CREATE HASH VALUES

Pedro Osorio Lopez.

1. High-Level Project Description:

*In this project, you'll create and evaluate the hash values for two files. You will use Linux commands to calculate the hash of two files and observe any differences in the hashes produced. Then, you'll determine if the files are the same, or different.

2. Project Setup and Required Tools:

*Only requirement for this project is setting up a Linux CLI to execute the required commands.

3. Step by Step Project Walkthrough

*We will complete multiple tasks in this project: List the contents of the home directory, compare the plain text of the two files presented for hashing, compute the sha256sum hash of the two separate files, and compare the hashes provided to identify the differences,

*We will start by using the command ls to list the contents of the home directory

```
analyst@78dbc732e0de:~$ ls
file1.txt  file2.txt
```

*We can see that we are working with two files named file1.txt and file2.txt respectively. Now we will proceed to display the contents of these two files with the cat command:

```
analyst@78dbc732e0de:~$ cat file1.txt
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
analyst@78dbc732e0de:~$ cat file2.txt
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

*At first glance, the contents of the two files seem identical. However, to determine if both files are actually the same, it is necessary to compute and evaluate their respective hashes. We will so this by using the sha256sum command and displaying the contents as follows:

```
analyst@78dbc732e0de:~$ sha256sum file1.txt >> file1hash
analyst@78dbc732e0de:~$ sha256sum file2.txt >> file2hash
analyst@78dbc732e0de:~$ cat file1hash
131f95c51cc819465fa1797f6ccacf9d494aaaff46fa3eac73ae63ffbdfd8267  file1.txt
analyst@78dbc732e0de:~$ cat file2hash
2558ba9a4cad1e69804ce03aa2a029526179a91a5e38cb723320e83af9ca017b  file2.txt
```

*We can already see that the hashes look different, so the files are not the same. Lastly, there is a command called cmp that actually compares the hashes itself, which can be useful for checking large hashes:

```
analyst@78dbc732e0de:~$ cmp file1hash file2hash
file1hash file2hash differ: char 1, line 1
analyst@78dbc732e0de:~$
```

4. Summary and/or Recommendations:

*In this project, we practiced how to compute hashes with the sha256sum command, display their contents with ls, and compare them using cmp, which are valuable tools to protect and validate data integrity in our organization.