Experiment- 1

Evidence Collection

a) Linux: Capturing RAM dump using fmem https://github.com/NateBrune/fmem
· dcfldd if=/dev/fmem of=memory.dump hash=sha256 sha256log=memory.dump.sha256 bs=1MB count=1000
b) Linux: Capturing Disk using dfldd https://www.obsidianforensics.com/blog/imaging-using-dcfldd
· dcfldd if=/dev/sdb1 of=/media/disk/test_image.dd hash=md5, sha1hashlog=/media/disk/hashlog.txt
c) Windows: Capture RAM dump of a windows system a. Hint: FTK Imager or RAMCapture
d) Windows: Capture Disk Image of a windows system Hint: FTK Imager

Aim

To collect volatile and non-volatile digital evidence from Linux and Windows systems using forensic tools while maintaining data integrity using hash values.

Tools Required

- Linux system (Kali / Ubuntu)
- Windows system
- dcflld
- fmem
- FTK Imager
- External storage device (USB)

What is Evidence Collection?

Evidence collection means **copying data (RAM or Disk)** from a computer **without altering the original data**, so it can be analyzed later in a forensic investigation.

There are **two types of evidence**:

1. **Volatile evidence** → RAM (lost if system is turned off)
2. **Non-volatile evidence** → Hard disk (permanent storage)

PART (a): Linux – Capturing RAM Dump using fmem

What is RAM Dump?

A **RAM dump** is a complete copy of the system's **main memory**, which may contain:

- Running processes
- Passwords
- Encryption keys
- Network connections

Tools Used

- fmem → Access physical memory
- dcflld → Forensic copying tool (advanced dd)

Step 1: Install Required Packages

sudo apt update

sudo apt install git build-essential dcflld -y

PROCESS

- Downloads Git, compiler, and forensic tool
- Required to build and run fmem

Step 2: Download and Load fmem Kernel Module

git clone https://github.com/NateBrune/fmem.git

cd fmem

make

sudo insmod fmem.ko

PROCESS

- Downloads memory access module
- Compiles kernel module
- Loads it into Linux kernel

OUTPUT

/dev/fmem

✓ Means RAM device is available

Step 3: Capture RAM Dump

INPUT

sudo dcflld if=/dev/fmem of=memory.dump \

hash=sha256 sha256log=memory.dump.sha256 \

bs=1M count=1000

COMMAND EXPLANATION

Parameter	Meaning
if=/dev/fmem	Input = RAM
of=memory.dump	Output file
hash=sha256	Generate SHA-256 hash
sha256log	Save hash to file
bs=1M	Read 1MB at a time
count=1000	Capture 1000MB (example)

OUTPUT (Example)

1000 blocks copied

SHA256: 9a7c...f21b

Files created:

memory.dump

memory.dump.sha256

Step 4: Verify Hash

INPUT

cat memory.dump.sha256

OUTPUT

9a7c3b91d8e4c1a1f0... memory.dump

✓ Confirms data integrity

PART (b): Linux – Capturing Disk Image using dcflld

What is Disk Imaging?

Disk imaging means **creating an exact bit-by-bit copy** of a storage device.

Step 1: Identify Disk

INPUT

lsblk

OUTPUT

NAME SIZE TYPE

sda 100G disk

└─sda1 100G part

sdb 16G disk

└─sdb1 16G part

➞ Evidence disk = /dev/sdb1

Step 2: Capture Disk Image

INPUT

sudo dcflld if=/dev/sdb1 of=/media/usb/test_image.dd \

hash=md5,sha1 sha1hashlog=/media/usb/hashlog.txt

PROCESS

- Reads disk sector by sector
- Saves forensic image
- Generates hash values

OUTPUT

MD5: 3f1a9e...7b2
SHA1: 9d3a...e21
Files created:
test_image.dd
hashlog.txt

View Hash Log

INPUT

cat /media/usb/hashlog.txt

OUTPUT

SHA1 (test_image.dd) = 9d3a...e21

PART (c): Windows – Capture RAM Dump

Tool: FTK Imager

Steps with Input & Output

INPUT (User Action)

- Run **FTK Imager as Administrator**
- File → Capture Memory
- Select destination (USB)

PROCESS

- Reads physical RAM
- Saves memory image

OUTPUT

memory.mem
memory.mem.md5
Example hash:
MD5: 8c7a4f9e12...

PART (d): Windows – Capture Disk Image

Using FTK Imager

INPUT

- Create Disk Image
- Physical Drive
- Select Drive 0
- Image Type: Raw (dd)
- Enable hashing

PROCESS

- Bit-by-bit disk copy
- Hash verification

OUTPUT

disk_image.dd

disk_image.dd.md5

disk_image.dd.sha1

IMPORTANT FORENSIC CONCEPTS

Hashing Example

Original File Hash = A1B2C3

Copied File Hash = A1B2C3

✓ Integrity maintained

Chain of Custody (Example)

Date	Evidence	Collected By	Hash
10-10-25	RAM Dump	Student	SHA256