My Project Using CAT and CMP Command

Maria Audu

This scenario portrays a security analyst tasked to implement security controls in order to protect an organization against a range of threats.

A malicious program may mimic an original program. If one code line is different from the original program, it produces a different hash value. Security teams can then identify the malicious program and work to mitigate the risk.

In this activity, the task is to create hash values for two files and use Linux commands to manually examine the differences.

Scenario

In this scenario, the task is to investigate whether two files are identical or different.

First, to display the contents of two files and create hashes for each file. **Next**, to examine the hashes and compare them.

Task 1. Generate hashes for files

My home directory, /home/analyst, is the current working directory. This directory contains two files file1.txt and file2.txt, which contains same data.

The task is to display the contents of each of these files, then generate a hash value for each of these files and send the values to new files, then examine the differences in these values later.

1. The command to list the contents of the directory is **ls**

The command:

ls

Two files, file1.txt and file2.txt, are listed.

2. The cat command displays the contents of the file1.txt file:

cat file1.txt

3. The cat command displays the contents of the file2.txt file:

cat file2.txt

4. Reviewing the output of the two file contents:

   analyst@4fb6d613b6b0:-$ cat file1.txt

   X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

   analyst@4fb6d613b6b0:-$ cat file2.txt

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

The question is, do the contents of the two files appear identical when you use the cat command?

The answer is **Yes.** The contents of the two files appear identical when I use the cat command to display the file contents.

Although the contents of both files appear identical when you use the cat command, I move on to generate the hash for each file to determine if the files are actually different.

5. Using the sha256sum command to generate the hash of the file1.txt file:

sha256sum file1.txt

I followed the same step for the file2.txt file.

6. Using the sha256sum command to generate the hash of the file2.txt file:

sha256sum file2.txt

7. Reviewing the generated hashes of the contents of the two files:

   analyst@4fb6d613b6b0:-$ sha256sum file1.txt

   131f95c51cc819465fa1797f6ccacf9d494aaaff46fa3eac73ae63ffbdfd8267 file1.txt

   analyst@4fb6d613b6b0:-$ sha256sum file2.txt

2558ba9a4cad1e69804ce03aa2a029526179a91a5e38cb723320e83af9ca017b file2.txt

The question is, do both files produce the same generated hash value?

The answer is **No.** The generated hash value for file1.txt is different from the generated hash value for file2.txt, which indicates that the file contents are not identical.

I have completed this task and used the sha256sum command to generate hash values for the file1.txt and file2.txt files.

Task 2. Compare hashes

In this task, the assignment is to write the hashes to two separate files and then compare them to find the difference.

1. Using the sha256sum command to generate the hash of the file1.txt file, and then send the output to a new file called file1hash:

sha256sum file1.txt >> file1hash

I followed same step for the file2.txt file.

2. Using the sha256sum command to generate the hash of the file2.txt file, and sent the output to a new file called file2hash:

sha256sum file2.txt >> file2hash

I now have two hashes written to separate files. The first hash was written to the file1hash file, and the second hash was written to the file2hash file.

I can manually display and compare the differences.

3. Using the cat command to display the hash values in the file1hash and file2hash files.

The command:

cat file1hash

cat file2hash

4. Inspecting the output and noting the difference in the hash values.

Although the content in file1.txt and file2.txt previously appeared identical, the hashes written to the file1hash and file2hash files are completely different.

Next task is to use the **cmp** command to compare the two files byte by byte. If a difference is found, the command reports the byte and line number where the first difference is found.

5. Using the **cmp** command to highlight the differences in the file1hash and file2hash files:

cmp file1hash file2hash

6. Reviewing the output, which reports the first difference between the two files:

analyst@4fb6d613b6b0:-$ cmp file1hash file2hash

file1hash file2hash differ: char1, line 1

The output of the cmp command indicates that the hashes differ at the first character in the first line.

The question is, based on the hash values, is file1.txt different from file2.txt?

The answer is **Yes;** the contents of the two files are different because the hash values of each file are different.

I completed this task and used the **cat** and **cmp** commands to compare the hashes in the file1hash and file2hash files.

.

I practiced how to:

- Compute hashes using sha256sum,
- Display hashes using the cat command, and
- Compare hashes using the cmp command.