



Cybersecurity 701

Collision Lab



Collision Materials

- Take a look at examples of collision and why it can be dangerous
- Materials needed
 - Kali Linux Machine
- Software Tools used
 - md5sum (Linux command to use MD5 to hash a file)
 - sha256sum (Linux command to use SHA256 to hash a file)



Objectives Covered

- Security+ Objectives (SY0-701)
 - Objective 2.4 - Given a scenario, analyze indicators of malicious activity.
 - Cryptographic Attacks
 - Birthday
 - Collision



What is Collision?

- Collision is when the hashes of two different hashed files match
 - Two different files
 - Same hash



ship.jpg



plane.jpg

Collision →

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ sha256sum plane.jpg ship.jpg
91e34644af1e6c36166e1a69d915d8ed5dbb43ffd62435e70059bc76a742daa6 plane.jpg
caf110e4aebef1fe7acef6da946a2bac9d51edcd47a987e311599c7c1c92e3abd ship.jpg

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ md5sum plane.jpg ship.jpg
253dd04e87492e4fc3471de5e776bc3d plane.jpg
253dd04e87492e4fc3471de5e776bc3d ship.jpg

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ sha512sum plane.jpg ship.jpg
6451d0a8082905f5910a981e89185446c3f03912bf0f62c216e082f0f37d51f2ec6735553590a5
50520483655165b022863d7e61b54a45534f37448493908e12 plane.jpg
434425fd7ad9ef91ab9805e1a87dc35ecfc4db00b085c4cf2d5984e1c10bda5d0c3b85499b9f36
2dcd66d389ad374ca16bfc902c45fb777a80f09b3b7a773e55 ship.jpg

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$
```

Collision Lab Overview

1. Set up Environment
2. Get Collision Lab Files
3. Collision Example
4. Why Collision should not happen
5. Why is Collision Bad?
6. SHA256 Partial Collision
7. How to Avoid Collision?

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
$ md5sum plane.jpg
253dd04e87492e4fc3471de5e776bc3d  plane.jpg

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
$ md5sum ship.jpg
253dd04e87492e4fc3471de5e776bc3d  ship.jpg
```



Set up Environment

- Log into your range
- Open the Kali Linux Environment
 - You should be on your Kali Linux Desktop



Get Collision Lab Files

- Open the terminal and navigate to the Collision Lab
`cd CourseFiles/Cybersecurity/`
- Navigate into this directory
`cd collision-lab`

```
(kali@10.15.4.52) - [~]
└─$ cd CourseFiles/Cybersecurity/

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity]
└─$ ls
autopsy-lab      cuckoo-lab      metadata-lab
backdoor-shortcut  documents      ransomware-lab
collision-lab    honeypot-lab   steganography-lab

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity]
└─$ cd collision-lab/

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$
```



Collision Example

- View the plane.jpg and ship.jpg

```
xdg-open plane.jpg
```

```
xdg-open ship.jpg
```

 - These Terminal commands will open the images in an image viewer
- Notice how they are two different images
- Exit out of the image viewer (back to terminal) (you may need to use **ctrl+c** to exit the current command)
- Run a SHA-256 checksum on the images

```
sha256sum plane.jpg
```

```
sha256sum ship.jpg
```

 - Notice the different SHA-256 checksum for the images

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ xdg-open plane.jpg

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ xdg-open ship.jpg

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ sha256sum plane.jpg
91e34644af1e6c36166e1a69d915d8ed5dbb43ffd62435e70059bc76a742daa6 plane.jpg

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ sha256sum ship.jpg
caf110e4aebef7e7acef6da946a2bac9d51edcd47a987e311599c7c1c92e3abd ship.jpg
```



Collision Example

- What happens if we run a MD5 checksum on the images?

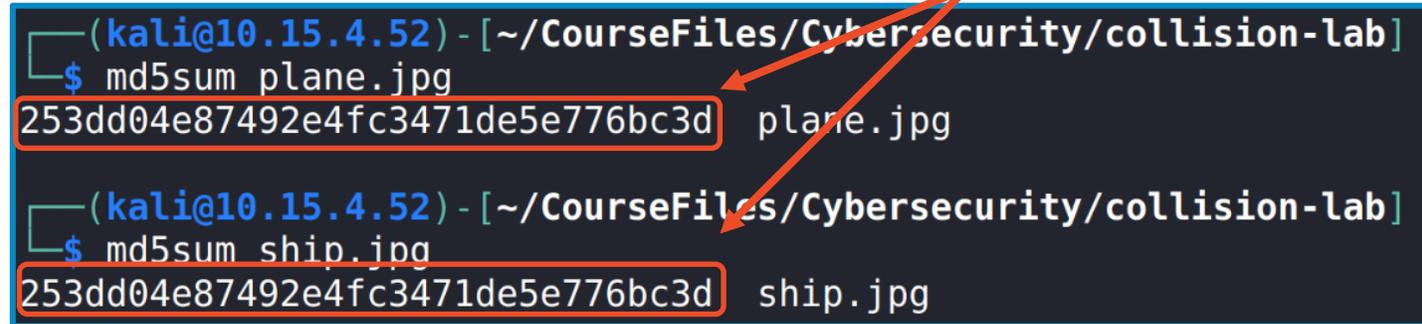
```
md5sum plane.jpg
```

```
md5sum ship.jpg
```

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ md5sum plane.jpg
253dd04e87492e4fc3471de5e776bc3d plane.jpg

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ md5sum ship.jpg
253dd04e87492e4fc3471de5e776bc3d ship.jpg
```

Collision



- Notice the two checksums are the same! We have a collision.

Why Collision should not happen

- Open the copy of Tale of Two Cities with a text editor*

```
leafpad TaleofTwoCities.txt
```

- Notice, this is a copy of the novel A Tale of Two Cities

- Exit out of Leafpad

- Create a copy of TaleofTwoCities.txt

```
cp TaleofTwoCities.txt TaleofTwoCitiesCopy.txt
```

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
$ leafpad TaleofTwoCities.txt
/usr/share/themes/Kali-Dark/gtk-2.0/gtkrc:39: Unable to find include file: "aps.rc"
/usr/share/themes/Kali-Dark/gtk-2.0/gtkrc:40: Unable to find include file: "hacks.rc"
/usr/share/themes/Kali-Dark/gtk-2.0/gtkrc:41: Unable to find include file: "hacks-dark.rc"

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
$ cp TaleofTwoCities.txt TaleofTwoCitiesCopy.txt

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
$
```



Why Collision should not happen

- Check to make sure the MD5 checksums are the same

```
md5sum TaleofTwoCities.txt
md5sum TaleofTwoCitiesCopy.txt
```
- You should notice they have the same checksum
 - This is because they are the same file

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
$ md5sum TaleofTwoCities.txt
6373ee9db053f480be62803c9ff4d561 TaleofTwoCities.txt

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
$ md5sum TaleofTwoCitiesCopy.txt
6373ee9db053f480be62803c9ff4d561 TaleofTwoCitiesCopy.txt
```



Why Collision should not happen

- Make a tiny change in the copy
- Open the copy in a text editor
`leafpad TaleofTwoCitiesCopy.txt`
- Make a minor change and save
- Exit leafpad

```
TaleofTwoCitiesCopy.txt
File Edit Search Options Help
The Project Gutenberg EBook of A Tale of Two Cities, by Charle
This eBook is for the use of anyone anywhere at no cost and wi
almost no restrictions whatsoever. You may copy it, give it a
re-use it under the terms of the Project Gutenberg License inc
with this eBook or online at www.gutenberg.org

Title: A Tale of Two Cities
      A Story of the French Revolution

Author: Charles Dickens
```

```
*TaleofTwoCitiesCopy.txt
File Edit Search Options Help
The Project Gutenberg EBook of A Tale of Two Cities, by Charles Dickens
This eBook is for the use of anyone anywhere at no cost and with
almost no restrictions whatsoever. You may copy it, give it away or
re-use it under the terms of the Project Gutenberg License included
with this eBook or online at www.gutenberg.org.

Title: A Tale of Two Cities
      A Story of the French Revolution

Author: Charles Dickens
```

Added a period after
www.gutenberg.org

Why Collision should not happen

- Check the MD5 hashes again

```
md5sum TaleofTwoCities.txt
md5sum TaleofTwoCitiesCopy.txt
```
- Are the hashes the same? How different are they?
 - Notice one minor change should completely change the hash

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ md5sum TaleofTwoCities.txt
6373ee9db053f480be62803c9ff4d561 TaleofTwoCities.txt

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ md5sum TaleofTwoCitiesCopy.txt
be6beb9d9c3bf495796d5302a5c3fed9 TaleofTwoCitiesCopy.txt
```



Why is Collision Bad?

- If you've run through the File Hashing Lab, you should know that checksums is a way to verify that a file has not been changed or tampered with
- What if a malicious program can have the same checksum of a program that the user wanted to download?
- Take a look at an example of a good program and a malicious program that was made by Peter Selinger*

*Read more about these programs at the following website:

<https://www.mscs.dal.ca/~selinger/md5collision/>



Why is Collision Bad?

- Check the MD5 hashes of the two programs

```
md5sum hello
```

```
md5sum erase
```

- You should notice these two programs have the same MD5 checksums

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ md5sum hello
da5c61e1edc0f18337e46418e48c1290  hello

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ md5sum erase
da5c61e1edc0f18337e46418e48c1290  erase
```

- This is another example of a collision!



Why is Collision Bad?

- Try to run the two programs
- Make the programs executable

```
chmod +x hello
chmod +x erase
```
- Run the programs

```
./hello
./erase
```
- What if a user was trying to download the hello program, verified it with the MD5 checksum, but it was actually the erase program?

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ chmod +x hello

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ chmod +x erase

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ ./hello
Hello, world!

(prompt enter to quit)

(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
└─$ ./erase
This program is evil!!!
Erasing hard drive...1Gb...2Gb... just kidding!
Nothing was erased.

(prompt enter to quit)
```



SHA256 Partial Collision

- While there are still no known SHA256 collisions, there are examples of partial collisions
- Check the SHA256 checksums of the Frank text files

```
sha256sum Frank1.txt Frank2.txt
```

Partial Collision →

```
(kali@10.15.4.52) - [~/CourseFiles/Cybersecurity/collision-lab]
$ sha256sum Frank1.txt Frank2.txt
6026d9b31ba780bb9973e7cfc8c9f74a35b54448d441a61cc9bf8db0fcae5280 Frank1.txt
6026d9b323898bcd7ecdbcbcd575b0a1d9dc22fd9e60074aefcbaade494a50ae Frank2.txt
```

Why are partial collisions still dangerous?

Sometimes a script will only check the first few characters in the checksum, this is to save time and not have to check all of the characters. Also, a human might only glance at the first couple of characters and not the entire checksum.



How to Avoid Collision?

- Avoid using MD5 hashes! (and other weak encryptions)
 - MD5 checksum have been proven to have collisions
- Use stronger encryption methods
 - Check using SHA-256
 - Check using SHA-512
- How else can you avoid collisions?

