Project Title: Breaking Bytes – A Forensic Adventure with Walter White and Jesse Pinkman

---

Overview

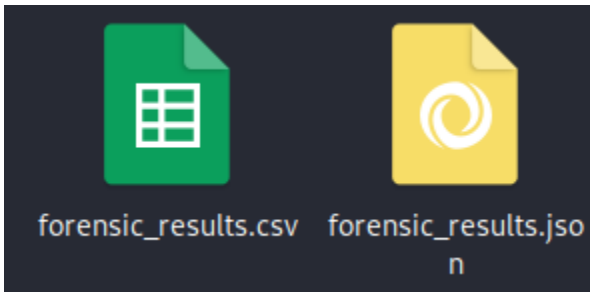
In this project, I bring a bit of *Breaking Bad* flair to the field of digital forensics. With my code duo, **Walter White** (the analytical mastermind) and **Jesse Pinkman** (the presentation genius), I've created a methodical approach to extract, analyze, and present file metadata with Heisenberg's precision.

Project Roles and Functionality

1. **Walter White** (walterwhite.py): The "master chemist" of the setup, Walter meticulously extracts metadata from images and PDFs. His data analysis skills match his chemistry expertise, and he's capable of retrieving everything from file creation and modification dates to detecting tampering in images using SSIM (Structural Similarity Index).
    - **Core Functions:**
        - **File Hashing:** Generates SHA-256 hash values to ensure data integrity.
        - **Image Metadata Extraction:** Uses exiftool to retrieve essential image metadata (like EXIF data and camera details).
        - **PDF Metadata Extraction:** Utilizes PyMuPDF and PyPDF2 to gather high-level information about PDFs.
        - **Tampering Detection for Images:** Implements **SSIM** with OpenCV to detect possible alterations in images.
    - **Batch Processing:** Walter doesn't work alone; I set up multiprocessing so he can handle file batches efficiently.
    - **Output:** Stores all forensic insights as both JSON (forensic_results.json) and CSV (forensic_results.csv), delivering organized data on a silver platter—Walter style.

```
┌──(mrbluesky69㉿kali)-[~/Desktop]
└─$ python3 walterwhite.py
Enter the file or directory path to process: /home/mrbluesky69/Desktop/IMG_20241026_115507.jpg
```

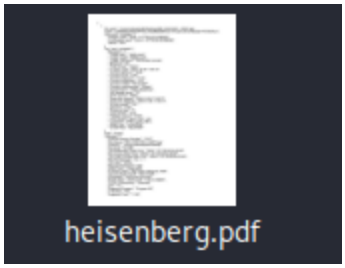
```
Results saved to forensic_results.json and forensic_results.csv
```



2. **Jesse Pinkman** (jessiepinkman.py): Once Walter has extracted the metadata, Jesse steps in with flair. He formats the data for readability and lets me decide if I want to store it as a PDF (aka "Heisenberg.pdf").
    - **Core Functions:**
        - **Data Display:** Formats JSON data with json.dumps() and outputs it in a structured, indented format.
        - **PDF Storage:** Optionally saves the JSON data as a PDF using ReportLab. If I say "Y," he'll turn metadata into a stylish PDF—fondly named **"Heisenberg.pdf"** during testing.

```
┌──(mrbluesky69㉿kali)-[~/Desktop]
└─$ python3 jessiepinkman.py
Enter the path of the JSON file: /home/mrbluesky69/Desktop/forensic_results.json
```

```
        "tampering_analysis": {
            "tampering_suspected": true,
            "ssim_scores": [
                0.17424496754877505,
                0.17149763583158004,
                0.20932297220821808,
                0.19451322269479976,
                0.25915232942325567,
                0.2493658796644092
            ]
        }
    }
]
Would you like to save this JSON as a PDF? (y/n): y
Enter output PDF filename (default is json_output.pdf): heisenberg.pdf
PDF saved as heisenberg.pdf
```



Purpose and Use Cases

In the field of forensics, metadata often holds the *hidden secrets* of digital artifacts. The "Breaking Bytes" project arms me with tools to:

- **Verify Data Authenticity:** With SHA-256 hashing, I can validate file integrity over time.
- **Uncover Tampering:** Image SSIM-based tampering analysis can reveal if photos have been modified.
- **Streamline Investigations:** Through batch processing, I can quickly process large datasets.
- **Generate Reports:** With optional PDF storage, the metadata results are primed for courtrooms or client presentations.

Importance in Cybersecurity

Digital forensics plays a critical role in cybersecurity by helping analysts detect data breaches, track unauthorized changes, and assess the authenticity of files. By extracting metadata and identifying tampering, this project aids in verifying the integrity of digital evidence, a crucial aspect in cybersecurity investigations. Tools like *Breaking Bytes* provide transparency and help in tracing the origins and history of files, enabling effective incident response and legal compliance.

Technical Libraries and Their Roles

- **Pillow (PIL):** For initial image handling and conversions.
- **PyMuPDF & PyPDF2:** Extract metadata from PDF files.
- **OpenCV & SSIM (scikit-image):** Detect tampering in image files using the Structural Similarity Index.
- **hachoir.metadata & hachoir.parser:** Accesses low-level metadata for forensic details.
- **ReportLab:** Converts JSON metadata into PDF format (the Jesse Pinkman finishing touch).

- **JSON & CSV Modules:** Serializes metadata output into structured formats.

Usage Flow (aka How to Cook the Perfect Batch)

1. **Run `walterwhite.py`:** Input the directory or file path, and Walter will extract metadata, detect any image tampering, and save everything to **forensic_results.json** and **forensic_results.csv**.
2. **Run `jessiepinkman.py`:** Jesse takes the JSON output, formats it for readability, and prompts:
    - Say **Y** to save as a PDF (default name: **Heisenberg.pdf**).
    - Say **N** to skip PDF storage but still view the readable JSON output.

For more details and to see the code in action, check out the project on my GitHub: [GitHub link].