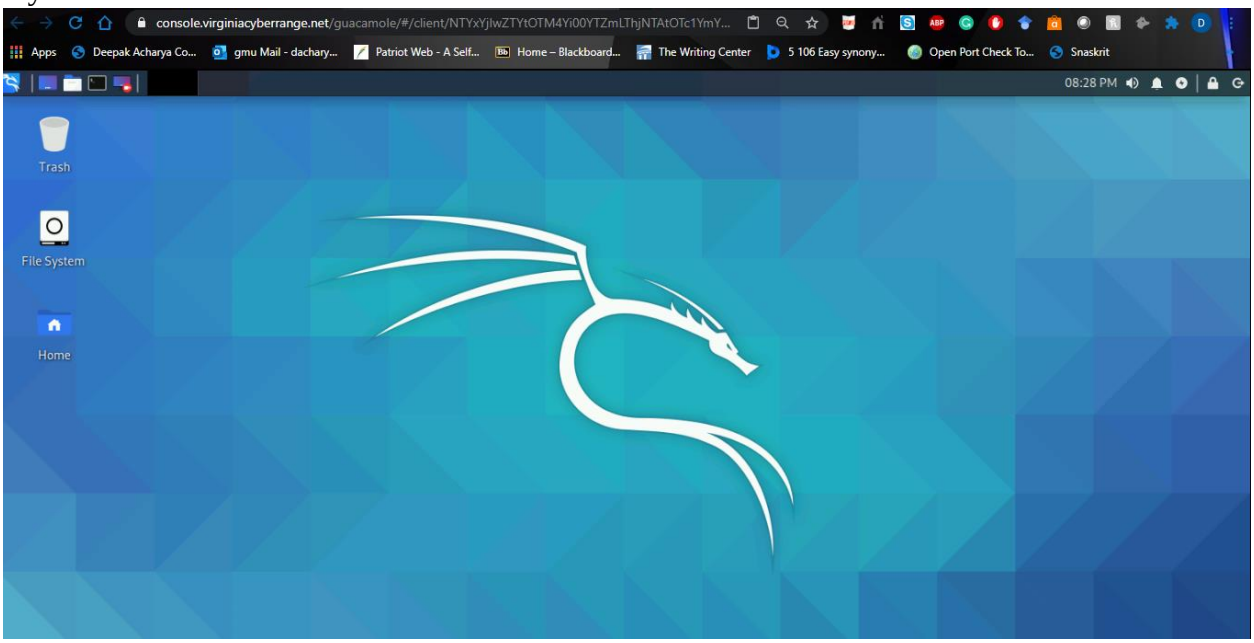
Hashing Collision Lab

1. **Overview**
   This lab will provide the concept of Hashing and Hashing Collision. Hashing is the process of converting a given key into another value by using the hash function. When the wo hashes match then it is known as hashing Collision. We will

2. **Resources Required**
   This lab requires the access to Virginia Cyber Range Linux machine. The tools that we will be using in this lab are md5sum, sha256sum, and sha512sum to calculate different hashes.

3. **Initial Setup**
   For this lab, I logged into the Virginia Cyber Range and selected the course 230. Then I selected the Cyber Basics Environment and started the Linux machine. The screenshot of my Linux machine is shown below.



4. **Tasks**
   The main purpose of this lab is to see if the hashes of two different hashed files matches. The first step will be hashing using SHA 256. For this, I went to the Linux terminal and downloaded the GitHub repository from this link https://github.com/cyber-org/collision-lab. We will get a directory called collision-lab shown in the screenshot.

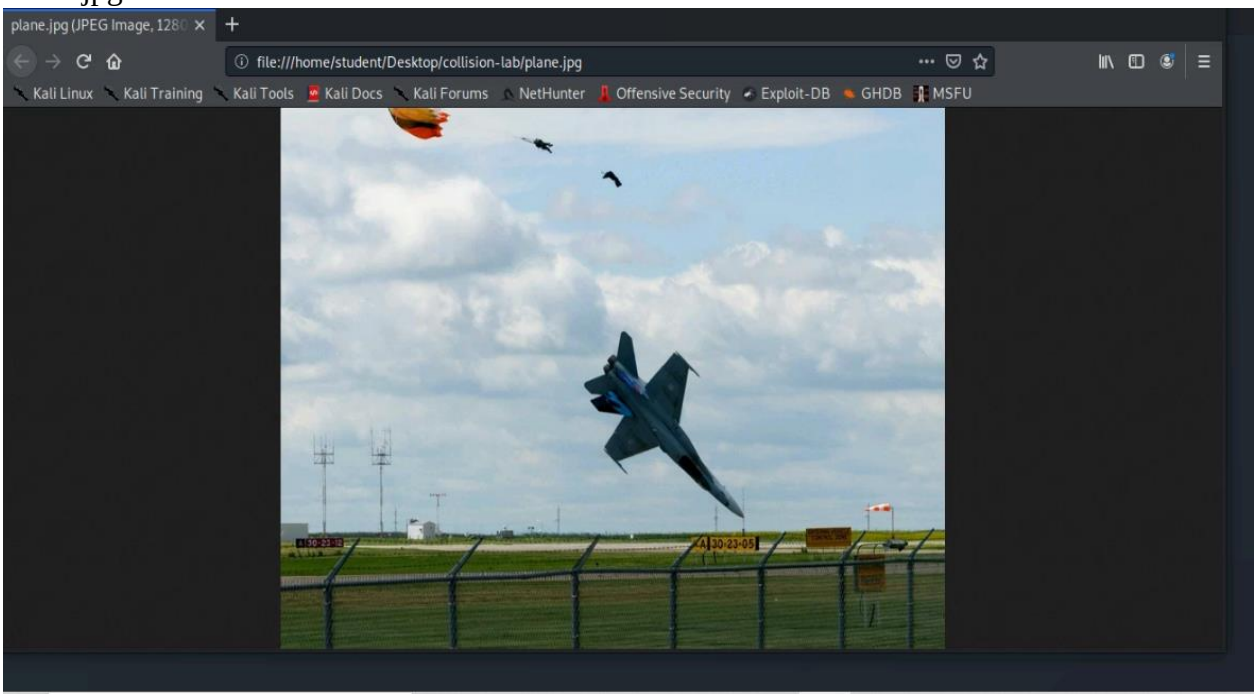
Lab 3: Hashing Collision Lab

```
student@kali:~/Desktop$ git clone https://github.com/cyber-org/collision-lab
Cloning into 'collision-lab'...
remote: Enumerating objects: 26, done.
remote: Counting objects: 100% (26/26), done.
remote: Compressing objects: 100% (25/25), done.
remote: Total 26 (delta 8), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (26/26), 1.01 MiB | 13.08 MiB/s, done.
student@kali:~/Desktop$ ls
collision-lab
student@kali:~/Desktop$
```

With the command "**xdg-open fileneame**", we got the following two different images

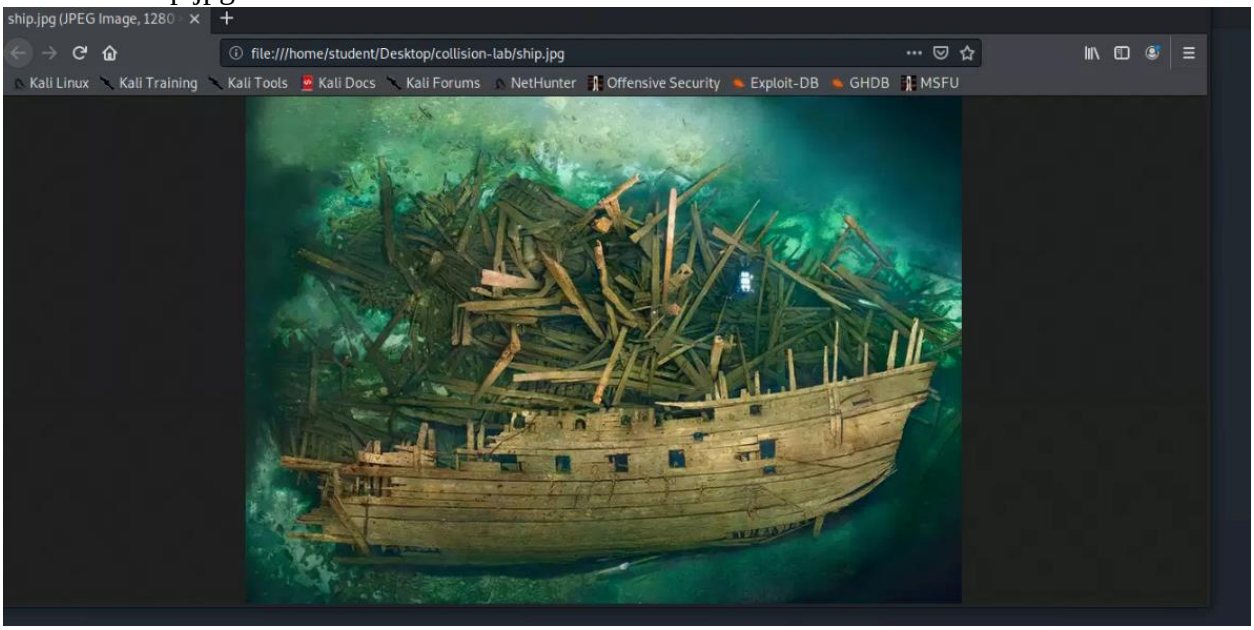
```
student@kali:~/Desktop$ cd collision-lab
student@kali:~/Desktop/collision-lab$ ls
README.md  TaleofTwoCities.txt  erase  hello  plane.jpg  ship.jpg
student@kali:~/Desktop/collision-lab$ xdg-open plane.jpg
student@kali:~/Desktop/collision-lab$ Sandbox: seccomp sandbox violation: pid 1419, tid 1419, syscall 315, args 1419 140257949793664 56 0 10 140257949793664.
Sandbox: seccomp sandbox violation: pid 1448, tid 1448, syscall 315, args 1448 140398354967552 56 0 10 140398354967552.
Sandbox: seccomp sandbox violation: pid 1502, tid 1502, syscall 315, args 1502 139671094168256 56 0 10 139671094168256.
^C
student@kali:~/Desktop/collision-lab$ xdg-open ship.jpg
student@kali:~/Desktop/collision-lab$ Sandbox: seccomp sandbox violation: pid 1654, tid 1654, syscall 315, args 1654 139723046968832 56 0 10 139723046968832.
Sandbox: seccomp sandbox violation: pid 1692, tid 1692, syscall 315, args 1692 139877049504192 56 0 10 139877049504192.
Sandbox: seccomp sandbox violation: pid 1725, tid 1725, syscall 315, args 1725 140046184655232 56 0 10 140046184655232.
^C
student@kali:~/Desktop/collision-lab$
```

Plane.jpg



Lab 3: Hashing Collision Lab

Result of Ship.jpg



Now we will try to see the SHA 256 hashing of these two different files.

```
student@kali:~/Desktop/collision-lab$ sha256sum plane.jpg
91e34644af1e6c36166e1a69d915d8ed5dbb43ffd62435e70059bc76a742daa6  plane.jpg
student@kali:~/Desktop/collision-lab$ sha256sum ship.jpg
caf110e4aebe1fe7acef6da946a2bac9d51edcd47a987e311599c7c1c92e3abd  ship.jpg
student@kali:~/Desktop/collision-lab$
```

The above is the result of SHA 256, which are different for two different images.

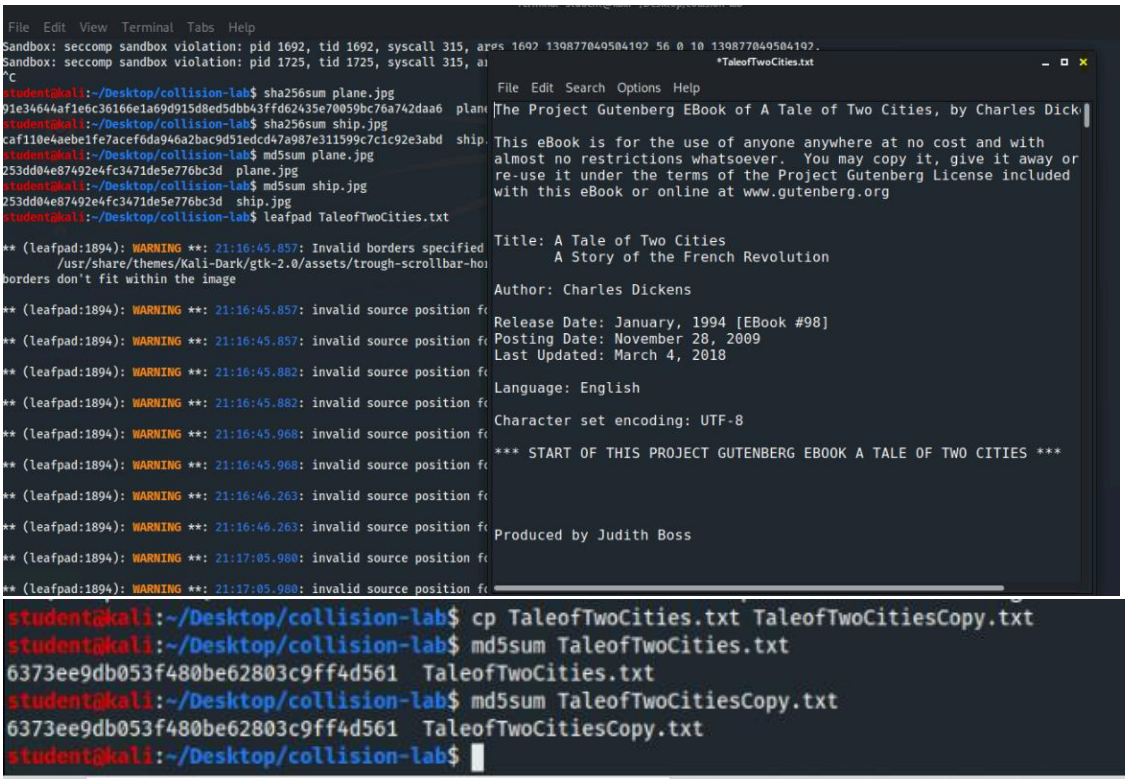
Next, we will see the MD5 hashes for these two files and the result is as follows:

```
student@kali:~/Desktop/collision-lab$ md5sum plane.jpg
253dd04e87492e4fc3471de5e776bc3d  plane.jpg
student@kali:~/Desktop/collision-lab$ md5sum ship.jpg
253dd04e87492e4fc3471de5e776bc3d  ship.jpg
student@kali:~/Desktop/collision-lab$
```

From the above screenshot, we can see that both hashes are same, which is hashing collision.

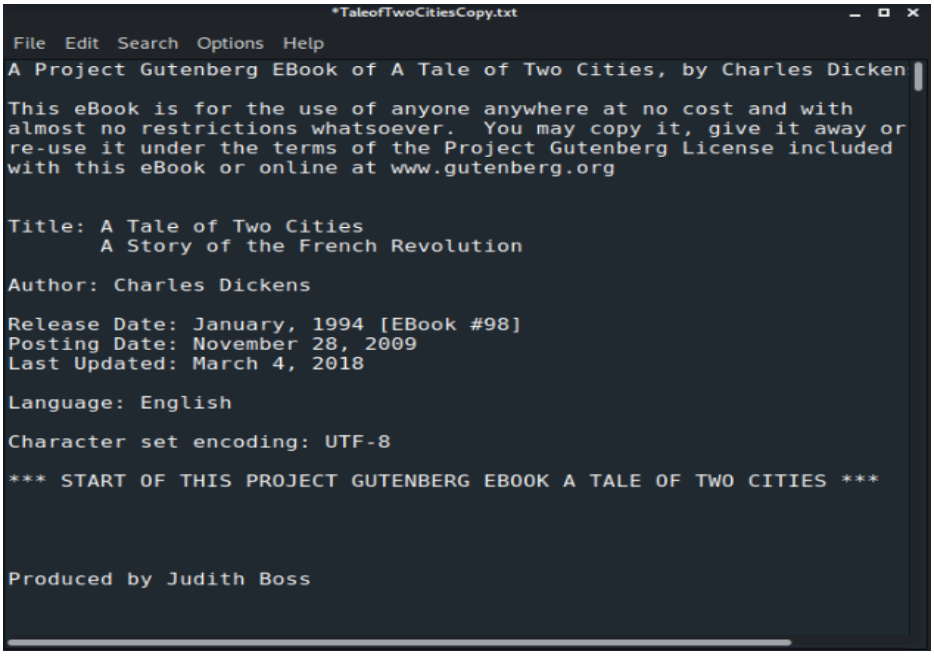
Next step is to look at the hashes of original file and the copy of the same file .

Lab 3: Hashing Collision Lab



The MD5 hash was same for both files as shown in the screenshot above. It is not the hashing collision because this is the copy of same file.

Now, I will make a small change on the copy file and will check the sum again. I changed first word “The” to “A”.



I checked both MD5 and SHA256 hash sum for both original and the edited copy files. And the results were as follows.

Finally, the hash sums were changed.

5. **Lesson Learned**

From the above lab, we can conclude that the MD-5 hashing has shorter key length than SHA-265 (SHA-265 returns 256 bits/ 16 bits hexadecimal hash so SHA is stronger) which makes MD-5 weaker. Thus, MD-5 is too weak to use for confidentiality. Thus, for the cryptographic integrity check needs, SHA-256 is a better choice than MD-5. From this lab we also learned that, if the hash of two different files are same then it is called hashing collision, but it is not collision if different file have same hash for the copy of a file and original file without any edits.