Lab 6: Covert Channels & Metadata Analysis Using Python

Course: FSCT 8561 – Security Applications
Instructor: Dr. Maryam R. Aliabadi
Lab Duration: 3 Hours

Overview

Images, documents, and files often carry metadata—information about how, when, and where they were created—that can leak privacy-sensitive information or act as covert channels. In this lab, you will use Python to extract, analyze, and detect anomalies in image metadata, explore hidden messages embedded in EXIF fields, and simulate forensic anti-tampering techniques to detect image manipulation. You will also encounter a multi-step secret hidden across multiple images, encouraging collaboration, investigation, and problem-solving.

Learning Objectives

- Extract and interpret EXIF metadata from images using Python libraries.
- Detect potential **covert channels** or hidden messages in metadata.
- Compare **EXIF timestamps with filesystem MAC times** to identify tampering.
- Identify traces of **image editing** using compression and quantization artifacts.
- Develop a Python-based script to **automate metadata analysis** and assign risk scores.

Pre-requisite

- Completed Labs 1–5
- Understanding of EXIF metadata structure and file system timestamps
- Familiarity with image editing concepts and JPEG compression

Required Reading & Tutorials

- Mastering Python for Networking and Security – Chapter 13
    - https://learning.oreilly.com/library/view/mastering-python-for/9781839217166/
    - Source code on: https://github.com/PacktPublishing/Mastering-Python-for-Networking-and-Security-Second-Edition
- Python EXIFRead Documentation: https://pypi.org/project/ExifRead/
- Pillow (Python Imaging Library) Tutorial: https://pillow.readthedocs.io/en/stable/

Lab Scenario

You are a forensic analyst investigating suspicious images suspected of containing hidden messages and covert channels. Your task is to extract metadata, analyze EXIF fields for embedded secrets, identify privacy leaks such as GPS data and timestamps, detect signs of image tampering, and reconstruct any multi-step secret distributed across multiple images.

Environment Setup

1. Install required libraries:

```
pip install exifread pillow
```

3. Download **sample images** provided for Lab 6.

Part 1 – Metadata Extraction

1. Load each image using ExifRead or Pillow.
2. Extract the following fields:
    a. GPSLatitude / GPSLongitude
    b. DateTimeOriginal, CreateDate, ModifyDate
    c. Camera make/model
    d. Software used for editing
    e. UserComment / ImageDescription
3. Record the metadata in a structured table.

Part 2 – Detecting Covert Channels

In this part, your goal is to identify potential hidden messages in the images' metadata and reconstruct the multi-step secret.

1. Check unusual metadata fields
    - Examine fields such as UserComment, ImageDescription, MakerNote, Software, and Copyright.
    - Look for values that seem out of place or unusually long.
2. Decode encoded content (if any)
    - Some hidden messages may be encoded in formats like Base64 rather than stored as plain text.
    - Use Python's base64 library or similar tools to decode these fields.
    - This simulates real-world scenarios where attackers obfuscate information to avoid detection.
3. Note anomalies that may indicate intentional covert messaging.
    - Each image contains one step of a multi-step secret. Extract the partial secret from each image and note it in your table.
4. Reconstruct the full secret
    - Once all partial secrets are extracted, combine them in the correct order to reveal the full multi-step message.

Part 3 – Consistency Checks

The goal of this part is to identify tampering via timestamp comparison.

1. Compare EXIF timestamps with file system MAC times (Created, Modified).
2. Look for inconsistencies or impossible sequences.

Expected Observations:

- Images edited post-capture may show mismatched EXIF vs filesystem times.
- Timestamp anomalies indicate potential tampering.

Part 4 – Quantization / Editing Detection

1. Examine images for double JPEG compression artifacts.
2. Look for inconsistencies in DCT coefficient patterns or block artifacts.
3. Record any detected editing traces.

Expected Observations:

- Edited images show subtle quantization artifacts.
- Original images are consistent across compression blocks.

Part 5 – Automation & Risk Scoring

The goal of this part is to build a Python tool to automate metadata analysis.

- Write a script that iterates over all images.
- Extract and log all metadata fields.
- Flag suspicious fields (GPS, editing software, anomalous timestamps).
- Assign a **risk score** to each image based on detected anomalies. For each image, assign points based on these **risk categories**:

Risk Type	Condition	Suggested Points
Hidden Secret	Covert channel field (Software, UserComment, GPSDestDistance, MakerNote, ImageDescription, Copyright) contains the secret	10
GPS / Privacy Leak	GPSLatitude or GPSLongitude present	5
Timestamp Anomaly	EXIF timestamp ≠ filesystem MAC time OR impossible sequence	5
Editing / Compression	Software tag indicates editing, or double JPEG compression detected	5

Expected Output: Table or CSV with extracted metadata and risk scores, and identification of images containing secret steps.

Part 6 - Reflection Questions

1. How could GPS coordinates in images compromise privacy in real-world scenarios?
2. Why is comparing EXIF timestamps to filesystem MAC times effective in detecting tampering?
3. What makes detecting double JPEG compression nearly impossible to hide?
4. How could an organization prevent covert channels in images shared publicly?
5. How does automating metadata extraction improve forensic investigation efficiency?

Deliverables

Submit **one PDF file** containing:

- `metadata_scanner.py-` Python script for metadata extraction and risk scoring
- Tables of extracted metadata for all images
- Screenshots of images containing secrets and their extracted partial messages
- Reconstructed full secret
- Answers to reflection questions

Filename format: Lab**6**-FirstName-LastName-StudentNumber.pdf

Lab 6 Grading Rubric (Total: 100 points)

Component	**Full Credit**	**Partial Credit**	**Minimal/No Credit**
Metadata Extraction Script (20 points)	Extracts all fields correctly; logs clearly	Minor missing fields or logging issues	Script missing or incorrect
Covert Channel Detection (25 points)	Detects all hidden messages; identifies anomalies	Partial detection or misidentified fields	Missing or incorrect detection
Consistency & Editing Analysis (15 points)	Correctly identifies timestamp mismatches and compression artifacts	Partial identification	Missing or incorrect
Multi-Step Secret Reconstruction (20 points)	Correctly reconstructs full secret from all images	Partial reconstruction or incorrect order	Missing or incorrect
Reflection & Security Discussion (20 points)	Answers show understanding of privacy, tampering, covert channels	Partial discussion	Missing or incomplete

Good Luck!