

LAB 2: FORENSIC DISK COPYING

INTRODUCTION

The purpose of this lab is to learn how to take a forensic copy (image) of a disk, and to understand the role of hash functions in securing the integrity of our forensic image.

RESOURCES AND LINKS

- SANS Investigative Forensic Toolkit (SIFT) Workstation
<https://computer-forensics.sans.org/community/downloads>
- EXIF Tool
<http://www.sno.phy.queensu.ca/~phil/exiftool/>

LAB OUTLINE

1. SIFT Workstation Setup and Linux Re-Familiarisation	2
SIFT Workstation Setup	2
SHARED FOLDER SETUP	3
Linux Re-Familiarisation and SIFT Workstation Familiarisation	4
2. Hashing Functions	6
3. DD and Forensic Imaging	8
Preparing the Forensic OS and Storage Media	8
Creating the Forensic Image	11
4. Hashing and Verifying Disk Images	12
5. Basic Forensic Examination	14
Examining the Raw Disk Image	14
Mounting a Forensic Disk Image	15
Examining Files	15

LAB 2: FORENSIC DISK COPYING

1. SIFT WORKSTATION SETUP AND LINUX RE-FAMILIARISATION

In this session, we'll use a slightly different forensic environment to what we used last week – SIFT Workstation. SIFT Workstation is a VMWare based environment that lets us take forensic images of devices, and analyse those images all within the one virtual machine. We can also move files in and out of the virtual machine and use windows tools to analyse files if we need.

In this part, we'll start the SIFT Workstation and re-familiarise ourselves with some basic Linux commands.

SIFT WORKSTATION SETUP

- 1) Start the VMWare Launcher and locate the “SANS SIFT Workstation 3.0” from either the ICT30010 or ICT70006 Folder

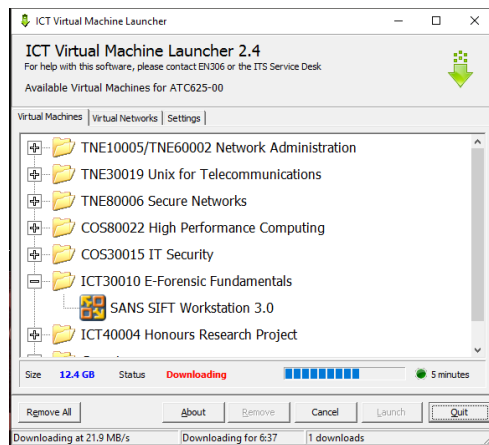


Figure 1. Virtual Machine Launcher Window

- 2) Once SIFT has started successfully, you will see a logon screen as below. Click on “sansforensics”, and enter the password “forensics”

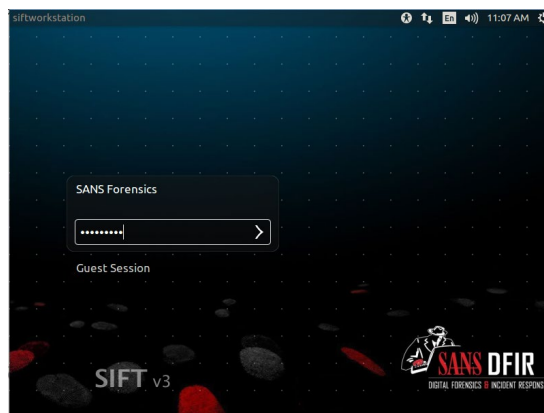


Figure 2. SIFT Workstation Logon Screen

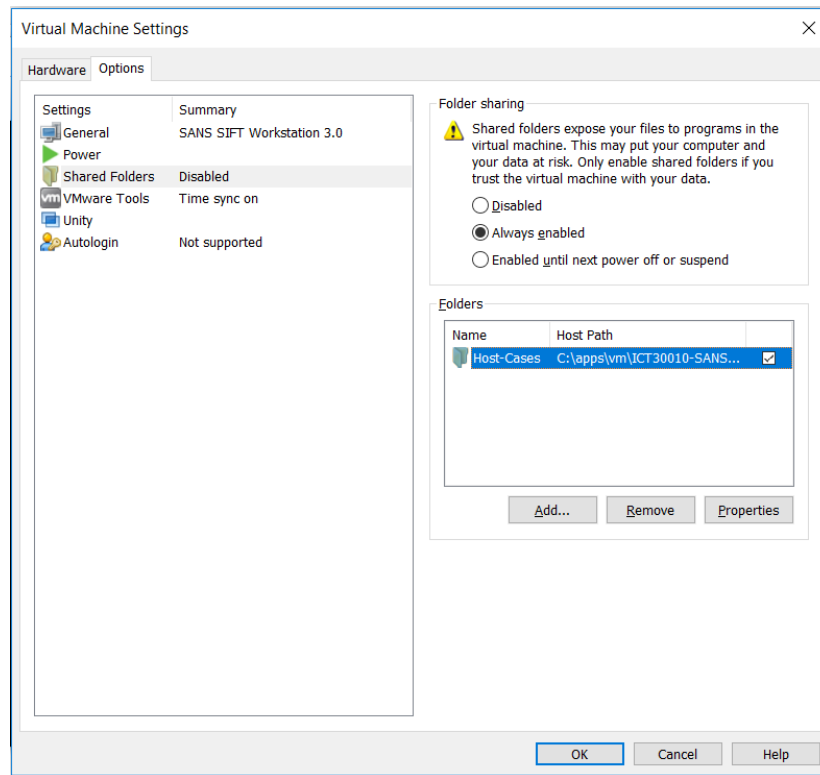
- 3) Once logged on, you'll be presented with the SIFT desktop, including a command line window.

LAB 2: FORENSIC DISK COPYING

SHARED FOLDER SETUP

Verify VMWare shared folder configuration

- 4) Navigate to the folder “C:\apps\vm\ICT30010-SANS-SIFT\” on the host Windows computer
- 5) Ensure that the folder named “Host-Cases” exists, if not create this folder
- 6) On the VMWare window click on Player -> Manage -> Virtual Machine Settings
- 7) Click on the Options tab and then click “Shared Folders”
- 8) Ensure that Folder sharing is set to “Always Enabled” and that the path is set to “C:\apps\vm\ICT30010-SANS-SIFT\Host-Cases”



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

LAB 2: FORENSIC DISK COPYING

LINUX RE-FAMILIARISATION AND SIFT WORKSTATION FAMILIARISATION

Since we'll be using SIFT Workstation a lot during the new few weeks, let's try running a few commands to become familiar with some of the main directories in the VM.

- 10) Check the current working directory by typing "`pwd`" at the command prompt. Note the result. This is the "home" directory for the user "sansforensics" (the Linux equivalent of C:\Documents and Settings\sansforensics\)

.....

- 11) Change directories to the desktop by typing "`cd Desktop`"

- 12) Get a long directory listing of the current directory by typing "`ls -l`" at the command prompt.

- 13) Observe the existence of the "cases" directory. This is, in fact a symbolic link (similar to a shortcut in windows). What directory does this link point to?

.....

- 14) Another directory we should become familiar with is "`/mnt/hgfs`". This directory contains all the folders that have been shared between VMWare and the SIFT Workstation. Sharing folders allows us to access files on the host windows system from within SIFT Workstation, thereby making it easy to copy files in and out of the Virtual Machine. Change to the directory now by running the command "`cd /mnt/hgfs`".

- 15) Find out the name of any folders we have shared with the Windows operating system by listing all the files in the hgfs directory (run "`ls`"). Note the name of the folder:

.....

- 16) Change into this directory ("`cd directoryname`"), then create a new file by running the command "`sudo touch myname`" (use your own name instead of "myname" – e.g. "`sudo touch jobloggs`")

Linux Command Cheat Sheet:

ls	list files in the current directory
ls -al	list all files (including hidden) with additional details
cd dir	change directory to <i>dir</i>
cd	change to home
pwd	show current directory
mkdir dir	create a directory <i>dir</i>
rm file	delete <i>file</i>
rm -r dir	delete directory <i>dir</i>
cp file1 file2	copy <i>file1</i> to <i>file2</i>
mv file1 file2	rename or move <i>file1</i> to <i>file2</i>
more file	output the contents of <i>file</i>
head file	output the first 10 lines of file
tail file	output the last 10 lines of file
touch file	create or update the time on <i>file</i>
sudo command	run <i>command</i> as root
date	show the current date and time
man cmd	show the manual for <i>cmd</i>
ping host	ping <i>host</i> and output results
ifconfig	show IP and MAC address info
df	show disk usage
fdisk -lu	list all disk partitions (run as root)
hdparm -I /dev/sda	show info about disk <i>sda</i>

LAB 2: FORENSIC DISK COPYING

17) Exit full-screen mode by pressing “Ctrl+Alt+Enter”, or pressing the maximize button at the top of the screen.

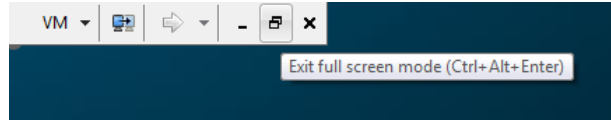


Figure 3. "Exit full screen mode" button in VMWare

18) Navigate to the folder “C:\apps\vm\ICT30010-SANS-SIFT\Host-Cases” on the Windows machine.

19) Verify that the file you created with your name is in this directory.

✓ Using the Host-Cases directory, we can easily move files in and out of the virtual machine. We'll use this in most of the later labs to bring forensic images into the VM, and other files out.


LAB 2: FORENSIC DISK COPYING

2. HASHING FUNCTIONS

Hashing functions are mathematical functions which take an arbitrary set of data (e.g. a computer file) and map it to a unique (for all practical purposes) “hash value”. This “hash value” is often described as a “digital fingerprint”. A good hash function should be easy to calculate from a set of data, but difficult to reverse. In other words, it should be impossible to determine or reconstruct the original data from the hash value.

Hash values are often used in forensics to quickly identify files. For example, hash values may be used to exclude common operating system files from a case, or to quickly highlight files known to be relevant (e.g. stolen intellectual property or contraband). Hash values are also commonly used to verify that the data contained within a forensic disk image is identical to the original disk, and to verify that the image does not change over time.

In this section, we will try two hash functions commonly used in forensics: MD5 and SHA1

✓ Make sure you’re in a new terminal window for these tasks. You don’t want to accidentally cancel the forensic imaging process. Open a new terminal window by clicking the  icon in the top menu bar.

- 20) Create a directory for this “case”. Run the command “`mkdir /cases/lab2`”
- 21) Change to the “cases” directory by typing “`cd /cases/lab2`” at the command prompt.
- 22) Create a text file with the contents “abcdefg” by typing “`echo abcdefg > file1.txt`”
- 23) View the contents of this file by running the command “`xxd file1.txt`”. The xxd command displays the hex values of each of the characters in a file on the left, and the ASCII representation of those characters on the right.

```
sansforensics@SIFT-Workstation:/cases/lab2$ xxd file1.txt
00000000: 6162 6364 6566 670a                abcdefg.
```

Offset from
beginning of file

Hex (base 16)
representation of bytes in
file (2 hex characters per
byte)

ASCII version of
file contents

Figure 4. Example output from xxd showing hex and ASCII representations

- 24) Note the hex values of the characters on the left-hand side (each character, or byte, is represented by two hex characters). “a” = 61, “b” = 62, etc. Note that the “echo” command added an extra character to the end of our file (hex “0a”). This is a new line character.
- 25) Create a second text file with the contents “abcdefh” (note we’ve changed the “g” to an “h” – only one letter difference) by typing “`echo abcdefh > file2.txt`”

LAB 2: FORENSIC DISK COPYING

- 26) View the contents of this file with the xxd command (`xxd file2.txt`)
- 27) There should only be one byte that different from file1.txt. What is the hex value for the new character ("h")?
-
- 28) Calculate the MD5 (Message Digest 5) values for the two files by running the command: `md5sum file?.txt` at the prompt (the question mark is a wildcard character, and asks the md5 command to run on any file with the name file^(any single character).txt).
- 29) Despite the similarity of the two input files (file1.txt and file2.txt), what do you notice about the two MD5 values returned?
-
- 30) Make a duplicate copy of file1.txt by running the command `cp file1.txt file3.txt`
- 31) Again, calculate the MD5 values of all three files by running `md5sum file?.txt`. What do you notice about the hash values for file1.txt and file3.txt?
-
- 32) MD5 values are typically presented in Hex (base 16). How many individual characters make up a MD5 value?
-
- 33) Since two hex characters represent one byte, how many bytes long is an MD5 value?
-
- 34) Hashing functions are more commonly described by the number of bits. Given that there are 8 bits in a byte, how many bits long is an MD5 value?
-
- 35) Calculate the SHA1 (Secure Hash Algorithm) values by running the command `sha1sum file?.txt`. How many individual characters make up a SHA1 value?
-
- 36) How many bytes long is a SHA1 value?
-
- 37) How many bits long is an SHA1 value?
-

LAB 2: FORENSIC DISK COPYING

3. DD AND FORENSIC IMAGING

The basic foundation of all forensic imaging tools is the ability to read every byte from an evidentiary disk (e.g. suspect's hard drive) and store a copy of this data in such a way that every byte can be traced back to an exact location on the original evidence. This is called a "disk level" or "physical" copy.

In Windows, specialised tools are required to access and copy a disk in this way, but every version of Linux comes with a simple tool to make these types of copies – "dd".

In this section, we will take a forensic image of a supplied hard disk drive, and verify a hash value for the forensic image. We'll use a slightly modified version of dd (dcfldd) in this exercise (since dd doesn't provide feedback until it's completed copying), but all the options we use in this exercise will also work with the normal dd command.

PREPARING THE FORENSIC OS AND STORAGE MEDIA

- 38) Determine what disks are connected to your SIFT workstation by running the command "sudo fdisk -lu". The output should be similar to this, this disk sizes and partitions can be different depending on host system configuration.

```
sansforensics@SIFT-Workstation:/cases$ sudo fdisk -lu

Disk /dev/sda: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders, total 62914560 sectors
Units = sectors of 1 * 512 = 512 bytes
Disk identifier: 0x0007eca3

   Device Boot      Start         End      Blocks    Id  System
/dev/sda1 *          63         481949      240943+   83   Linux
/dev/sda2            481950      4482134      2000092+   82   Linux swap / Solaris
/dev/sda3            4482135      62910539     29214202+   83   Linux

Disk /dev/sdb: 32.2 GB, 32212254720 bytes
255 heads, 63 sectors/track, 3916 cylinders, total 62914560 sectors
Units = sectors of 1 * 512 = 512 bytes
Disk identifier: 0x000c3c7e

   Device Boot      Start         End      Blocks    Id  System
/dev/sdb1            63         62910539     31455238+   83   Linux
```

Figure 5. Example output from fdisk -lu

- 39) Make a note of the currently connected disks. This will help you identify the name of the disk you are about to connect.

Remember that Linux typically names disks "/dev/sda" or "/dev/hda", while partitions are typically named "/dev/sda1", "/dev/hda2", etc.

LAB 2: FORENSIC DISK COPYING

NOTE: Due to COVID this lab has been modified. Normally we would complete the following tasks:

- 1. Note the physical details of a USB thumb drive**
- 2. Create a forensic image of the USB thumb drive**
- 3. Hash the forensic image to ensure data integrity**
- 4. Analyse the contents of the forensic image**

The lab will continue from the hashing phase but the instructions have been left as it is strongly recommended you complete steps 40 through to 51 in your own time with your own USB thumb drive, this is not required for the assessment

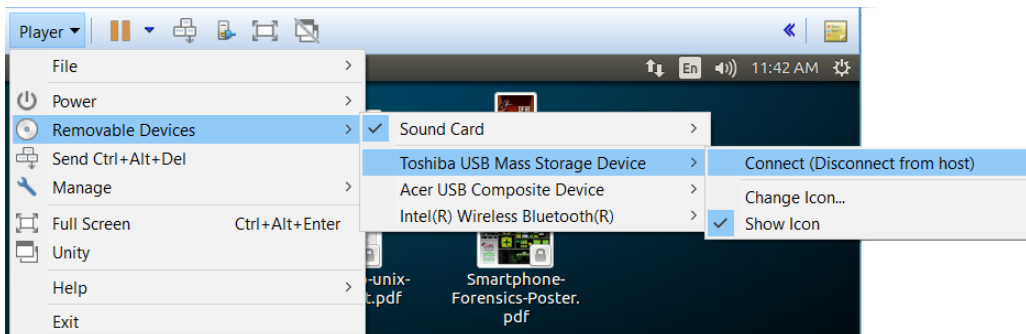
40) Note down the details of the USB drive you have been supplied. You should get into the habit of doing this any time you start examining a disk. This way, can always link your notes, or any information you obtain from the drive, back to the original evidence. If some of the details aren't available on the device (e.g. it doesn't have a serial number, or the size isn't written on it), make a note of that, too (e.g. write "unknown").

a) USB Drive:

- i) Make:
- ii) Size: MB /GB / TB
- iii) Serial #:

41) Connect the supplied USB drive to the host computer

42) On the VMWare window click Player -> Removable Devices -> Toshiba USB Mass Storage Device -> Connect (Disconnect from host)



43) Click "OK" on the warning message

44) Re-run the fdisk command to determine the name of the newly connected disk ("`sudo fdisk -lu`"). The output should show the disks you noted previously, and one new disk. If the disk doesn't appear straight away (and you've selected "Connect" from the VMWare console menu) you may need to wait for a minute while the operating system detects the drive)

LAB 2: FORENSIC DISK COPYING

- a) Identify the new drive and note the device name for this disk.

.....

- b) Note the sector size for the new disk drive: bytes

- c) Note the total size for the new disk drive: MB / GB

- d) Note the details of the partition reported by fdisk below (you will need a calculator to work out the total size – there’s one built into SIFT Workstation!). Remember the Total Sectors should be: EndSector-StartSector + 1 and Total Size in MB is (TotalSectors * SectorSize) / 1024 / 1024

Partition (e.g. /dev/hda1)	Partition Type (e.g. NTFS/FAT)	Start Sector	End Sector	Total Sectors	Total Size (MB)

- 45) Compare the total size of the partition to the total size of the disk. How much extra space is unused on this disk?

..... GB

✓ For this lab, we’re only interested in the data up to the end of the partition, and will ignore the extra available space. We should keep in mind, however, that in a real investigation it may be possible to have a hidden partition, or there may be remnants of previous partitions in the unused part of the drive that could contain critical information.

- 46) Check you have enough disk space on your /cases partition to store a forensic image of this drive by running the command “df -h”. Note the number in the “available” column. Record the available space

..... GB

```
sansforensics@SIFT-Workstation:~$ df -h
Filesystem      Size  Used Avail Use% Mounted on
/dev/sda3       28G   4.4G   22G   17% /
/dev/sdb1       30G   11G   18G   39% /cases
/dev/sda1       228M   37M   180M   17% /boot
```

Figure 6. Example output from df

- 47) Make sure you are still in the cases directory, type “cd /cases/lab2”

LAB 2: FORENSIC DISK COPYING

CREATING THE FORENSIC IMAGE

- 48) Run the command “`sudo dcfldd if=/dev/sdx of=image.dd bs=512 count=xxxxx`”. Be sure to replace “/dev/sdx” with the device name for the usb disk that you noted earlier in step 44a), and make sure xxxxx is the same as the end sector for the partition you located earlier in step 44d. This command will image only the number of blocks requested in the count parameter (of size 512 bytes) from the start of the drive (to speed up time in the imaging process, we are not interested in the space at the end of the partition).

✓ Often, we may want to prepare our storage media before making a forensic image. If we were planning on storing our images on an external hard drive, we would typically “sanitise” it by writing zeros to the entire disk surface. We can do this using “`dd if=/dev/zero of=/dev/ourstoragedrive`”. This prevents possible arguments about data contamination down the track.

- 49) This may take a while to complete (you’ll be able to see how far through it is by watching the output – the total number of Mb to be written should be *approximately* the same as the partition size you noted earlier).

```
sansforensics@SIFT-Workstation:/cases/lab2$ sudo dcfldd if=/dev/sdx of=image.dd
184832 blocks (5776Mb) written.
```

Figure 7. Example output from dcfldd – part way through imaging

LAB 2: FORENSIC DISK COPYING

4. HASHING AND VERIFYING DISK IMAGES

Once we've completed taking an image of a disk, we want to verify that it is identical to the original, and create the digital equivalent of a tamper-evident seal on our evidence. We use hash functions for this.

- 50) Once the forensic image process has completed you should see something like the following (x records in, x records out).

```
sansforensics@SIFT-Workstation:/cases/lab2$ sudo dcfldd if=/dev/sdx of=image.dd
256768 blocks (8024Mb) written.
256912+0 records in
256912+0 records out
sansforensics@SIFT-Workstation:/cases/lab2$
```

Figure 8. Example output after dcfldd has completed imaging

- 51) Note the total number of Mb written. Note that this may be slightly larger than the total partition size you noted earlier (because we've included the data before the partition)

.....

NOTE: Continue from step 52 using the provided forensic image

- 52) Verify that you're still in the /cases/lab2 directory by typing "pwd"
(if you're not, type "cd /cases/lab2")

- 53) 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 "image.dd.gz" into this folder (located in the T:\HED\ICT30010 folder)

- 54) Prepare Lab Files

- a) Check that you can access the image.dd.gz file by listing files in the VMWare shared folder:
"ls /mnt/hgfs/Host-Cases"
- b) Unzip the lab files to the cases directory:
"gzip -dkc /mnt/hgfs/Host-Cases/image.dd.gz > /cases/lab2/image.dd"

LAB 2: FORENSIC DISK COPYING

- 55) Check the permissions on the image by getting a long directory listing (`ls -l`). Observe the existence of the “w” flag in the left column. Linux allows the user to control three permission types (read, write, execute) for the owner of the file, the group and anyone else. In this case, the owner flags are “rw-”, the group flags are “r--” and the everyone else flags are set to “r--”.

```
-rw-r--r-- 1 root root 1073741824 2011-04-15 09:33 image.dd
```

Figure 9. Permissions for image.dd file with write flag for owner

- 56) Make the file read-only by running the following command `sudo chmod a-w image.dd`. Verify that this worked correctly by getting a long directory listing (`ls -l`), and checking that no “w” flags exist for the image file.

```
-r--r--r-- 1 root root 1073741824 2011-04-15 09:33 image.dd
```

Figure 10. Permissions for read-only image.dd file showing no “w” (write) flags

- 58) Calculate an MD5 hash for the forensic image and view the results with the command `md5sum image.dd > image.dd.md5 ; cat image.dd.md5`.

Hash value:

- 59) The `fdisk` command can be used to identify the partitions contained within a forensic image with the command `sudo fdisk -lu image.dd`
- 60) Note the details of the partition reported by `fdisk` below (you will need a calculator to work out the total size –there’s one built into SIFT Workstation!). Remember the Total Sectors should be: $\text{EndSector} - \text{StartSector} + 1$ and Total Size in MB is $(\text{TotalSectors} * \text{SectorSize}) / 1024 / 1024$

Partition (e.g. /dev/hda1)	Partition Type (e.g. NTFS/FAT)	Start Sector	End Sector	Total Sectors	Total Size (MB)

LAB 2: FORENSIC DISK COPYING

5. BASIC FORENSIC EXAMINATION

EXAMINING THE RAW DISK IMAGE

One way to examine a forensic disk image is to simply run searches over the disk image file itself. While this might not give us the full picture, it can be a quick way of identifying if evidence we're looking for is actually present, particularly if we're just looking for text content.

In this section, we'll use various standard Linux commands like "grep", "strings", and "xxd" to examine the forensic image.

61) Check that you're in the "lab2" directory by running the command "pwd". If you're not in the right directory, run "cd /cases/lab2"

62) Search for the string "secret" within the forensic image

- a) Run "xxd image.dd | grep secret". This command takes the hex output from xxd and searches it for the word "secret". Note the offset of the first occurrence:

.....

(press "CTRL-C" to cancel the search once the first result has displayed, as we don't need to search the whole disk)

- b) View the text surrounding the offset noted above by running "xxd -s 0xNNNN -l 200 image.dd" (where NNNN is the offset noted above). You can try increasing the "-l" value from 200 if you wish to read more. What is the "Super secret document" about?

.....
.....

63) Search for text strings that look like file names

- a) Find strings that look like image file names (ending with a ".jpg") by running "strings --encoding=l --radix=d image.dd | grep "\.jpg\$" (note the "" included in the command). This searches the image file for Unicode strings (encoding = l), showing the file offset in decimal (radix=d), and only displays results matching the grep expression (looks like a jpg filename). What is the name of the JPEG file?

.....

(again, press CTRL-C to cancel the search once a result has been displayed – no need to search the entire disk)

LAB 2: FORENSIC DISK COPYING

MOUNTING A FORENSIC DISK IMAGE

Usually, we examine a forensic image by “mounting” the image so we can view the files and directories as they would have appeared to the user. We either do this by using a specialised forensic tool which understands the file system on the forensic image (e.g. FAT, NTFS) or we can mount the drive in an operating system which natively supports the file system.

In this case, we’re going to mount the disk image using the SIFT Workstation’s native support of the NTFS file system.

- 64) Verify that you’re in the “/cases/lab2” directory by running the “`pwd`” command.
- 65) We’ll now need to know how many bytes into the drive the first partition starts. Given the sector size you noted earlier, and the starting sector of the partition (both values should be written in the table), calculate this value:

..... sectors x bytes per sector = bytes offset

- 66) Mount the first partition in the forensic image
- a) Run “`sudo mount -t ntfs -o ro,loop,offset=x image.dd /mnt/windows_mount/`” (replace “x” with the byte offset you just calculated)
 - b) Change to the /mnt/windows_mount directory and get a directory listing to ensure the drive mounted correctly (“`cd /mnt/windows_mount`” then “`ls`”)
- 67) Note the filenames you see in this directory (there should be five files and two directories):

.....
.....
.....

EXAMINING FILES

- 68) Verify that you’re in the “/mnt/windows_mount” directory by running the “`pwd`” command.
- 69) Examine the JPEG Image, and manually locate metadata (EXIF) within the image file
- a) Examine the start of the image.jpg file in xxd by running the command “`xxd image.jpg | head -15`”
 - b) What model camera/phone was used to take this image?
 - c) What date/time was the image taken?

LAB 2: FORENSIC DISK COPYING

70) Examine the EXIF data within SIFT workstation, using the “exif” command line tool.

- a) Run the command “`exif image.jpg`”
- b) Confirm the “Model”, “Manufacturer” and “Date and Time” values you obtained above
- c) Also note the Latitude and Longitude values contained within the image (values returned by exif are in Hours, Minutes, Seconds – in that order, each separated by a single ,). Also make sure you note if the values are North/South, or East/West

		Hours	Minutes	Seconds
Latitude:	North / South			
Longitude:	East / West			

Let’s find out where this image was taken. We’d like to put the values we’ve located into google maps, but Google only accepts decimal latitudes and longitudes. Thankfully, our suspect USB drive has on it a script to convert latitudes and longitudes from Degrees, Minutes and Seconds into decimal values.

- d) Run the command “`python dms.py h,m,s h,m,s`” (where h,m,s are the hours, minutes and seconds for latitude and longitude) to convert to decimal. Your command might be “`python dms.py xx,xx,xx yy,yy,yy`”. Note the output from this script:

```
sansforensics@SIFT-Workstation:/mnt/windows_mount$ python /images/dms.py 37,20,40
144,30,2
Input:      (37,20,40), (144,30,2)
Output:     (37.344444, 144.500556)
```

Figure 11. Example Output from dms.py

- e) Adjust your answer from the dms script to account for direction (North/South or East/West)
 - i) If your latitude was South, make the first value negative (i.e. 37.344444 becomes -37.444444)
 - ii) If your longitude was West, make the second value negative (i.e. 144.500556 becomes -144.50556.
- f) Open a web browser (back in your Swinburne windows workstation), and enter your answer from the above question into Google maps (<http://maps.google.com>). Where was this photo taken?

LAB 2: FORENSIC DISK COPYING

71) Let's now examine the Microsoft Word Document

- a) Run the command `xxd -s 137900 -l 300 word.doc` (hex dump beginning at byte 137900 for 300 bytes). There should be two names present in this document. What are they?
 - i)
 - ii)
- b) The command "exiftool" can do more than just examine EXIF data in Image files, but can examine metadata contained within more than 100 different file types. Run the "exiftool" command on the Microsoft word document by typing `exiftool word.doc`
 - i) When was the document last printed?
 - ii) Who is listed as the "Author" of the document?
 - iii) What user last saved the document?