Adrián Conesa Guirao

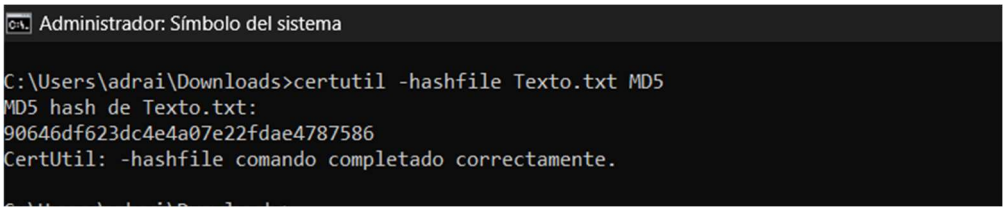
U3 T0 - Verificación de la integridad de archivos mediante funciones hash

Contenido

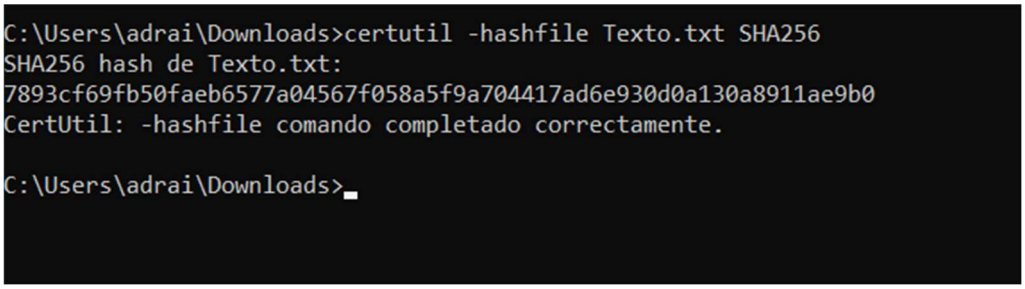
Práctica en Windows:

Utilizando la herramienta **CertUtil** en Windows, calcula el hash de un archivo que tengas disponible en tu ordenador

Resultados Powershell MD5:

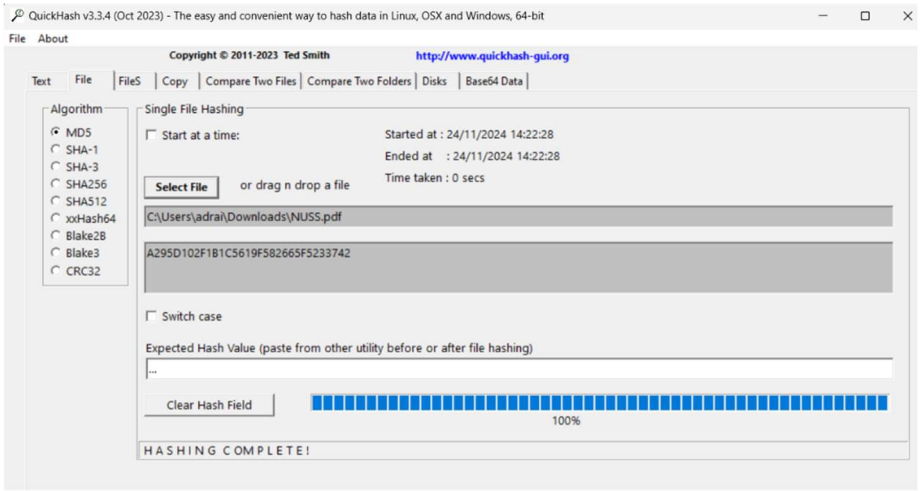


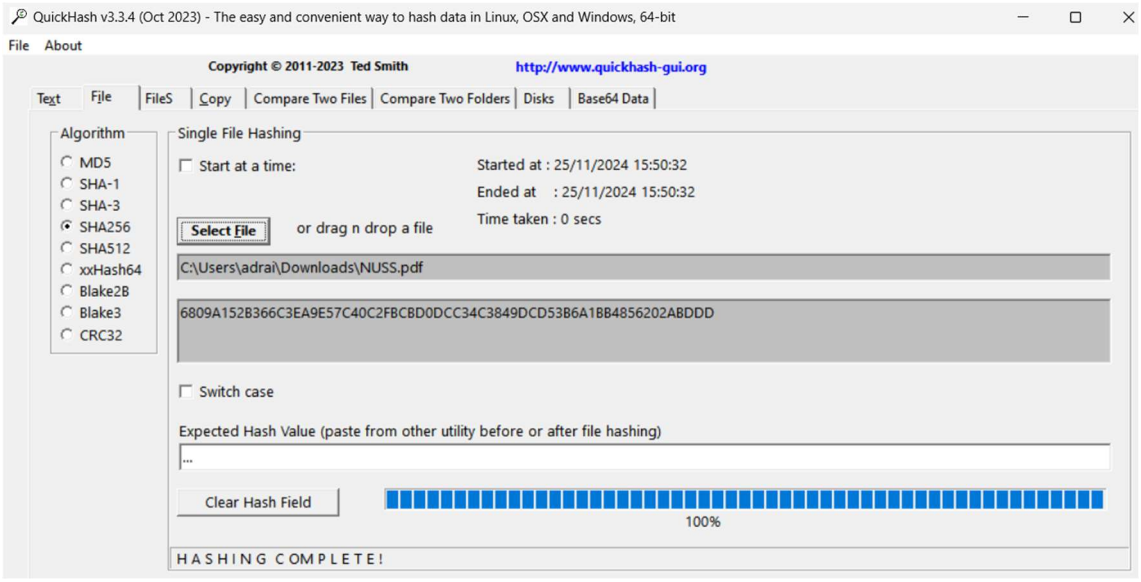
Resultados Powershell SHA256:



Instala **HashTab** o **QuickHash GUI** y genera el hash de un archivo usando uno de estos programas. Explica brevemente tu experiencia con estas herramientas

Resultados QuickHash GUI:





¿Te parecieron fáciles de usar?

Sí es una herramienta bastante básica en cuanto a funcionamiento y muy fácil de entender sin haberla usado antes.

¿Cuál es la ventaja de tener una interfaz gráfica?

La comodidad a la hora de hacerlo

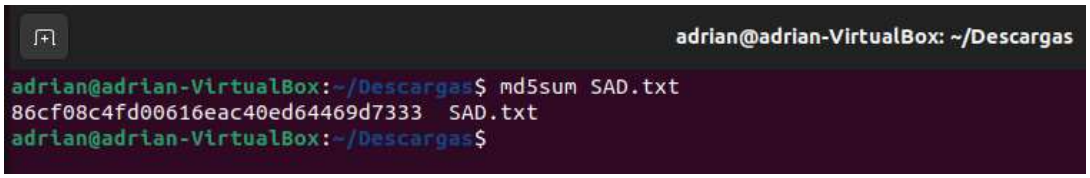
Práctica en Linux:

Utilizando una máquina virtual con Linux o una distribución en modo live, calcula el hash de un archivo usando los siguientes comandos:

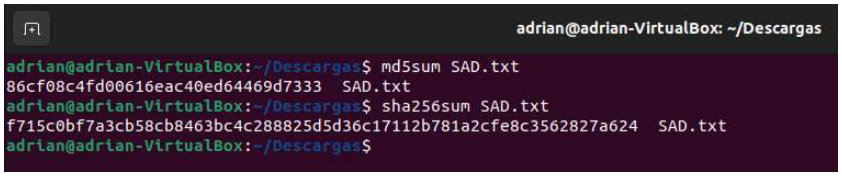
md5sum para generar el hash MD5.

sha256sum para generar el hash SHA-256.

Resultado md5sum:



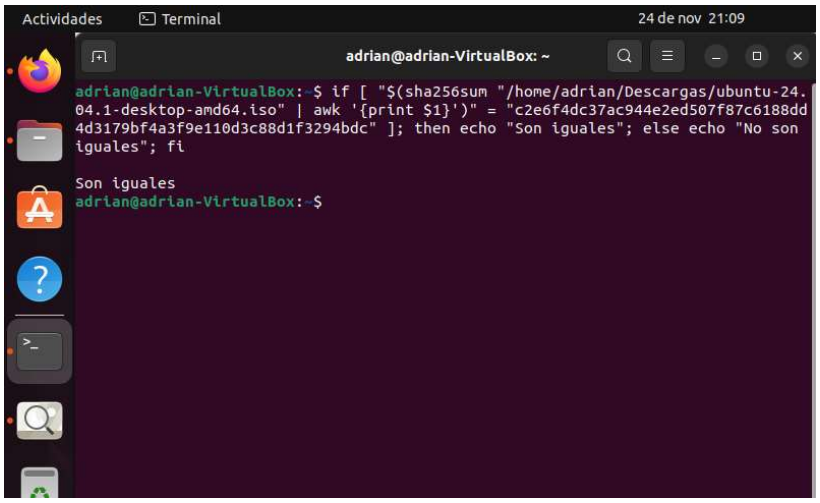
Resultado sha256sum:



Explica cómo podrías utilizar estos comandos para verificar la integridad de un archivo descargado de internet.

Por ejemplo Ubuntu proporciona sus hash.

Para que los compruebes cuando has descargado la iso de instalación, lo mejor sería que descargases el .txt de su página web donde indica el hash y el tipo de algoritmo, con este sencillo script lo podrías comprobar sin necesidad de descargar el archivo:



Comparación de Algoritmos

Investiga y responde: ¿Por qué los algoritmos **MD5** y **SHA-1** ya no son recomendados para aplicaciones críticas? Da un ejemplo de una situación en la que el uso de estos algoritmos podría representar un riesgo.

MD5 y SHA-1 ya no son seguros debido a su vulnerabilidad a los ataques de colisión. El uso de estos algoritmos en aplicaciones críticas puede poner en riesgo la integridad de los datos, la autenticidad de las comunicaciones y la seguridad de los sistemas en general. Por lo tanto, es fundamental migrar a funciones hash más modernas y robustas como SHA-2 o SHA-3.

Ejemplo:

Una empresa utiliza SHA-1 para verificar que los archivos descargados de un proveedor sean los mismos archivos que el proveedor coloca en el servidor. En este caso, un atacante podría generar un archivo malicioso que genere el mismo valor hash que un archivo legítimo, luego puede reemplazar el archivo legítimo en el servidor del proveedor con el archivo que ha creado él.

Cuando la empresa obtenga el archivo del servidor del proveedor, el valor hash coincidirá y, por lo tanto, la empresa creerá que el archivo es legítimo y ejecutará el archivo, exponiendo así su sistema al compromiso.