

LAB 3: BASIC FILE SYSTEM FORENSICS

INTRODUCTION

In this lab, we will examine a forensic image that contains a number of different files - images, documents, etc. We'll look at how a forensic tool can make deleted file recovery very easy if file records still exist. We'll also look at ways to identify file types (without relying on file extensions alone), and also look at basic file carving techniques to recover deleted files when the file records have been overwritten.

RESOURCES AND LINKS

- SANS Investigative Forensic Toolkit (SIFT) Workstation - <https://computer-forensics.sans.org/community/downloads>
- The Sleuth Kit - <http://www.sleuthkit.org>

LAB OUTLINE

1. GoodChem Scenario	2
2. SIFT Workstation Setup	3
Shared Folder Setup	3
Workstation Setup	4
Image Verification	4
Determine Partitions on Disk Image	4
Mount The Partition	5
3. Review Deleted File Allocation Table Entries	6
4. File Signature Analysis	9
5. Recovering Deleted Files	11
6. Re-Verify	14
7. Report	15
8. Results	17
Undeleted Files	17
Renamed Files	17
Carved File	17

LAB 3: BASIC FILE SYSTEM FORENSICS

1. GOODCHEM SCENARIO

You have been hired by GoodChem (a chemical manufacturing company) to investigate a possible theft of intellectual property. An IT administrator, Fred Thompson, at the company was found in possession of a concealed USB drive as he was being escorted from the building after being terminated for ongoing performance issues. Employees at GoodChem are expressly forbidden from using their own storage devices at work. He responded smugly when questioned about the device's contents, and claims he's never used it at work, and that it only contains personal files and pictures he's downloaded from the Internet. He has given his permission for the device to be analysed.

You will be provided with a compressed archive containing the following:

1. A forensic image of the USB drive.
2. A text file containing the MD5 hash value of the forensic image
3. A text file containing a list of known files and their MD5 hash value

Your job is to uncover what, if any, documents belonging to the company might be located on it. The company is expecting a full report detailing any files you were able to recover, as well as information about the dates and times the files may have been copied to the USB device and/or deleted. As Fred worked in IT, it is likely he has used his knowledge to make recovery of these files difficult.

GoodChem has provided you with details of entries from their server access logs that they deemed suspicious. These logs suggest that Fred has recently accessed four files which he had no legitimate reason to access.

Date/Time	User	File Accessed	Description
1/Mar/2010 3:04pm	FThompson	GoodChemClientList.xls	a List of all GoodChem's current clients
31/Mar/2010 3:29pm	FThompson	GoodChem Inventory.xls	a list of GoodChem's inventory and current prices
17/Mar/2010 4:15pm	FThompson	GoodChem Users Summary.xls	a summary of GoodChem user accounts by department
23/Mar/2010 10:38am	FThompson	GoodChemUsers.xls	a list of all usernames and passwords to GoodChem's main database

It is up to you, however, to discover if he has copied these files to his USB device.

Note: It is good practice to commence an investigation log outlining the current date and time and any steps you have undertaken while undertaking the investigation, this log will form part of the contents of your final report

LAB 3: BASIC FILE SYSTEM FORENSICS

2. SIFT WORKSTATION SETUP

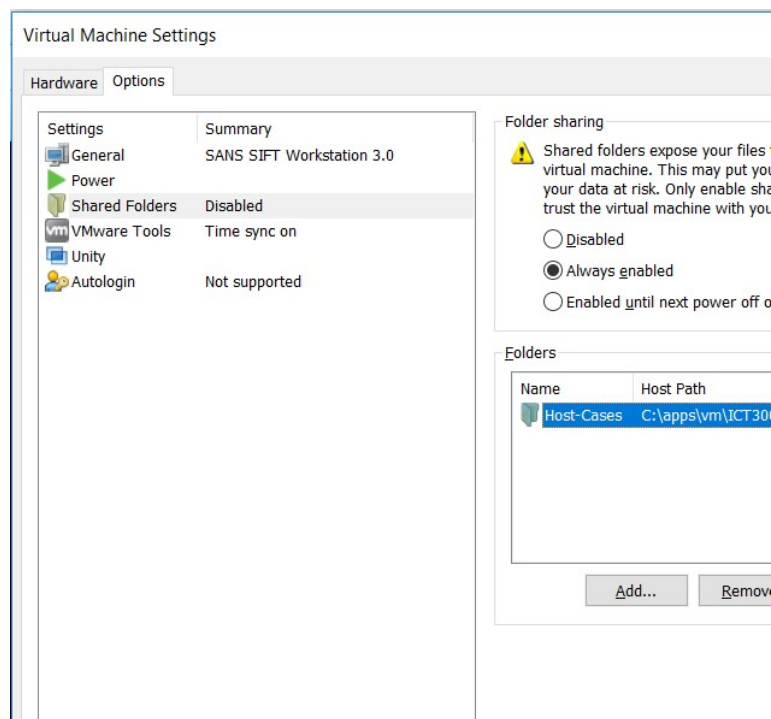
For this lab, you'll need:

1. SIFT Workstation 3.0 (ICT Virtual Machine Launcher)
2. Disk image for the lab named "lab3.tar.gz" (via Canvas)

SHARED FOLDER SETUP

Verify VMWare shared folder configuration

- 1) Navigate to the folder "C:\apps\vm\ICT30010-SANS-SIFT\" on the host Windows computer (if working from home create this folder structure)
- 2) Ensure that the folder named "Host-Cases" exists, if not create this folder
- 3) On the VMWare window click on Player -> Manage -> Virtual Machine Settings
- 4) Click on the Options tab and then click "Shared Folders"
- 5) Ensure that Folder sharing is set to "Always Enabled" and that the path is set to "C:\apps\vm\ICT30010-SANS-SIFT\Host-Cases"



- 6) Restart the SIFT workstation to enable the shared folder configuration

LAB 3: BASIC FILE SYSTEM FORENSICS

WORKSTATION SETUP

- 7) Copy the lab files to the SIFT Workstation
 - a) Open the directory "C:\apps\vm\ICT30010-SANS-SIFT\Host-Cases" on your windows machine.
 - b) Copy the file "lab3.tar.gz" into this folder
- 8) Start SIFT Workstation
 - a) Start the VMWare Launcher and locate the "SANS SIFT Workstation 3.0" from the ICT30010 Folder
 - b) Once SIFT has started successfully log on as "sansforensics", with the password "forensics"
- 9) Prepare Lab Files
 - a) Check that you can access the lab3.tar.gz file by listing files in the VMWare shared folder:
`ls /mnt/hgfs/Host-Cases`
 - b) Create a new directory for this lab:
`mkdir /cases/lab3`
 - c) Unzip the lab files to this new directory:
`tar -xzf /mnt/hgfs/Host-Cases/lab3.tar.gz -C /cases/lab3/`

IMAGE VERIFICATION

- 10) Since we've just made a copy of a forensic image, it's probably a good idea to verify that nothing went wrong with the copying/unzipping process. Since our forensic image has been provided to us with a hash, it should be simple to verify.
 - a) Change to the "lab3" directory, and get a listing of all the files (run `cd /cases/lab3`, then `ls`)
 - b) Verify that the hash of "lab3.dd" matches that of the hash value provided in the "lab3.dd.md5" file:
 - i) `md5sum lab3.dd` Value:
 - ii) `cat lab3.dd.md5` Value:

DETERMINE PARTITIONS ON DISK IMAGE

We need to identify what partition(s) are on the forensic image, so let's examine the partition table using a couple of "easy" tools. If you get time at the end of the lab, you may also want to check these values "the hard way".

- 11) The "fdisk" command we've used before on disks can also work directly from a raw disk image file (since everything in linux is a file anyway), so run `fdisk -lu lab3.dd`.

LAB 3: BASIC FILE SYSTEM FORENSICS

- a) Note the sector size for the forensic image: bytes
- a) Note the details of the partition reported by fdisk (you will need a calculator to work out the total size. Remember the Total Sectors should be: $\text{EndSector} - \text{StartSector} + 1$ and Total Size in MB is $(\text{TotalSectors} * \text{SectorSize}) / 1024 / 1024$

Partition	Partition Type (e.g. NTFS/FAT)	Start Sector	End Sector	Total Sectors	Total Size (MB)
lab3.dd 1					

- b) There's also a specially designed forensic tool which forms part of the "sleuth kit" that performs a similar function - mmls (Media Management LS). Run the command "`mmls lab3.dd`". The output from this tool not only shows you the FAT32 Partition, but also what other areas of the disk have been allocated for use (e.g. the first sector should be listed as containing the primary partition table)

```
sansforensics@SIFT-Workstation:/cases/lab3$ mmls lab3.dd
DOS Partition Table
Offset Sector: 0
Units are in 512-byte sectors

   Slot   Start      End          Length      Description
00:  Meta   0000000000    0000000000    0000000001    Primary Table (#0)
01:  ----- 0000000000    0000000062    0000000063    Unallocated
02:  00:00  0000000063    0003968054    0003967992    Win95 FAT32 (0x0B)
03:  ----- 0003968055    0003970047    0000001993    Unallocated
```

Figure 1. Partition layout information from mmls

MOUNT THE PARTITION

- 12) Mount the partition by determining the partition's offset from within the forensic image, then running the mount command.
- a) Calculate the starting offset in bytes (starting sector x sector size):
- Starting Offset: bytes
- b) Create a mount point for the disk image by running "`sudo mkdir /mnt/lab3`"
- c) Mount the partition read-only by running
"`sudo mount -t vfat -o ro,loop,offset=xxxx lab3.dd /mnt/lab3`"
(don't forget to replace xxxx with the offset you just calculated in)

LAB 3: BASIC FILE SYSTEM FORENSICS

3. REVIEW DELETED FILE ALLOCATION TABLE ENTRIES

13) Open the /mnt/lab3 directory in the GUI, and browse the directory contents.

- a) Double click the “mount_points” icon on the desktop then navigate to “lab3”.
- b) Note the directory names, and the number of files visible in each directory:
 - i) Name: Number of Files:
 - ii) Name: Number of Files:
 - iii) Name: Number of Files:
 - iv) Name: Number of Files:

14) Now let’s try some other tools to see what other information we can get.

- a) Back in the terminal, run the sleuth kit version of the ls command, “fls” (“**fls lab3.dd**”). This command fails, because fls doesn’t know where the partition begins (the offset from the start of the disk image), and therefore doesn’t know what type of file system is being used.
- b) Run the command “**man fls**” to show the fls manual. What is the option we need to add to the fls command to specify the offset? (note: Press “Q” to exit the man command)

.....

- c) And does the offset need to be specified? In bytes, or sectors?


✓ When we don’t know exactly what options to use for a linux command, we can use the built-in manual (“man”) to find out. This will work on almost all linux commands!

- d) Re-Run the fls command - “**fls offsetoptions lab3.dd**” - with the options you worked out from the manual (and the partition offsets you worked out earlier). You should see the folders you identified earlier, and some additional files/folders, including the “Volume Label Entry”, which is the name given to the volume by the user, and some \$... files which are hidden files which contain information about the files on the volume.

```
r/r 3: VOLUMELABEL      (Volume Label Entry)
d/d 13: Directory 1
v/v 63395347: $MBR
```

Figure 2. Example output from fls (entry type, inode number, name)

LAB 3: BASIC FILE SYSTEM FORENSICS

- e) The numbers next to each entry are the “inode” numbers for the file or directory (basically a unique number for each entry). Note the inode numbers of each of the folders you identified earlier
- i) Name: iNode Number:
- ii) Name: iNode Number:
- iii) Name: iNode Number:
- iv) Name: iNode Number:
- f) The “Word Docs” folder had an inode number of “9” (refer to the fls output from earlier). Get a listing of the files in this folder using the Sleuth Kit command `“fls -o 63 lab3.dd 9”` (this command may take a few seconds to complete)
- g) Note that two file have an asterix (*) next to their names. This means the file is deleted but still recoverable by our forensic tools. Write down the details for these files.
- i) Name: iNode Number:
- ii) Name: iNode Number:
- h) Attempt to recover each of the deleted files (i.e. run through the next few steps twice - once for each deleted file you found, and listed above):
- i) Attempt to recover the deleted files by running the command `“icat -o 63 lab3.dd inode > filename”` (remember to replace “inode” with the inode number for each file you are attempting to recover, and filename with the original name of the file. The “>” character tells icat to output the contents of the file to *filename* instead of the screen)
- ii) Launch the OpenOffice application calc from the quick launch bar. 
- iii) Click File -> Open and navigate to the folder `“/cases/lab3”`.
- iv) Attempt to open the two files recovered earlier to ensure that they are valid.
- v) Calculate the MD5 hash of the file, and check if the file is on the list of “notable” files by running `“md5deep -m notablefiles.txt -w filename”` (if the file is not notable, md5deep will return nothing. Otherwise, it should return the name of the original file who’s hash matches)
- vi) If the file matches, make a note of the inode number and filename in the “Undeleted Files” table at the end of the worksheet.
- vii) Check the dates and times associated with the file by running `“istat -o 63 lab3.dd inode”` (replacing *inode* with the inode number of the file)

LAB 3: BASIC FILE SYSTEM FORENSICS

viii) The `istat` output tells you exactly where on the disk the file's contents are by listing each sector used by the file.

ix) Note the file's size, and the Written, Accessed and Created dates for the file at the end of the worksheet, too.

✓ FAT file systems only store three dates - Created, Modified and Accessed. Also, the accessed date on a FAT file system is only stored as a date (no time!)

15) Check the other directories to make sure there are no other deleted files that can be recovered this way

- a) List all the files in the directory: `fls -o 63 lab3.dd inode`. (replacing *inode* with the inode number of the directory to interrogate)
- b) Check if any files are marked as deleted (i.e. have an asterix next to their name)

LAB 3: BASIC FILE SYSTEM FORENSICS

4. FILE SIGNATURE ANALYSIS

In this section, we attempt to identify files which may have been renamed by the user in an attempt to hide them from forensic examination.

The SIFT tool “Sorter” allows us to sort through a large number of files in an image, and group them by category (document, image, etc.) based on signature, rather than just file extension. This can help uncover files that whose “header” or “signature” does not match their file extension (e.g. a word document renamed to “picture.jpg”)

16) Use the SIFT tool “sorter” to examine file signatures.

- a) Make a directory for sorter to place its reports - `mkdir sorter`
- b) Run the SIFT “sorter” command `sorter -o 63 -d ./sorter lab3.dd` (“-d ./sorter” tells sorter to output to the sorter directory you just created).

17) Examine the output from the sorter tool

- a) Change to the sorter directory (`cd sorter`), then look at the files that sorter created (`ls`)
- b) The “sorter.sum” file contains a summary of the file types that sorted discovered. View its contents by running `more sorter.sum` (use the arrow keys to view the file, and press “q” to exit).
- c) Note that twelve (12) files are marked as “images”, and eight (8) files of type “document” have been located. Sorter also saved more details about these files in the files named “images.txt” and “documents.txt”.

18) Examine the “images” files

- a) View the contents of the “images.txt” file by running `more images.txt`. The twelve files appear to be images, as no errors have been reported. Let’s make sure they really are images by opening one of them.
- b) Double click the “cases” icon on the desktop then navigate to “computer”, “Mnt”, “lab3”, then “Internet Pics”. Verify that the twelve Image files mentioned in the images.txt file can be viewed.

19) Examine the “Document” files

- a) Back in the terminal, view the contents of the “documents.txt” file by running `more documents.txt`. Note that sorter wasn’t able to extract details from the three files in the “Powerpoints” directory, but that metadata was successfully extracted from the Word Docs (and the two excel files we’ve already recovered). Let’s make sure that the three PowerPoint files are what they appear to be.
- b) Double click the “cases” icon on the desktop then navigate to “computer”, “Mnt”, “lab3”, then “Powerpoint”. Verify that the three PowerPoint files mentioned in the documents.txt file can be opened by double clicking each icon.

LAB 3: BASIC FILE SYSTEM FORENSICS

c) Which document doesn't open as a powerpoint document, instead opening in calc?

i) Name:

20) Let's take a closer look at the PowerPoint files:

a) Back in the terminal, change to the /mnt/lab3/Powerpoints directory ("cd /mnt/lab3/Powerpoints"), and list the files in the directory (ls)

✓ You can save typing full path and file names in linux by pressing the "Tab" key on the keyboard after writing a few letters of each part of the path

e.g. typing "/mnt/lab3/Po" then pressing "Tab" will auto-complete the "Powerpoint" directory name

b) View the first 16 bytes of all three files by running "xxd -l 16 filename" (replacing *filename* with the name of each file in turn)

c) You should see that all three signatures match. The first few bytes "D0CF 11E0" are typical of Microsoft Office documents (xls, doc, ppt, etc.), but also many other document types which use the "CDF" (Compound Document Format).

d) Let's take a look at the end of the offending "PowerPoint" file. Run "tail awesomejokes.pps | xxd -a" (tail will display the last section of a file, and the -a flag tells xxd to ignore lines containing only zeros). Do you see any words in the output from xxd that might give you a clue as to the *real* document type? What do you think it is?

.....

e) Making a copy of the file in your cases directory with the correct extension (run "cp awesomejokes.pps /cases/lab3/awesomejokes.pps.xls")

f) Check if the file is on the list of "notable" files by going back to the lab3 directory ("cd /cases/lab3") and running "md5deep -m notablefiles.txt -w awesomejokes.pps.xls" (if the file is not notable, md5deep will return nothing. Otherwise, it should return the name of the original file who's hash matches)

g) If the file matches, let's add it to the "Renamed Files" table at the end of the worksheet. But first, we need to collect some more information:

i) Find the inode number for the awesomejokes.pps file by running fls again - the inode number for the Powerpoints folder was 7 (you noted this down earlier). So run "fls -o 63 lab3.dd 7" and note the inode number:

(1) Name: iNode Number:

ii) Get the dates and times for the awesomejokes.pps file by running "lsstat -o 63 lab3.dd *inode* | more" (replacing *inode* with the number you just discovered). Again, note the file size, and the Written, Accessed and Created Dates in the "Renamed Files" table at the end of the worksheet.

LAB 3: BASIC FILE SYSTEM FORENSICS

5. RECOVERING DELETED FILES

So far, the deleted files we've wanted to recover have (thankfully) had the file allocation table (FAT) entries which outline their location on disk intact. This makes the files easy to recover.

Some times a file may still exist on the filesystem without a FAT entry. We can try to locate them by looking for known information such as the file header or the file contents. If you can locate a file of interest you can attempt to manually recover the file using a process know as file carving.

We have now recovered three out of four suspected files, but we haven't found any traces of the GoodChemInventory.xls file yet (if it's there at all!)

Let's try to find it with a keyword search.

GoodChem has told you that Keith Falce initially wrote the document, and that some of the items they have on their inventory list were "Tegin Pellets" and "Tegobetaine" and various peanut products (such as peanut oil).

21) Create a keyword list

- a) In the cases directory ("`cd /cases/lab3`") create a file called "keyword.txt" by running the command "`gedit keywords.txt`". Enter the terms "Keith Falce", "Tegin Pellets", "Tegobetaine", and "peanut" (each on a new line, without quotes). Save the file, and close gedit.
- b) Verify you entered the keywords correctly by running "`cat keywords.txt`".

22) Collect all the text strings from the image and their location within the image

- a) Create a searchable list of strings contained within the forensic image by running the command "`srch_strings -a -t d lab3.dd > lab3.str`". This command simply searches for *any* groups of characters which may comprise a word or sentence, and saves the list of strings to lab3.str

23) Run your search across the indexed strings file

- a) Search the strings file for your keywords by running "`grep -i -f keywords.txt lab3.str`"
- b) The number next to the search hit indicates the number of bytes offset from the start of the forensic image. Note the file offset:

.....

- c) We can now use the ifind command to locate the inode number that refers to the sector containing the keyword hit. First, though, we need to calculate the sector number (from the start of the volume):

$$((\text{File offset in bytes}) / (\text{Sector size in bytes})) - (\text{Sector offset of Volume Boot Record})$$

$$= (\text{answer above in 23b} / 512) - 63 = \text{Logical Sector: }$$

LAB 3: BASIC FILE SYSTEM FORENSICS

- d) Putting this value into `ifind`, we should be able to obtain the inode number of the file which contained the search hit. Run `"ifind -o 63 lab3.dd -d sector"` (replacing sector with your answer above).

inode:

- e) Now we can use another program `"ffind"` to locate the file name for the inode number.
Run `"ffind -o 63 lab3.dd -d inode"`, using the inode number you were just given by `ifind`.

- f) Which file did we find our search hit in?

File Name:

Since it appears this search hit was *not* for our Inventory file (it seems that someone working for GoodChem just likes peanuts), this is known as a false positive. Let's try simply searching for the filename, to see if we can find any other traces of the GoodChemInventory.xls file.

24) Search for the GoodChemInventory file name:

- a) Search the strings file for the filename `"goodcheminventory.xls"` by running
`"grep -i goodcheminventory.xls lab3.str"`
- b) You should see two "hits" along with their byte offset. Convert the two byte offsets into logical sector numbers (number of sectors from the start of the volume, as opposed to the disk) as above - $(bytes/512)-63$:
- i) Byte Offset: Logical Sector:
- ii) Byte Offset: Logical Sector:
- c) Run `ifind` against the two logical sector numbers you just calculated -
`"ifind -o 63 lab3.dd -d sector"`. You'll notice that `ifind` reports that neither sector is associated with a file/inode. This means that no current (or deleted and recoverable) file is associated with the data we just found.

25) Let's take a look at the data surrounding the two search hits we located

- a) Use `"blkcat"` to extract the contents from the sector - run `"blkcat -o 63 lab3.dd sector -h"` (the `-h` flag tells blockcat to return the output in a hex format similar to `xxd`). Run this command for both sector values you calculated before

You should see the `"GoodChemInventory.xls"` file name in the output from both commands. You may also notice the letters `"PK"` appearing just before and just after the name.

LAB 3: BASIC FILE SYSTEM FORENSICS

```
sansforensics@SIFT-Workstation:/cases/lab3$ blkcat -o 63 lab3.dd 27200 -h
0      504b0304 14000000 08002c4b 713cc7fd      PK.. .... ..K q<..
16     704f1e86 020000ce 07001500 0000476f      pO.. .... ..Go
32     6f644368 656d496e 76656e74 6f72792e      odCh emIn vent ory.
48     786c73ec 5a0b6c1c c7799ea3 f8383e24      xls. Z.l. .y.. .8>$
```

Figure 3. Output from blkcat showing first GoodChemInventory.xls Search Hit

The “PK” header is commonly associated with zip files. Perhaps our suspect zipped up the inventory file before he copied it? This also explains why our keyword search produced no hits - the contents of the file (including the words we were looking for) were compressed!

- b) A ZIP file typically begins with “PK” followed by the hex values 0405, 0506 or 0607. Since our first search hit begins with PK\x04\x05, let’s try to copy out the data beginning there. This time, we’re going to need the *physical* sector number (the number of sectors from the start of the disk), and the number of sectors to copy out (length).

- i) Calculate the *physical* sector number of the first search hit (logical sector + 63)

(1) Physical Sector:

- ii) For now, let’s assume that the second search hit is at the end of the file. Calculate the number of sectors we need to copy out to ensure we include this second search hit. In other words, calculate “(sector of search hit #2) - (sector of search hit #1) + 1”

(1) Sector Count:

- c) Copy out the contents of the sectors we just calculated using dd:

“dd skip=physicalsector count=sectorcount if=lab3.dd of=inventory.zip”

- d) This process is known as file carving, we are copying out the physical contents of the sectors on the disk in an attempt to recover a file that has been deleted
- e) Attempt to unzip the contents of the extracted file by running “unzip inventory.zip”. Unzip will show you the name(s) of the file(s) that are being extracted. Note that a file named “GoodChemInventory.xls” is being extracted

```
sansforensics@SIFT-Workstation:/cases/lab3$ unzip inventory.zip
Archive:  inventory.zip
  inflating: GoodChemInventory.xls
```

Figure 4. Output from Unzip showing filename of extracted file

- f) Check that this is the file we were looking for by running, seeing if the file is on the list of “notable” files by running “md5deep -m notablefiles.txt -w GoodChemInventory.xls”, then add the file’s details to the “Carved File” table at the end of the worksheet. You can check the file size and the date/time associated with the XLS file by running a long directory listing (ls -l)

LAB 3: BASIC FILE SYSTEM FORENSICS

6. RE-VERIFY

At the end of a forensic examination, it's a good idea to confirm (to yourself, and for the purposes of explaining in court later) that none of your tools changed the disk or the forensic image you were working on.

- 26) Re-Verify that the hash of "lab3.dd" matches that in "lab3.dd.md5" by running
"md5sum lab3.dd ; cat lab3.dd.md5"

LAB 3: BASIC FILE SYSTEM FORENSICS

7. REPORT

For this lab, you'll need to write a report for GoodChem outlining your process and findings. This report may be used in court if GoodChem decides to take action against Fred Thompson for his actions.

Your report should be factual and include enough information for another forensic examiner to replicate your results, but does not need to detail exact settings and commands issued to obtain the information (in other words, "I used a forensic tool to determine the created date of the file" is better than "I ran the command 'istat -o 63 lab3.dd 123'").

The audience for your report will be lawyers, judges and other non-technical people.

It is important that opinions and other suppositions are not part of your report. Statements like "Fred saved the file at 12:03pm" is a conclusion not based entirely on the evidence you were provided. A better statement might be "the file had a created date of 12:03pm".

You will be marked on:

1. Your explanation what you received, how and when received the forensic evidence, how you handled continuity of the evidence. *(1 mark)*
2. The details of the partition you located within the forensic image. *(1 mark)*
3. Your ability to locate the two deleted files, detailing the details of the files, how they were hidden within in the forensic image and the explanation of the processes you used to locate them. *(2 marks)*
4. *Your ability to locate the renamed file, detailing the details of the files, how it was hidden within in the forensic image and the explanation of the processes you used to locate it. (2 marks)*
5. *Your ability to locate the carved file, detailing the details of the file, how it was hidden within in the forensic image and the explanation of the processes you used to locate it. (2 marks)*
6. Your ability to stick to the facts and explain your processes and findings in non-technical (but still accurate!) language and the overall presentation of the report *(2 marks)*

You are not required to submit your completed lab sheet, this a written assessment.

LAB 3: BASIC FILE SYSTEM FORENSICS

While the exact format of the report is up to you, a simple way of presenting the report might be in dot point form.

For example:

1. On Wednesday 4th of May, 2011 at approximately 7:03pm, I
2. At 7:06pm, I
23. On Friday 6th of May, 2011 at approximately 9:27am, I recovered the file details are as follows:

File Name	iNode Number	Modified Date	Accessed Date	Created Date	File Size

The lecture next week will cover the topic of forensic report writing.

LAB 3: BASIC FILE SYSTEM FORENSICS

8. APPENDIX - RESULTS

UNDELETED FILES

These files were marked as deleted, but all the details of the files were recoverable.

File Name	iNode Number	Modified Date	Accessed Date	Created Date	File Size

RENAMED FILES

This file was present on the disk but was renamed requiring additional forensic examination to identify.

File Name	iNode Number	Modified Date	Accessed Date	Created Date	File Size

CARVED FILE

These files were deleted from the disk, and no metadata could be located in the file table (or directory entries)

CARVED FILE

File Type	Starting Sector	Length (Sectors)	Size (Bytes)

CONTENTS OF CARVED FILE

File Name	File Size	Modified Date