Práctica de Laboratorio - Convertir elementos en hashes (Hashing)
Objetivos

Parte 1: Crear un hash de un Archivo de texto con OpenSSL

Parte 2: Verificar hashes

Aspectos básicos / Escenario

Las funciones de hash son algoritmos matemáticos diseñados para tomar datos como entrada y generar una cadena de caracteres única de tamaño fijo, también conocida como hash. Diseñadas para que sean veloces, las funciones de hash son muy difíciles de revertir; es muy difícil recuperar los datos que crearon un hash determinado, basándose solamente en el hash. Otra de las propiedades importantes de las funciones hash es que hasta el cambio más pequeño realizado en los datos de entrada genera un hash completamente diferente.

Aunque se puede utilizar OpenSSL para generar y comparar hashes, hay otras herramientas disponibles. Algunas de estas herramientas también están incluidas en esta práctica de laboratorio.

Recursos necesarios

- Máquina virtual CyberOps Workstation

Instrucciones

Parte 1: Crear un hash de un Archivo de texto con OpenSSL

OpenSSL se puede utilizar como una herramienta independiente para hashing. Siga los pasos que se indican a continuación para crear un hash de un archivo de texto:

a. Abra una ventana de terminal en la máquina virtual CyberOps Workstation.

b. Como el archivo de texto del cual se quiere generar un hash se encuentra en el directorio /home/analyst/lab.support.files/, ingrese a ese directorio:

```
[analyst@secOps ~]$ cd /home/analyst/lab.support.files/
```

c. Escriba el siguiente comando para generar una lista del contenido del archivo de texto letter_to_grandma.txt en la pantalla:

```
[analyst@secOps lab.support.files]$ cat letter_to_grandma.txt
Hi Grandma,
I am writing this letter to thank you for the chocolate chip cookies you sent me. I
got them this morning and I have already eaten half of the box! They are absolutely
delicious!


I wish you all the best. Love,
Your cookie-eater grandchild.
```

d. En la misma ventana del terminal, ingrese el siguiente comando para generar un ha sh del archivo de texto. El comando utilizara SHA-2-256 como algoritmo de hashing para generar un hash del archivo de texto. El hash aparecerá en la pantalla después de que OpenSSL lo haya computado.

```
[analyst@secOps lab.support.files]$ openssl sha256 letter_to_grandma.txt
```

```
SHA256(letter_to_grandma.txt)=
deff9c9bbece44866796ff6cf21f2612fbb77aa1b2515a900bafb29be118080b
```

Observe el formato de la salida. OpenSSL muestra el algoritmo de hashing utilizado, SHA-256, seguido por el nombre del archivo utilizado como datos de entrada. El hash SHA-256 se muestra a si mismo después del signo igual (´=´).

e. Las funciones de hash son útiles para verificar la integridad de los datos independientemente de que se traten de una imagen, una canción o un simple archivo de texto. El cambio más pequeño produce un hash completamente diferente. Los hashes se pueden calcular antes y después de la transmisión, para luego compararlos. Si los hashes no coinciden, los datos fueron modificados durante la transmisión.

Modifique el archivo de texto letter_to_grandma.txt y vuelva a calcular el hash MD5. Ingrese el siguiente comando para abrir **nano**, un editor de texto operado por línea de comandos (command-line text editor).

```
[analyst@secOps lab.support.files]$ nano letter_to_grandma.txt
```

Utilice **nano** para cambiar la primera oración de **'Hi Grandma'** a **'Hi Grandpa'**. Observe que solo estamos cambiando un carácter: la **'m'** a una **'p'**. Después de hacer el cambio, presione las teclas **<CONTROL+X>** para guardar el archivo modificado. Presione **'Y'** (Sí) para confirmar el nombre y guardar el archivo. Presione la tecla **<Enter>** y saldrá del editor nano para continuar con el siguiente paso.

f. Ahora que se ha modificado y guardado el archivo, vuelva a ejecutar el comando para generar un hash SHA-2-256 del archivo.

```
[analyst@secOps lab.support.files]$ openssl sha256 letter_to_grandma.txt
SHA256(letter_to_grandma.txt)=
43302c4500b7c4b8e574ba27a59d83267812493c029fd054c9242f3ac73100bc
```

¿El hash nuevo, es diferente al calculado en el punto (d.)? ¿Qué tan diferente?

g. También se puede usar un algoritmo de hashing con una longitud de bits más larga, como SHA-2-512. Utilice el siguiente comando para generar un hash SHA-2-512 del archivo letter_to_grandma.txt :

```
[analyst@secOps lab.support.files]$ openssl sha512 letter_to_grandma.txt
SHA512(letter_to_grandma.txt)=
7c35db79a06aa30ae0f6de33f2322fd419560ee9af9cedeb6e251f2f1c4e99e0bbe5d2fc32ce5
01468891150e3be7e288e3e568450812980c9f8288e3103a1d3
[analyst@secOps lab.support.files]$
```

h. Utilice **sha256sum** y **sha512sum** para generar un hash de SHA-2-256 y SHA-2-512 del archivo letter_to_grandma.txt:

```
[analyst@secOps lab.support.files]$ sha256sum letter_to_grandma.txt
43302c4500b7c4b8e574ba27a59d83267812493c029fd054c9242f3ac73100bc
letter_to_grandma.txt


[analyst@secOps lab.support.files]$ sha512sum letter_to_grandma.txt
7c35db79a06aa30ae0f6de33f2322fd419560ee9af9cedeb6e251f2f1c4e99e0bbe5d2fc32ce5
01468891150e3be7e288e3e568450812980c9f8288e3103a1d3 letter_to_grandma.txt
```

¿Los hashes generados con **sha256sum** y **sha512sum**,coinciden con los hashes generados en los puntos (f.) y (g.) respectivamente? Explique.

Nota: SHA-2 es el estándar recomendado para hashing. Si bien SHA-2 aún no se ha visto comprometido de manera efectiva, las computadoras se vuelven cada vez más poderosas. Se espera que esta evolución natural pronto permita que los atacantes comprometan (vulneren) SHA-2.

SHA-3 es el algoritmo de hashing más nuevo, y eventualmente será el reemplazo de la familia de hashes SHA-2.

Nota: La VM CyberOps Workstation solo soporta SHA-2-224, SHA-2-256 y SHA-2-512 (**sha224sum, sha256sum**y **sha512sum**, respectivamente).

Parte 2: Verificar Hashes

Como se mencionó antes, un uso común de los hashes es verificar la integridad de los archivos. Sigan los pasos que se detallan a continuación para utilizar hashes SHA-2-256 y verificar la integridad de sample.img, un archivo que se descargó de Internet.

a. Junto con sample.img, también se descargó sample.img_SHA256.sig, el cual es un archivo que contiene el SHA-2-256 computado (calculado) por el sitio web. Primero, utilice el comando cat para mostrar el contenido del archivo sample.img_SHA256.sig:

```
[analyst@secOps lab.support.files]$ cat sample.img_SHA256.sig
c56c4724c26eb0157963c0d62b76422116be31804a39c82fd44ddf0ca5013e6a
```

b. Utilicen SHA256sum para calcular el hash SHA-2-256 del archivo sample.img:

```
[analyst@secOps lab.support.files]$ sha256sum sample.img
c56c4724c26eb0157963c0d62b76422116be31804a39c82fd44ddf0ca5013e6a sample.img
```

¿Se descargó el archivo sample.img sin ningún tipo de error? Explique.

Nota: Si bien comparar hashes un método relativamente robusto para detectar errores de transmisión, hay mejores formas de garantizar que el archivo no ha sido modificado. Herramientas, como **gpg**, proporcionan métodos mucho mejores para garantizar que el archivo descargado no haya sido modificado por ningún tercero, y que es efectivamente el archivo que el editor quería publicar.