Lahocine José Ait Aliourdellah Bravo

SAD UT-3 T-0
Verificación de la integridad de archivos mediante funciones hash

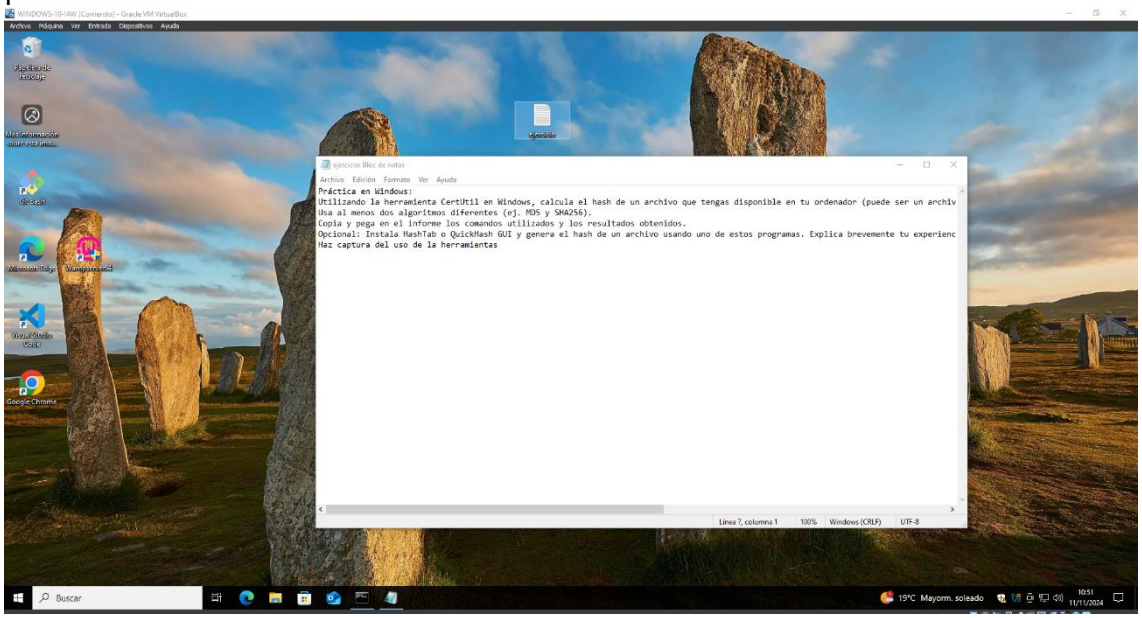
ÍNDICE

Introducción

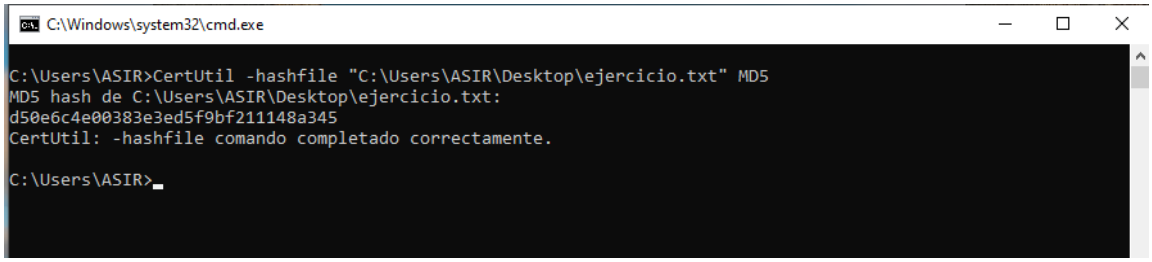
En esta práctica, se aprenderá a verificar la integridad de archivos utilizando funciones hash, que son fundamentales para garantizar que los archivos no hayan sido alterados durante su transferencia o almacenamiento. A través de esta tarea, aprenderemos conocimientos prácticos sobre cómo generar y verificar los valores hash de archivos utilizando diferentes herramientas y algoritmos en dos sistemas operativos: Windows y Ubuntu.

Windows

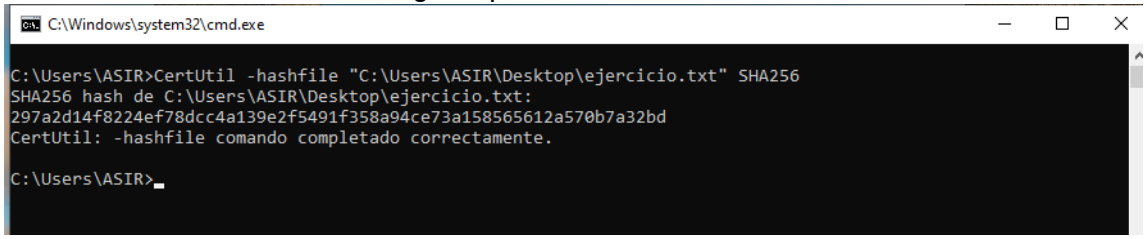
Lo primero que haremos es crear un archivo “txt” que tenga texto en su interior, para poder verificar su hash.



Ejecutamos el comando “CertUtil -hashfile “C:\Users\ASIR\Desktop\ejercicio.txt” MD5” para calcula el valor hash MD5 del archivo “ejercicio.txt” ubicado en “C:\Users\ASIR\Desktop\”. El comando genera un valor MD5 de 128 bits, mostrado en formato hexadecimal.



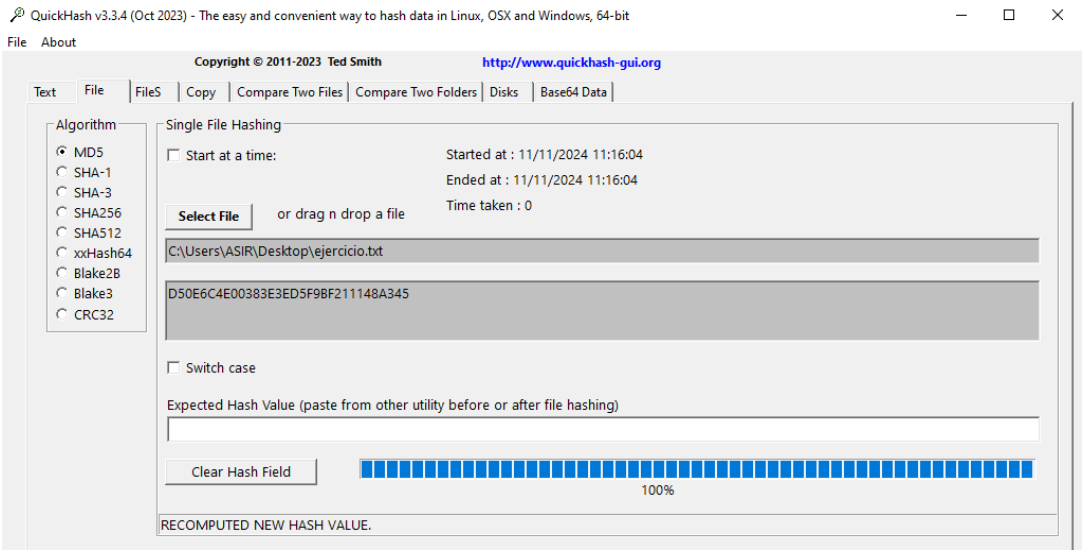
Ejecutando el comando “CertUtil -hashfile "C:\Users\ASIR\Desktop\ejercicio.txt" SHA256” calcula el hash SHA-256 del archivo “ejercicio.txt”, generando una cadena única de 64 caracteres, más seguro que MD5.



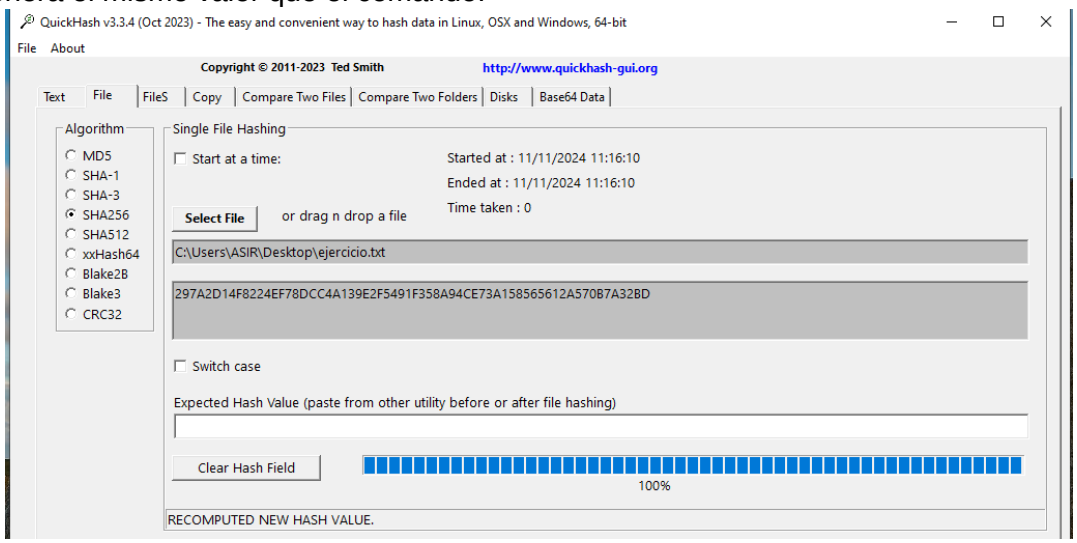
QuickHash GUI

QuickHash GUI es una herramienta gráfica de código abierto que permite generar y comparar hash de archivos usando diferentes algoritmos como MD5, SHA-1, SHA-256, entre otros. Esta aplicación proporciona una interfaz visual que facilita la generación de hash sin necesidad de utilizar la línea de comandos.

Abrimos QuickHash GUI y nos dirigimos a la pestaña “File” donde seleccionaremos el archivo “ejercicio.txt” en la parte izquierda podremos seleccionar el algoritmo de hash. Elegimos “MD5” y podemos ver el valor que nos genera, podemos comprarlo con el que nos ha dado con el comando y ver que es el mismo.



Y repetimos el paso con el algoritmo “SHA256” y de nuevo podemos ver que nos genera el mismo valor que el comando.

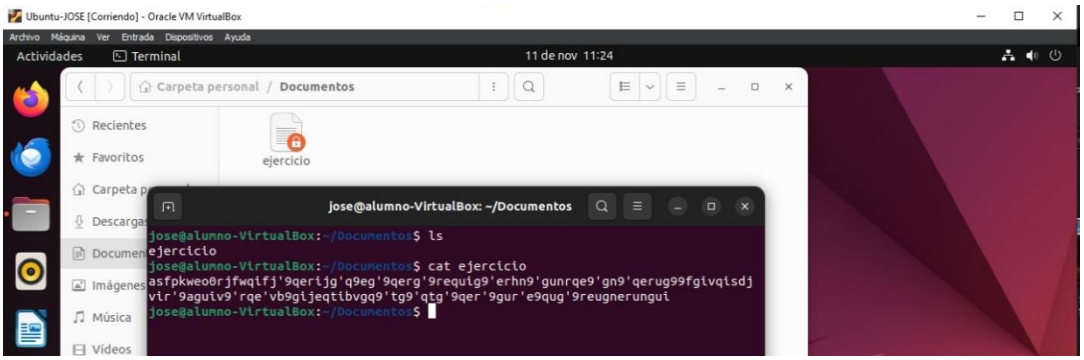


Experiencia con QuickHash GUI

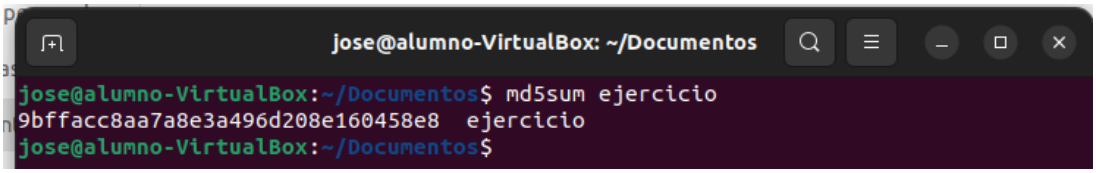
QuickHash GUI es fácil de usar gracias a su interfaz gráfica, que simplifica la selección de archivos, elección de algoritmos y visualización de resultados sin necesidad de usar la línea de comandos. La ventaja principal es que permite realizar tareas técnicas de forma rápida y sin errores, siendo ideal para usuarios no expertos en programación. Además, ofrece varias opciones de algoritmos de hash, lo que la hace versátil y accesible.

Ubuntu

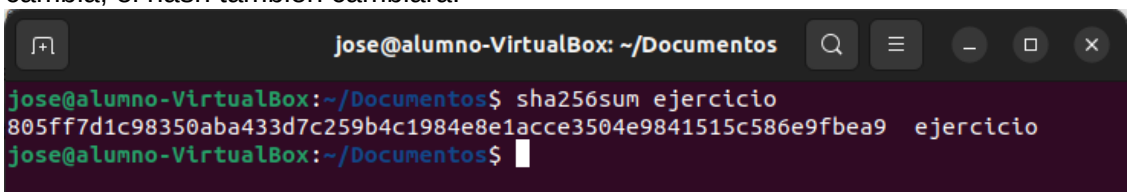
Lo primero que vamos a hacer es la creación del archivo de texto que verificaremos su hash.



Ejecutamos el comando “md5sum ejercicio” que calcula el hash MD5 del archivo llamado “ejercicio”, genera un valor MD5 de 128 bits, mostrado en formato hexadecimal. Este valor único se utiliza para verificar la integridad del archivo; si el archivo cambia, su hash MD5 también cambiará.



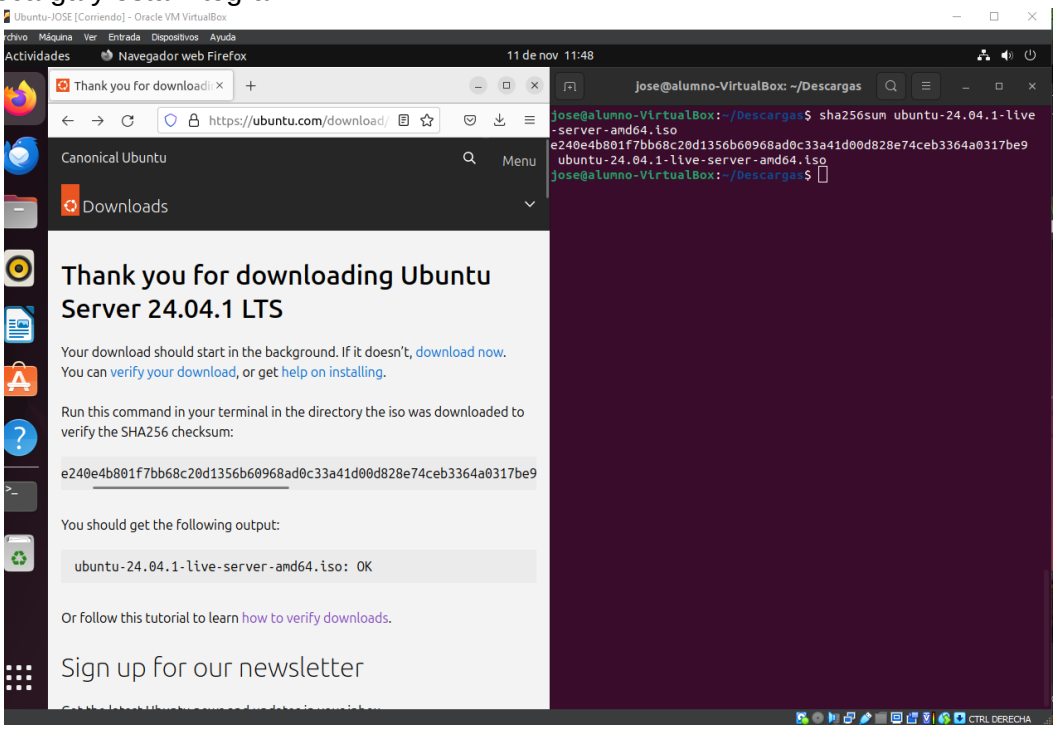
Y con el comando “sha256sum ejercicio” calcula el hash SHA-256 del archivo “ejercicio”, que es una representación única de su contenido. El resultado es un valor de 64 caracteres que se usa para verificar la integridad del archivo. Si el archivo cambia, el hash también cambiará.



Verificar la integridad de un archivo descargado de internet

Para poder probar el verificar la integridad de un archivo descargado de internet vamos a hacer un ejemplo práctico.

Nos dirigimos a la pagina oficiar de Ubuntu descargamos cualquier versión de el en mi caso “Ubuntu server”. En la pagina de descarga esta la opción de “verify your download” y nos aparecerá el hash de la iso. Una vez tengamos la ISO ejecutamos el comando “sha256sum Ubuntu-24.04.1-live-server-amd64.iso” y podemos comprobar que el hash es igual lo que significa que la ISO no se ha corrompido durante la descarga y está íntegra.



Análisis

¿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 recomendados porque un atacante puede aprovechar las colisiones para manipular datos. Una colisión sucede cuando dos entradas diferentes generan el mismo valor hash. Esto permite que un atacante sustituya un archivo, mensaje o certificado válido con otro malicioso, sin que el sistema detecte el cambio, ya que ambos tendrán el mismo hash.

Ejemplo de riesgo:

Muchos sitios ofrecen descargas de software acompañadas de un hash MD5 o SHA-1 para que los usuarios verifiquen que el archivo descargado no ha sido alterado. Sin embargo, un atacante podría crear una versión del software con malware que genere el mismo hash que el original. Al verificar, el usuario vería el hash coincidir y pensaría que el software es legítimo, cuando en realidad ha sido comprometido.

Indica en qué situaciones podría ser aceptable utilizar MD5 en lugar de algoritmos más seguros como SHA-256 o SHA-512.

MD5 podría usarse en lugar de SHA-256 o SHA-512 cuando la seguridad no sea crítica, como en la verificación de integridad de archivos en entornos controlados, la detección de duplicados en bases de datos, o la generación de identificadores rápidos donde no se necesite alta resistencia a ataques.