Case Name: Introduction to Single Purpose Forensic Tools

Couse Name: IST402

Instructor: Robert Price

Date: 03/17/2025

Examiner Name: Jaspreet Singh

Table of Contents

LIST OF ILLUSTRATIVE MATERIALS	3
TABLES	3
FIGURES	3
EXECUTIVE SUMMARY	4
BACKGROUND	4
EVIDENCE	4
COLLECTION AND ANALYSIS	5
COLLECTION	5
ANALYSIS	5
Hashing	5
Mounting	8
Foremost	9
HEX Editor	12
CONCLISION	13
Hashing	13
Mounting	14
Foremost	14
HEX Editor	14

List of Illustrative Materials

Table	es1
Tak	ole 1: Evidence
Figur	es5
Hash	ing
	Figure 1: Access Forensic Files & Viewing Flag6
	Figure 2: Viewing FTK Imager Hash File
	Figure 3: Cat Command Hash Verification
	Figure 4: Cat MD5 Checksum
	Figure 5: Cat SHA1 Checksum
Mour	nting
	Figure 6: Viewing Forensics Folder and Files
	Figure 7: Mount Contents 8
	Figure 8: Mounting9
Forer	nost
	Figure 9: Foremost Menu9
	Figure 10: Foremost Configuration File10
	Figure 11: Foremost JPG Recovery10
	Figure 12: Foremost GIF Recovery11
	Figure 13: Foremost AVI Recovery11
	Figure 14: Foremost EXE Recovery11
	Figure 16: Foremost PNG Recovery11
	Figure 16: Foremost DOC Recovery12
	Figure 16: Foremost PDF Recovery12
HEX	Editor
	Figure 18: Opening HEX Editor in Command Prompt12
	Figure 19: JPG HEX Editor13

EXECUTIVE SUMMARY

Background

The examiner will demonstrate an understanding of forensic methodology, key forensic concepts, identifying types of evidence on current Windows operating systems and be familiar with the structure and composition of modern Windows file systems. The examiner will demonstrate an understanding of the methodologies and tools used to collect and process digital forensic evidence.

Hashing is the process of taking in a stream of plain text and transforming the data into a hashed text using a hashing algorithm. You can use the hash to make sure that a message was not modified during transmission. That hash can make sure that the disk image was not tampered with. Hashed images are used in forensics investigations. Hashing is also used on files, passwords, and other pieces of data. In this lab, the examiner is going to image a disk and create a hash of that disk, verify integrity using file hashing tools, use Foremost to carve and recover deleted files from a disk, and use a hex editor to review files. Foremost is a file carving tool used to recover files based on known headers and file signatures, even after they have been deleted from the file system. A hex editor is a tool that allows the examiner to inspect the raw hexadecimal content of files, which is helpful in verifying file types and identifying data fragments.

Evidence

Description	Hash	Hash Value	Examiner
	Algorithm		
Evidence	MD5	6958437CFB625D29A17121893E07402C	Jaspreet
			Singh
Evidence	SHA1	FEE3A78ADF5DD06D048BC90345CA7C546CF38D09	Jaspreet
			Singh

COLLECTION AND ANALYSIS

Collection

The examiner began by verifying the integrity of the disk image called image.dd by using and utilizing the md5 and sha1 hash values to configure that they match the original FTK generated hash values. Which were verified though, .txt files which contained the original hash values hence, once checked to ensure an exact match for the hash values the examiner would know that the image's integrity is maintained, and the image has not been altered.

The examiner then mounted the disk image using the mount command in the command prompt hence allowing the examiner to see once mounted that there are no visible files present. Therefore, suggesting that there has been deleted data, and it was not overwritten. The examiner utilized the following command "ls -la" to check for hidden files and verify that the disk image was mounted in read-only mode so that the integrity of the image is preserved through this process.

To then recover this deleted data the examiner uses a forensic tool called Foremost which is a file carving tool to extract specific types of files from the disk image. In this case the examiner did separate command prompt output directories using Foremost for the following file types: JPG, GIF, AVI, PNG, EXE, DOC, and PDF. After each file type was carved, the examiner was able to review each directory to verify that the forensic tool was able to successfully recover files based on the file headers and signatures. Then the examiner was able to see audit.txt in the command prompt which showed the examiner a log for each file carving and showed the number and types of files recovered from each type.

Then the examiner went into the directory and containing the carved JPG files and opened the files using a hex editor. By examining the raw hexadecimal content, the examiner was able to confirm the file signature of "JFIF" which indicates a JPEG image. From that the examiner was able to see that raw hex analysis can also be used to verify the integrity of the recovered files from the system.

Analysis

Forensic File Viewing

The examiner was able to use the command prompt to navigate to the forensics directory and viewed its contents using the ls command. Then the examiner uses an application called leafpad to open the sampleflag.txt file hence showing that the flag value is 999818.

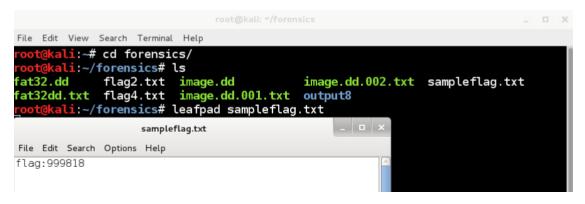


Figure 1: Access Forensic Files & Viewing Flag

FTK Imager Hash Verification

The examiner was able to open a .txt file by using the ls command "ls image.dd.001.txt" in the prompt hence allowing this file to open and show the examiner data about the disk image as well as the hash values. The examiner can see that both the MD5 and SHA1 checksums are both marked as verified, confirming that the image's integrity is maintained.

```
File Edit Search Options Help
Created By AccessData® FTK® Imager 3.1.3.2
flag:311226
Case Information:
Acquired using: ADI3.1.3.2
Case Number:
Evidence Number:
Unique Description:
Examiner:
Notes:
Information for H:\image.dd:
Physical Evidentiary Item (Source) Information:
[Device Info]
Source Type: Logical
[Drive Geometry]
Bytes per Sector: 512
Sector Count: 202,752
[Physical Drive Information]
Removable drive: False
Source data size: 99 MB
Sector count:
                  202752
[Computed Hashes]
MD5 checksum:
                  6958437cfb625d29a17121893e07402c
SHA1 checksum: fee3a78adf5dd06d048bc90345ca7c546cf38d09
Image Information:
                       Mon Dec 02 15:19:04 2013
Acquisition started:
 Acquisition finished: Mon Dec 02 15:19:14 2013
Segment list:
 H:\image.dd.001
Image Verification Results:
 Verification started: Mon Dec 02 15:19:14 2013
Verification finished: Mon Dec 02 15:19:14 2013
MD5 checksum: 6958437cfb625d29a17121893e07402c : verified
SHA1 checksum: fee3a78adf5dd06d048bc90345ca7c546cf38d09 : verified
```

Figure 2: Viewing FTK Imager Hash File

Cat Command Hash Verification

The examiner can use the "cat" command which is used to display the contents of a file directly in the terminal to verify that both the MD5 and SHA1 hash values match and are verified.

```
Information for H:\image.dd:
Physical Evidentiary Item (Source) Information:
[Device Info]
Source Type: Logical
[Drive Geometry]
Bytes per Sector: 512
Sector Count: 202,752
[Physical Drive Information]
Removable drive: False
Source data size: 99 MB
Sector count:
                  202752
[Computed Hashes]
MD5 checksum:
                  6958437cfb625d29a17121893e07402c
SHA1 checksum:
                  fee3a78adf5dd06d048bc90345ca7c546cf38d09
Image Information:
Acquisition started:
                        Mon Dec 02 15:19:04 2013
Acquisition finished: Mon Dec 02 15:19:14 2013
Segment list:
 H:\image.dd.001
Image Verification Results:
Verification started: Mon Dec 02 15:19:14 2013
Verification finished: Mon Dec 02 15:19:14 2013
MD5 checksum:
                  6958437cfb625d29a17121893e07402c : verified
SHA1 checksum:
                  \texttt{fee} 3a 78 ad \texttt{f} 5dd 06d 048 bc 90345 ca 7c 546 cf 38d 09 : \texttt{verified}
```

Figure 3: Cat Command Hash Verification

Cat Command MD5

The examiner uses the cat command in the prompt to extract the MD5 has value from the image.dd.001.txt file and the checksum value to verify integrity. The examiner was able to use "grep MD5" in the prompt to extract only information on the MD5 hash values.

```
root@kali:~/forensics# cat image.dd.001.txt | grep MD5
MD5 checksum: 6958437cfb625d29a17121893e07402c
MD5 checksum: 6958437cfb625d29a17121893e07402c : verified
root@kali:~/forensics# md5sum image.dd
6958437cfb625d29a17121893e07402c image.dd
```

Figure 4: Cat MD5 Checksum

Cat Command SHA1

The examiner uses the cat command in the prompt to extract the SHA1 has value from the image.dd.001.txt file and the checksum value to verify integrity. The

examiner was able to use "grep SHA1" in the prompt to extract only information on the SHA1 hash values.

```
root@kali:~/forensics# cat image.dd.001.txt | grep SHA1
SHA1 checksum: fee3a78adf5dd06d048bc90345ca7c546cf38d09
SHA1 checksum: fee3a78adf5dd06d048bc90345ca7c546cf38d09 : verified
root@kali:~/forensics# sha1sum image.dd
fee3a78adf5dd06d048bc90345ca7c546cf38d09 image.dd
```

Figure 5: Cat SHA1 Checksum

Files and Directory before Mounting

The examiner can use the following command "ls -l" to view information about the forensic folder and the details of the files within the folder.

```
-/torensics# ls -l
total 2456072
rwxrw-rw- 1 root root 2411168256 Dec 22
                                         2013 fat32.dd
                                          2019 fat32dd.txt
                              122 Jan 14
rwxrw-rw- 1 root root
                               12 Jun 20
                                         2018 flag2.txt
rw-r--r-- 1 root root
   r--r-- 1 root root
                               12 Jun 20
                                          2018 flag4.txt
rwxrw-rw- 1 root root 103809024 Dec 2
                                          2013 image.dd
rwxrw-rw- 1 root root
                             1105 Jun 20
                                          2018 image.dd.001.txt
rwxr--r-- 1 root root
                             1105 Jun 20
                                          2018 image.dd.002.txt
                             4096 Jun 20 2018 output8
drwxr-xr-x 3 root root
                             4096 Mar 17 10:06 partition
drwxr-xr-x 2 root root
-rw-r--r-- 1 root root
                               12 Jun 20 2018 sampleflag.txt
```

Figure 6: Viewing Forensics Folder and Files

Checking Contents of Mount

The examiner runs the following command "ls -la partition" to check for hidden files and confirm the partition was empty prior to mounting.

```
root@kali:~/forensics# ls -la partition
total 5
drwxr-xr-x 2 root root 1024 Dec 31 1969 .
drwxr-xr-x 4 root root 4096 Mar 17 10:06 ..
```

Figure 7: Mount Contents

Mount the Disk Image

The examiner uses the following command to mount the disk image "mount | grep vfat" this then confirms that image.dd was mounted to the directory using the VFAT file system. Then after the examiner made a mistake by misspelling the umount

partition command by saying "unmount" but after executing the command correctly with umount the examiner safely unmounts the image.

```
root@kali:~/forensics# mount | grep vfat
/root/forensics/image.dd on /root/forensics/partition type vfat (rw,relatime,fma
sk=0022,dmask=0022,codepage=cp437,iocharset=utf8,shortname=mixed,errors=remount-
ro)
root@kali:~/forensics# unmount partition
bash: unmount: command not found
root@kali:~/forensics# umount partition
```

Figure 8: Mounting

Viewing Foremost Help Menu/Options

The examiner uses the "foremost -h" command to display available options and instructions for the Foremost forensic file carving tool. This menu provides a reference for the examiner for specifying file types, input files, and output directories during the file recovery process.

Figure 9: Foremost Menu

Viewing Foremost Configuration File

The examiner used the following command "head -n 20 /etc/foremost.conf" to preview the Foremost configuration file. Within the file defines how Foremost identifies file types based on headers and footers and allows customization for supported file formats.

```
root@kali:~/forensics# head -n 20 /etc/foremost.conf

# Foremost configuration file

# Note the foremost configuration file is provided to support formats which

# don't have built-in extraction functions. If the format is built-in to foremo

st

# simply run foremost with -t <suffix> and provide the format you wish to extrac

t.

# The configuration file is used to control what types of files foremost

# searches for. A sample configuration file, foremost.conf, is included with

# this distribution. For each file type, the configuration file describes

# the file's extension, whether the header and footer are case sensitive,

# the maximum file size, and the header and footer for the file. The footer

# field is optional, but header, size, case sensitivity, and extension are

# not!

# Any line that begins with a '#' is considered a comment and ignored. Thus,

# to skip a file type just put a '#' at the beginning of that line

# Headers and footers are decoded before use. To specify a value in
```

Figure 10: Foremost Configuration File

JPG Recovery

The examiner was able to execute the following command "foremost -i image.dd -t jpg -o output1" to recover the deleted JPG files from the disk image. In this the command the "-i" is specifying the input file which is image.dd. Then the "-t" tells the type of file to search and look for which is JPG. Finally, the "-o" tells the output directory will be stored in output 1. The examiner then executes the command "cat output1/audit.txt" to then find that 83 JPG images were recovered and the details for the JPG images as well.

```
root@kali:~/forensics# foremost -i image.dd -t jpg -o outputl
Processing: image.dd
|*|
```

Figure 11: Foremost JPG Recovery

GIF Recovery

The examiner executes the same foremost command as done before for JPG, "foremost -i image.dd -t gif -o output2" but this time for GIF's then saving it as output2. The examiner then executes the command "cat output2/audit.txt" to then find that 10 GIFs were recovered and the details for the GIFS as well.

```
root@kali:~/forensics# foremost -i image.dd -t gif -o output2
Processing: image.dd
|*|
root@kali:~/forensics# cat output2/audit.txt
```

Figure 12: Foremost GIF Recovery

AVI Recovery

The examiner executes the same foremost command but for avi as shown "foremost -i image.dd -t avi -o output3". The examiner then executes the command "cat output3/audit.txt" to then find that 5 AVIs were recovered and the details for the AVI's as well.

```
root@kali:~/forensics# foremost -i image.dd -t avi -o output3
Processing: image.dd
|*|
root@kali:~/forensics# cat output3/audit.txt
```

Figure 13: Foremost AVI Recovery

EXE Recovery

The examiner executes the same foremost command but for exe as shown "foremost -i image.dd -t exe -o output4". The examiner then executes the command "cat output4/audit.txt" to then find that 3 EXEs were recovered and the details for the EXE's as well.

```
root@kali:~/forensics# foremost -i image.dd -t exe -o output4
Processing: image.dd
|*|
root@kali:~/forensics# cat output4/audit.txt
```

Figure 14: Foremost EXE Recovery

PNG Recovery

The examiner executes the same foremost command but for png as shown "foremost -i image.dd -t png -o output5". The examiner then executes the command "cat output5/audit.txt" to then find that 230 PNGs were recovered and the details for the PNG's as well.

```
root@kali:~/forensics# foremost -i image.dd -t png -o output5
Processing: image.dd
|*|
root@kali:~/forensics# cat output5/audit.txt
```

Figure 15: Foremost PNG Recovery

DOC Recovery

The examiner executes the same foremost command but for png as shown "foremost -i image.dd -t doc -o output6". The examiner then executes the command "cat output6/audit.txt" to then find that 3 doc files were recovered and the details for the doc files as well.

```
root@kali:~/forensics# foremost -i image.dd -t doc -o output6
Processing: image.dd
|*|
root@kali:~/forensics# cat output6/audit.txt
```

Figure 16: Foremost DOC Recovery

PDF Recovery

The examiner executes the same foremost command but for png as shown "foremost -i image.dd -t pdf -o output7". The examiner then executes the command "cat output7/audit.txt" to then find that 3 pdf files were recovered and the details for the pdf files as well.

```
root@kali:~/forensics# foremost -i image.dd -t pdf -o output7
Processing: image.dd
|*|
root@kali:~/forensics# cat output7/audit.txt
```

Figure 17: Foremost PDF Recovery

Command Prompt Opening HEX Editor

The examiner executes the following cod below to navigate to the jpg folder within the output1 directory, which contains the recovered JPG files from the disk image. Then the examiner uses the hex editor command to then begin a hexadecimal analysis on the carved images files from Foremost.

```
root@kali:~/forensics# pwd
/root/forensics
root@kali:~/forensics# cd output1
root@kali:~/forensics/output1# cd jpg
root@kali:~/forensics/output1/jpg# hexeditor
```

Figure 18: Opening HEX Editor in Command Prompt

JPG HEX Editor

The examiner opens the recovered JPG file called 00043125.jpg which is a unique identifier assigned by Foremost in the hex editor application and confirmed the presence of the JFIF file signature which helps verify that the file is a valid JPEG file.

root@kaii: "/rorensics/output1/jpg																	
File Edit View Search Terminal Help																	
File: 000	4312	5. i	pa					1	SCII	0f	fset	t: 0	x000	0000	000	/ 0x	000060EE (%00)
00000000	GF.	De	FF	ΕŪ	00	10	4A	46	49	46	00	01	01	01	00	60	JFIF`
00000010	_		00	00	FF	E1	00	36	45	78	69	66	00	00	49	49	.`6ExifII
00000020	2A	00	08	00	00	00	02	00	01	03	05	00	01	00	00	00	*
0000030	26	00	00	00	03	03	01	00	01	00	00	00	00	4B	3E	06	&K>.
0000040	00	00	00	00	A0	86	01	00	8F	В1	00	00	FF	DB	00	43	
00000050	00	08	06	06	07	06	05	08	07	07	07	09	09	08	0A	0C	
0000060	14	0D	0C	0B	0B	0C	19	12	13	0F	14	1D	1A	1F	1E	1D	
00000070	1A	1C	10	20	24	2E	27	20	22	20	23	1C	1C	28	37	29	\$.' ",#(7)
00000080	2C :	30	31	34	34	34	1F	27	39	3D	38	32	3C	2E	33	34	,01444.'9=82<.34
0000090	32	FF	DB	00	43	01	09	09	09	0C	0B	0C	18	0D	0D	18	2C
00000A0	32	21	10	21	32	32	32	32	32	32	32	32	32	32	32	32	2!.!22222222222
00000B0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	22222222222222
00000C0	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	32	22222222222222
00000D0	32	32	32	32	32	32	FF	C0	00	11	08	00	88	02	85	03	222222
00000E0	01	22	00	02	11	01	03	11	01	FF	C4	00	1F	00	00	01	."
00000F0	05	01	01	01	01	01	01	00	00	00	00	00	00	00	00	01	
0000100	02	03	04	05	06	07	98	09	0A	0B	FF	C4	00	B5	10	00	
00000110	02	01	03	03	02	04	03	05	05	04	04	00	00	01	7D	01	}.
00000120	02	03	00	04	11		12	21	31	41	06	13	51	61	07	22	!1AQa."
00000130	71	14	32	81	91	A1	98	23	42	В1	C1	15	52	D1	F0	24	q.2#BR\$
00000140			72	82	09	0A	16	17	18		1A	25	26		28	29	3br%&'()
00000150	2A :	34	35	36	37	38	39	34	43	44	45	46	47	48	49	4A	*456789:CDEFGHIJ
0000160		54	55	56	57	58	59	5A	63	64	65	66	67	68	69	6A	STUVWXYZcdefghij
00000170			75	76	77	78	79	7A	83	84	85	86	87	88	89	88	stuvwxyz
0000180	92		94	95	96	97	98	99	9A	A2	AЗ	A4	Α5	A6	Α7	8A	
00000190			B2	B3	В4	B5	В6	В7	B8	В9	BA	C2	C3	C4	C5	C6	
000001A0		C8	C9	CA	D2	D3	D4	D5	D6	D7	D8	D9	DA	E1	E2	E3	
000001B0			E6	E7	E8	E9	ΕA	F1	F2	F3	F4	F5	F6	F7		F9	
00001C0			C4	00	1F	01	00	03	01	01	01	01	01	01	01	01	
00001D0			00	00	00	00	00	01	02	03	04	05	06		08	09	
`G Help	^C	Exi	it ((No	Save	≥)	^1	∫ go	To 01	ffs€	et	^X	Ex	it a	and	Save	^W Search

Figure 19: JPG HEX Editor

Conclusion

Overall, through this lab the examiner was able to gain hands-on experience using single-purpose forensic tools to verify data integrity, recover deleted files and analyze file content at a byte level. The examiner used the following forensic tools such as sha1sum, md5sum, mount, Foremost, and HEX editor to simulate and learn real-world forensic techniques and maintain evidence integrity throughout the entire investigation process.

Hashing

The examiner utilized the md5sum and sha1sum in to generate and compare hash values for the disk image image.dd file. Therefore, through these checksums of md5 and sha1 the examiner was able to confirm the disk images integrity against the

original FTK hash values. Hence ensuring that the disk image was reliable to use for forensic analysis and the examiner can continue with the forensic investigation.

Mounting

The examiner was able to create a mount in the command prompt then mounted the disk image in read-only mode using the mount command. From there the examiner was able to see that no visible or hidden files were present. Hence indicating to the examiner that there is deleted data that needs to be recovered using additional forensic tools.

Foremost

Through this lab the examiner utilized a forensic tool called Foremost which is used to help carve out deleted file types such as JPG, GIF, AVI, PNG, EXE, DOC, and PDF from the disk image. Then the examiner was able to output each file type into their own output directories and audit log each file so that the examiner can confirm the successful completion and recovery of files.

HEX Editor

Finally, the examiner uses HEX editor to examine and recovered JPF files from output 1 and confirmed that the JPEG pictures are verified due to the JFIF signature. This usage of the HEX editor application highlights the importance of byte level analysis when detecting a files authenticity and trying to find hidden data.