Threat Intelligence Extraction Script

1. Overview

This script extracts threat intelligence information from various file formats, including PDF, DOCX, and TXT, using SpaCy's Named Entity Recognition (NER), regex patterns, and transformer-based models. It processes a single file or a folder containing multiple supported files, extracting details such as Indicators of Compromise (IoCs), Tactics, Techniques, and Procedures (TTPs), malware names, threat actors, and targeted entities. The extracted information is saved in JSON format for further analysis.

2. Requirements

2.1 Key Dependencies

The necessary Python packages are mentioned at the beginning of the script. Install them to ensure the script works correctly.

The following Python libraries are required:PyMuPDF (for PDF text extraction)

- docx2txt (for DOCX text extraction)
- re (for regex-based text extraction)
- json (for JSON file generation)
- pathlib (for path manipulation)
- requests (for API calls to VirusTotal)
- SpaCy (for Named Entity Recognition)
- transformers (for LLM-based NER and classification)
- werkzeug.utils (for secure filename handling)
- datetime (for timestamping output files)

2.2 Input Requirements

- **File Formats:** PDF, DOCX, and TXT files with OCR support (machine-readable text).
- **Avoid Content:** PDFs should be machine-readable and free from excessive non-text elements such as images, charts, or tables to ensure accurate text extraction and processing.
- **Input Types:**
  - Single supported file (PDF, DOCX, or TXT).
  - Folder containing multiple supported files.

2.3 Output

- The script generates JSON output files in the same directory as the input.
- Each JSON file corresponds to an input file and is named: output_<filename>.json.

3. How to Run the Script

3.1 Ensure all dependencies are installed:

- Install Python 3.7+ from Python's official site.
- Install the required libraries using pip:

Install all the dependencies listed in the file by running the following command in your terminal:

```
pip install -r requirements.txt
```

3.2 Prepare the Input Files:

- Ensure that all input files are machine-readable and free from non-text elements such as images, charts, or tables.
- Place multiple files in a single directory if processing multiple files.

3.3 Execute the Script via CLI

- Save the script as threat_extractor.py
- Open the terminal/command prompt and execute the script:

```
python threat_extractor.py
```

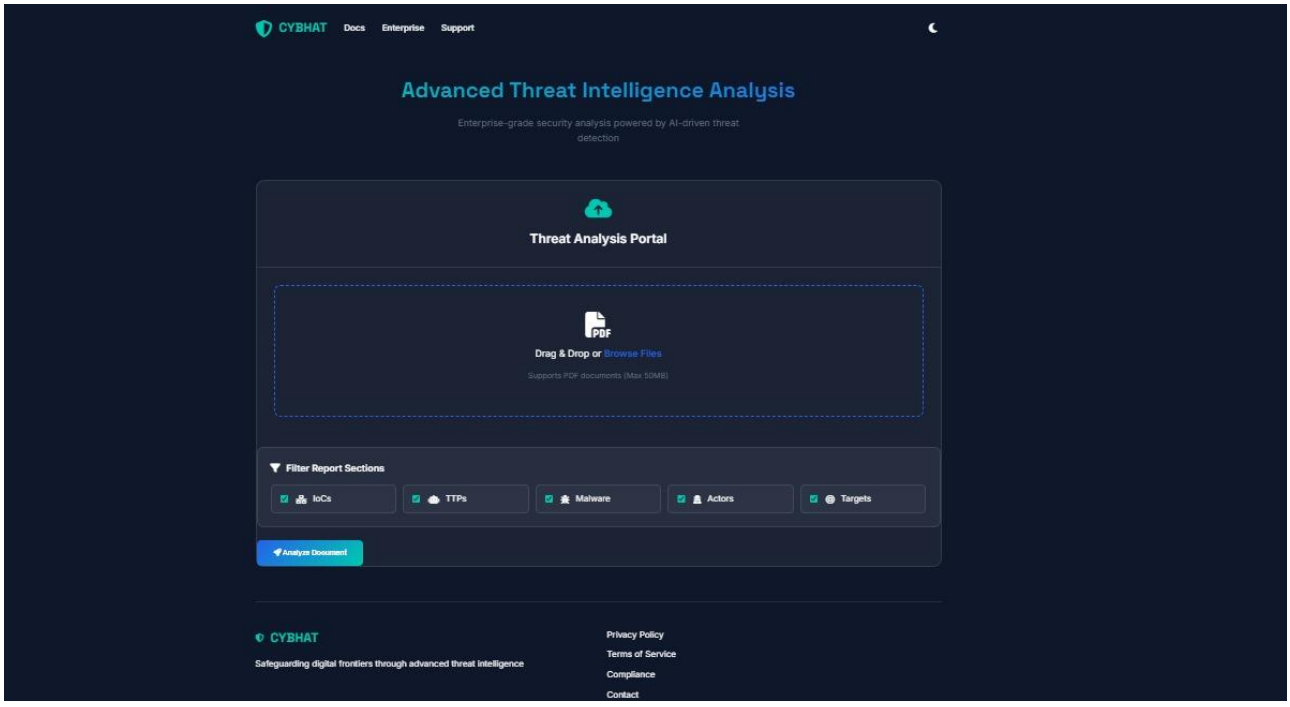
- Follow the prompts to enter a file or folder path.

3.4 Execute the Script via Web-Based Dashboard

Access the dashboard at provided link in a web browser to upload files, analyze extracted data, and download results.

The dashboard provides an intuitive interface for users to interact with the threat intelligence extraction script. The following features are supported:

- **File Upload**: Users can upload single or multiple files (PDF, DOCX, TXT) for processing.
- **Automated Extraction**: The dashboard processes the uploaded files and extracts threat intelligence information.
- **Filtering & Sorting**: Users can filter and sort extracted data based on categories such as IoCs, TTPs, malware names, and threat actors.
- **Download Options**: Extracted data can be downloaded in JSON format for further analysis
- **Real-time Processing**: Displays results dynamically as files are processed.



3.5 View the Output

JSON files will be generated in the same directory as the input PDFs. Each JSON file will correspond to an input PDF and be named **output_<pdf_file_name>.json**

4. Logic Behind the Extraction Process

The script combines regex patterns, SpaCy's NER, and transformer-based models to identify key threat intelligence elements.

4.1 Indicators of Compromise (IoCs):

Regex patterns identify critical IoCs such as:

- **IP Addresses**: Extracts IPv4 addresses like 192.168.1.1.
- **MAC Addresses**: Finds MAC addresses in formats like 00:1A:2B:3C:4D:5E.
- **Domains**: Captures domain names like example.com.
- **URLs**: Identifies URLs with protocols like https://example.com.
- **File Hashes**: Extracts MD5 (32 chars), SHA-1 (40 chars), and SHA-256 (64 chars).
- **Email Addresses**: Finds emails like user@example.com.
- **Registry Keys**: Matches Windows registry keys starting with HKEY_.
- **File Paths**: Detects Windows file paths like C:\folder\file.txt.
- **GUIDs**: Captures unique identifiers like {12345678-1234-1234-1234-123456789012}.
- **Filenames**: Extracts common file types like .exe, .pdf, .jpg.

4.2 Tactics, Techniques, and Procedures (TTPs):

- Searches the text for MITRE ATT&CK framework tactics (e.g., Initial Access) and techniques (e.g., Phishing, T1566).
- Matches text with a predefined list of TTPs and their corresponding IDs.

4.3 Malware Names:

- Scans the text and identifies potential malware references from the input report using SpaCy's NER.
- Matches text with a predefined list of for known malware names
- Enriches results with metadata from VirusTotal API.

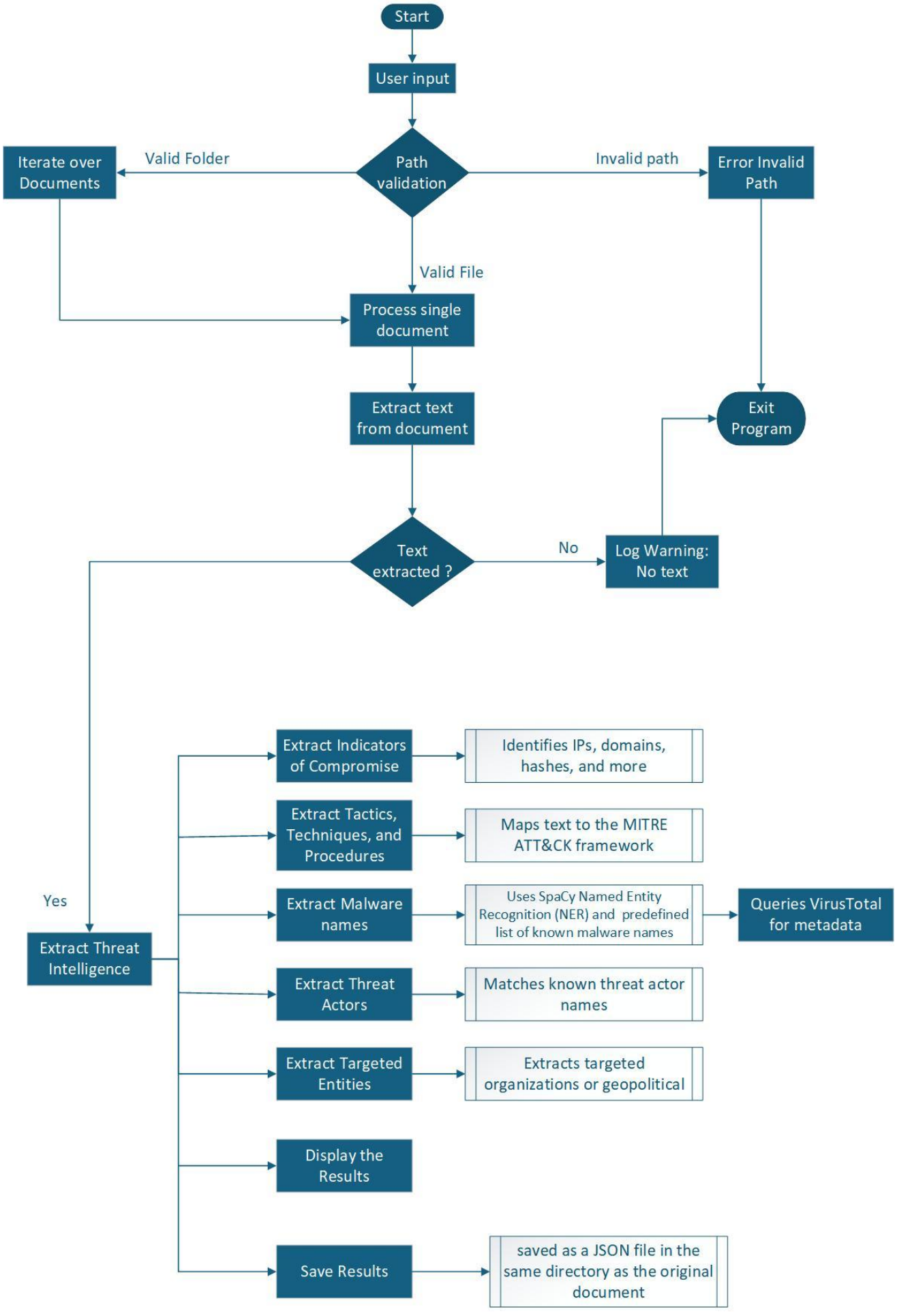
4.4 Threat Actors:

- Scans the text and identifies potential threat actors' references from the input report.
- Uses transformer-based models to identify potential threat actors.
- Extracts organizations, individuals, or nation-state actors involved in cyber threats.

4.5 Targeted Entities:

- Uses transformer-based Named Entity Recognition (NER) to extract targeted industries or organizations.

Program FlowChart



5. Dataset Preprocessing

- **Preprocessing for Scanned PDFs:**
  Use OCR tools (e.g., Tesseract or Adobe Acrobat) to convert scanned PDFs to machine-readable text.
- **Text Cleaning:**
  Remove unnecessary spaces, special characters, or formatting issues.

6. Limitations and Improvements

6.1 Limitations:

- **OCR Dependency:** Scanned PDFs require OCR preprocessing.
- **Formatting Issues:** Extraction accuracy may be affected by documents with complex layouts.
- **Accuracy:** Regex and NER-based methods may result in false positives or negatives.
- **Incomplete mappings:** MITRE ATT&CK techniques may need regular updates.

6.2 Improvements:

- Automate enrichment using APIs like VirusTotal or Recorded Future.
- Expand regex patterns and NER training datasets for better coverage.