Lab - Hashing Things Out

Objectives

Hashing a Text File with OpenSSL

Part 2: Verifying Hashes

Background / Scenario

Hash functions are mathematical algorithms designed to take data as input and generate a fixed-size, unique string of characters, also known as the hash. Designed to be fast, hash functions are very hard to reverse; it is very hard to recover the data that created any given hash, based on the hash alone. Another important property of hash function is that even the smallest change done to the input data yields a completely different hash.

While OpenSSL can be used to generate and compare hashes, other tools are available. Some of these tools are also included in this lab.

Required Resources

- CyberSec Workstation virtual machine

Instructions

Hashing a Text File with OpenSSL

OpenSSL can be used as a standalone tool for hashing. To create a hash of a text file, follow the steps below:

a. In the CyberSec Workstation virtual machine, open a terminal window.

b. Use you preferred text editor to create a text file in your current directory, the following example use echo to create a file:

```
[CyberSec@ubuntu:~]$ echo “This is a plaintext file” > plaintext.txt
```

c. Type the command below to list the contents of the plaintext.txttext file on the screen:

```
[CyberSec@ubuntu:~]$ cat plaintext.txt
This is a plaintext file
```

d. From the terminal window, issue the command below to hash the text file. The command will use SHA-2-256 as the hashing algorithm to generate a hash of the text file. The hash will be displayed on the screen after OpenSSL has computed it.

```
[CyberSec@ubuntu:~]$ openssl sha256 plaintext.txt
SHA256(plaintext.txt)=
deff9c9bbece44866796ff6cf21f2612fbb77aa1b2515a900bafb29be118080b
```

Notice the format of the output. OpenSSL displays the hashing algorithm used, SHA-256, followed by the name of file used as input data. The SHA-256 hash itself is displayed after the equal (‘=’) sign.

e. Hash functions are useful for verifying the integrity of the data regardless of whether it is an image, a song, or a simple text file. The smallest change results in a completely different hash. Hashes can be calculated before and after transmission, and then compared. If the hashes do not match, then data was modified during transmission.

Let’s modify the plaintext.txt text file and recalculate the MD5 hash.

```
[CyberSec@ubuntu:~]$ echo “new line” >> plaintext.txt
```

f. Now that the file has been modified and saved, run the same command again to generate a SHA-2-256 hash of the file.

```
[CyberSec@ubuntu:~]$ openssl sha256 plaintext.txt
SHA256(plaintext.txt)=
43302c4500b7c4b8e574ba27a59d83267812493c029fd054c9242f3ac73100bc
```

Is the new hash different that hash calculated in item (d)? How different?

g. A hashing algorithm with longer bit-length, such as SHA-2-512, can also be used. To generate a SHA-2-512 hash of the plaintext.txtfile, use the command below:

```
[CyberSec@ubuntu:~]$ openssl sha512 plaintext.txt
SHA512(plaintext.txt)=
7c35db79a06aa30ae0f6de33f2322fd419560ee9af9cedeb6e251f2f1c4e99e0bbe5d2fc32ce5
01468891150e3be7e288e3e568450812980c9f8288e3103a1d3
[CyberSec@ubuntu:~]$
```

h. Use **sha256sum** and **sha512sum** to generateSHA-2-256 and SHA-2-512 hash of the plaintext.txtfile:

```
[CyberSec@ubuntu:~]$ sha256sum plaintext.txt
43302c4500b7c4b8e574ba27a59d83267812493c029fd054c9242f3ac73100bc
plaintext.txt

[CyberSec@ubuntu:~]$ sha512sum plaintext.txt
7c35db79a06aa30ae0f6de33f2322fd419560ee9af9cedeb6e251f2f1c4e99e0bbe5d2fc32ce5
01468891150e3be7e288e3e568450812980c9f8288e3103a1d3  plaintext.txt
```

Do the hashes generated with **sha256sum** and **sha512sum** match the hashes generated in items (f) and (g), respectively? Explain.

Note: SHA-2 is the recommended standard for hashing. While SHA-2 has not yet been effectively compromised, computers are becoming more and more powerful. It is expected that this natural evolution will soon make it possible for attackers to break SHA-2.

SHA-3 is the newest hashing algorithm and eventually be the replacement for SHA-2 family of hashes.