## **Task 1. Generate hashes for files**

The lab starts in your home directory, /home/analyst, as the current working directory. This directory contains two files file1.txt and file2.txt, which contain same data.

In this task, you need to display the contents of each of these files. You’ll then generate a hash value for each of these files and send the values to new files, which you’ll use to examine the differences in these values later.

1. Use the ls command to list the contents of the directory.

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

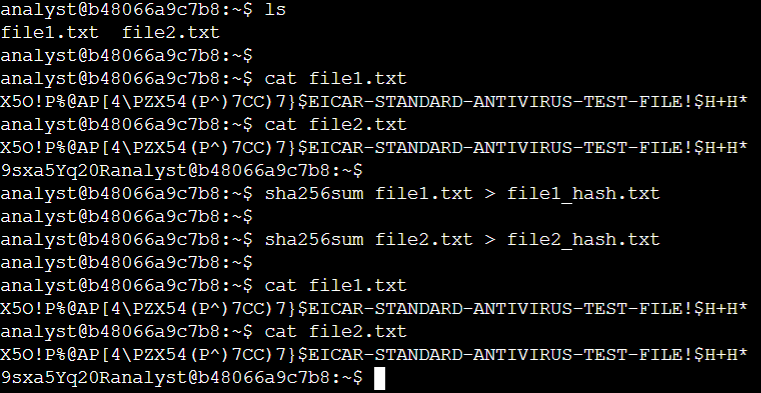
1. Use the cat command to display the contents of the file1.txt file:

cat file1.txt

***Note:*** *If you enter a command incorrectly and it fails to return to the command-line prompt, you can press* ***CTRL+C*** *to stop the process and force the shell to return to the command-line prompt.*

1. Use the cat command to display the contents of the file2.txt file:

cat file2.txt



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

1. Use the sha256sum command to generate the hash of the file1.txt file:

sha256sum file1.txt

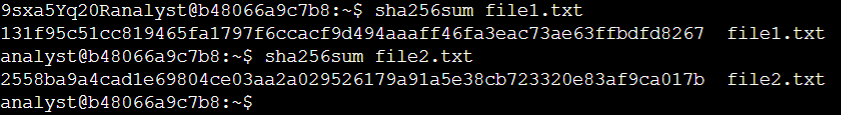
Copied!

content\_copy

You now need to follow the same step for the file2.txt file.

1. Use the sha256sum command to generate the hash of the file2.txt file:

sha256sum file2.txt



## **Task 2. Compare hashes**

In this task, you’ll write the hashes to two separate files and then compare them to find the difference.

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

sha256sum file1.txt >> file1\_hash.txt

You now need to complete the same step for the file2.txt file.

1. Use the sha256sum command to generate the hash of the file2.txt file, and send the output to a new file called file2hash:

sha256sum file2.txt >> file2\_hash.txt

Now, you should 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.

You can manually display and compare the differences.

1. Use the cat command to display the hash values in the file1\_hash.txt and file2\_hash.txt files.
2. Inspect the output and note the difference in the hash values.

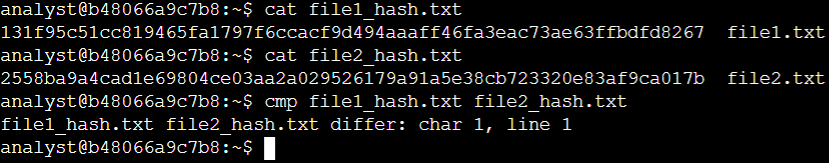
***Note:*** *Although the content in file1.txt and file2.txt previously appeared identical, the hashes written to the file1hash and file2hash files are* ***completely*** *different.*

Now, you can 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.

1. Use the cmp command to highlight the differences in the file1hash and file2hash files:

cmp file1\_hash.txt file2\_hash.txt

1. Review the output, which reports the first difference between the two files:

**Conclusion**

I practiced how to

* compute hashes using sha256sum,
* display hashes using the cat command, and
* compare hashes using the cmp command.

These are valuable tools you can use to validate data integrity as you contribute to the control of your organization’s security.