# Digital Forensics Lab Notes. File Systems, Metadata, File Carving, and Hash Integrity

## Overview

This lab focused on four core forensics skills that show up constantly in investigations. Identify file systems from disk images, extract metadata, carve deleted files, and validate integrity with cryptographic hashes. The goal was a clean, repeatable process where every result is verifiable.

## Analyst summary

Classification: Forensics triage and evidence handling exercise
Priority: Medium
Confidence: High

### Core outputs

File systems identified across three images
Metadata extracted from a PDF and a JPG
One deleted image recovered via file carving and validated with an MD5 hash
Hashing workflow demonstrated on Linux and Windows PowerShell

## Tools and platforms used

FTK Imager
ExifTool
Scalpel
md5sum, sha1sum, sha256sum
PowerShell Get FileHash

## Part 1. Identify file systems with FTK Imager

### Goal

Determine the file system used by each provided image.

### Method

I loaded each image as an evidence item in FTK Imager, expanded the evidence tree, then checked the partition and file system labeling shown by FTK.

### Findings

carve1.img. NTFS
carve2.img. FAT32
disk1.img. EXT3

### Why it matters

File system type determines where artifacts live and how you approach recovery. NTFS and FAT32 change how you interpret timestamps and deleted file behavior, EXT3 shifts the investigation to Linux artifacts and structures.

Part 2. Metadata analysis with ExifTool

Goal

Extract the author from a PDF, and the camera model from a JPG.

Working directory steps used

cd ~/Desktop

cd "Metadata and File Carving"

PDF metadata, author

Command used

exiftool dummy.pdf | grep -i author

Result

Author. Evangelos Vlachogiannis

Image metadata, camera model

Command used

exiftool picture.jpg | grep -i "camera model|model"

Result

Camera model. iPhone 6

Why it matters

Metadata can confirm ownership, tools used to create or edit a file, device origin, and timeline consistency. It is one of the fastest ways to validate or challenge a story.

Part 3. File carving a deleted image with Scalpel

Goal

Carve a disk image, recover a deleted image, and report the MD5 hash of the recovered file.

Method

I enabled only JPG and PNG in the Scalpel configuration to keep the carve focused, ran Scalpel into a clean output directory, then hashed the recovered file.

Commands used

sudo nano /etc/scalpel/scalpel.conf

rm -rf scalpel_out

mkdir scalpel_out

sudo scalpel -o scalpel_out carve1.img

sudo find scalpel_out -type f -exec md5sum {} ;

Result

Recovered image MD5

2fcab62f58b320f16032914b89fe96a1

Why it matters
Carving recovers data that is not visible through normal file browsing. Hashing proves the carved file you recovered is the same file you analyzed and reported on.

Part 4. Hashing and integrity validation
Goal
Generate hashes for strings and files, and understand how to do it on Linux and PowerShell.

Linux patterns used
Hashing a string, using echo -n to avoid adding a newline
echo -n "text here" | md5sum
echo -n "text here" | sha256sum

Hashing a file
md5sum hashthis.jpg
sha1sum hashthis.jpg
sha256sum hashthis.jpg

PowerShell pattern used
Get-FileHash .\hashthis.jpg -Algorithm SHA1

Why it matters
Hashing is the foundation of integrity. It supports chain of custody, repeatability, and confidence that evidence has not been altered.

Key takeaways
Identify the file system first, it shapes the investigation.
Pull metadata early, it can confirm origin and timeline.
Carve when deletion or hidden data is suspected.
Hash anything you plan to report, integrity is what makes evidence defensible.

# LAB COMPLETED

## AHMED ATTIA

has successfully completed

**DATA ACQUISITION**

Use FTK Imager, ProcDump, and KAPE to acquire data for later analysis.