**Project: Brute Force Detection**

**Output/Explanation :**

This project **simulates, processes, and detects brute-force attacks** from **authentication logs**. It consists of three main scripts:

1. **simulated\_logs.py** – Generates simulated authentication logs.
2. **process\_logs.py** – Processes logs to count failed login attempts per IP per minute.
3. **brute\_force\_detection.py** – Identifies potential brute-force attacks based on thresholds.

These scripts can be executed together using **run\_all.py**, which automates the entire process.

1. **simulated\_logs.py - Log Simulation :**

**Purpose:**

* + - Generates 500 login attempt logs.
    - Includes random IP addresses, usernames, passwords, timestamps, and statuses (success/failure)

**Code :**

****

**Output :**

****

1. **process\_logs.py - Log Processing :**

**Purpose:**

* Reads **simulated\_logs.csv**.
* Filters failed login attempts.
* Counts failures per IP per minute.
* Saves results to **failed\_attempts\_count.csv**

**Code :**

****

**Output :**

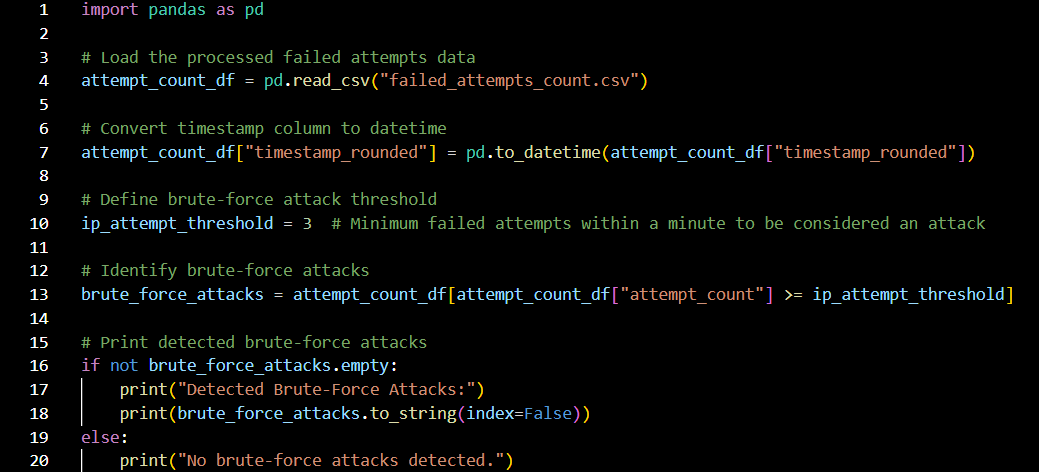
****

1. **brute\_force\_detection.py - Brute-Force Attack Detection :**

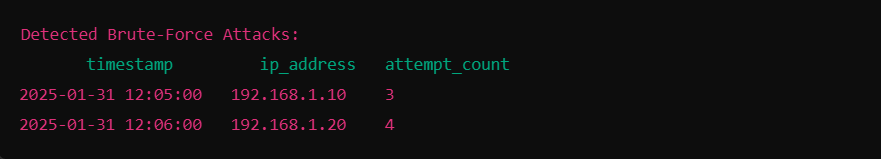
**Purpose:**

* Reads **failed\_attempts\_count.csv**.
* Detects brute-force attacks where an IP exceeds 3 failed attempts within 1 minute.
* Displays detected attacks.

**Code :**

****

**Output :**

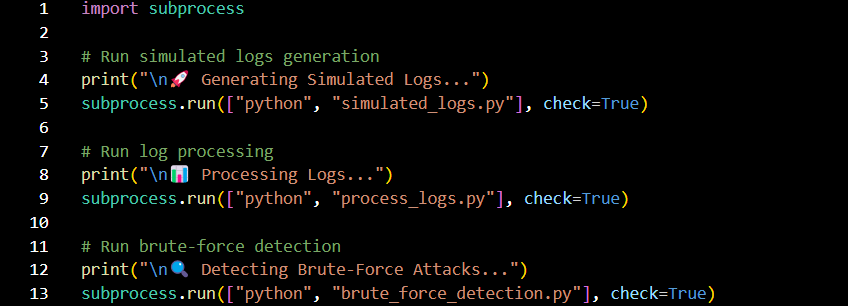
****

1. **run\_all.py - Automating the Execution :**

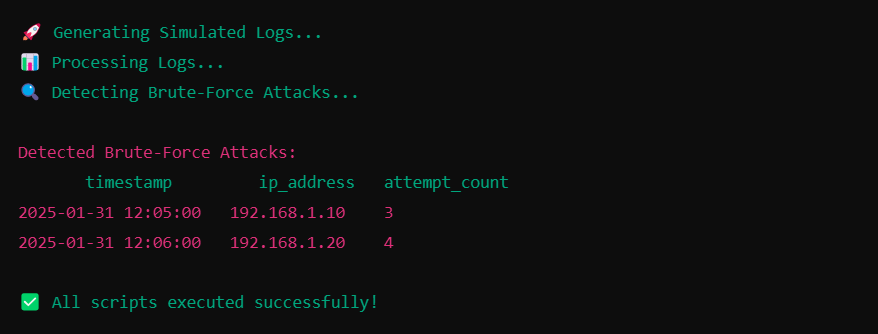
**Purpose:**

* Runs all scripts in sequence.
* Automates log **generation, processing, and brute-force detection**.

**Code :**

****

**Output :**

****

**Conclusion & Future Updates :**

* This project serves as a foundational system for detecting brute-force attack patterns by **simulating, processing, and analysing login attempts**.
* While it provides a reliable detection mechanism based on thresholds and simulated data, there is significant potential **for real-world applications and future enhancements.**