|  |  |
| --- | --- |
| Course | Computer Forensic |
| Lab | Lab 02 |
| Name of the Student | Maisha Khatoon |
| Student ID | 100899259 |

# **LAB: 02**

Overview of hashing, file header information, disk images

Objective:

* **Learn about Data integrity, File Headers, view metadata and File Manipulation**

**Activities:**

**Task 1**

File Hashes

**Leaning Activities:**

At the end of these activities, you should understand:

* How to calculate and compare hashes of the files
* How detect NTFS stream data
* How to display contents in steganography

Tools to use:

-Windows 10 Host (VM or host PC doesn’t matter)

-WinHex (available from: <http://www.x-ways.net/winhex.zip>)

-HxD Hex Editor (available from: https://mh-nexus.de/en/downloads.php?product=HxD20)

-Nirsoft Hashmyfiles (available from: <https://www.nirsoft.net/utils/hashmyfiles-x64.zip>)

**Note: Please include a screenshot of each step. Corp your screenshot to show relevant information only. Uncropped screenshots will result in a 10% deduction from your marks.**

**Sample files contained within Lab2.zip**

**Task 1:**

**Hashes**

A Hash (MD5, Sha1, etc) also known as ‘a Cryptographic hash function’ is a “One-Way” function that describes the applicable data in a numeric string of a fixed length. This is often much smaller than the original data. By design a Hash function is not feasible to reverse to derive the described original data.

Diffusion property of hashes: if you change a small amount of the original data (1 bit) due to the construction of the hashing algorithm a large change occurs to the hashing output.

* 1. Open notepad and type the following text:

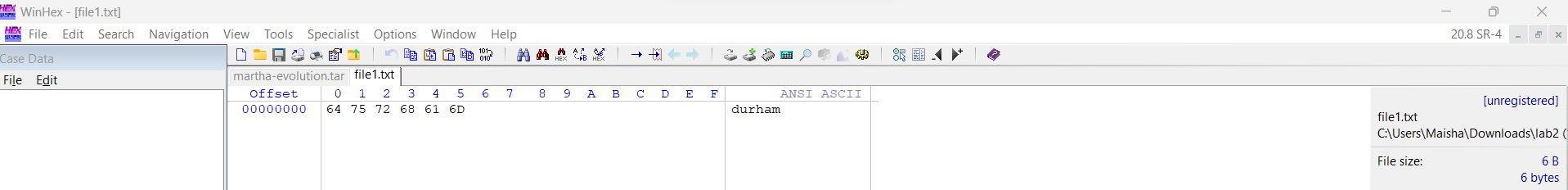
It just says a single word ‘Durham’.

* 1. Save as file1.txt

1.3 Open file1.txt in Winhex and view the contents

In WinHex click ‘file’ then ‘open’ then select file1.txt. On the left is the data contained within the file in hex code, on the right is the ANSI ASCII translation of that hex code.

**Take a screenshot of the message for your lab report.**

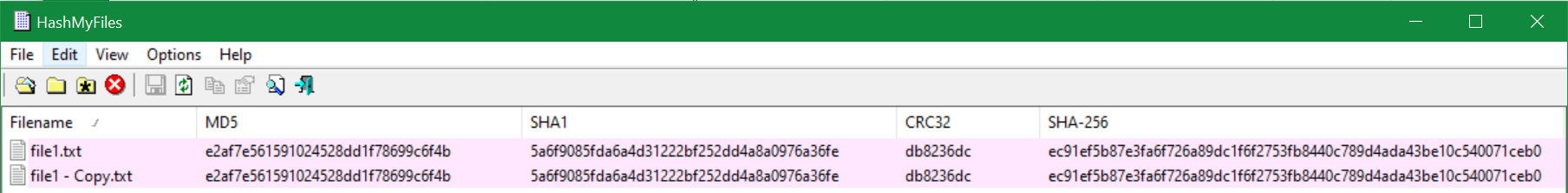
****

1.4 Now, let’s copy file1.txt and paste it. We should now have file1.txt and file1 – Copy.txt in

our folder.

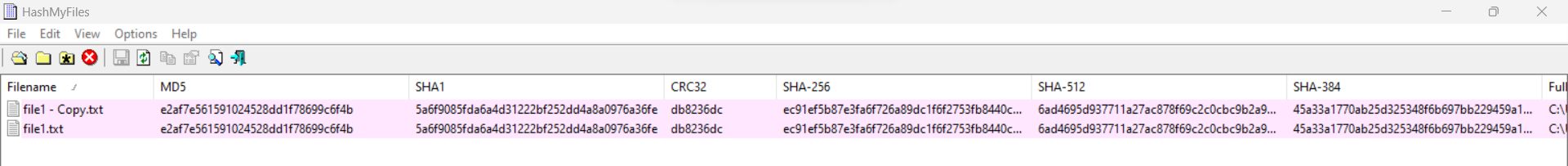
1.5 Let’s use the hash tool (Nirsoft Hashmyfiles) to calculate the hash of both of these files.

At this point, the hash should be the same for both files (exact copies of one-another).



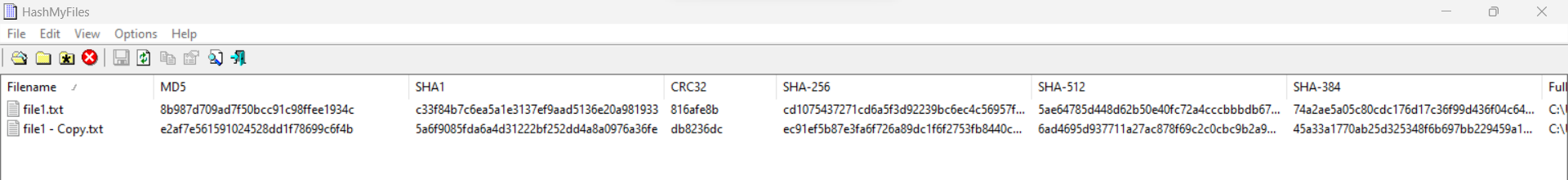
In fact, your file’s hash should match mine. Hashes are deterministic (Figure 1.5).

**Take a screenshot of the message for your lab report.**



1.6 Now open file1.txt in notepad and add a period, save, and rehash the file and compare it to our copy of the original. Notice the difference in the resulting hash value.

**Take a screenshot of the message for your lab report.**



**Task 2**:

A txt file is very simple and doesn’t contain any header detail aside from the content of the

text file (Durham). However, most files have a lot more information and details baked in.

2.1 Let’s take for instance file2.rtf this file is called a rich text file. The file has a lot more features

than a simple txt file and since that’s the case, more information is contained within it. Let’s look at file2.rtf in WinHex.

More complex files often have more functionality, in this case the rtf file contains information

regarding the stored data such as the keyset, font, and other control verbs to help the rtf

reader render the document correctly with the correct format.

2.2 Let’s open up both file2.rtf and file 3.rtf in Wordpad and look at the contents.

Both files say ‘durham’.

2.3 Now let’s run a hash against these files, they appear to be copies of one another, so do

their hashes match?

Why or Why don’t the hashes match? (include your answer in your lab write up)

The reason why the hashes don’t match for both files is that one of the bytes in the hex code of file2 and file3 was not similar.

2.4 Let’s use WinHex or HxD Editor to look a bit deeper into these two files.

Use the Tools/Compare data feature in WinHex or HxD to determine what’s going on with these two files.

2.5 Something is strange on offset (the number of bytes from the beginning of the file) 173

what is it?

The hexcode in file 3 it was “1” while for file2 at offset 173 was “0”

2.7 Let’s open file 4.rtf in wordpad.

We see it says durham, again, looks the same as file 3 and file 2. However, this file is

hiding a dark secret!! Open the file in WinHex to see a hidden message.

**Take a screenshot of the message for your lab report.**

**A screenshot of a computer

Description automatically generated**

This is a form a steganography, that is, hiding information or a message within another file.

2.8 Why would steganography be important to identify in digital forensics?

Digital forensics must be able to recognise steganography because it aids in the discovery of concealed evidence, detection of clandestine communication, prevention of data exfiltration, and defence against cutting-edge cyber threats.

**Task 3**

1. Download s-tools (<http://www.jjtc.com/Steganography/tools.html>)

Save the zip file in C:\task3 folder.

1. Unzip the zip file in C:\task3\steg folder.
2. Drag the zebras.bmp file to your desktop. Do not make a shortcut. The file itself must be moved there.
3. Start S-tools.exe by double clicking on the icon on the desktop. A window will appear.
4. Drag the zebras.bmp file to the S-tools window.
5. Right click on the zebras pictures and select Reveal from the menu.
6. Fill in the 3-character pass phrase 'abc' (without the quotes) in two places. Leave IDEA as the encryption algorithm. Click on OK.
7. Wait until the Revealed Archive dialog box appears. This may take a minute or two.
8. Right click on any item and select Save As to save the file. Repeat for the other ones. These are the hidden files. **Take a screenshot of the message for your lab report.**

A screenshot of a computer

Description automatically generated

**Task 4**

1. Create a s1.txt file and add the following message and save the file:

The hidden phone#:

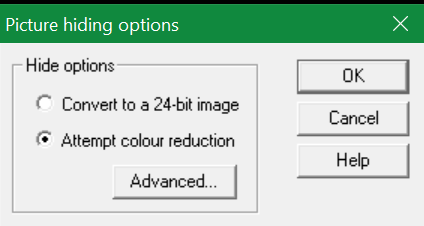
123 455 6666

443 555 3444

890 343 3333

1. Launch s-tools
2. Drag task4.bmp file into the open windows
3. Now drag the “s1.txt” over the top of the BMP image. The next window that you see will be asking you to enter the passphrase and to select your encryption algorithm. Let’s leave the algorithm as the default “IDEA” and type the passphrase "123456" in, If you typed correctly you’ll now see a second picture of the BLACK only this time it will have Hidden Data in the top blue bar.

If the following msg showed up, click OK



1. Now let’s right click that new “hidden data” picture and then Save As and name it “task4\_1.bmp”
2. Now close out of everything and re-open S-Tools.exe. Next, drag the “task4\_1.bmp” file onto the S-Tools window, move your mouse over the picture and right-click. Select the option “Reveal” and you will see the same window you saw before requesting the passphrase. Type in your passphrase, 123456 and click OK.
3. Wait until the Revealed Archive dialog box appears. This may take a minute or two.
4. Right click on any item and select Save As to save the file to desktop.

**Take a screenshot of the message for your lab report.**

A screenshot of a computer

Description automatically generated

**Task5**

1.create a new directory c:\task5

2. Go to a task5 directory and create a txt file

3. Right Click, select New > Text document, enter the name "lab2\_task5.txt"

4. Open the file in notepad and enter text "This is a demo file for ADS.", Save and close.

5. Open a command prompt Start > Run, enter "cmd", click OK

6. In the command window, go to your task5 directory.

7. Type the following to confirm that you have some text in the file: notepad lab2\_task5.txt

A notepad window will pop up and show your lab2\_task5.txt text.

This has all been setup. Now we get to the ADS part.

8.Type the following to save secret notes to a different text file and hide the reference to that text file inside of "lab3\_task6.txt", echo my swiss bank account no 654321 > lab2\_task6.txt:secret.txt

9. Now if you type the following command you will see only your original file:

"notepad lab3\_task6.txt"

Try the following command to see the secret message:

"lab3\_task6.txt:secret.txt"

**Take a screenshot of the message for your lab report.**

A screenshot of a computer

Description automatically generated

**Task6**

Text book, Page 150 & 151

Steps 1-7

**Take a screenshot of the message for your lab report.**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Reflective statements (end-of-exercise):**

You should reflect on these questions:

1. Steganalysis is the process of \_\_\_\_detecting\_\_\_\_\_\_\_\_\_\_\_\_\_\_steganography and \_\_\_recovering\_\_\_\_\_hidden evidence.
2. What is the difference between cryptography and steganography?

Transforming information into a safe code in order to preserve its secrecy and integrity during transmission or storage is known as cryptography. On the other hand, steganography entails concealing the existence of hidden information by enclosing it in media that appears benign. The primary distinction between the two is that while steganography concentrates on disguising the presence of the message, cryptography focuses on protecting the content.

1. What is unique about messages transmitted using steganography?

Steganographic messages are distinct because they are cloaked behind cover media that appears benign, virtually blending in with regular data. The communication itself remains secret, adding another level of secrecy and security, and the buried information is not visible to casual onlookers.

1. What are the basic steganographic concepts or inputs and outputs?

The innocent-appearing data or file, such as an image, audio file, video file, or text document, that hides the secret information.

Secret message: The private or delicate information concealed in the cover materials.

The process or technique utilised to include the hidden message into the cover material is known as steganography.

Stego-media: The finished product once the cover media has been modified to include the hidden message.

Cover media and the covert message are inputs.

Stego-media with the secret message contained inside.

**References**

<https://blogs.msmvps.com/alunj/2020/06/24/revisiting-ntfs-alternate-data-streams/>

<https://docs.microsoft.com/en-us/windows/win32/fileio/file-streams>