Lab Practical: File Carving and Recovery Using pyFileCarving

Objective:

To understand the concept of file carving and perform data recovery from a raw disk image using the open-source pyFileCarving tool.

Background:

When files are deleted from a file system, their metadata is removed — but the file content remains on disk until overwritten. File carving is a forensic technique that recovers files directly from raw data by identifying unique signatures (headers and footers).

Common File Signatures:

File Type	Header (Hex)	Footer (Hex)
JPEG	FF D8	FF D9
PDF	25 50 44 46	25 25 45 4F 46
PNG	89 50 4E 47 0D 0A 1A 0A	49 45 4E 44 AE 42 60 82
ZIP	50 4B 03 04	50 4B 05 06

Tools Required:

1. Operating System: Windows, Linux, or macOS
2. Python 3.8+
3. Git
4. pyFileCarving
5. Sample disk image (.dd or .E01)

Procedure:

1. Install pyFileCarving using pip: pip install pyFileCarving
2. Clone from GitHub if needed: git clone https://github.com/wahlflo/pyFileCarving.git
3. Download a sample image (e.g., from DigitalCorpora).
4. Run the tool: pyFileCarving -i sample.dd -o carved_output
5. Inspect carved files and verify recovered data.

Discussion Questions:

1. How does file carving differ from normal file recovery?
2. Why can deleted files often be recovered?
3. What are the limitations when files are fragmented?
4. Which file types are easiest to carve and why?
5. How can forensic investigators validate carved files?

Expected Output:

A folder containing recovered files (e.g., .jpg, .pdf) and a student report including tools used, commands executed, screenshots, and answers to discussion questions.

Learning Outcome:

Students will understand how file signatures help recover data, learn about unallocated space, practice using forensic tools, and understand limitations of file carving on fragmented or encrypted files.