Alumno: Dragos Cornel Iván Cornel

Actividad 2: Adquisición de evidencias digitales

-Preparación de las herramientas forenses: Conviene **formatear nuestro USB-herramienta de trabajo en formato NTFS**, y asegurarnos de que no estén las siguientes carpetas: en **Windows** eliminar System Volume Information, $RECYCLE.BIN, desktop.ini, autorun.inf.
En Linux/macOS eliminar.Trash-*, .fseventsd, .Spotlight-V100, .DS_Store, autorun.inf.

Justo antes de manipular cualquier evidencia, y tras detectar qué dispositivos contienen evidencia potencial, tendremos que identificar y etiquetar las pruebas: en mi caso un (1) USB, y un (1) ordenador al que le voy a extraer la RAM y la imagen de disco, según petición del ejercicio. De modo que se **etiquetan los dispositivos**, físicamente y/o digitalmente:

- Se coloca una etiqueta física con: Número de caso, fecha/hora de adquisición, iniciales del responsable, descripción breve del dispositivo.
- Se registra en una **hoja de cadena de custodia**

-También se crean **hashes** (SHA-256, etc.) de la evidencia para asegurar su integridad.

-Desactivamos montaje automático de USB y bloqueamos contra escritura:

-En Linux, siendo mi caso, creamos una nueva regla en nano con: sudo nano /etc/udev/rules.d/100-no-automount.rules, y escribimos y guardamos: ACTION=="add", SUBSYSTEM=="block", ENV{UDISKS_IGNORE}="1", y finalmente udevadm control --reload. El paso final es conectar el USB a analizar y montarlo solo lectura desde terminal, con sudo mount -o ro /dev/sdX1 /mnt/usb, donde “sdX1” es nuestra partición que podremos averiguar con lsblk -f.

-En Windows desde CMD como admin el comando sería: reg add "HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer" /v NoDriveTypeAutoRun /t REG_DWORD /d 255 /f para desactivar el montaje automático de USB, y este otro comando: reg add "HKLM\SYSTEM\CurrentControlSet\Control\StorageDevicePolicies" /v WriteProtect /t REG_DWORD /d 1 /f para establecer modo solo lectura del USB.

PARTE 1: Adquisición en frío

Para esta parte se ha usado el propio equipo con sistema operativo Ubuntu (como herramienta de análisis), y un pendrive (la prueba a analizar). La herramienta usada es **GuyMager**.

Establecido esto, el primer paso sería verificar el sistema de archivos con el que está formateado el pendrive. En **Linux** en mi caso lo haría con lsblk -f, y en Windows wmic logicaldisk get name,filesystem,volumename,size desde CMD, o con Get-Volume | Format-Table -AutoSize desde PowerShell. La salida en mi caso sería:

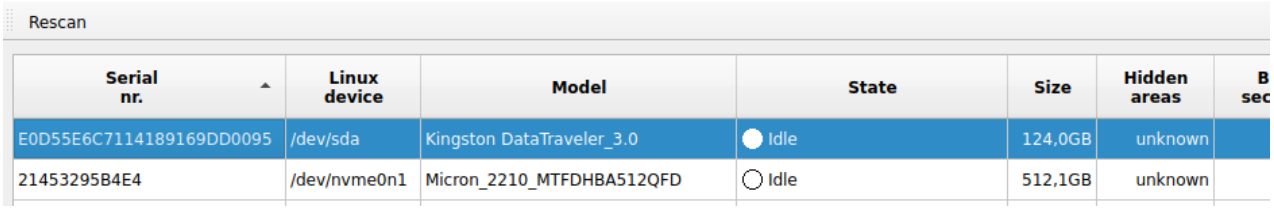
```
sda
└─sda1
    ntfs             B4723A3C723A0420                    32,1G    72% /media/chocolate/B4723A3C723A0420
```

Observamos que mi **USB** sería **sda**, la partición ocupada con datos sería **sda1**, formateada en **NTFS**, con el identificador único del sistema de archivos de una partición **UUID** que vemos en la imagen, seguido del espacio libre y el porcentaje utilizado del USB.

Otras opciones serían dmesg o dmesg | tail -20 para identificar el nombre del dispositivo. Abajo se observa la salida del disco sin montar.

```
sda           8:0    1    15G  0 disk
└─sda1        8:1    1    15G  0 part
```

-Abrimos GuyMager con sudo guymager. Seleccionamos nuestro USB, luego **click derecho, adquirimos imagen** y rellenamos con los datos de nuestro caso.



-He seleccionado el pendrive y elegido "crear imagen": en este caso opté por "**linux dd raw image**". Sin embargo, para un caso real me hubiera planteado la opción "**Expert Witness**" con la extensión **Encase** para mayor profesionalidad y adaptabilidad.

-Al final del análisis he guardado los hashes generados para validación futura.

```
SHA256 hash                : 50f8d3ca4203160ee4a946bf5becc28f684dc0f729229f421293c96a4e7c5f8a
SHA256 hash verified source: 50f8d3ca4203160ee4a946bf5becc28f684dc0f729229f421293c96a4e7c5f8a
SHA256 hash verified image : 50f8d3ca4203160ee4a946bf5becc28f684dc0f729229f421293c96a4e7c5f8a
Source verification OK. The device delivered the same data during acquisition and verification.
Image verification OK. The image contains exactly the data that was written.

Acquisition started : 2025-05-21 18:15:19 (ISO format YYYY-MM-DD HH:MM:SS)
Verification started: 2025-05-21 18:30:34
Ended               : 2025-05-21 18:45:09 (0 hours, 29 minutes and 49 seconds)
Acquisition speed   : 16.81 MByte/s (0 hours, 15 minutes and 14 seconds)
Verification speed  : 17.55 MByte/s (0 hours, 14 minutes and 35 seconds)
```

Así, en esta imagen se observan los **hashes** de la imagen: el **principal**, el hash de la **imagen fuente** verificada, y el hash de la **imagen verificada**. Se observan además los tiempos y la duración del proceso.

Es importante no modificar nunca la evidencia original. De modo que clonamos bit a bit la imagen original 2 veces y creamos hashes de cada clon y original. Trabajamos sobre una de las copias, y si volvemos a necesitar otra copia volvemos sobre la clon nr. 1, intentando tocar la imagen original lo menos posible.

-Anotar todo en un "registro de cadena de custodia" (nombre del operador, acciones realizadas, lugar, versiones de las herramientas forenses, etc.). En mi caso:

Versión herramienta forense **GuyMager: 0.8.13-1**
Versión timestamp : **2021-08-13-12.57.42 UTC**; Compilado con: **gcc 11.2.0**. Host name: **intel**
System: Linux intel 6.8.0-59-generic #61~22.04.1-**Ubuntu** SMP PREEMPT_DYNAMIC T**ue Apr 15 17:03:15 UTC** 2 x86_64

PARTE 2: **Adquisición en caliente de la memoria RAM:** herramienta **DumpIt**.

-Descargamos el .exe en: https://www.toolwar.com/2014/01/dumpit-memory-dump-tools.html
-Guardamos la herramienta en nuestro pendrive de trabajo forense. Si observamos las propiedades de esta veremos qué poca memoria ocupa. Esto es clave para sobrescribir lo menos posible la RAM a analizar.
-Conectamos pendrive al equipo donde vamos a extraer la RAM, **asegurando evitar** su montaje automático, como hemos mencionado. **Creamos el directorio** que será el punto de montaje del USB, y lo haremos en la RAM del equipo a analizar (/tmp/), con el comando mkdir /tmp/usb-forense, y luego con mount -o ro /dev/sdX1 /tmp/usb-forense (Linux) montamos solamente en modo lectura. Desde ahí ejecutamos la herramienta dumpit.exe como administrador.
NOTA: Para un caso real, es más recomendable usar USB booteable con Sistema Operativo CAINE.
-Ejecutamos nuestra herramienta forense, y en la siguiente pestaña aceptamos. Y al cabo de pocos minutos tendremos nuestro archivo .raw en la misma carpeta del pendrive, en mi caso **disco D:**.

```
* Destination = \??\D:\forensics\RAM\dumpIt\DESKTOP-3VU71PH-20250522-132035.raw

--> Are you sure you want to continue? [y/n] y
+ Processing... Success.
```

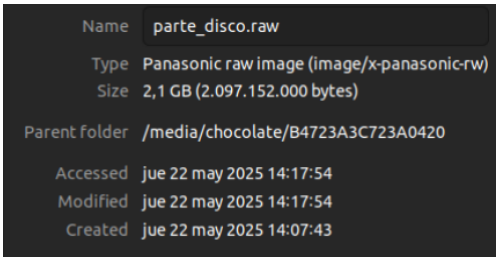
Abajo la RAM extraída directamente en nuestro pendrive (*/media*), el hash de esta y el contenido de la RAM en formato .raw todo en nuestro equipo forense listo para trabajar con ello:

```
chocolate@intel:/media/chocolate/B4723A3C723A0420/forensics/RAM/dumpIt$ sha256sum DESKTOP-3VU71PH-20250522-13203
5.raw
967d07181f235cc453d9611e3ef8c3c46c30f379ae5087af45705d4075f94ecd  DESKTOP-3VU71PH-20250522-132035.raw
```

PARTE 3: Adquisición en caliente del disco duro

-Descargamos la herramienta **.dd for Windows**, de la web http://www.chrysocome.net/dd.
-Abrimos CMD como Administrador en el equipo a analizar, y ejecutamos wmic diskdrive list brief para identificar el número de disco físico. Con esto averiguamos que nuestro USB-herramienta forense es \\.**PhysicalDrive1**. Esto es importante a la hora de seleccionar el lugar de almacenamiento de la prueba.
De modo que nuestro comando sería: dd if=\\.\PhysicalDrive1 of=D:\parte_disco.raw bs=4M skip=100 count=500 - -progress

Siendo el caso de este ejercicio la toma de una muestra cualquiera, se han elegido estos parámetros pudiendo perfectamente ser otros. Aquí esto significa: bs=4M copiamos bloques de 4MB, skip=100 saltamos 100 bloques al principio (opcional), count=500 es copiar 500 bloques después de eso, y finalmente guardar en D:\ nuestro pendrive-herramienta de trabajo. Con - -progress vemos el avance de la extracción..
He aquí nuestra adquisición parcial de un disco de un equipo a analizar, y abajo el hash generado. Para este ejercicio me he ahorrado los detalles técnicos del PC a analizar para no pasarme aún más de extensión.




```
chocolate@intel:/media/chocolate/B4723A3C723A0420$ sha256sum parte_disco.raw
a86afa6d37056526b2cc4bc721d7e0ad8631112e1f26ce903712aea97d0036c1  parte_disco.raw
```