FORENSICS
LAB SERIES

Lab 13: Data Carving

Material in this Lab Aligns to the Following Certification Domains/Objectives	
Certified Cyber Forensics Professional (CCFP) Objectives	Computer Hacking Forensic Investigator (CHFI) Objectives
5: Application Forensics	10: Recovering Deleted Files and Deleted Partitions

Document Version: 2016-08-17

Contents

Introduction

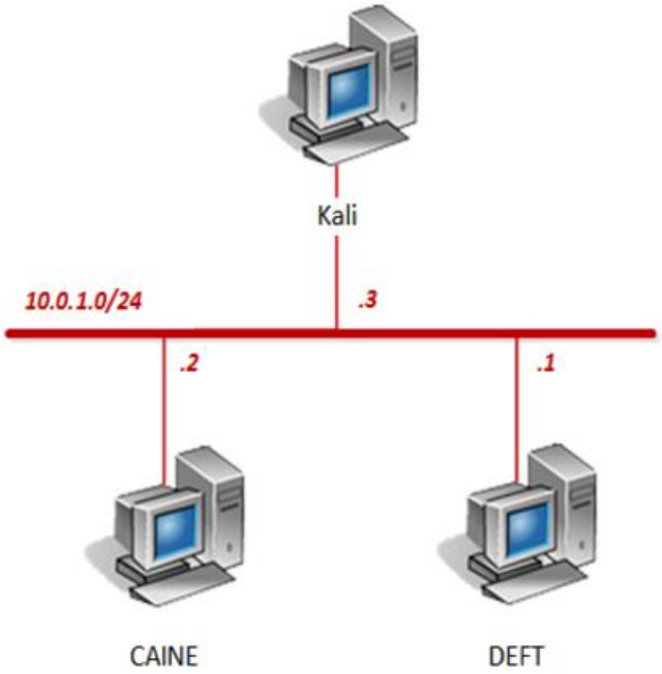
This lab will demonstrate ways to recover files that are deleted, partially deleted or corrupted. The techniques will be demonstrated by using several tools and comparing the results.

Objective

In this lab, you will be conducting forensic practices using various tools. You will be performing the following tasks:

1. Using Scalpel to Carve Files
2. Using Foremost to Carve Files
3. View the Carved Files

Pod Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

Virtual Machine	IP Address	Account (if needed)	Password (if needed)
DEFT	10.0.1.1	deft	password
CAINE	10.0.1.2	caine	
Kali	10.0.1.3	root	toor

1 Using Scalpel to Carve Files

1. Click on the **CAINE** graphic on the *topology page* to open the VM.
2. Open a new terminal by clicking on the **MATE Terminal** icon located in the bottom tool pane.

3. Using the terminal, type the command below followed by pressing the **Enter** key. to view the available arguments for the *Scalpel* tool.

```
scalpel -h
```

```
caine@Caine01:~$ scalpel -h
Scalpel version 2.1
Written by Golden G. Richard III and Lodovico Marziale.
Scalpel carves files or data fragments from a disk image based on a set of
file carving patterns, which include headers, footers, and other information.

Usage: scalpel [-b] [-c <config file>] [-d] [-e] [-h] [-i <file>]
[-n] [-o <outputdir>] [-O] [-p] [-q <clustersize>] [-r]
[-v] [-V] <imgfile> [<imgfile>] ...

Options:
-b  Carve files even if defined footers aren't discovered within
    maximum carve size for file type [foremost 0.69 compat mode].
-c  Choose configuration file.
-d  Generate header/footer database; will bypass certain optimizations
    and discover all footers, so performance suffers.  Doesn't affect
    the set of files carved.  **EXPERIMENTAL**
-e  Do nested header/footer matching, to deal with structured files that may
    contain embedded files of the same type.  Applicable only to
    FORWARD / NEXT patterns.
-h  Print this help message and exit.
-i  Read names of disk images from specified file.  Note that minimal parsing of
    the pathnames is performed and they should be formatted to be compliant C
    strings; e.g., under Windows, backslashes must be properly quoted, etc.
```

Scalpel is designed to recover file fragments or files from a disk image using headers, footers, and other information about the files. Each file type has a unique hex signature that defines their type, for example, "*doc*", "*docx*", etc... A useful website for looking up different file types is: http://www.filesignatures.net/

4. Enter the command below to view the configuration file for *scalpel* that is located in the */etc/scalpel/* directory and take notice of the several premade pattern searches for *gif* and *jpg* file signatures.

```
cat /etc/scalpel/scalpel.conf | less
```

With the *less* command, use the **Enter** key to skip to the next line item or use the **spacebar** to skip by page. When finished, press the **q** character to quit.

```
#-----------------------------------------------------------------
# GRAPHICS FILES
#-----------------------------------------------------------------
#
#
# AOL ART files
#       art     y       150000  \x4a\x47\x04\x0e        \xcf\xc7\xcb
#       art     y       150000  \x4a\x47\x03\x0e        \xd0\xcb\x00\x00
#
# GIF and JPG files (very common)
        gif     y       5000000         \x47\x49\x46\x38\x37\x61        \x00\x3b
        gif     y       5000000         \x47\x49\x46\x38\x39\x61        \x00\x3b
        jpg     y       200000000       \xff\xd8\xff\xe0\x00\x10        \xff\xd9
#
#
# PNG
#       png     y       20000000        \x50\x4e\x47?   \xff\xfc\xfd\xfe
#
```

5. Using the terminal, navigate to the */home/caine/Downloads/11-carve-fat/* directory. Type the command below followed by pressing the **Enter** key.

```
cd Downloads/11-carve-fat
```

```
caine@Caine01:~$ cd Downloads/11-carve-fat/
caine@Caine01:~/Downloads/11-carve-fat$
```

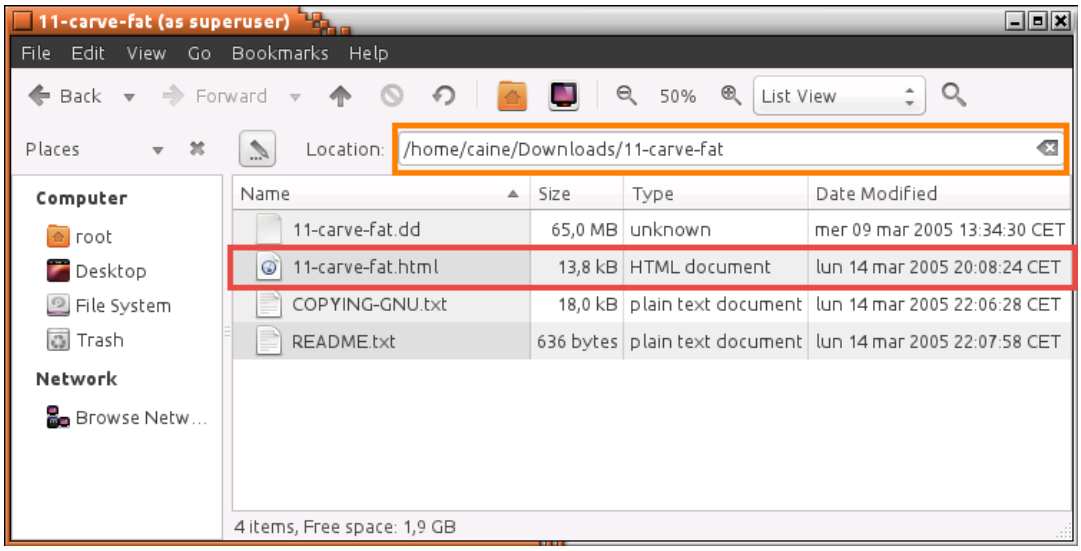
6. List the files in the current directory, enter the command below.

```
ls
```

```
caine@Caine01:~/Downloads/11-carve-fat$ ls
11-carve-fat.dd  11-carve-fat.html  COPYING-GNU.txt  README.txt
caine@Caine01:~/Downloads/11-carve-fat$
```

7. Notice the *11-carve-fat.dd* file. Open the file manager by clicking on the **Caja** icon in the bottom tool pane.

8. Using the file manager, navigate to **/home/caine/Downloads/11-carve-fat/** and double-click on the **11-carve-fat.html** file to open it.



9. Notice that the *Firefox* browser appears. Scroll towards the bottom of the page and briefly analyze the series of files. These are the files that will be carved out of the image.

Files

The following files and the MD5 hash and description were created on the file system.

Num	Name	MD5	Size	Note	Sectors
1	2003_document.doc	e72f388b36f9370f19696b164c308482	19968	A Valid DOC file	(0-38) 281 -320
2	enterprise.wav	7629b89adade055f6783dc1773274215	318895	A valid WAV file	(0-622) 16021 -16644
3	haxor2.jpg	84e1dceac2eb127fef5bfdcb0eae324b	24367	An invalid JPEG with only 1 header byte corrupted. This byte is located at offset 19 within the file.	(0-47)16645 -16692
4	holly.xls	7917baf0219645afef8b381570c41211	23040	A valid XLS file	(0-44) 16693-16738
5	lin_1.2.pdf	e026ec863410725ba1f5765a1874800d	1399508	A linearized PDF	(0-2733) 16741 -19475
6	nlin_14.pdf	5b3e806e8c9c06a475cd45bf821af709	122434	A non-linearized PDF	(0-239) 19477 -19716
7	paul.jpg	37a49f97ed279832cd4f7bd002c826a2	29885	A valid jpeg	(0-58) 19717 -19776

10. Change focus to the **terminal**.

11. Enter the command below to carve a raw image for testing purposes and view the output.

```
scalpel -c /etc/scalpel/scalpel.conf 11-carve-fat.dd
```

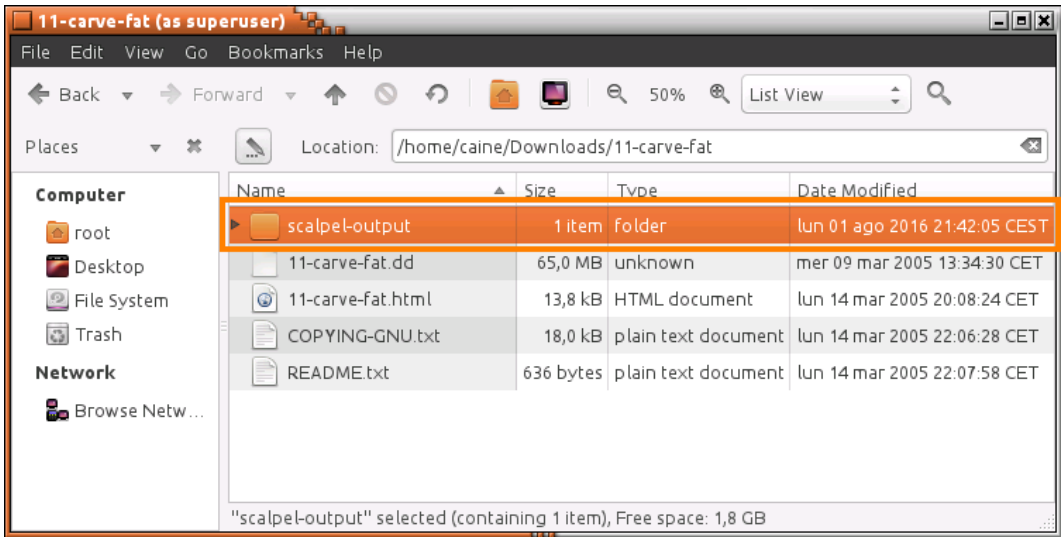
```
caine@Caine01:~/Downloads/11-carve-fat$ scalpel -c /etc/scalpel/scalpel.conf 11-
carve-fat.dd
Scalpel version 2.1
Written by Golden G. Richard III and Lodovico Marziale.
Multi-core CPU threading model enabled.
Initializing thread group data structures.
Creating threads...
Thread creation completed.

Opening target "/home/caine/Downloads/11-carve-fat/11-carve-fat.dd"

Image file pass 1/2.
11-carve-fat.dd: 100.0% |*******************************|   62.0 MB    00:00 ETA
```

At the end of the process, *Scalpel* should successfully finish with 19 files carved.

```
Carving files from image.
Image file pass 2/2.
11-carve-fat.dd: 100.0% |*******************************|   62.0 MB    00:00 ETA
Processing of image file complete. Cleaning up...
Done.
Scalpel is done, files carved = 19, elapsed  = 3 secs.
caine@Caine01:~/Downloads/11-carve-fat$
```


2 Using Foremost to Carve Files

1. Change focus to the **file manager**.
2. Using the file manager, make sure to be viewing the **/home/caine/Downloads/11-carve-fat/** directory and notice that there is now a *scalpel-output* folder. This folder contains the output from *Scalpel*.



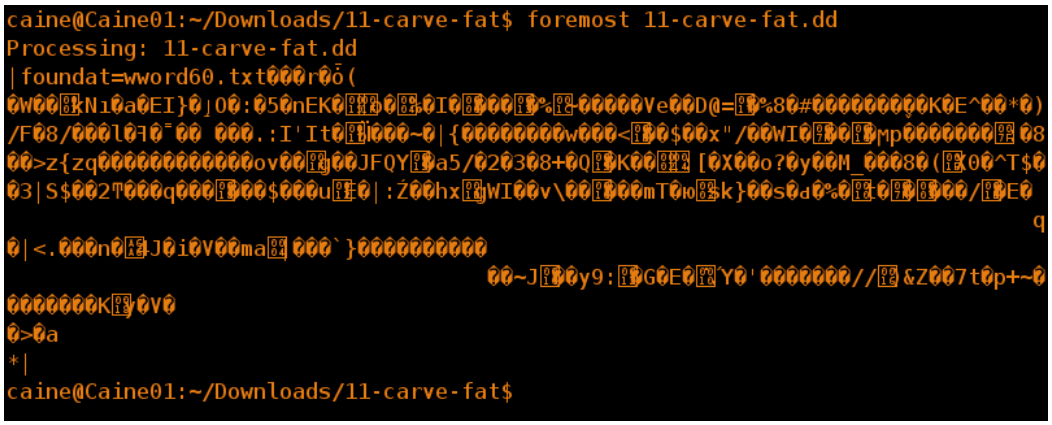
3. Change focus back to the **terminal**.
4. Using the terminal, briefly analyze the configuration file contents for the *Foremost* tool by entering the command below. This file has been modified for this lab.

```
cat /etc/foremost.conf | less
```

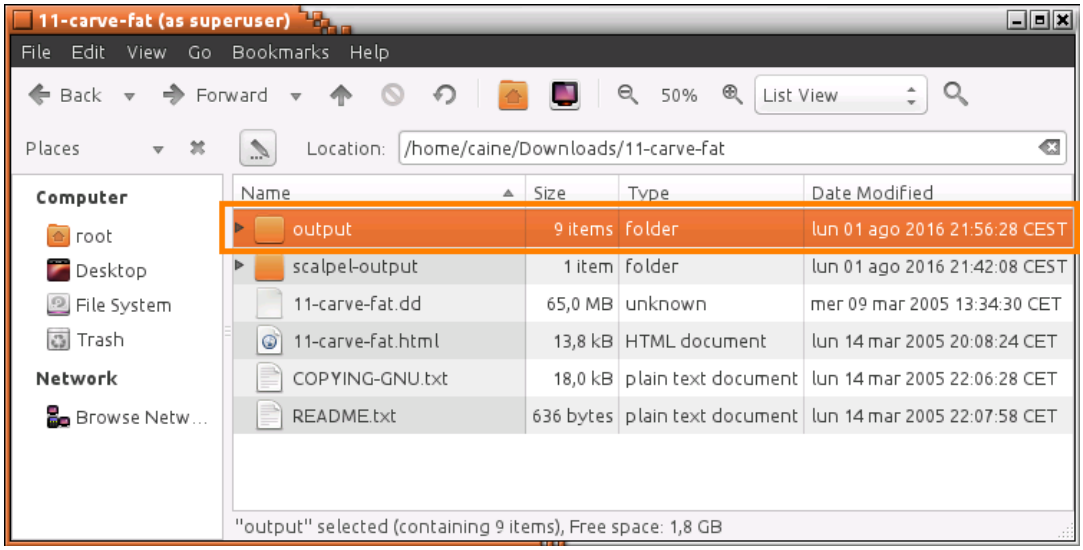
```
#-------------------------------------------------------------------
# GRAPHICS FILES
#-------------------------------------------------------------------
#
#
# AOL ART files
#       art     y       150000  \x4a\x47\x04\x0e        \xcf\xc7\xcb
#       art     y       150000  \x4a\x47\x03\x0e        \xd0\xcb\x00\x00
#
# GIF and JPG files (very common)
#       (NOTE THESE FORMATS HAVE BUILTIN EXTRACTION FUNCTION)
        gif     y       155000000       \x47\x49\x46\x38\x37\x61        \x00\x3b
        gif     y       155000000       \x47\x49\x46\x38\x39\x61        \x00\x00
\x3b
        jpg     y       20000000        \xff\xd8\xff\xe0\x00\x10        \xff\xd9
        jpg     y       20000000        \xff\xd8\xff\xe1 \xff\xd9
        jpg     y       20000000        \xff\xd8        \xff\xd9
#
# PNG   (used in web pages)
#       (NOTE THIS FORMAT HAS A BUILTIN EXTRACTION FUNCTION)
#       png     y       200000  \x50\x4e\x47?   \xff\xfc\xfd\xfe
```


5. Type the command below followed by pressing the **Enter** key to use the Foremost tool on the *11-carve-fat.dd* file to carve again.

```
foremost 11-carve-fat.dd
```



6. Change focus to the **file manager**.
7. Notice now that an *output* folder appears which contains what the *foremost* tool found using its configuration file.



8. Change focus to the **terminal** window.

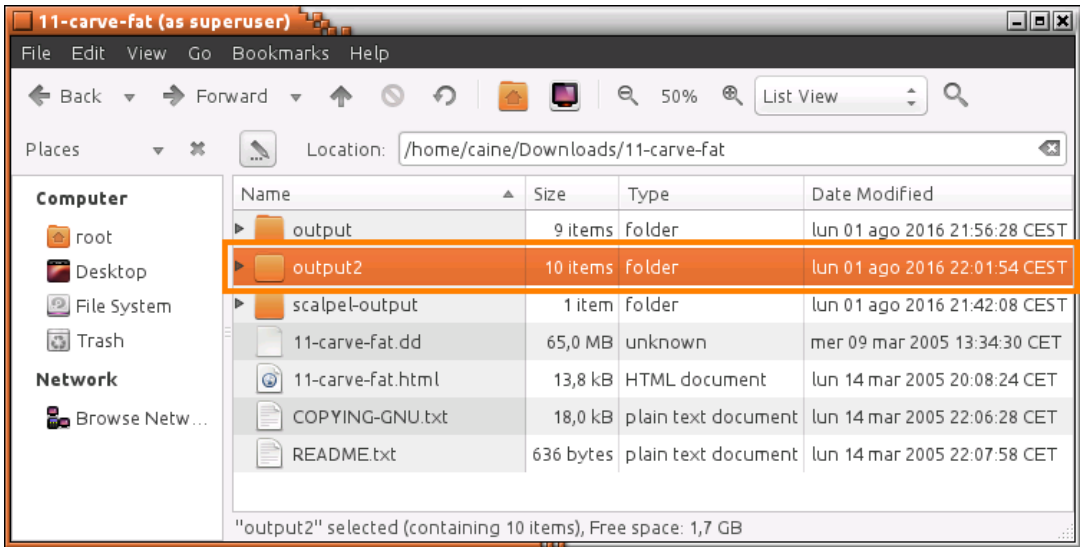
9. Enter the command below to use the same configuration file that was used with scalpel to use with foremost to see if the output changes.

```
foremost -c /etc/scalpel/scalpel.conf 11-carve-fat.dd -o output2
```

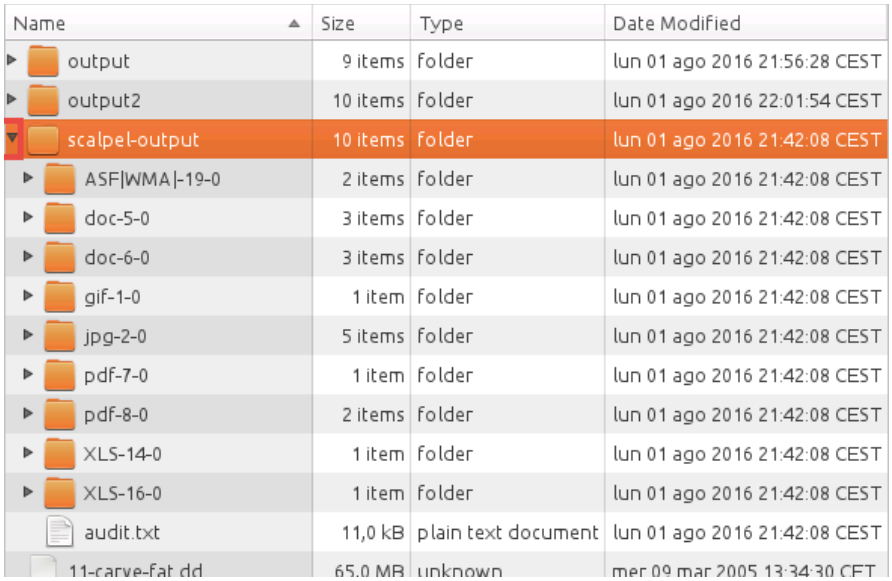
```
caine@Caine01:~/Downloads/11-carve-fat$ foremost -c /etc/scalpel/scalpel.conf 11
-carve-fat.dd -o output2
Processing: 11-carve-fat.dd
|*|
caine@Caine01:~/Downloads/11-carve-fat$
```


3 View the Carved Files

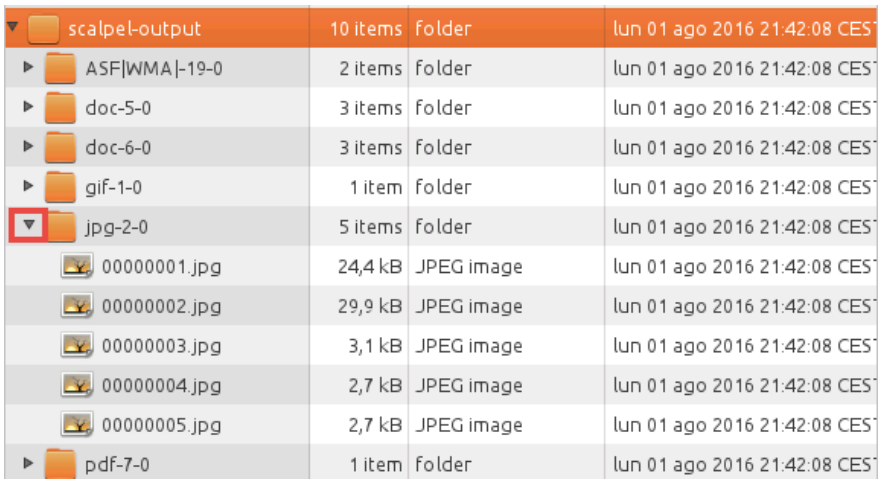
1. Change focus to the **file manager** and notice an *output2* folder is visible.



2. Using the file manager, explore the *scalpel-output* folder to view the carved files from the *Scalpel* tool. Begin by expanding the folder, click on the **arrow** next to *scalpel-output*.



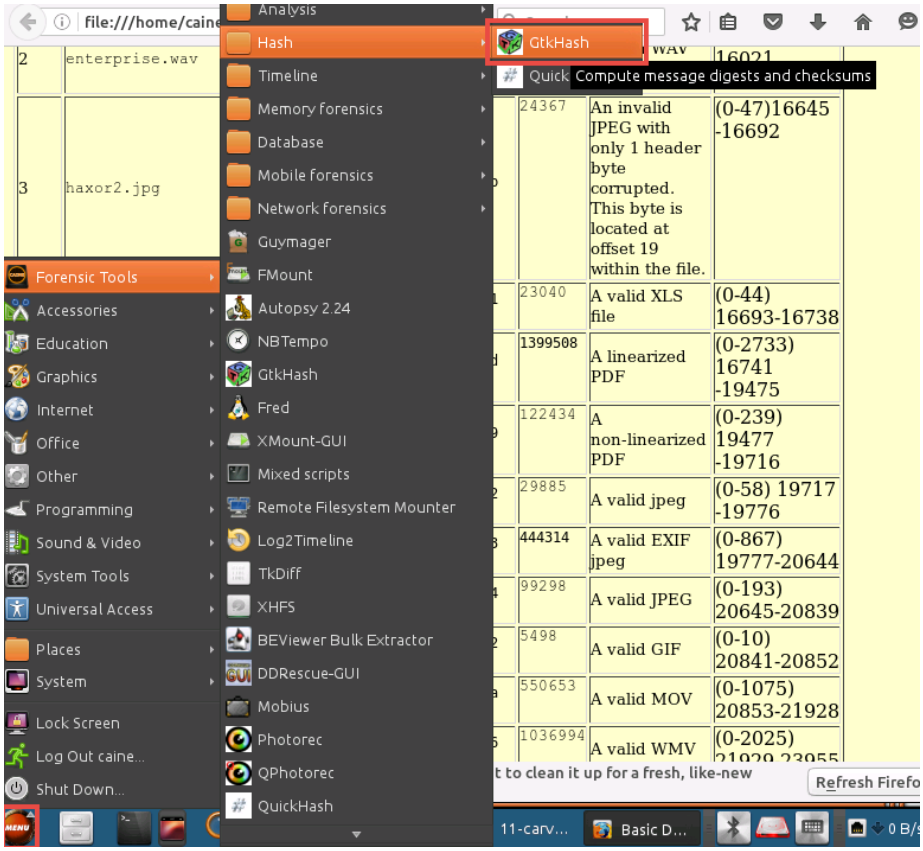
3. Notice each type of file is placed into its own folder. Click on the **arrow** next to *jpg-2-0*.



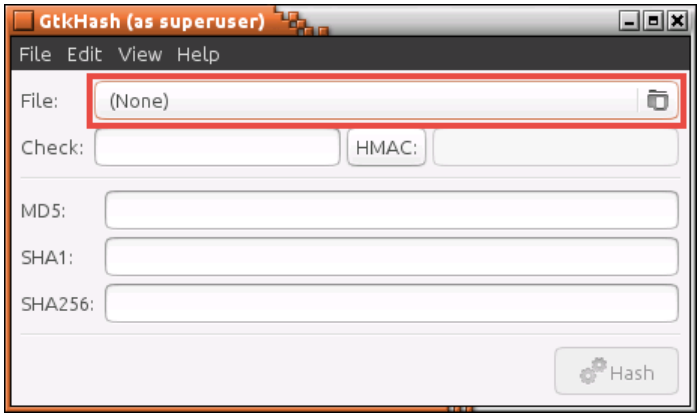
scalpel-output	10 items	folder	lun 01 ago 2016 21:42:08 CES
ASF|WMA|-19-0	2 items	folder	lun 01 ago 2016 21:42:08 CES
doc-5-0	3 items	folder	lun 01 ago 2016 21:42:08 CES
doc-6-0	3 items	folder	lun 01 ago 2016 21:42:08 CES
gif-1-0	1 item	folder	lun 01 ago 2016 21:42:08 CES
jpg-2-0	5 items	folder	lun 01 ago 2016 21:42:08 CES
00000001.jpg	24,4 kB	JPEG image	lun 01 ago 2016 21:42:08 CES
00000002.jpg	29,9 kB	JPEG image	lun 01 ago 2016 21:42:08 CES
00000003.jpg	3,1 kB	JPEG image	lun 01 ago 2016 21:42:08 CES
00000004.jpg	2,7 kB	JPEG image	lun 01 ago 2016 21:42:08 CES
00000005.jpg	2,7 kB	JPEG image	lun 01 ago 2016 21:42:08 CES
pdf-7-0	1 item	folder	lun 01 ago 2016 21:42:08 CES

When comparing the 5 jpg files found by *Scalpel* against the HTML (*11-carve-fat.html*) document opened previously, notice that there are supposed to be 4 jpg files. There might be a duplicate or the carver tool found a false positive (something that appears to be a jpg but actually isn't).

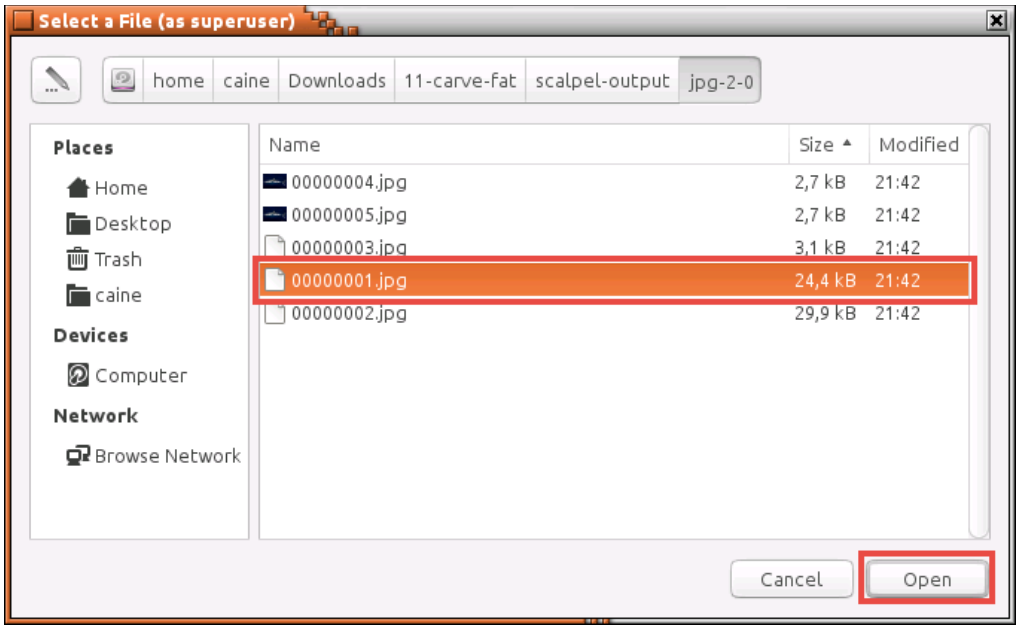
4. Navigate to **Menu > Forensic Tools > Hash > GtkHash** to open the hashing tool.



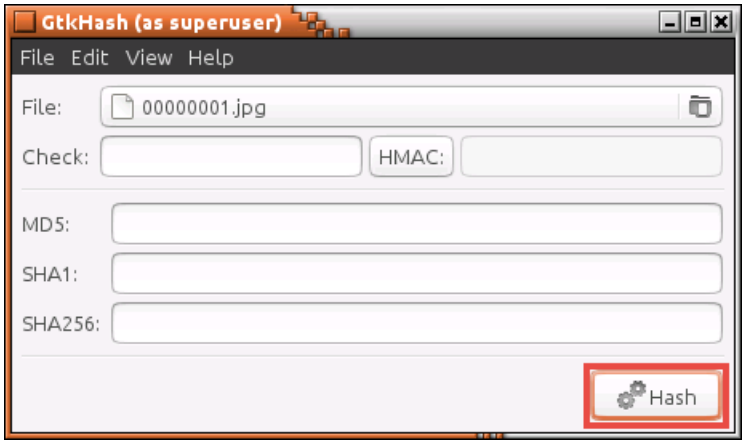
5. Using the *GtkHash* tool, click on the **File** icon.



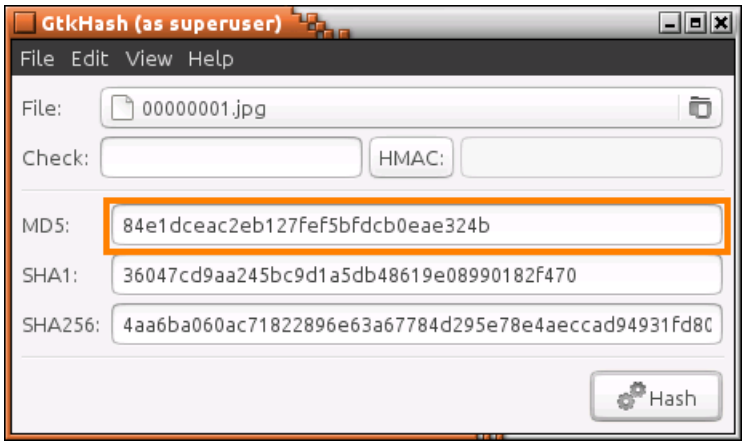
6. In the *Select a File* window, navigate to **/home/caine/Downloads/11-carve-fat/scalpel-output/jpg-2-0/** and select the **00000001.jpg** file. Click **Open**.



7. Using the *GtkHash* tool, verify that the file is loaded and click **Hash**.



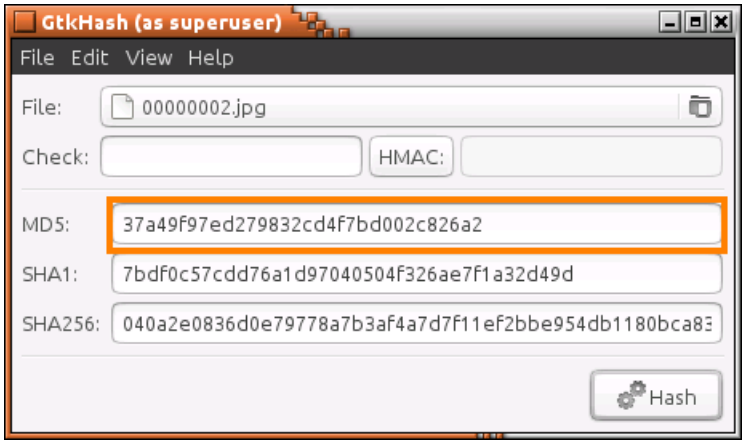
8. Compare the *MD5* hash to the *MD5* hashes found in the *11-carve-fat.html* file. Notice that it matches with the *haxor2.jpg* file.



3	haxor2.jpg	84e1dceac2eb127fef5bfdcb0eae324b	24367	An invalid JPEG with only 1 header byte corrupted. This byte is located at offset 19 within the file.	(0-47)16645 -16692

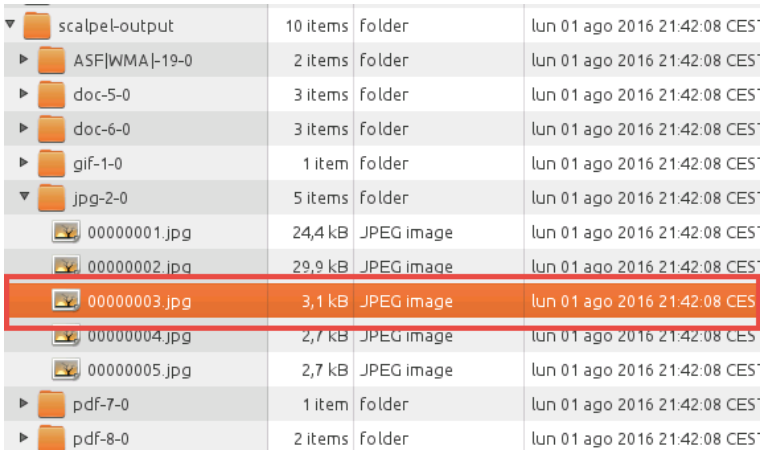
9. Repeat **Task 3**, **Steps 5-7,** but this time for the **00000002.jpg** file.

10. When comparing the *MD5* hash, notice that it matches with the *paul.jpg* file.

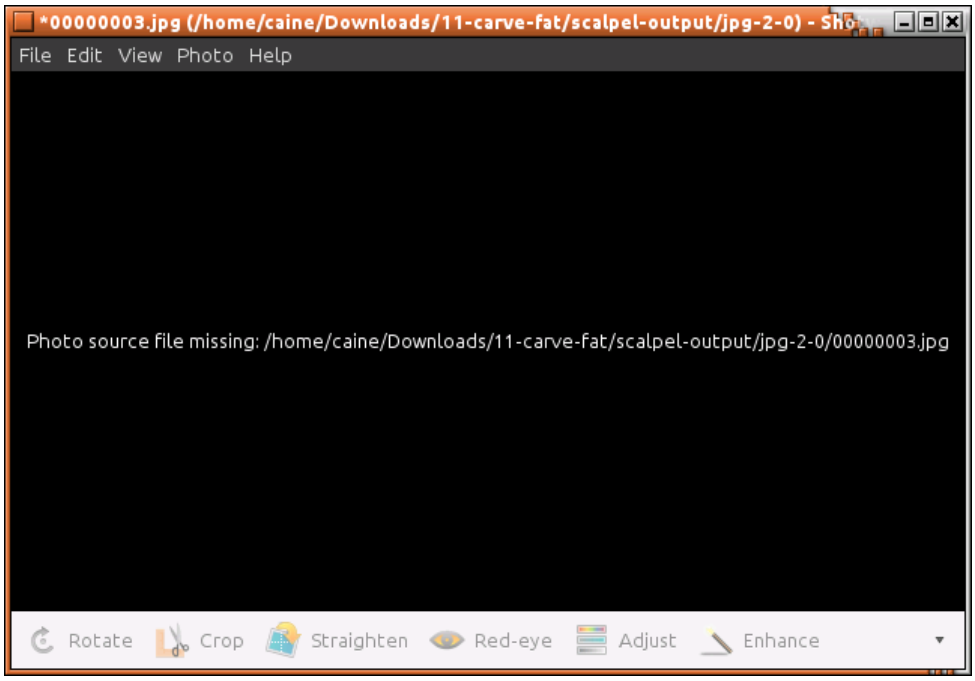


7	paul.jpg	37a49f97ed279832cd4f7bd002c826a2	29885	A valid jpeg	(0-58) 19717 -19776

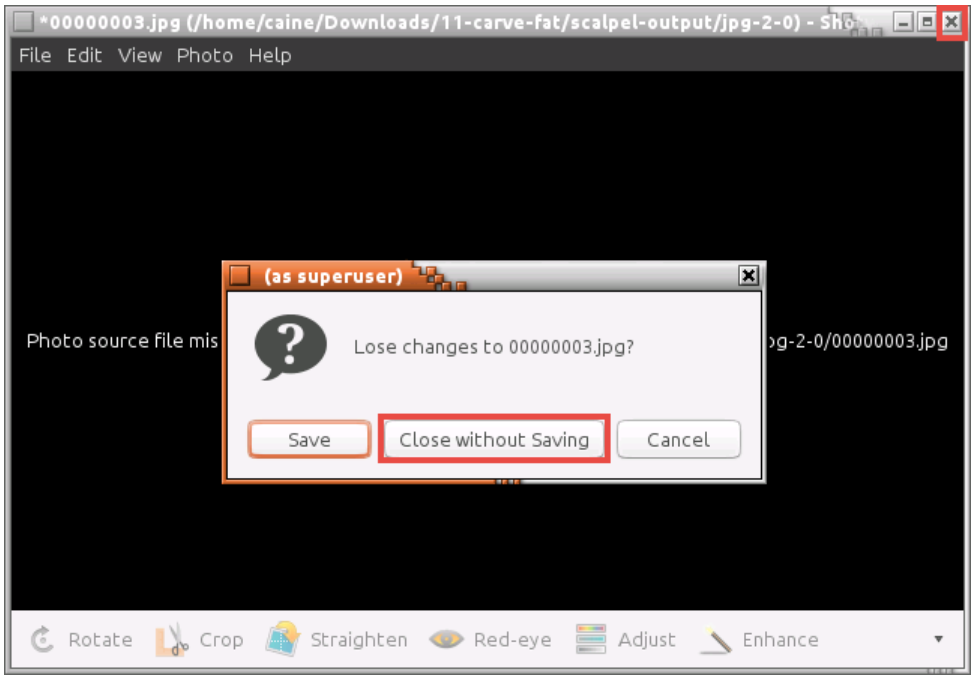
11. When comparing the *MD5* hash values for *00000003.jpg*, *00000004.jpg*, and *000000005.jpg*, the hashes don't match. Using the *file manager*, double-click on the **00000003.jpg** file to open it.



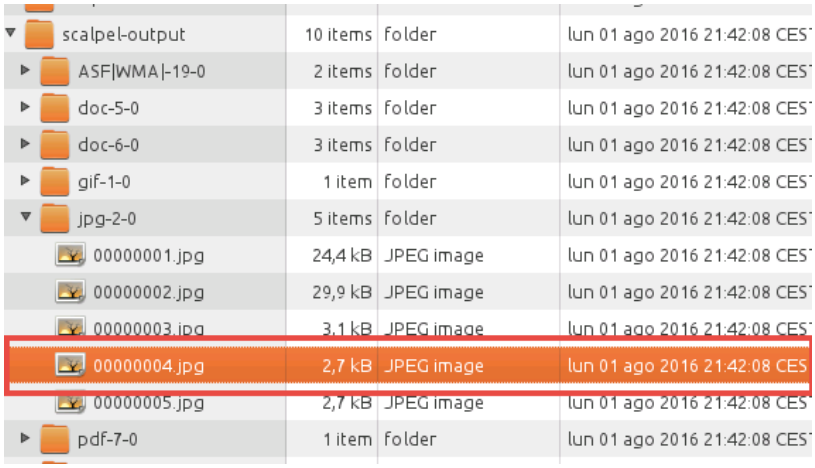
12. Notice the message received for the *00000003.jpg* file. The file is either damaged or a false positive.



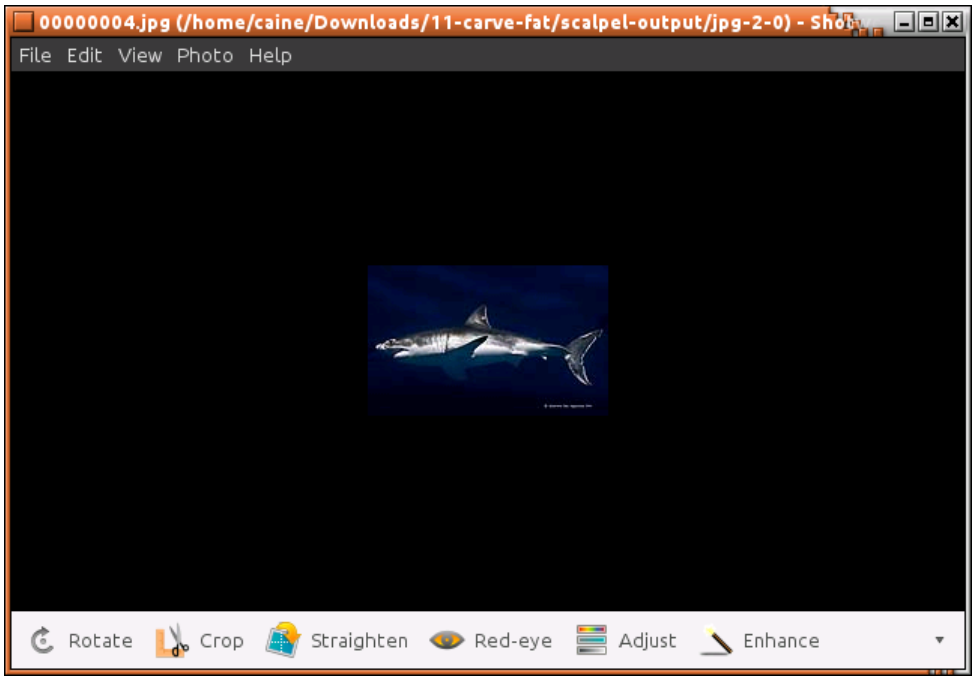
13. Close the file window for *00000003.jpg*. If prompted to save, click **Close without Saving**.



14. Using the *file manager*, double-click on the **00000004.jpg** file to open it.



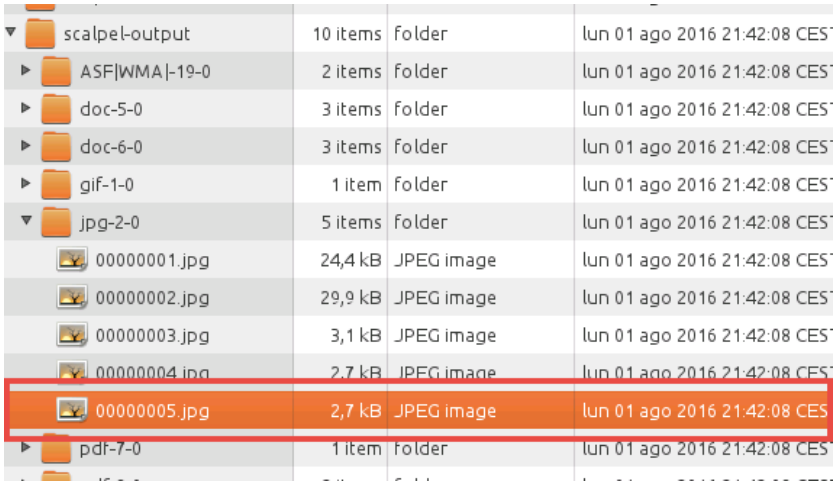
15. Notice the image shown for *00000004.jpg*. When comparing to the **11-carve-data.html** file, there is supposed to be a *shark.jpg* present.



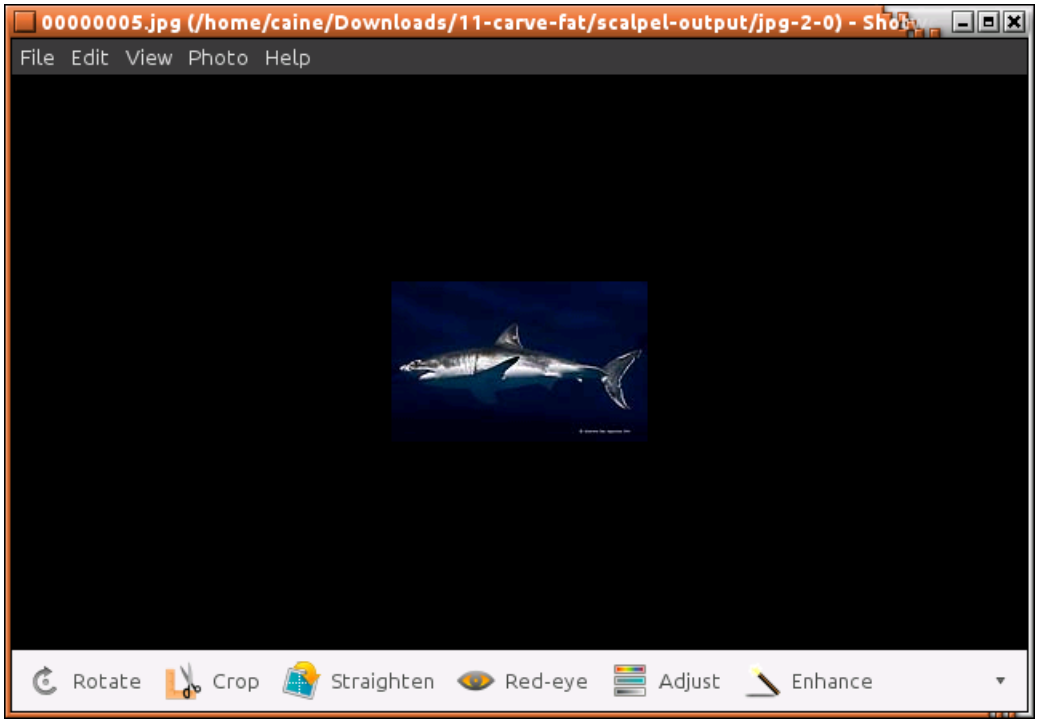
9	shark.jpg	d83428b8742a075b57b0dc424cd297c4	99298	A valid JPEG	(0-193) 20645-20839

16. Close the file window for *00000004.jpg*.

17. Using the *file manager*, double-click on the **00000005.jpg** file to open it.



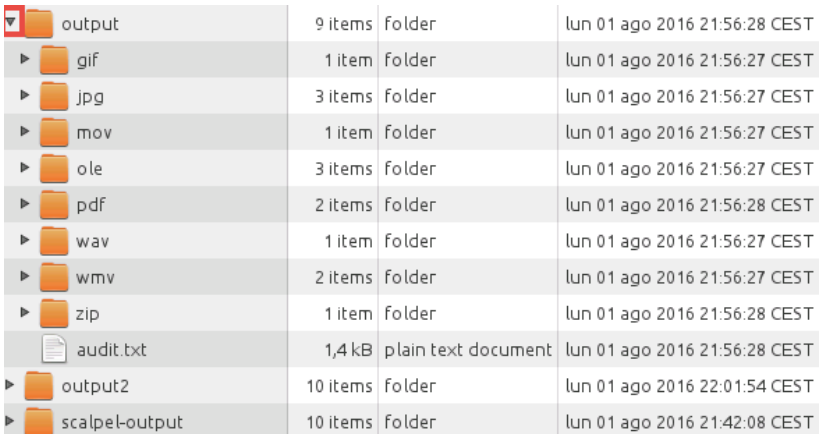
18. Notice the image shown for *00000005.jpg*.



It appears that the *shark.jpg* image is shown twice and the missing image is *pumpkin.jpg* when viewing the *11-carve-fat.html* file. The hashes don't match any of the known images.

19. Close the file window for *00000005.jpg*.

20. Change focus to the **file manager** window and expand the **output** folder by clicking on its **arrow**.

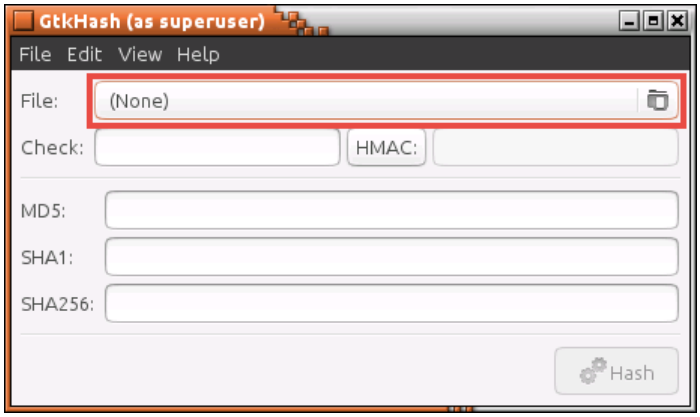


21. Expand the **jpg** folder by clicking on its **arrow**.

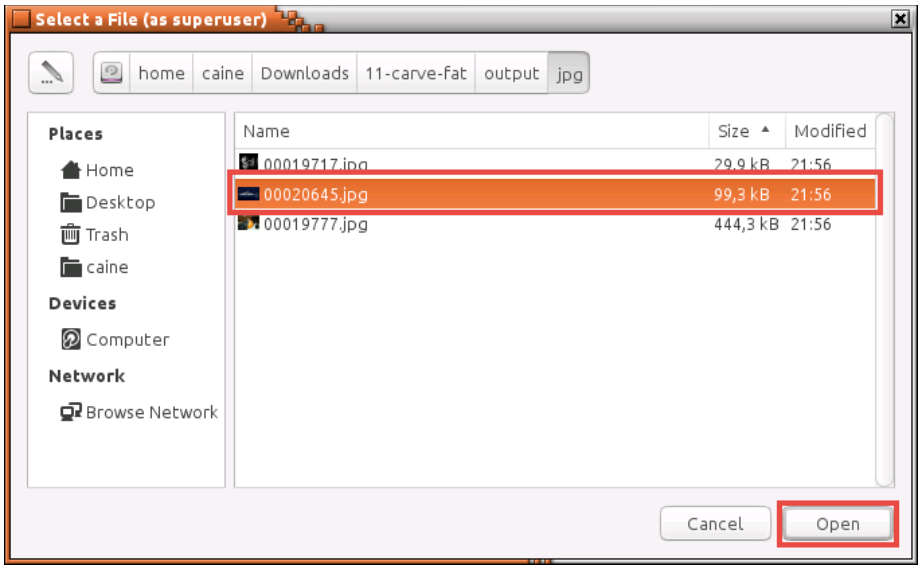
Notice 3 *jpg* files appear.

22. Change focus to the **GtkHash** application.

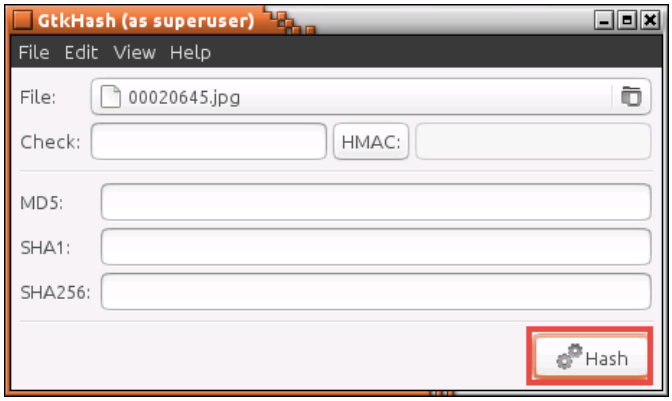
23. Using *GtkHash*, click on the **File** icon.



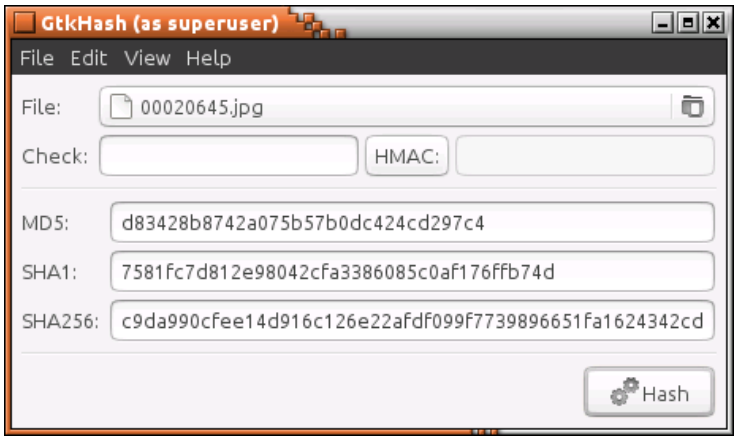
24. In the *Select a File* window, navigate to **/home/caine/Downloads/11-carve-fat/output/jpg/** and select the **00020645.jpg** file. Click **Open**.



25. Using the *GtkHash* tool, verify that the file is loaded and click **Hash**.

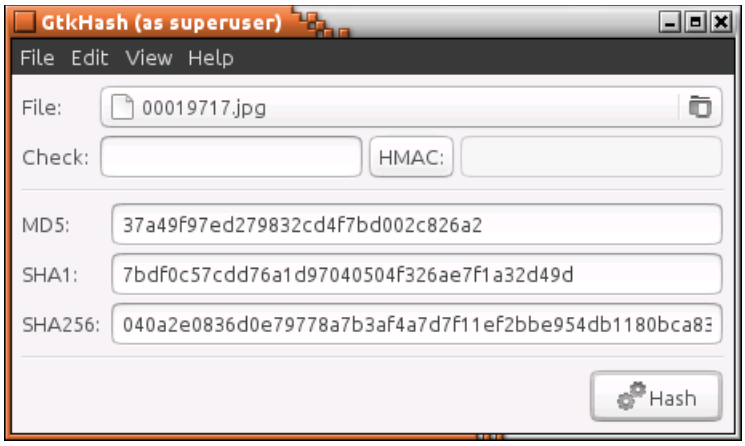


26. When comparing the *MD5* hash to the *MD5* hashes found in the *11-carve-fat.html* file. Notice that it matches with the *shark.jpg* file.



9	shark.jpg	d83428b8742a075b57b0dc424cd297c4	99298	A valid JPEG	(0-193) 20645-20839

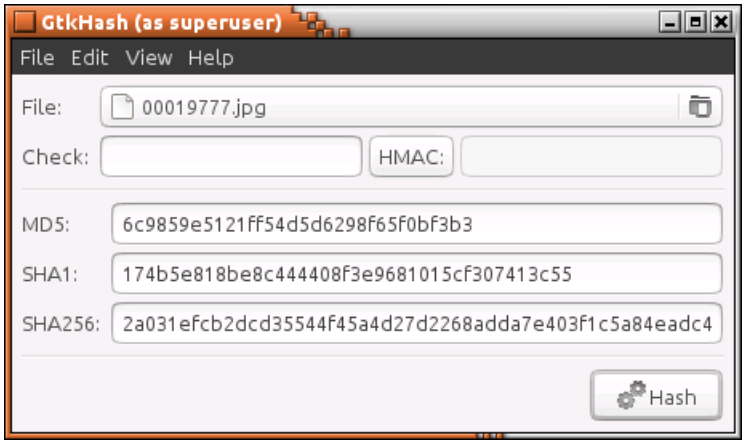
27. Repeat **Task 3**, **Steps 23-25** but this time for the **00019717.jpg** file.
28. When comparing the *MD5* hash, notice that it matches with the *paul.jpg* file.



7	paul.jpg	37a49f97ed279832cd4f7bd002c826a2	29885	A valid jpeg	(0-58) 19717 -19776

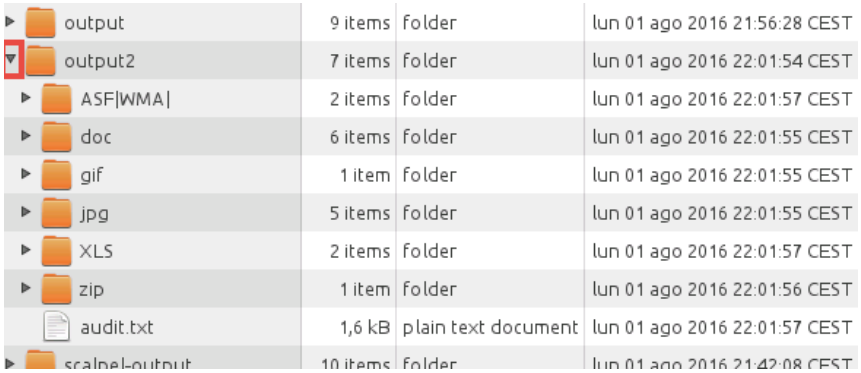
29. Repeat **Task 3**, **Steps 23-25** but this time for the **00019777.jpg** file.

30. When comparing the *MD5* hash, notice that it matches with the *pumpkin.jpg* file.



8	pumpkin.jpg	6c9859e5121ff54d5d6298f65f0bf3b3	444314	A valid EXIF jpeg	(0-867) 19777-20644

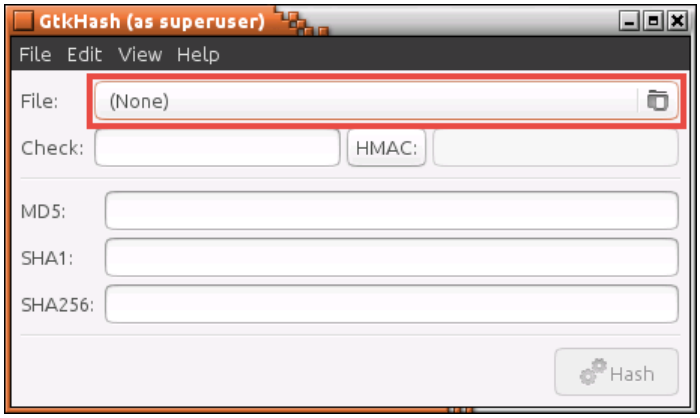
31. Notice that the *haxor2.jpg* is missing from the *output* folder when compared to the *scalpel-output* folder. Change focus to the **file manager** window and expand the **output2** folder by clicking on its **arrow**.



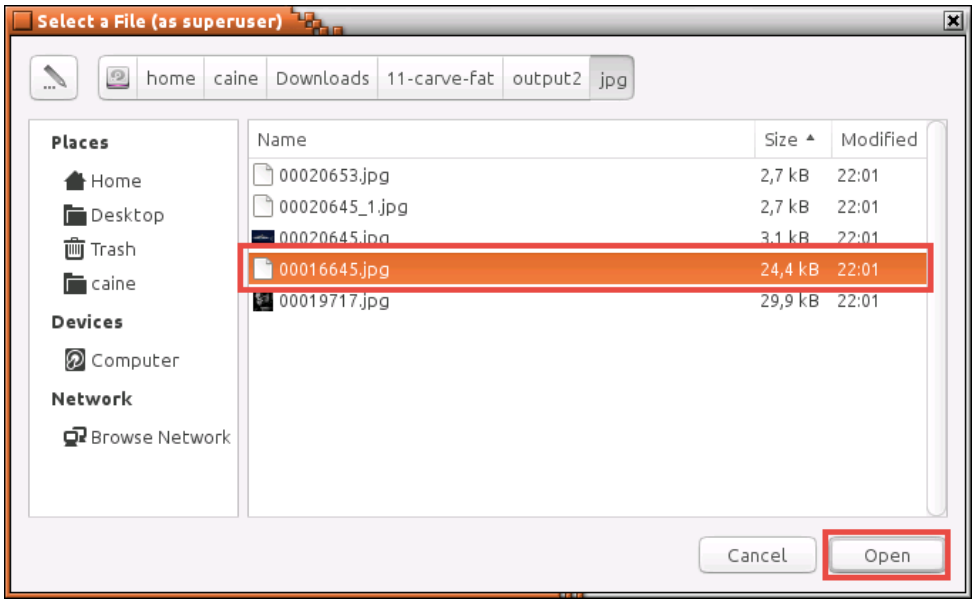
32. Expand the **jpg** folder by clicking on its **arrow**.

Notice 5 *jpg* files appear.

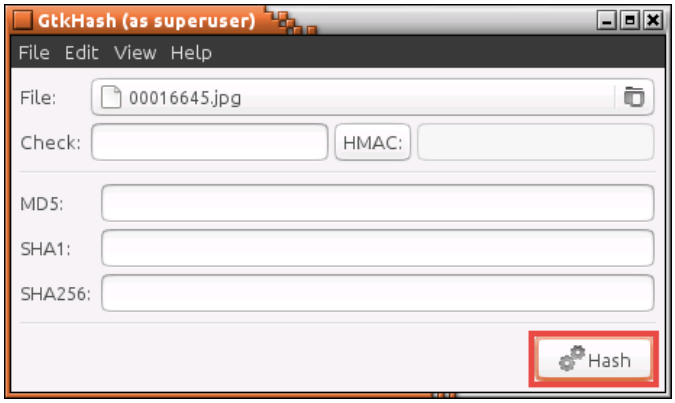
33. Change focus to the **GtkHash** application.
34. Using *GtkHash*, click on the **File** icon.



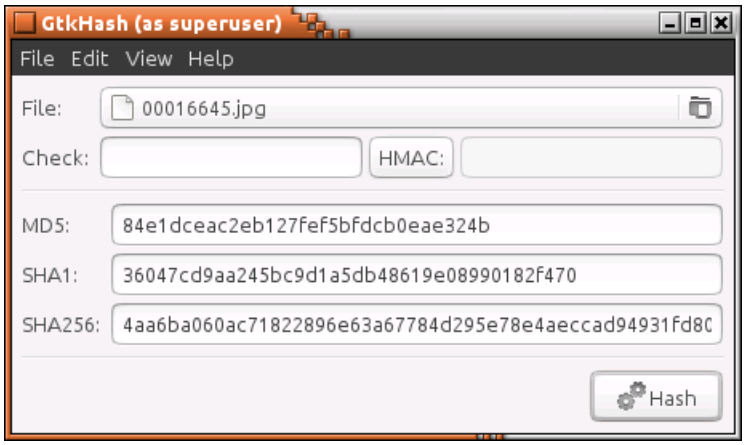
35. In the *Select a File* window, navigate to **/home/caine/Downloads/11-carve-fat/output2/jpg/** and select the **00016645.jpg** file. Click **Open**.



36. Using the *GtkHash* tool, verify that the file is loaded and click **Hash**.

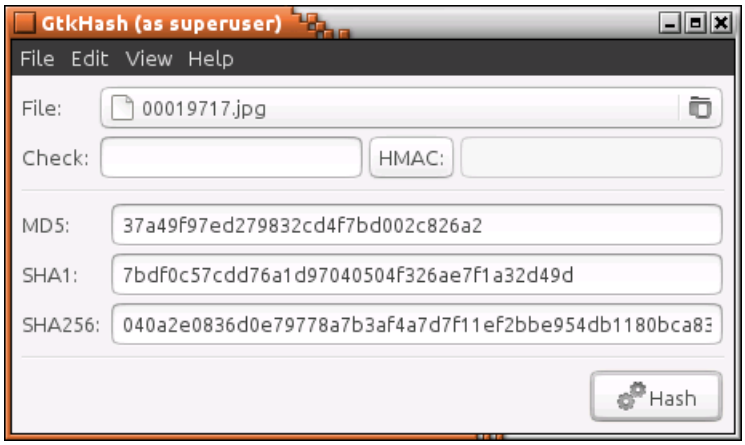


37. When comparing the *MD5* hash to the *MD5* hashes found in the *11-carve-fat.html* file. Notice that it matches with the *haxor2.jpg* file.



3	haxor2.jpg	84e1dceac2eb127fef5bfdcb0eae324b	24367	An invalid JPEG with only 1 header byte corrupted. This byte is located at offset 19 within the file.	(0-47)16645 -16692

38. Repeat **Task 3**, **Steps 34-36** but this time for the **00019717.jpg** file.
39. When comparing the *MD5* hash, notice that it matches with the *paul.jpg* file.



7	paul.jpg	37a49f97ed279832cd4f7bd002c826a2	29885	A valid jpeg	(0-58) 19717 -19776

40. Close all **PC Viewers** and end the reservation to complete the lab.

