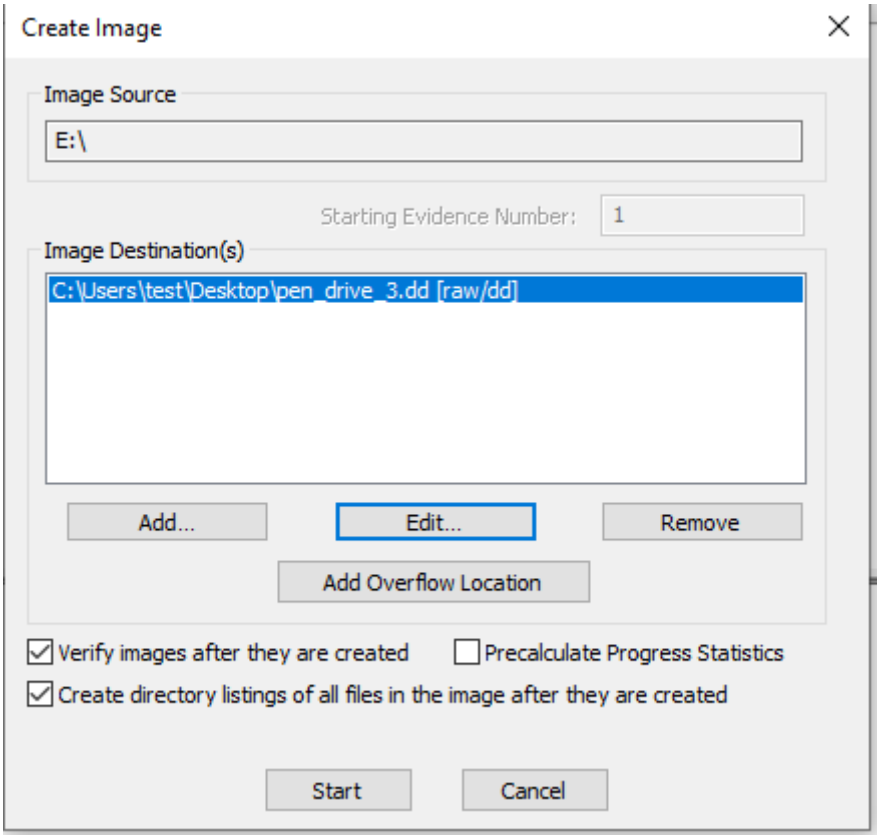
5. Después de finalizar, añadimos donde queremos guardar la imagen, el formato Raw (dd).
6. En Evidence Item Information, completar:
    a. **Número de caso:** Identificador único.
    b. **Descripción:** Ej: "Clonación forense pendrive Verbatim".
    c. **Examinador:** Nombre del responsable.
    d. **Notas:** Observaciones relevantes (ej: "USB bloqueado por políticas de grupo").



7. Elegimos la ruta en el que queremos guardar la imagen y nombre de la imagen y extensión (en mi caso, lo guardaré en el escritorio y con el nombre de pen_drive_3.dd). También elegimos el tamaño de la imagen, siempre superior a la partición para que al crear la imagen solo sea una en vez de dos imágenes.



8. Y procedemos a crear la imagen en “start”.



2.2.2.3. Validación de Integridad

Tras la clonación, calcular los hashes con Powershell:

2.2.2.3.1. pen_drive_3.dd (SHA512):

Con el siguiente comando en powershell:

Get-FileHash -Path "C:\Users\test\Desktop\pen_drive_3.dd.001" -Algorithm SHA256 | Format-List

Hash*:*

BCD9DD153B1C71D6D2C8D89D46E0895A66CC6FCB2ADC57D846F94C4471BA36FD

2.2.2.3.2. pen_drive_3.dd (SHA512):

Con el siguiente comando en powershell:
Get-FileHash -Path "C:\Users\test\Desktop\pen_drive_3.dd.001" -Algorithm SHA512 | Format-List

Hash:

609DD3BA82E724BBB9FC008B95262E12CD1C80AB17961BF074DF5F03C4DFBB855C9
3978239EE016FEB87B7CBFF8EC0C0BF5FDDF4B5F33C6BDDB4E1CDF76A2B14.

3. Conclusiones y Recomendaciones

3.1. Conclusiones y Recomendaciones

Durante la práctica, se realizaron múltiples clonaciones del mismo pendrive utilizando diferentes herramientas y sistemas operativos. Los resultados obtenidos permiten extraer las siguientes conclusiones:

3.1.1. Coincidencia de Hashes en Linux

Coincidencia de Hashes en Linux

Herramientas utilizadas:

- **dc3dd:** Generó pen_drive_1.dd con hashes SHA-256 y SHA-512 integrados.
- **dd:** Generó pen_drive_2.dd, con hashes calculados manualmente.

Resultados:

- SHA-256:
  0eb8a3a10639ea5695a135308eeccd09e859f782795727ab37189c278818be04

- SHA-512:
  f82c4dbde0aec0d8e4c860918c97e531b824515eccbc6e335e27172550e3eb5397a86
  8ca2bc4038f620730d0b13d2aba14fc15d66cfcf21807887a8ea44c8969

Implicaciones forenses:

- Las imágenes son idénticas bit-a-bit.
- Validación exitosa del principio de no alteración de la evidencia.

Diferencia de Hashes en Windows (FTK Imager)

Imagen generada: pen_drive_3.dd.
Hashes obtenidos:

- **SHA-256:**
  BCD9DD153B1C71D6D2C8D89D46E0895A66CC6FCB2ADC57D846F94C4471BA3
  6FD

- SHA-512:
  609DD3BA82E724BBB9FC008B95262E12CD1C80AB17961BF074DF5F03C4DFBB
  855C93978239EE016FEB87B7CBFF8EC0C0BF5FDDF4B5F33C6BDDB4E1CDF76
  A2B14

Causas probables:

1. Montaje automático previo (alteración de metadatos).
2. FTK Imager no clonó sectores defectuosos de la misma forma que Linux.
3. Diferencias en el manejo de FAT32 entre sistemas operativos.

3.2. Comparación de Métodos para Evitar el Montaje Automático

Método	Sistema	Comando/Configuración	Efectividad
gsettings (GNOME)	Linux	gsettings set org.gnome.desktop.media-handling automount false	Baja
Reglas udev	Linux	`ACTION=="add change", SUBSYSTEM=="block", ENV(UDISKS_IGNORE)="1"`	Baja
Desactivar udisks2	Linux	sudo systemctl disable --now udisks2	Alta
diskpart	Windows	automount disable + automount scrub	Baja
mountvol /N	Windows	mountvol /N	Baja
Registro de Windows	Windows	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MountMgr\NoAutoMount = 1	Baja
Directivas de Grupo	Windows	Habilitar las 3 políticas de discos extraíbles en gpedit.msc	Alta

Conclusión:

- Linux ofrece mayor control mediante servicios de bajo nivel (udisks2).
- Windows requiere combinar métodos (ej: directivas de grupo + diskpart) para obtener resultados óptimos.

3.3 Recomendaciones Finales

Para Linux

Clonación:

- Usar dc3dd por su integración de hashes y logs.
- Ejemplo:
  sudo dc3dd if=/dev/sdb1 of=imagen.dd hash=sha256,sha512 log=registro.log

Prevención de montaje:

- Deshabilitar udisks2:
  sudo systemctl disable --now udisks2

Para Windows

Clonación:

- Usar FTK Imager en formato RAW (.dd) con verificación de hashes.

Prevención de montaje:

- Habilitar las tres políticas de discos extraíbles en gpedit.msc.
- Ejecutar gpupdate /force para aplicar cambios.

Buenas Prácticas Generales

- **Documentación:** Registrar todos los comandos, hashes y configuraciones.
- **Almacenamiento:** Guardar imágenes en medios externos (evitar sobrescritura).
- **Validación:** Comparar hashes entre herramientas/sistemas para detectar anomalías.

3.4 ¿Por qué a veces no coinciden los hashes de dos clonaciones del mismo dispositivo?

Causas principales:

- **Modificación accidental:** El sistema operativo alteró metadatos (timestamps, journals).
- **Herramientas diferentes:** dd (Linux) y FTK Imager (Windows) manejan sectores defectuosos de forma distinta.
- **Hardware:** Sectores dañados en el pendrive.

Conclusión forense:

  - Si los hashes coinciden, la clonación es válida.
  - Si difieren, se debe investigar si hubo errores en el proceso o contaminación de la evidencia.

Anexo Técnico: Comandos Utilizados en el Proceso Forense

Linux (SIFT Workstation)

Verificación de montaje automático:

- Detectar conexión USB: dmesg | tail -20
- Listar particiones: sudo fdisk -l
- Verificar puntos de montaje: mount | grep sd

Desactivar montaje automático:
Método A: gsettings (GNOME)

- Verificar estado: gsettings get org.gnome.desktop.media-handling automount
- Desactivar: gsettings set org.gnome.desktop.media-handling automount false

Método B: Reglas udev

- Añadir reglas: sudo nano /etc/udev/rules.d/10-deviceBlocker.rules
- Recargar reglas: sudo udevadm control --reload

Método C: Desactivar udisks2

- Detener servicio: sudo systemctl stop udisks2.service
- Deshabilitar permanentemente: sudo systemctl disable udisks2.service

Clonación forense

- Con dc3dd
    - sudo dc3dd if=/dev/sdb1 of=pen_drive_1.dd hash=sha256,sha512 log=pen_drive_1.log verb=on
- Con dd
- sudo dd if=/dev/sdb1 of=pen_drive_2.dd bs=4M status=progress conv=noerror

Validación de hashes

- sha256sum pen_drive_1.dd
- sha512sum pen_drive_2.dd

Windows 10 Professional

Desactivar montaje automático
Diskpart:

- diskpart
- automount disable
- automount scrub

mountvol

- mountvol /N

Registro de Windows:

- Clave: HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\MountMgr
- # Valor DWORD: "NoAutoMount" = 1

Directivas de Grupo
gpedit.msc : Habilitar políticas de discos extraíbles
- gpupdate /force # Aplicar cambios

Clonación con FTK Imager
Pasos gráficos:
File > Create Disk Image > Logical Drive > Seleccionar partición > RAW (.dd)

Validación de hashes (PowerShell)
- Get-FileHash -Path "C:\ruta\pen_drive_3.dd" -Algorithm SHA256 | Format-List
- Get-FileHash -Path "C:\ruta\pen_drive_3.dd" -Algorithm SHA512 | Format-List

Notas Clave

- **Linux:**
    - udisks2 es el método más efectivo para evitar montaje.
    - dc3dd incluye hashes integrados (recomendado).
- **Windows:**
    - Las Directivas de Grupo son la opción más robusta (requiere edición Pro/Enterprise).
    - mountvol /N y el Registro mostraron baja efectividad en pruebas.