

Part 1. Disk imaging using FTK Imager

Task 1:

(Note: Answers are in **RED**)

1. After the imaging process was complete, what files did FTK Imager create? (Select all that apply.)
 - **The image file with an extension of .001**
 - **A text file for image summary**
 - An image file with an extension of .Ex01
 - No files
2. During the imaging process, you should have noticed that "Verify images after they are created" is checked by default. What is the result of having this option checked? (choose a single best answer)
 - FTK imager will compute the hash value of the image
 - FTK imager will compute the hash value of the USB drive
 - **FTK imager will compute the MD5 and SHA1 hashes of the USB drive and the MD5 and SHA1 hashes of the image, and verify the hashes match.**
 - FTK imager will compute the MD5 hash of the USB drive and the MD5 hash of the image, and verify the hashes match
3. How many hash algorithms did the FTK imager use to verify the image has not been altered? (choose a single best answer)
 - One hash algorithm
 - **Two hash algorithms**
 - Three hash algorithms
 - Four hash algorithms

Task 2: Screenshot of FTK imager after you have loaded the USB image was loaded into the FTK imager.

The screenshot displays the FTK Imager interface with a loaded USB image. The interface is divided into several panes:

- Evidence Tree:** Shows the hierarchical structure of the image. It includes Partition 1 (14793MB) with a FAT32 file system, containing a root directory with System Volume Information and [unallocated space]. Partition 2 (0MB) is also shown with a FAT12 file system and a root directory containing various files and folders, including SteamLibrary, System Volume Information, and [unallocated space].
- File List:** A table listing the files found in the image. The columns are Name, Size, Type, and Date Modified.
- Hex View:** A pane showing the raw data of the selected file, with columns for address, hex values, and ASCII characters.
- Custom Content Sources:** A pane at the bottom left showing the evidence source as 'Evidence:File System|Path|File'.

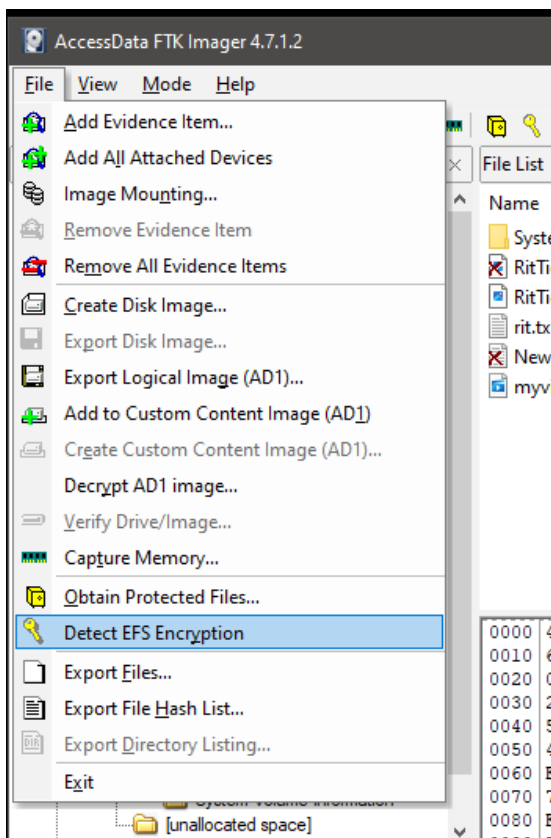
Name	Size	Type	Date Modified
System Volume Information	8	Directory	2/3/2024 12:13:28 PM
RitTigers.jiff	0	Regular File	2/3/2024 5:48:28 PM
RitTigers.jiff	31	Regular File	2/3/2024 5:47:58 PM
rit.txt.txt	1	Regular File	2/3/2024 5:47:12 PM
New Text Document.txt	0	Regular File	2/3/2024 5:46:34 PM
myvideo.mp4	7,962	Regular File	2/3/2024 6:03:16 PM

Address	Hex	ASCII
0000	42 20 00 49 00 6E 00 66 00 6F 00 0F 00 72 72 00	B .I.n.f.o...rr.
0010	6D 00 61 00 74 00 69 00 6F 00 00 00 6E 00 00 00	m.a.t.i.o...n...
0020	01 53 00 79 00 73 00 74 00 65 00 0F 00 72 6D 00	.S.y.s.t.e...rm.
0030	20 00 56 00 6F 00 6C 00 75 00 00 00 6D 00 65 00	.V.o.l.u...m.e.
0040	53 59 53 54 45 4D 7E 31 20 20 20 16 00 14 AD 61	SYSTEM~1a
0050	43 58 43 58 00 00 AE 61 43 58 03 00 00 00 00 00	CXCX...eacX....
0060	E5 6D 00 65 00 6E 00 74 00 2E 00 0F 00 9F 74 00	âme...nt...t...
0070	78 00 74 00 00 00 FF FF FF FF 00 00 FF FF FF FF	x.t...ÿÿÿÿ...ÿÿÿÿ
0080	E5 4E 00 65 00 77 00 20 00 54 00 0F 00 9F 65 00	âN...ew...T...e...
0090	78 00 74 00 20 00 44 00 6F 00 00 00 63 00 75 00	x.t...D...o...c...u...
00a0	E5 45 57 54 45 58 7E 31 54 58 54 20 00 AD D0 8D	âEWTEX~1TXT ~D...
00b0	43 58 43 58 00 00 D1 8D 43 58 00 00 00 00 00 00	CXCX...N...CX....
00c0	41 72 00 69 00 74 00 2E 00 74 00 0F 00 93 78 00	Ar...i...t...t...x...
00d0	74 00 2E 00 74 00 78 00 74 00 00 00 00 00 00 00	t...t...x...t...ÿÿ
00e0	52 49 54 54 58 54 7E 31 54 58 54 20 00 AD D0 8D	RITITXT~1TXT ~D...
00f0	43 58 43 58 00 00 E6 8D 43 58 06 00 2C 00 00 00	CXCX...e...CX....
0100	E5 66 00 00 00 FF FF FF FF FF FF 0F 00 80 FF FF	âf...ÿÿÿÿÿÿ...ÿÿ
0110	FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	ÿÿÿÿÿÿÿÿÿÿÿÿ...ÿÿÿÿ
0120	E5 52 00 69 00 74 00 54 00 69 00 0F 00 80 67 00	âR...i...t...i...g...
0130	65 00 72 00 73 00 2E 00 6A 00 00 00 66 00 69 00	e...r...s...j...f...i...
0140	E5 49 54 54 49 47 7E 31 4A 46 49 20 00 4E 0D 8E	âITTIG~1JFI ~N...
0150	43 58 43 58 00 00 0E 8E 43 58 00 00 00 00 00 00	CXCX...CX....
0160	42 66 00 00 00 FF FF FF FF FF FF 0F 00 80 FF FF	Bf...ÿÿÿÿÿÿ...ÿÿ
0170	FF FF FF FF FF FF FF FF FF 00 00 FF FF FF FF	ÿÿÿÿÿÿÿÿÿÿÿÿ...ÿÿÿÿ
0180	01 52 00 69 00 74 00 54 00 69 00 0F 00 80 67 00	.R...i...t...i...g...
0190	65 00 72 00 73 00 2E 00 6A 00 00 00 66 00 69 00	e...r...s...j...f...i...
01a0	52 49 54 54 49 47 7E 31 4A 46 49 20 00 4E 0D 8E	RITITIG~1JFI ~N...
01b0	43 58 43 58 00 00 FD 8D 43 58 07 00 09 7A 00 00	CXCX...ÿ...CX...z...
01c0	4D 59 56 49 44 45 4F 20 4D 50 34 20 18 97 61 90	MYVIDEO MP4 ~a...
01d0	43 58 43 58 00 00 68 90 43 58 0B 00 EA 64 7C 00	CXCX...h...CX...êd .
01e0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
01f0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0200	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0210	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0220	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0230	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0240	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0250	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0260	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0270	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0280	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

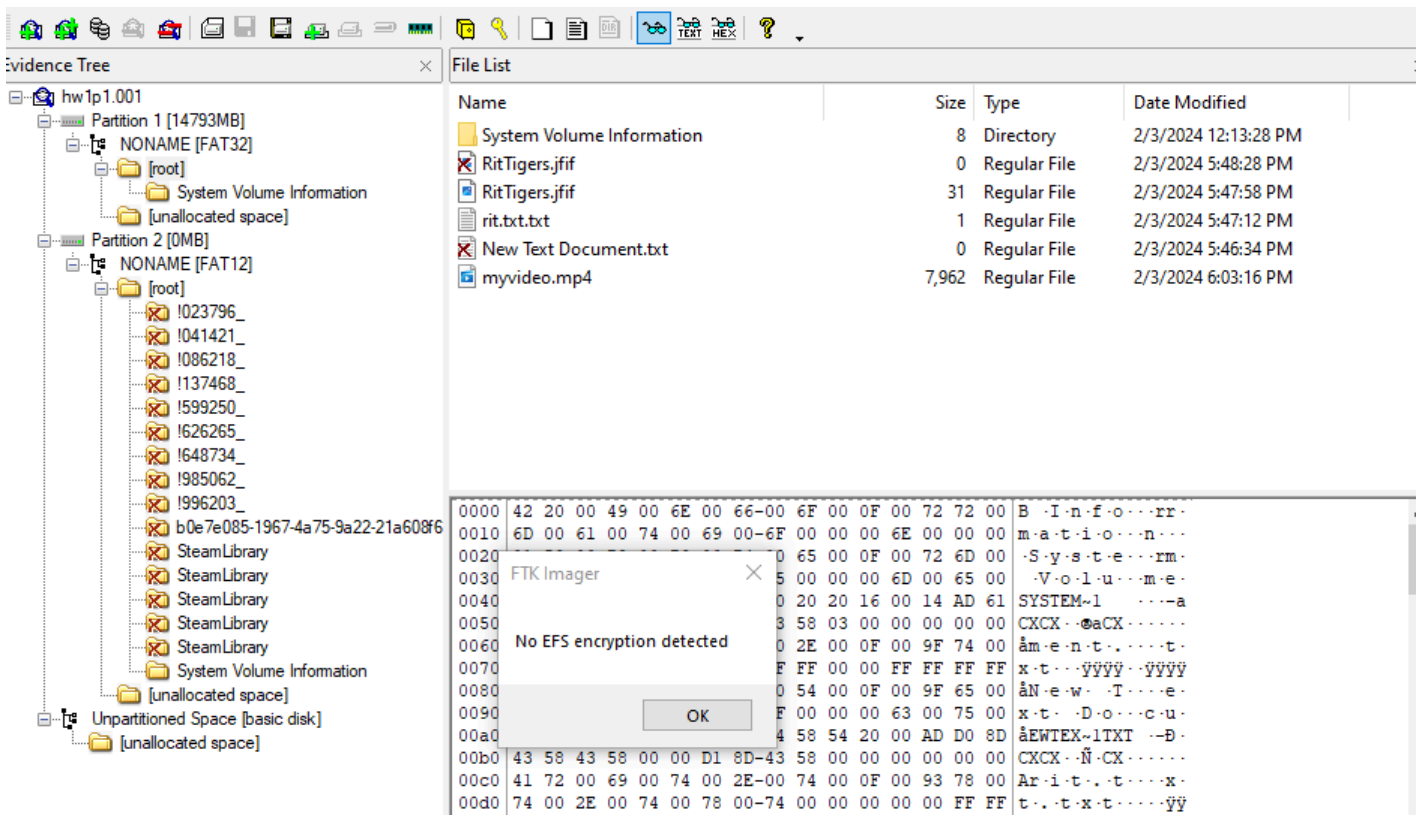
Cursor pos = 0; dus = 2; log sec = 32768; phy sec = 32832

For User Guide, press F1

Task 3: One of the FTK Imager features is Detecting EFS Encryption. Basically, you can check for encrypted data on a physical drive or an image with the FTK Imager. When using this feature, the program scans the evidence and notifies if encrypted files are located.



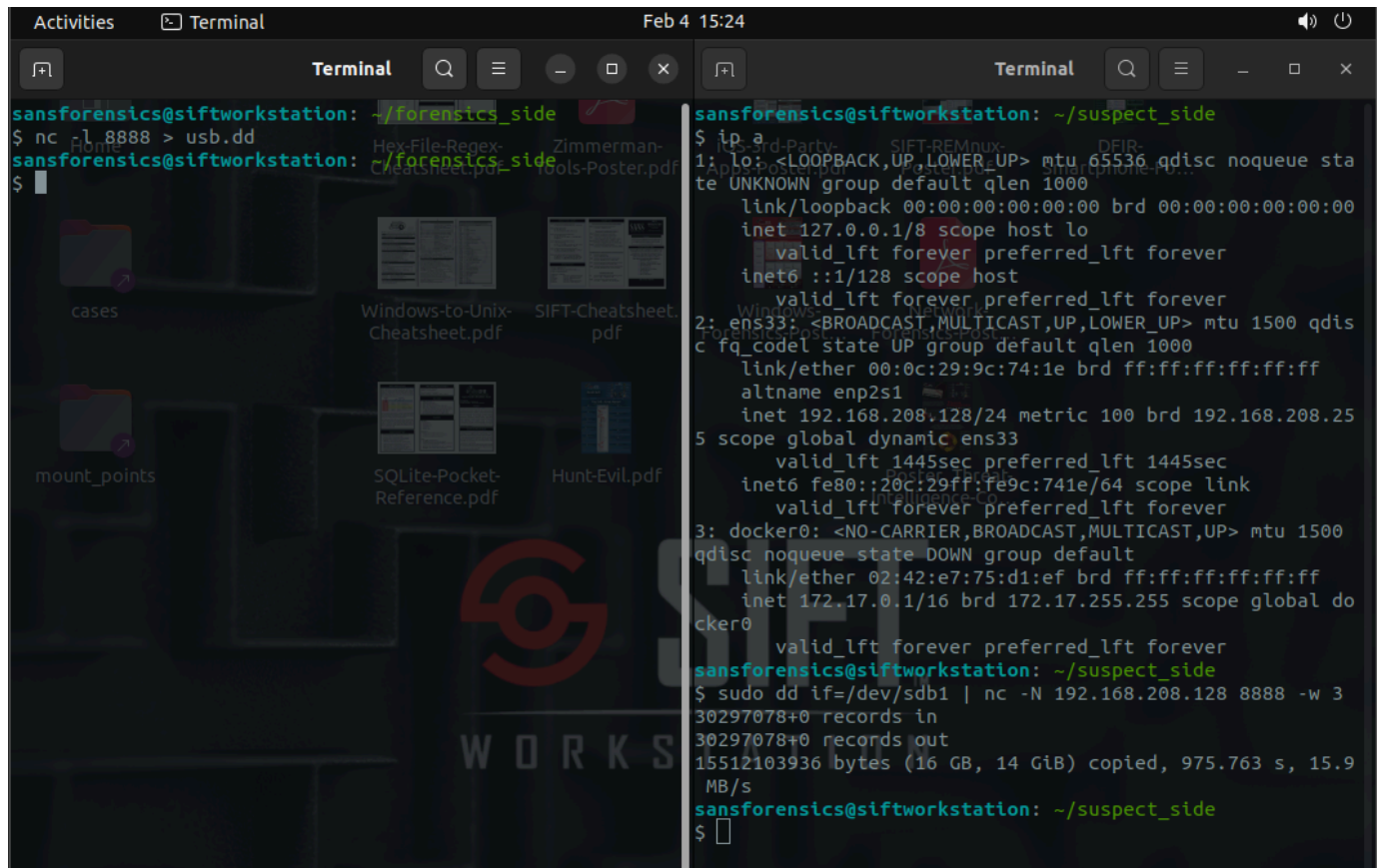
However, for me there was no EFS encryption detected.



Part 2. Imaging with dd and netcat (nc)

Task 4:

1. I used the command `nc -l 8888 > usb.dd` to save receive data over the network to usb.dd
2. I used the command `sudo dd if=/dev/sdb1 | nc -N 192.168.208.128 8888 -w 3` to send the USB image to the forensics machine terminal.

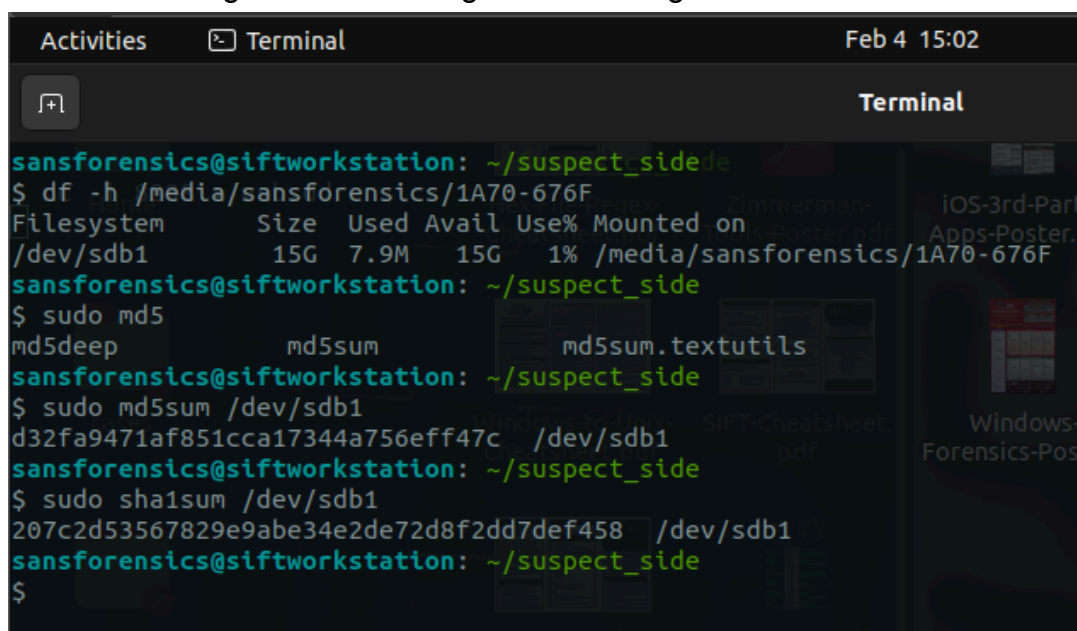


The screenshot shows two terminal windows. The left window, titled 'Terminal', shows the user `sansforensics@siftworkstation` in the `~/forensics_side` directory. It runs the command `nc -l 8888 > usb.dd`. The right window, also titled 'Terminal', shows the user `sansforensics@siftworkstation` in the `~/suspect_side` directory. It runs the command `ip a`, which displays network interface details for `lo` and `ens33`. The `ens33` interface is configured with IP `192.168.208.25`. Then, it runs the command `sudo dd if=/dev/sdb1 | nc -N 192.168.208.128 8888 -w 3`, which successfully transfers the USB image to the forensics machine, reporting `15512103936 bytes (16 GB, 14 GiB) copied, 975.763 s, 15.9 MB/s`.

```
sansforensics@siftworkstation: ~/forensics_side
$ nc -l 8888 > usb.dd
sansforensics@siftworkstation: ~/forensics_side
$

sansforensics@siftworkstation: ~/suspect_side
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: ens33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 00:0c:29:9c:74:1e brd ff:ff:ff:ff:ff:ff
    altname enp2s1
    inet 192.168.208.25/24 metric 100 brd 192.168.208.255 scope global dynamic ens33
        valid_lft 1445sec preferred_lft 1445sec
    inet6 fe80::20c:29ff:fe9c:741e/64 scope link
        valid_lft forever preferred_lft forever
3: docker0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc noqueue state DOWN group default
    link/ether 02:42:e7:75:d1:ef brd ff:ff:ff:ff:ff:ff
    inet 172.17.0.1/16 brd 172.17.255.255 scope global docker0
        valid_lft forever preferred_lft forever
sansforensics@siftworkstation: ~/suspect_side
$ sudo dd if=/dev/sdb1 | nc -N 192.168.208.128 8888 -w 3
30297078+0 records in
30297078+0 records out
15512103936 bytes (16 GB, 14 GiB) copied, 975.763 s, 15.9 MB/s
sansforensics@siftworkstation: ~/suspect_side
$
```

3. I used `sudo md5sum /dev/sdb1`, and `sudo sha1sum /dev/sdb1` to calculate the md5 and sha1 hash of the image before sending the USB image to the forensics machine.



The screenshot shows a terminal window titled 'Terminal' with the user `sansforensics@siftworkstation` in the `~/suspect_side` directory. It first runs `df -h /media/sansforensics/1A70-676F`, showing that `/dev/sdb1` has a size of 15G and is 1% full. Then, it runs `sudo md5` to install the md5sum utility. Finally, it runs `sudo md5sum /dev/sdb1`, resulting in the hash `d32fa9471af851cca17344a756eff47c`, and `sudo sha1sum /dev/sdb1`, resulting in the hash `207c2d53567829e9abe34e2de72d8f2dd7def458`.

```
sansforensics@siftworkstation: ~/suspect_side
$ df -h /media/sansforensics/1A70-676F
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdb1       15G   7.9M   15G   1% /media/sansforensics/1A70-676F
sansforensics@siftworkstation: ~/suspect_side
$ sudo md5
md5deep          md5sum          md5sum.textutils
sansforensics@siftworkstation: ~/suspect_side
$ sudo md5sum /dev/sdb1
d32fa9471af851cca17344a756eff47c /dev/sdb1
sansforensics@siftworkstation: ~/suspect_side
$ sudo sha1sum /dev/sdb1
207c2d53567829e9abe34e2de72d8f2dd7def458 /dev/sdb1
sansforensics@siftworkstation: ~/suspect_side
$
```

4. I used `sudo md5sum ./usb.dd`, and `sudo sha1sum ./usb.dd` commands to compute the md5 and sha1 hash of the usb.dd image.

```
sansforensics@siftworkstation: ~/forensics_side
$ nc -l 8888 > usb.dd
sansforensics@siftworkstation: ~/forensics_side
$ ls
usb.dd
sansforensics@siftworkstation: ~/forensics_side
$ sudo md5sum ./usb.dd
d32fa9471af851cca17344a756eff47c ./usb.dd
sansforensics@siftworkstation: ~/forensics_side
$ sudo sha1sum ./usb.dd
207c2d53567829e9abe34e2de72d8f2dd7def458 ./usb.dd
sansforensics@siftworkstation: ~/forensics_side
$
```

5. The hash value of usb.dd in linux by dd command is different from the hash values of the raw image created by FTK imager in part 1. The reasons could be the the metadata, FTK Imager includes metadata about the image in its output, which is not included in a raw dd image. This additional data can cause a difference in the hash values. Also, FTK Imager and dd might acquire the data differently. For example, FTK Imager might be doing a physical drive capture, while dd might be capturing only the logical drive. This difference in acquisition methods can lead to different hash values.

The screenshot shows a terminal window on the left with the same commands and output as in the previous block. On the right, a 'Drive/Image Verify Results' dialog box is open, displaying verification data for a file named 'hw1p1.001'.

Drive/Image Verify Results	
Name	hw1p1.001
Sector count	30297216
MD5 Hash	
Computed hash	2df5e24f04cdf95370ebf953b6e8efed
Report Hash	2df5e24f04cdf95370ebf953b6e8efed
Verify result	Match
SHA1 Hash	
Computed hash	2817aadd71f2464ef12dc4a9926c85225325a
Report Hash	2817aadd71f2464ef12dc4a9926c85225325a
Verify result	Match
Bad Blocks List	
Bad block(s) in image	No bad blocks found in image

The dialog box has a 'Close' button at the bottom right. The background of the terminal window shows various file icons and a large red 'S' logo.

Part 3. Linux memory acquisition using LiME

Task 5: Screenshot of lsmod | grep lime

```
sansforensics@siftworkstation: ~/lime/src
$ lsmod | grep lime
lime                16384  0
sansforensics@siftworkstation: ~/lime/src
$
```

Task 6: Two strings commands that was used

This is commands has been used to get the hashed version of the password for sansforensics.

`sudo strings -f -d ../Desktop/mh8872_memory_dump.bin > strings_output_1`

and

`cat strings_output_1 | grep password`

```
../Desktop/mh8872_memory_dump.bin: gdm-smart-card-sssd-01-password
../Desktop/mh8872_memory_dump.bin: hc_DES_read_password
../Desktop/mh8872_memory_dump.bin: password prompts for encrypted file systems.
../Desktop/mh8872_memory_dump.bin: password prompts for encrypted file systems.
../Desktop/mh8872_memory_dump.bin: Description: run common desktop actions without password
../Desktop/mh8872_memory_dump.bin: to run common actions without being asked for their password:
../Desktop/mh8872_memory_dump.bin: /etc/java-8-openjdk/management/jmxremote.password 7b46c291e7073c31d3ce0adae2f755
4f
../Desktop/mh8872_memory_dump.bin: password-manager-service:
../Desktop/mh8872_memory_dump.bin: dialog-password-symbolic
../Desktop/mh8872_memory_dump.bin: Description: runtime support for password checker library cracklib2
../Desktop/mh8872_memory_dump.bin: including programs to build the password dictionary databases used by
../Desktop/mh8872_memory_dump.bin: {"userName":"sansforensics","uid":1000,"gid":1000,"homeDirectory":"/home/sansfore
nsics","shell":"/bin/bash","privileged":{"hashedPassword":["$2a$10$xeAdkxMLN5dzg4l55bGKfuRy0TegEBPBzxlWq4edAWgl7rx0
znxi"]},"passwordChangeNow":false,"lastPasswordChangeUsec":1681862400000000,"passwordChangeMaxUsec":8639913600000000
,"passwordChangeWarnUsec":604800000000}
../Desktop/mh8872_memory_dump.bin: mount-op-ask-password
```

This commands was used to see and find what media was attached to the machine which is my USB.

`sudo strings -n 8 ../Desktop/mh8872_memory_dump.bin > strings_output_2`

and

`Cat strings_output_2 | grep GET`

```
'RGET=/media/sansforensics/1A70-676F ROOT=/ OPTS=uhelper=udisks2
NAUTILUS_VIEW_GET_IFACE (view)->get_view_id
NAUTILUS_VIEW_GET_IFACE (view)->set_templates_menu
```

Task 7:

Command used: `foremost -d -i ../Desktop/mh8872_memory_dump.bin -o ./foremost_output/`

```
132: 06582884.exe 781 KB 3370430736
133: 06769348.exe 781 KB 3465906304
134: 06807300.exe 781 KB 3485337728
135: 06837364.exe 781 KB 3500730496
136: 06927724.exe 781 KB 3546994816
137: 07787316.doc 11 MB 3987106064
138: 07824340.doc 11 MB 4006062240
Finish: Sun Feb  4 21:13:09 2024

139 FILES EXTRACTED

doc:= 9
exe:= 130
-----

Foremost finished at Sun Feb  4 21:13:09 2024
sansforensics@siftworkstation: ~/Documents/foremost_output
$ ls
audit.txt  doc  exe
sansforensics@siftworkstation: ~/Documents/foremost_output
$
```

With the command I found a lot of doc files and exe files. Some of the exe files seems to be system related files, and some of the docs files seems to be related to some of forensics pdf files in the linux.