Python Text Parser – Security Log Filtering (Section 8)

This script (parser.py) is a simple Python tool that automates security log review.

1. Purpose:

The script reads a plain-text log file and extracts only the lines that may be security-relevant, such as failed logins, errors, warnings, or unauthorized access messages. The filtered lines are written to a separate output file for easier review.

2. How it works:

- Takes two command-line arguments:

1) input_log.txt – the original log file

2) output_alerts.txt – the file that will contain only suspicious lines

- Defines a list of suspicious keywords:

FAILED LOGIN, Failed password, ERROR, WARNING, unauthorized, denied

- For each line in the input file, it checks whether any keyword appears in that line.

- If a keyword is found, the line is written to the output file.

3. Example usage:

python parser.py sample_input_log.txt sample_output_alerts.txt

4. Example input (sample_input_log.txt):

- Normal informational log messages (user logins, backups, etc.)

- One FAILED LOGIN event

- One WARNING about disk space

- One ERROR reaching the database

5. Example output (sample_output_alerts.txt):

Only the lines that contain:

- FAILED LOGIN

- WARNING

- ERROR

6. Value for cybersecurity:

This demonstrates how Python can be used to:

- Automate repetitive log review tasks

- Quickly isolate suspicious events from large log files

- Support security analysts in detecting potential incidents faster