Experiment Title

Evidence Collection from Linux and Windows Systems

Aim

To collect **volatile (RAM)** and **non-volatile (Disk)** digital evidence from **Linux and Windows systems** using forensic tools while maintaining data integrity using cryptographic hash values.

Tools & Software Used

- Linux OS
- Windows OS
- **fmem** (Linux RAM acquisition)
- **dcfldd** (Disk & memory imaging)
- **FTK Imager**
- **RAMCapture**
- External USB storage device

Theory

Evidence collection is the first and most critical step in **Digital Forensics**.
Digital evidence can be:

- **Volatile data** – RAM contents (lost when system is powered off)
- **Non-volatile data** – Hard disk contents

To preserve evidence integrity, **hash values (MD5, SHA1, SHA256)** are calculated during acquisition. Any change in data will change the hash, proving tampering.

Procedure

A) Linux: Capturing RAM Dump using fmem

1. Download fmem tool:

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

cd fmem

make

2. Load kernel module:

sudo insmod fmem.ko

3. Capture RAM dump:

sudo dcfldd if=/dev/fmem of=memory.dump hash=sha256 sha256log=memory.dump.sha256 bs=1MB count=1000

4. Unload module:

sudo rmmod fmem

B) Linux: Capturing Disk Image using dcfldd

1. Identify disk:

lsblk

2. Capture disk image:

sudo dcfldd if=/dev/sdb1 of=/media/disk/test_image.dd hash=md5,sha1 hashlog=/media/disk/hashlog.txt

C) Windows: Capturing RAM Dump

Using FTK Imager

1. Open FTK Imager as Administrator
2. File → Capture Memory
3. Select destination (USB drive)
4. Enable "Include Pagefile"
5. Click Capture Memory

D) Windows: Capturing Disk Image

Using FTK Imager

1. Open FTK Imager
2. File → Create Disk Image
3. Select Physical Drive
4. Choose image type (E01 / Raw)

5. Select destination
6. Enable Verify Image
7. Start imaging

Observation

- RAM dump files were successfully created.
- Disk images were acquired without modifying the source.
- Hash values were generated and stored separately.
- Evidence integrity was maintained throughout the process.

Result

Thus, volatile and non-volatile evidence was successfully collected from Linux and Windows systems using standard forensic tools while ensuring integrity using cryptographic hash functions.

Precautions

1. Always capture RAM before disk imaging.
2. Use write-protected external storage.
3. Verify hash values after acquisition.
4. Avoid working directly on original evidence.

Viva Voce Questions

1. What is volatile evidence?
2. Why hashing is required?
3. Difference between MD5 and SHA256?
4. Why RAM is captured first?
5. What is forensic disk imaging?

Sample Inputs and Outputs

A) Linux: RAM Acquisition using fmem

Sample Input

sudo insmod fmem.ko

sudo dcfldd if=/dev/fmem of=memory.dump hash=sha256 sha256log=memory.dump.sha256 bs=1MB count=1000

Sample Output

1000 blocks (1000 MB) written.

Hash (sha256):

9f3a8c5d7a4e1c6b8d0f2a1c4e6b9d7a8c5e4f3a2b1c9d8e7f6a5b4c3d2e1

Generated Files

memory.dump

memory.dump.sha256

B) Linux: Disk Imaging using dcfldd

Sample Input

sudo dcfldd if=/dev/sdb1 of=/media/disk/test_image.dd hash=md5,sha1 hashlog=/media/disk/hashlog.txt

Sample Output

102400 blocks written.

MD5 Hash:

d41d8cd98f00b204e9800998ecf8427e

SHA1 Hash:

da39a3ee5e6b4b0d3255bfef95601890afd80709

Generated Files

test_image.dd

hashlog.txt

C) Windows: RAM Capture (FTK Imager)

Sample Input (User Action)

- Tool: FTK Imager
- Option Selected: Capture Memory
- Destination: E:\Evidence\

Sample Output

Memory capture completed successfully.

Output File: memory.mem

Generated Files

memory.mem

pagefile.sys (optional)

D) Windows: Disk Image Capture (FTK Imager)

Sample Input (User Action)

- Source: PhysicalDrive0
- Image Type: E01
- Destination: E:\Evidence\

Sample Output

Image creation completed successfully.

Verification successful.

MD5 Hash: 3b5d5c3712955042212316173ccf37be

Generated Files

disk_image.E01

disk_image.E01.txt (hash & metadata)

Hash Verification (Optional – Linux)

Sample Input

sha256sum memory.dump

Sample Output

9f3a8c5d7a4e1c6b8d0f2a1c4e6b9d7a8c5e4f3a2b1c9d8e7f6a5b4c3d2e1 memory.dump