



Certificado de Profesionalidad en Seguridad Informática

IronHack - SOC

Módulo 4

Criptografía - Hashes

Práctica 2

Alumno: Julián Gordon

Indice

Enunciado.....	3
Introducción.....	4
Creación de fichero y cálculo de hash md5 y sha256.....	5
Cálculo de Hashes con PowerShell.....	5
Herramientas de Cálculo de Hashes con Interfaz Gráfica en Windows.....	7
Automatización del cálculo de hashes en Windows con PowerShell.....	9
Creación de fichero y cálculo de hash md5 y sha256 en Linux.....	13
Cálculo de hashes con herramientas propias del sistema.....	13
Automatización del Cálculo de Hashes de Ficheros en Kali Linux con Bash.....	14
Conclusiones.....	16

Enunciado

Actividad 2.- Cálculo de Hashes.

- En Windows, crea un fichero de texto y calcula su hash, md5, sha256, mediante PowerShell. Realiza cambios en el fichero y vuelve a calcular los hashes. Compara el resultado.
- Adicionalmente, busca herramientas para el cálculo de hashes mediante interfaz gráfica y repite el ejercicio.
- Mediante Powershell, automatiza el cálculo de hashes de todos los ficheros de un directorio.
- En Linux, realiza las mismas operaciones de cálculo de hash mediante las herramientas propias del sistema.
- Por último, en Linux, automatiza mediante Shell script, la obtención de los hashes de todos los ficheros de un directorio.

Introducción

El cálculo de hashes, una práctica fundamental en el ámbito de la seguridad informática, ha desempeñado un papel crucial en la verificación de la integridad de los datos y la detección de cambios no autorizados en los archivos desde hace décadas. Un hash es una función matemática que toma un conjunto de datos, ya sea un archivo, un texto o cualquier tipo de información, y genera una cadena de caracteres única e irrepetible que actúa como una "firma digital" para ese conjunto de datos. Este proceso transforma los datos de entrada en una cadena de longitud fija, independientemente de su tamaño original, lo que facilita la comparación y verificación de la integridad de los datos.

Una de las características más interesantes de los hashes es su propiedad de unidireccionalidad, lo que significa que es fácil calcular el hash de un conjunto de datos dado, pero es extremadamente difícil, sino imposible, reconstruir los datos originales a partir del hash. Además, cambios menores en los datos de entrada producirán un hash completamente diferente, lo que hace que los hashes sean ideales para detectar incluso las modificaciones más sutiles en los archivos.

Los algoritmos de hash más comunes incluyen MD5 (Message Digest Algorithm 5), SHA-1 (Secure Hash Algorithm 1), SHA-256 y SHA-512, entre otros. Cada uno de estos algoritmos tiene sus propias características y niveles de seguridad, con algunos considerados más seguros que otros debido a la longitud de su salida y la resistencia a los ataques.

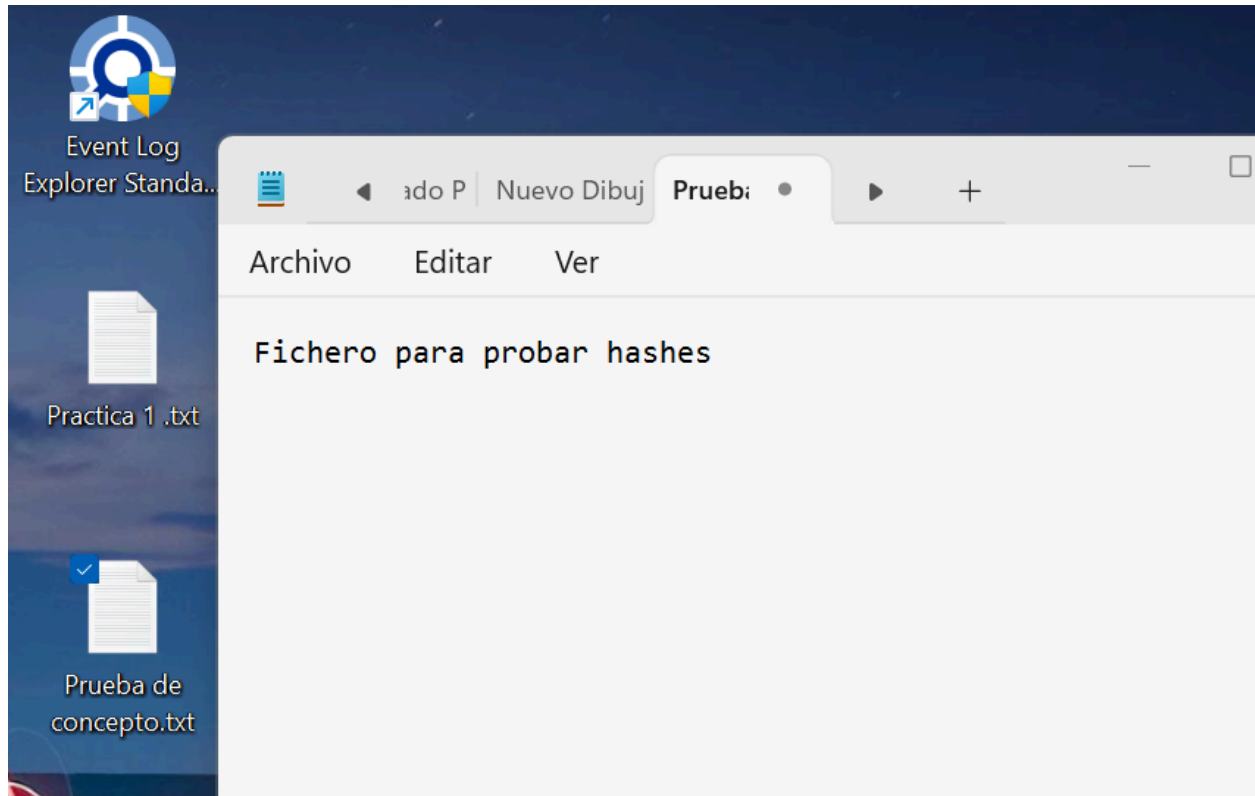
El cálculo de hashes se utiliza en una amplia gama de aplicaciones, desde la verificación de la integridad de los archivos descargados hasta la protección de las contraseñas almacenadas. Además, los hashes se utilizan en la criptografía para garantizar la autenticidad y la integridad de los mensajes y datos transmitidos.

En este trabajo práctico, explicaremos el proceso de cálculo de hashes en diferentes sistemas operativos, Windows y Linux, utilizando herramientas nativas y scripts automatizados. A través de este ejercicio, profundizaremos en la importancia de los hashes en la seguridad informática y su papel fundamental en la protección de los datos y la detección de posibles manipulaciones.

Creación de fichero y cálculo de hash md5 y sha256

Cálculo de Hashes con PowerShell

Comenzaremos este proceso creando un archivo de texto en el sistema operativo Windows.



Luego, se utiliza PowerShell, una interfaz de línea de comandos de Microsoft, para calcular el hash MD5 y SHA-256 del archivo recién creado. Con el comando `Get-FileHash + fichero`

```
PS C:\Users\Lenovo\OneDrive\Escritorio> Get-FileHash '.\Prueba de concepto.txt'
```

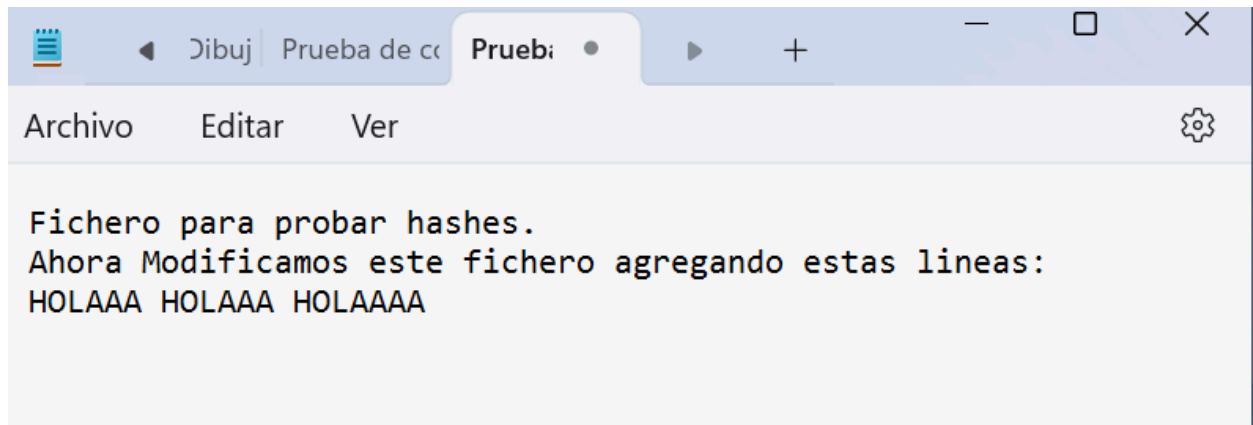
Algorithm	Hash	Path
-----	----	----
SHA256	21024069EB0D2EE0FCCA761BA27404B90F5D5B7EFE72E45A4CE789E5AE74B31F	C:\Users\Lenovo\OneDrive\Escr...

```
PS C:\Users\Lenovo\OneDrive\Escritorio> Get-FileHash '.\Prueba de concepto.txt' -Algorithm MD5
```

Algorithm	Hash	Path
-----	----	----
MD5	F9F9A2EF1B4FA6DEBA4E020C526A5EE0	C:\Users\Lenovo\OneDrive\Escr...

Estos algoritmos de hash son funciones criptográficas que convierten los datos en una cadena de longitud fija, lo que permite verificar la integridad de los archivos.

Después de calcular los hashes, realizamos cambios en el archivo de texto original. Estos cambios pueden incluir agregar, eliminar o modificar el contenido del archivo.



Una vez realizados los cambios, se vuelve a calcular los hashes MD5 y SHA-256 del archivo modificado.

```
PS C:\Users\Lenovo\OneDrive\Escritorio> Get-FileHash '.\Prueba de concepto.txt'
```

Algorithm	Hash	Path
SHA256	A020E2F43D36584C7D1AABEBA9CD56B7A0490540A72E958B77CF881A9DE9A7BB	C:\Users\Lenovo\OneDrive\Escr...

```
PS C:\Users\Lenovo\OneDrive\Escritorio> Get-FileHash '.\Prueba de concepto.txt' -Algorithm MD5
```

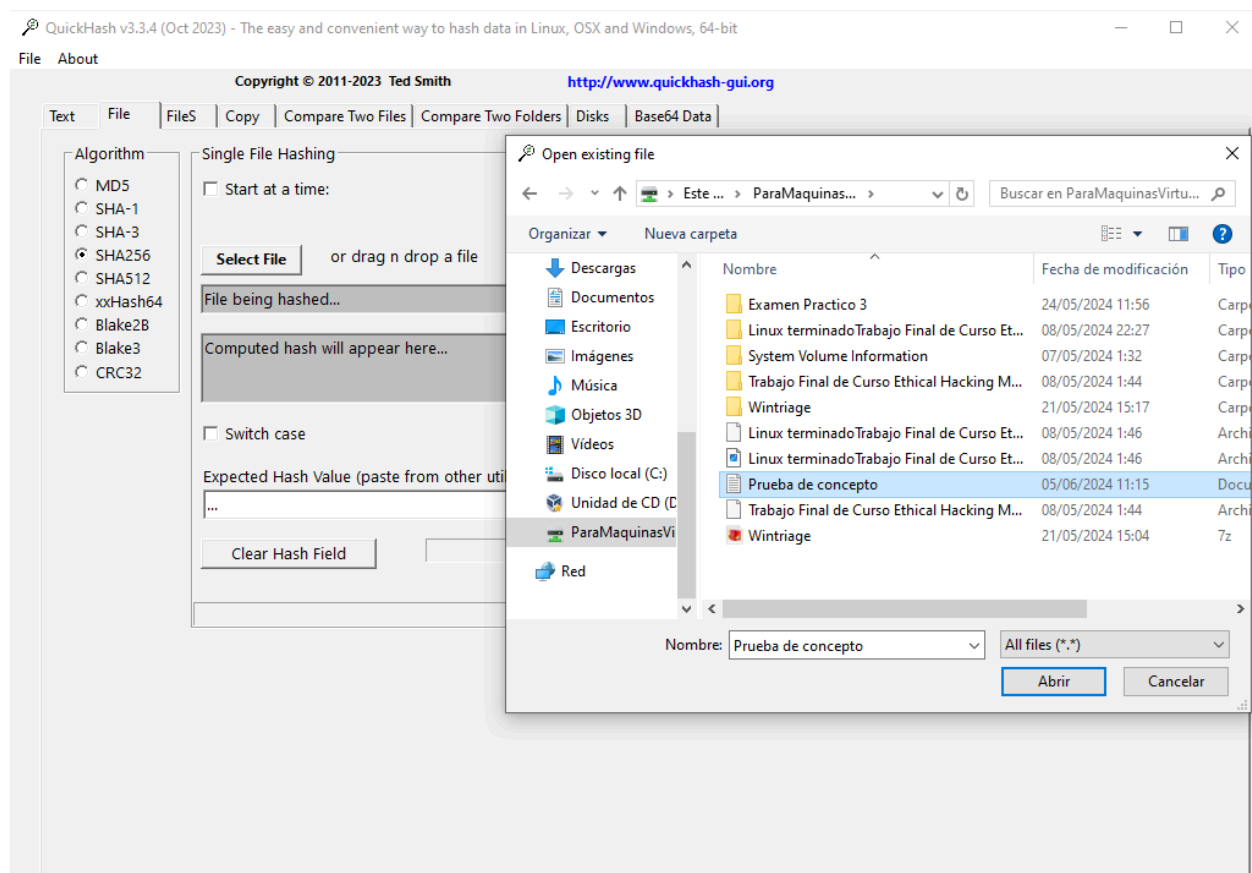
Algorithm	Hash	Path
MD5	D5D7AEDC0D0A4064EA0CD160D515682C	C:\Users\Lenovo\OneDrive\Escr...

Finalmente, se compara el resultado de los hashes originales con los hashes recalculados. Los hashes son diferentes, significa que el archivo ha sido modificado y su integridad puede haber sido comprometida.

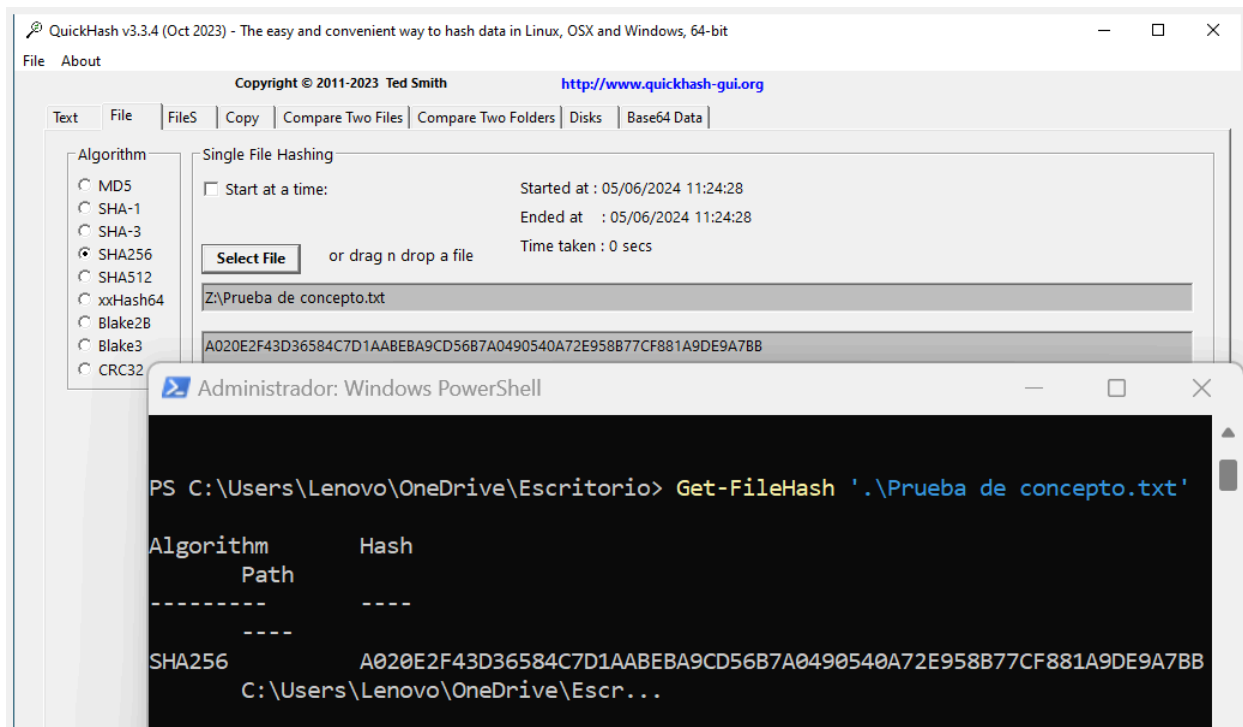
Herramientas de Cálculo de Hashes con Interfaz Gráfica en Windows

Además del uso de PowerShell, se buscan y utilizan herramientas de cálculo de hashes que proporcionan una interfaz gráfica de usuario (GUI). Estas herramientas permiten a los usuarios calcular fácilmente los hashes de los archivos utilizando una interfaz intuitiva en lugar de comandos de línea de comandos. El proceso de calcular los hashes es similar al realizado con PowerShell, pero la diferencia radica en la experiencia del usuario y la comodidad proporcionada por la GUI. En esta ocasión vamos a utilizar la herramienta Quickhash.

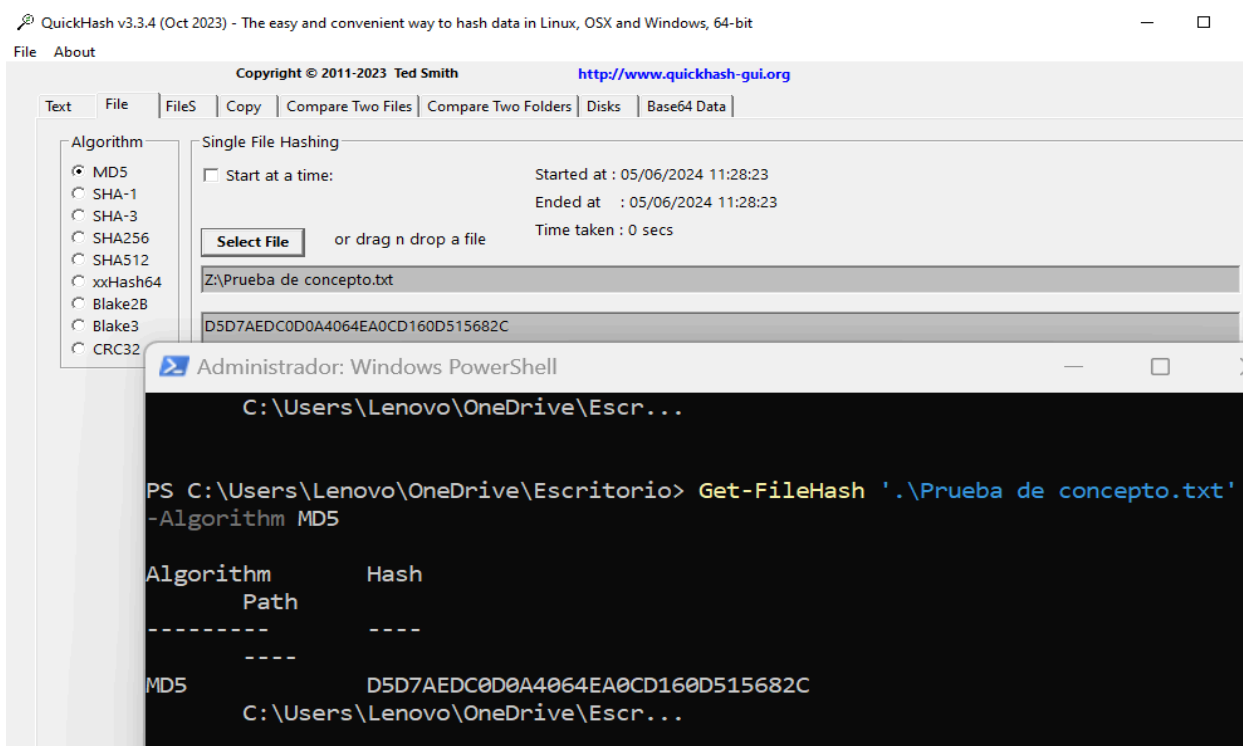
Seleccionamos el fichero que deseamos y el algoritmo a utilizar, en este caso SHA256.



Podemos verificar que los hashes son similares a los que nos dió PowerShell.



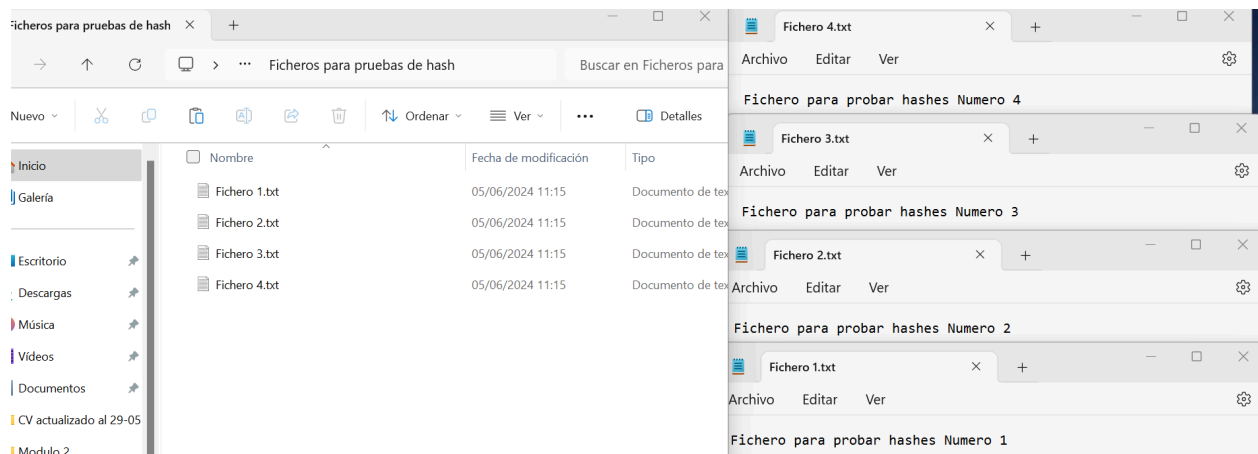
Ahora con MD5.



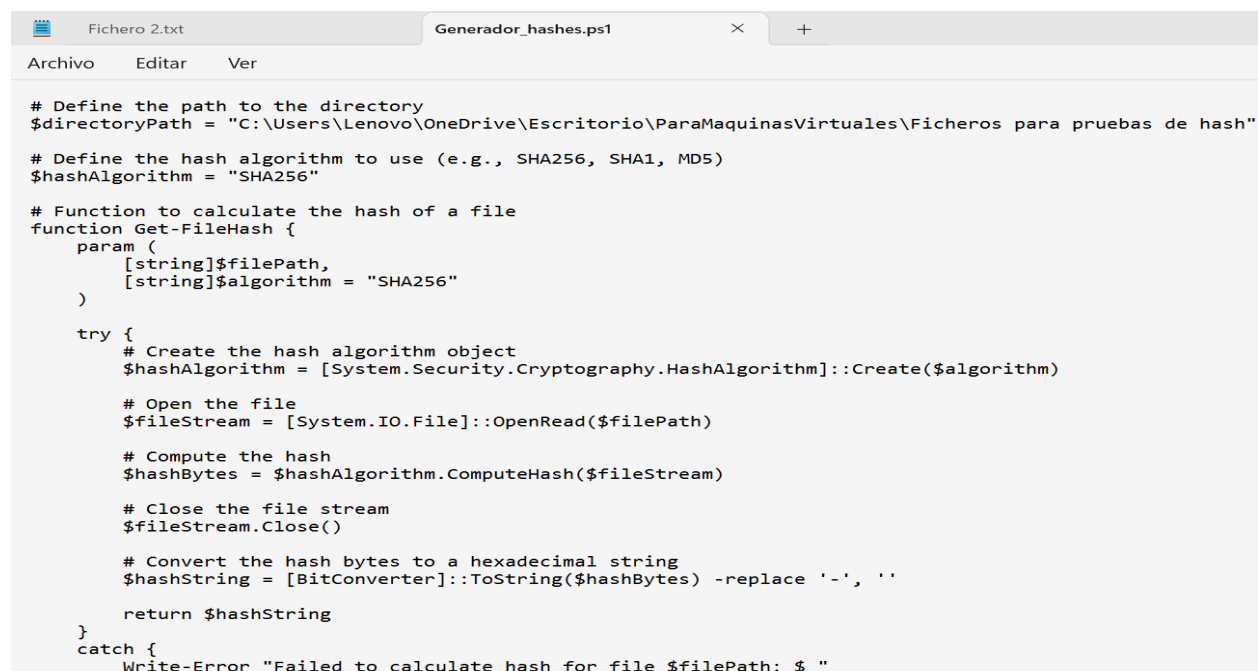
Automatización del cálculo de hashes en Windows con PowerShell

En esta parte, vamos a automatizar el cálculo de hashes de todos los archivos en un directorio utilizando PowerShell. Esto implica escribir un script de PowerShell que recorra cada archivo en el directorio especificado, calcule su hash y genere un informe con los resultados. La automatización ahorra tiempo y esfuerzo al realizar la tarea de calcular los hashes de manera eficiente para múltiples archivos.

Para esta etapa, crearemos 4 ficheros.



Escribimos el script en un txt y lo modificamos para que sea un .ps1 .



Aquí tenemos el script entero en texto plano.

```
# Define the path to the directory
$directoryPath = "C:\ruta\al\directorio"

# Define the hash algorithm to use (e.g., SHA256, SHA1, MD5)
$hashAlgorithm = "SHA256"

# Function to calculate the hash of a file
function Get-FileHash {
    param (
        [string]$filePath,
        [string]$algorithm = "SHA256"
    )
    try {
        # Create the hash algorithm object
        $hashAlgorithm = [System.Security.Cryptography.HashAlgorithm]::Create($algorithm)

        # Open the file
        $fileStream = [System.IO.File]::OpenRead($filePath)

        # Compute the hash
        $hashBytes = $hashAlgorithm.ComputeHash($fileStream)

        # Close the file stream
        $fileStream.Close()

        # Convert the hash bytes to a hexadecimal string
        $hashString = [BitConverter]::ToString($hashBytes) -replace '-', ' '
        return $hashString
    }
}
```

```

catch {
    Write-Error "Failed to calculate hash for file $filePath: $_"
    return $null
}
}

# Get all files in the directory
$files = Get-ChildItem -Path $directoryPath -File

# Loop through each file and calculate the hash
foreach ($file in $files) {
    $filePath = $file.FullName

    $hash = Get-FileHash -filePath $filePath -algorithm $hashAlgorithm

    if ($hash) {
        Write-Output "File: $filePath"

        Write-Output "Hash ($hashAlgorithm): $hash"

        Write-Output "-----"
    }
}

```

Primero, se define la ruta al directorio y el algoritmo de hash a utilizar (por ejemplo, SHA256, SHA1, MD5). Luego, se define una función llamada Get-FileHash que toma la ruta de un archivo y el algoritmo de hash como parámetros y devuelve el hash del archivo en formato hexadecimal.

El script recorre cada archivo en el directorio utilizando el cmdlet Get-ChildItem para obtener una lista de archivos. Para cada archivo, llama a la función Get-FileHash para calcular su hash utilizando el algoritmo especificado. Finalmente, imprime el nombre del archivo y su hash correspondiente en la consola de salida de PowerShell. Si ocurre algún error durante el proceso de cálculo del hash, se muestra un mensaje de error.

Al intentar ejecutar el script que generamos, nos aparece el primer error. No tenemos habilitada la función para ejecutar scripts en nuestro ordenador. Debemos modificarla.

```
PS C:\Users\Lenovo\OneDrive\Escritorio> .\Generador_hashes.ps1
.\Generador_hashes.ps1 : No se puede cargar el archivo C:\Users\Lenovo\OneDrive\Escritorio\Generador_hashes.ps1 porque
la ejecución de scripts está deshabilitada en este sistema. Para obtener más información, consulta el tema
about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170.
En línea: 1 Carácter: 1
+ .\Generador_hashes.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

Modificamos la política para ejecución de scripts.

```
PS C:\Users\Lenovo\OneDrive\Escritorio> Get-ExecutionPolicy -List
>>

Scope ExecutionPolicy
-----
MachinePolicy Undefined
UserPolicy Undefined
Process Undefined
CurrentUser Undefined
LocalMachine Undefined

PS C:\Users\Lenovo\OneDrive\Escritorio> Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass
>>

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías
exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en
https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda (el valor predeterminado es "N"): s
```

Ahora sí ejecutamos el script y nos devuelve el resultado de los hash de todos los ficheros de la carpeta.

```
PS C:\Users\Lenovo\OneDrive\Escritorio> .\Generador_hashes.ps1
File: C:\Users\Lenovo\OneDrive\Escritorio\ParaMaquinasVirtuales\Ficheros para pruebas de hash\Fichero 1.txt
Hash (SHA256): B2E9F287C3F39F108C2D1EB6826368EFF63C4ACB8FAAA727991E927A43CC7545
-----
File: C:\Users\Lenovo\OneDrive\Escritorio\ParaMaquinasVirtuales\Ficheros para pruebas de hash\Fichero 2.txt
Hash (SHA256): 933157868A3601D178F3759AB9B2EDFDB48BF683A3A6715D416A7BEC35162221
-----
File: C:\Users\Lenovo\OneDrive\Escritorio\ParaMaquinasVirtuales\Ficheros para pruebas de hash\Fichero 3.txt
Hash (SHA256): 521E3B86C5476D3B436EBC05F8A6B7DFDD0CA1C87210D288506F4252F22C569
-----
File: C:\Users\Lenovo\OneDrive\Escritorio\ParaMaquinasVirtuales\Ficheros para pruebas de hash\Fichero 4.txt
Hash (SHA256): 69E959AA383033BCE9B7C1994EC21F5281BB4337E0088218784A984298141D76
-----
PS C:\Users\Lenovo\OneDrive\Escritorio>
```

Por seguridad, volvemos a la configuración anterior.

```
PS C:\Users\Lenovo\OneDrive\Escritorio> Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy Restricted
>>

Cambio de directiva de ejecución
La directiva de ejecución te ayuda a protegerte de scripts en los que no confías. Si cambias dicha directiva, podrías
exponerte a los riesgos de seguridad descritos en el tema de la Ayuda about_Execution_Policies en
https://go.microsoft.com/fwlink/?LinkID=135170. ¿Quieres cambiar la directiva de ejecución?
[S] Sí [O] Sí a todo [N] No [T] No a todo [U] Suspender [?] Ayuda (el valor predeterminado es "N"): Sí
```

Confirmamos que está correctamente modificado.

```
PS C:\Users\Lenovo\OneDrive\Escritorio> .\Generador_hashes.ps1
.\Generador_hashes.ps1 : No se puede cargar el archivo C:\Users\Lenovo\OneDrive\Escritorio\Generador_hashes.ps1 porque
la ejecución de scripts está deshabilitada en este sistema. Para obtener más información, consulta el tema
about_Execution_Policies en https://go.microsoft.com/fwlink/?LinkID=135170.
En línea: 1 Carácter: 1
+ .\Generador_hashes.ps1
+ ~~~~~
+ CategoryInfo          : SecurityError: (:) [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
PS C:\Users\Lenovo\OneDrive\Escritorio>
```

Creación de fichero y cálculo de hash md5 y sha256 en Linux

Cálculo de hashes con herramientas propias del sistema

En el sistema operativo Linux, se realizan las mismas operaciones de cálculo de hashes utilizando las herramientas propias del sistema, como md5sum y sha256sum. Estas herramientas son comandos de línea de comandos que permiten calcular los hashes MD5 y SHA-256 de los archivos.

```
(root@kali)-[/home/kali]
# echo "probando hashes" > archivo_prueba_de_hashes.txt

(root@kali)-[/home/kali]
# md5sum archivo_prueba_de_hashes.txt
d741693e6d260f53b0c5402bcdff1cf  archivo_prueba_de_hashes.txt

(root@kali)-[/home/kali]
# sha256sum archivo_prueba_de_hashes.txt
92f97c70032c98c585f643071ee57451b483e5eb526a2937e855363e68c04ad5  archivo_prueba_de_hashes.txt

(root@kali)-[/home/kali]
# echo "probando hashes, agregamos algo" > archivo_prueba_de_hashes.txt

(root@kali)-[/home/kali]
# sha256sum archivo_prueba_de_hashes.txt
84bf822a114cd9e6d0e7503585562900586f1a86220e33e04c70b7ba70d9e230  archivo_prueba_de_hashes.txt

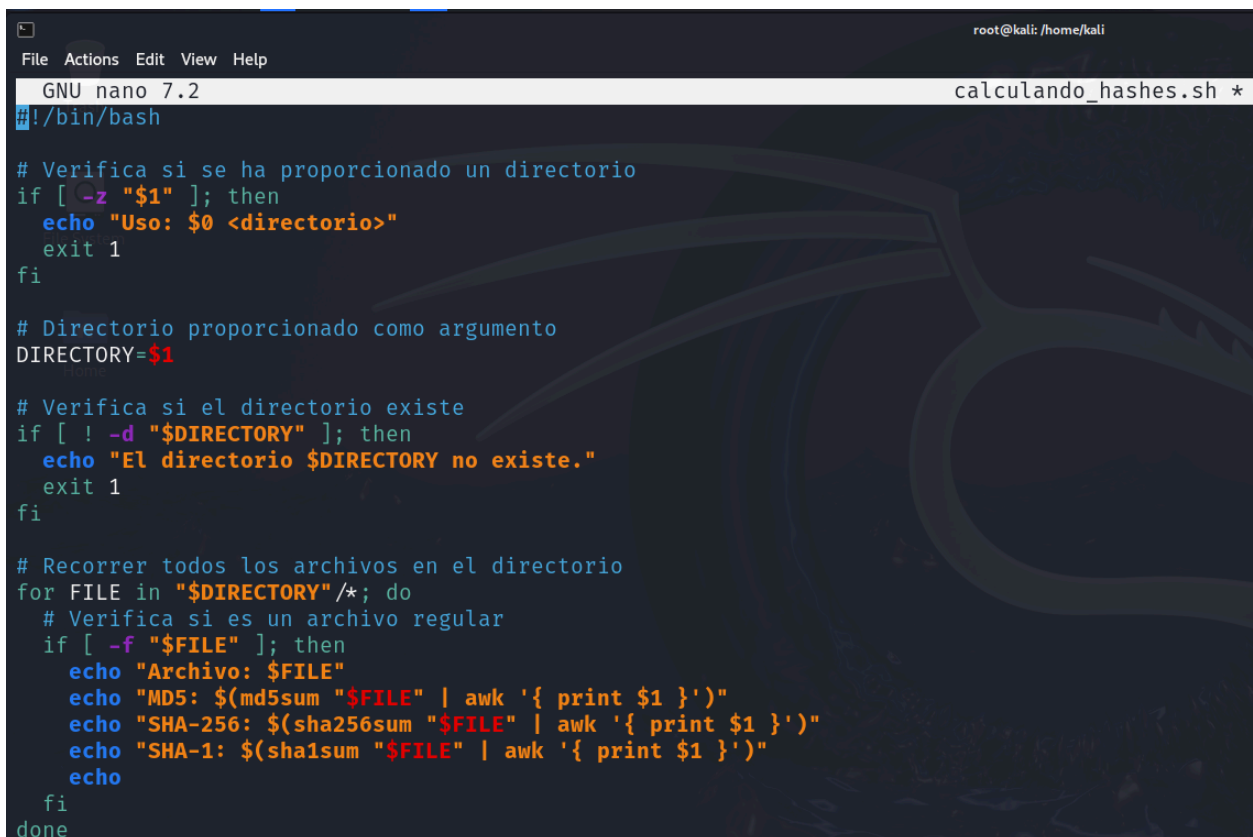
(root@kali)-[/home/kali]
# md5sum archivo_prueba_de_hashes.txt
1b400930f09db63bfa56a57d3469c6b2  archivo_prueba_de_hashes.txt
```

Repetimos el mismo proceso que con PowerShell, pero ahora esta vez en la consola de Kali Linux.

Automatización del Cálculo de Hashes de Ficheros en Kali Linux con Bash

En esta última parte, se automatiza el cálculo de hashes de todos los archivos en un directorio utilizando un script de shell en Linux. Similar al enfoque en Windows, se crea un script que recorre cada archivo en el directorio especificado, calcula sus hashes y genera un informe con los resultados. Esta automatización es útil para tareas repetitivas y facilita la gestión de archivos en sistemas Linux.

Creemos el script con el comando nano calculando_hashes.sh



```
GNU nano 7.2 calculando_hashes.sh *
#!/bin/bash

# Verifica si se ha proporcionado un directorio
if [ -z "$1" ]; then
    echo "Uso: $0 <directorio>"
    exit 1
fi

# Directorio proporcionado como argumento
DIRECTORY=$1

# Verifica si el directorio existe
if [ ! -d "$DIRECTORY" ]; then
    echo "El directorio $DIRECTORY no existe."
    exit 1
fi

# Recorrer todos los archivos en el directorio
for FILE in "$DIRECTORY"/*; do
    # Verifica si es un archivo regular
    if [ -f "$FILE" ]; then
        echo "Archivo: $FILE"
        echo "MD5: $(md5sum "$FILE" | awk '{ print $1 }')"
        echo "SHA-256: $(sha256sum "$FILE" | awk '{ print $1 }')"
        echo "SHA-1: $(sha1sum "$FILE" | awk '{ print $1 }')"
        echo
    fi
done
```

Ahora crearemos ficheros dentro de una carpeta para hacer la misma prueba que realizamos anteriormente en powershell.

```
(root@kali)-[/home/kali]
# mkdir carpeta_hashes

(root@kali)-[/home/kali]
# cd carpeta_hashes

(root@kali)-[/home/kali/carpeta_hashes]
# echo "test 1" > test1.txt

(root@kali)-[/home/kali/carpeta_hashes]
# echo "test 2" > test2.txt

(root@kali)-[/home/kali/carpeta_hashes]
# echo "test 3" > test3.txt

(root@kali)-[/home/kali/carpeta_hashes]
# cd ..

(root@kali)-[/home/kali]
# ls
archivo_prueba_de_hashes.txt  carpeta_hashes  Documents  Music  Public  snort3  Videos
calculando_hashes.sh          Desktop        Downloads  Pictures  RegRipper3.0  Templates  wazuh-agent_4.7.3-1_amd64.deb

(root@kali)-[/home/kali]
# cp calculando_hashes.sh carpeta_hashes

(root@kali)-[/home/kali]
# cd carpeta_hashes

(root@kali)-[/home/kali/carpeta_hashes]
# ls
calculando_hashes.sh  test1.txt  test2.txt  test3.txt
```

Ejecutamos el script y podemos ver el resultado obtenido de los hashes.

```
(root@kali)-[/home/kali/carpeta_hashes]
# ./calculando_hashes.sh /home/kali/carpeta_hashes
Archivo: /home/kali/carpeta_hashes/calculando_hashes.sh
MD5: 4bbbad05b1df79f67e48dfd8a3bb1d1d
SHA-256: 2e4c5a234fc55e8bec2ed328cc8d0bb91ba401f89aeb09a48e478f35bc7d430f
SHA-1: 95694c238c6d6d2c3c030cfa964a2978f5a24b3f

Archivo: /home/kali/carpeta_hashes/test1.txt
MD5: 2490a3d39b0004e4afeb517ef0ddbe2d
SHA-256: 3cd203ac11340842055a6de561c9d69ca4493e912bd4c3c440c80711e16d5aee
SHA-1: b54e43082887d1e7cdb10b7a21fe4a1e56b44b5a

Archivo: /home/kali/carpeta_hashes/test2.txt
MD5: b0b3b0dbf5330e3179c6ae3e0ac524c9
SHA-256: ef691f74bb2e7cb7e9b48b4d57e9e62fa535a0a6ea0100676c4fc492cca8b6d0
SHA-1: a096a9d3cb96fa4cf6c63bd736a84cb7a7e4b61e

Archivo: /home/kali/carpeta_hashes/test3.txt
MD5: 2244fbd6bee5dcbe312e387c062ce6e6
SHA-256: f332dc0b25c681863f10100f0fedd2b2e6ddcc9abe360d6d780d73b7322c9aa5
SHA-1: cde0c01b26634f869bb876326e4fbe969792bf94
```

Conclusiones

Durante este trabajo, realizamos una serie de operaciones para calcular hashes en diferentes sistemas operativos y automatizamos el proceso utilizando scripts. En Windows, utilizamos PowerShell para calcular los hashes MD5 y SHA-256 de un archivo, lo que nos permitió verificar la integridad del archivo y detectar cambios en su contenido. Además, exploramos herramientas de interfaz gráfica para calcular hashes, lo que proporciona una alternativa más amigable para los usuarios menos familiarizados con la línea de comandos.

En Linux, utilizamos las herramientas nativas del sistema, como md5sum y sha256sum, para calcular los hashes de los archivos, lo que demuestra la versatilidad y potencia de las herramientas integradas en el sistema operativo. Finalmente, automatizamos el cálculo de hashes en Linux mediante un script de shell, lo que nos permitió realizar la tarea de manera eficiente para múltiples archivos en un directorio.

Este trabajo práctico nos proporcionó una comprensión más profunda del proceso de cálculo de hashes y nos permitió explorar diferentes enfoques para realizar esta tarea en sistemas operativos Windows y Linux. La capacidad de verificar la integridad de los datos y detectar posibles manipulaciones es fundamental en la seguridad informática, y el cálculo de hashes es una herramienta esencial en este contexto.