OpenSSL guia practica

Ing. Fernando Boiero

Año 2023

Tabla de Contenidos

Parte 1: Hashing con OpenSSL

OpenSSL es una herramienta muy útil para generar hash de distintos tipos de archivos. En esta sección, te enseñaré cómo generar hash MD5 y SHA256 utilizando OpenSSL.

Ejercicio 1: Generar hash MD5 de un archivo

1. Crea un archivo llamado "archivo.txt" con algún texto dentro.
2. Abre la terminal de tu sistema operativo (Windows o Linux).
3. Ubícate en la carpeta donde se encuentra el archivo "archivo.txt".
4. Ejecuta el siguiente comando:
   openssl md5 archivo.txt
   Este comando generará el hash MD5 del archivo "archivo.txt".
5. Verifica que el hash generado coincida con el hash obtenido por algún otro medio (por ejemplo, mediante una herramienta online).

Ejercicio 2: Generar hash SHA256 de un archivo

1. Crea un archivo llamado "archivo.txt" con algún texto dentro.
2. Abre la terminal de tu sistema operativo (Windows o Linux).
3. Ubícate en la carpeta donde se encuentra el archivo "archivo.txt".
4. Ejecuta el siguiente comando:
   openssl sha256 archivo.txt
   Este comando generará el hash SHA256 del archivo "archivo.txt".
5. Verifica que el hash generado coincida con el hash obtenido por algún otro medio (por ejemplo, mediante una herramienta online).

Parte 2: Cifrado simétrico con OpenSSL

En esta sección, te enseñaré cómo cifrar y descifrar un archivo utilizando una clave simétrica con OpenSSL.

Ejercicio 1: Cifrar un archivo utilizando una clave simétrica

1. Crea un archivo llamado "archivo.txt" con algún texto dentro.
2. Abre la terminal de tu sistema operativo (Windows o Linux).
3. Ubícate en la carpeta donde se encuentra el archivo "archivo.txt".
4. Genera una clave simétrica utilizando el siguiente comando:
   openssl rand -base64 32 > clave.txt
   Este comando generará una clave de 32 caracteres y la almacenará en el archivo "clave.txt".
5. Cifra el archivo "archivo.txt" utilizando la clave generada con el siguiente comando:
   openssl enc -aes-256-cbc -salt -in archivo.txt -out archivo_cifrado.txt -pass file:clave.txt
   Este comando cifrará el archivo "archivo.txt" utilizando AES-256-CBC y la clave almacenada en el archivo "clave.txt". El archivo cifrado se guardará en el archivo "archivo_cifrado.txt".
6. Verifica que el archivo cifrado no es legible.

Ejercicio 2 (Descifrar un archivo utilizando una clave simétrica)

1. Abre una terminal en tu sistema operativo.

2. Ubica el archivo que quieres descifrar y asegúrate de tener la clave simétrica necesaria para descifrarlo.
3. Ingresa el siguiente comando en la terminal:
   openssl enc -d -aes256 -in archivo_cifrado.txt -out archivo_descifrado.txt -k clave_secreta

Reemplaza archivo_cifrado.txt por el nombre del archivo que quieres descifrar y archivo_descifrado.txt por el nombre que quieres darle al archivo descifrado.
Reemplaza clave_secreta por la clave simétrica que se utilizó para cifrar el archivo.

4. Presiona Enter para ejecutar el comando. El archivo debería descifrarse y guardarse en la ubicación que especificaste en el comando.

Ejercicio 3 (Cifrar un archivo utilizando una clave asimétrica)

1. Abre una terminal en tu sistema operativo.
2. Genera un par de claves asimétricas utilizando el siguiente comando:
   openssl genrsa -out clave_privada.pem 2048

Este comando generará una clave privada llamada clave_privada.pem con un tamaño de 2048 bits.

3. Genera un certificado autofirmado utilizando el siguiente comando:
   openssl req -new -x509 -key clave_privada.pem -out certificado_firmado.crt

Este comando generará un certificado autofirmado llamado certificado_firmado.crt utilizando la clave privada que acabas de generar.

4. Extraer la clave publica:
   openssl x509 -pubkey -noout -in certificado_firmado.crt > clave_publica.pem
5. Cifra el archivo utilizando la clave pública del par de claves que generaste anteriormente con el siguiente comando:
   openssl pkeyutl -encrypt -in archivo_a_cifrar.txt -out archivo_cifrado.txt -inkey clave_publica.pem -pubin

Reemplaza archivo_a_cifrar.txt por el nombre del archivo que quieres cifrar y archivo_cifrado.txt por el nombre que quieres darle al archivo cifrado.
Reemplaza clave_publica.pem por el nombre del archivo que contiene la clave pública del par de claves que generaste anteriormente.

5. Presiona Enter para ejecutar el comando. El archivo debería cifrarse y guardarse en la ubicación que especificaste en el comando.

Ejercicio 4 (Descifrar un archivo utilizando una clave asimétrica)

1. Abre una terminal en tu sistema operativo.
2. Descifra el archivo utilizando la clave privada del par de claves que generaste en el Ejercicio 3 con el siguiente comando:
   openssl rsautl -decrypt -in archivo_cifrado.txt -out archivo_descifrado.txt -inkey clave_privada.pem

Reemplaza archivo_cifrado.txt por el nombre del archivo que quieres descifrar y archivo_descifrado.txt por el nombre que quieres darle al archivo descifrado.
Reemplaza clave_privada.pem por el nombre del archivo que contiene la clave privada del par de claves que generaste en el Ejercicio 3.

3. Presiona Enter para ejecutar el comando. El archivo debería descifrarse y guardarse en la ubicación que especificaste en el comando.

Ejercicio Integrador

Supongamos que somos una empresa de mensajería que necesita enviar un archivo confidencial a un cliente. Este archivo contiene información muy sensible y por lo tanto, es necesario asegurarnos de que solamente el destinatario pueda acceder a ella.

Para ello, vamos a utilizar OpenSSL para realizar los siguientes pasos:

1. Generar un par de claves pública y privada para el destinatario.
2. Utilizar la clave pública del destinatario para cifrar el archivo.
3. Enviar el archivo cifrado al destinatario.
4. El destinatario utiliza su clave privada para descifrar el archivo.

Pasos a seguir:

1. Generación de claves:
   El destinatario debe generar un par de claves pública y privada. Para ello, se deben ejecutar los siguientes comandos en la terminal:
   openssl genpkey -algorithm RSA -out private_key.pem openssl rsa -pubout -in private_key.pem -out public_key.pem
   Estos comandos generarán un archivo private_key.pem que contiene la clave privada y otro archivo public_key.pem que contiene la clave pública del destinatario.
2. Cifrado del archivo:
   El remitente debe cifrar el archivo utilizando la clave pública del destinatario. Para ello, se debe ejecutar el siguiente comando en la terminal:
   openssl rsautl -encrypt -pubin -inkey public_key.pem -in archivo_a_cifrar.txt -out archivo_cifrado.bin
   Este comando cifrará el archivo archivo_a_cifrar.txt utilizando la clave pública del destinatario y lo guardará en el archivo archivo_cifrado.bin.
3. Envío del archivo:
   El remitente debe enviar el archivo cifrado archivo_cifrado.bin al destinatario.
4. Descifrado del archivo:
   El destinatario debe descifrar el archivo utilizando su clave privada. Para ello, se debe ejecutar el siguiente comando en la terminal:
   openssl rsautl -decrypt -inkey private_key.pem -in archivo_cifrado.bin -out archivo_descifrado.txt
   Este comando descifrará el archivo cifrado archivo_cifrado.bin utilizando la clave privada del destinatario y lo guardará en el archivo archivo_descifrado.txt.

Con este ejercicio, podrán aplicar los conocimientos adquiridos en el uso de OpenSSL para cifrado asimétrico.
Además, se les presenta un caso de uso real donde deben aplicar los conocimientos de seguridad y privacidad en el intercambio de información confidencial.