VRV Security's Python Intern

Assignment

VIGNESH MURUGESAN
B.TECH

ARTIFICIAL INTELLIGENCE
AND DATA SCIENCE

Assignment: Log Analysis Script

Objective

The goal of this assignment is to assess your ability to write a Python script that processes log files to extract and analyze key information.

```
# Import necessary libraries
import re
import csv
from collections import defaultdict
from google.colab import files

# Configurable threshold for failed login attempts
FAILED_LOGIN_THRESHOLD = 10

def parse_log_file(file_path):
    """
    Reads the log file and returns the lines as a list of strings.
    """
    with open(file_path, 'r') as file:
        return file.readlines()

def count_requests_per_ip(logs):
    """
```

```
    Counts requests made by each IP address and returns a sorted list of
tuples (IP, count).
    """
    ip_counts = defaultdict(int)
    for log in logs:
        ip_match = re.match(r"(\d+\.\d+\.\d+\.\d+)", log)
        if ip_match:
            ip = ip_match.group(1)
            ip_counts[ip] += 1
    return sorted(ip_counts.items(), key=lambda x: x[1], reverse=True)

def find_most_frequent_endpoint(logs):
    """
    Identifies the most frequently accessed endpoint and its count.
    """
    endpoint_counts = defaultdict(int)
    for log in logs:
        endpoint_match = re.search(r"\"(?:GET|POST|PUT|DELETE) (\/\S+)",
log)
        if endpoint_match:
            endpoint = endpoint_match.group(1)
            endpoint_counts[endpoint] += 1
    return max(endpoint_counts.items(), key=lambda x: x[1])

def detect_suspicious_activity(logs, threshold=FAILED_LOGIN_THRESHOLD):
    """
    Flags IP addresses with failed login attempts exceeding the threshold.
    """
    failed_attempts = defaultdict(int)
    for log in logs:
        if "401" in log or "Invalid credentials" in log:
            ip_match = re.match(r"(\d+\.\d+\.\d+\.\d+)", log)
            if ip_match:
                ip = ip_match.group(1)
                failed_attempts[ip] += 1
    return {ip: count for ip, count in failed_attempts.items() if count >
threshold}

def save_to_csv(requests_per_ip, most_accessed, suspicious_ips,
output_file):
```

```
    """
    Saves the analysis results to a CSV file in the specified format.
    """
    with open(output_file, 'w', newline='') as file:
        writer = csv.writer(file)

        # Write requests per IP
        writer.writerow(["Requests per IP"])
        writer.writerow(["IP Address", "Request Count"])
        writer.writerows(requests_per_ip)
        writer.writerow([])

        # Write most accessed endpoint
        writer.writerow(["Most Accessed Endpoint"])
        writer.writerow(["Endpoint", "Access Count"])
        writer.writerow([most_accessed[0], most_accessed[1]])
        writer.writerow([])

        # Write suspicious activity
        writer.writerow(["Suspicious Activity"])
        writer.writerow(["IP Address", "Failed Login Count"])
        writer.writerows(suspicious_ips.items())

def main():
    # Upload log file
    print("Please upload the log file (e.g., sample.log):")
    uploaded = files.upload()

    # Get the uploaded file name
    log_file_path = list(uploaded.keys())[0]
    output_file_path = "log_analysis_results.csv"

    # Parse log file
    logs = parse_log_file(log_file_path)

    # Perform analyses
    requests_per_ip = count_requests_per_ip(logs)
    most_accessed_endpoint = find_most_frequent_endpoint(logs)
    suspicious_ips = detect_suspicious_activity(logs)
```

```
    # Display results
    print("\nRequests per IP Address:")
    for ip, count in requests_per_ip:
        print(f"{ip}: {count}")
    print("\nMost Frequently Accessed Endpoint:")
    print(f"{most_accessed_endpoint[0]} (Accessed
{most_accessed_endpoint[1]} times)")
    print("\nSuspicious Activity Detected:")
    for ip, count in suspicious_ips.items():
        print(f"{ip}: {count} failed login attempts")

    # Save results to CSV
    save_to_csv(requests_per_ip, most_accessed_endpoint, suspicious_ips,
output_file_path)
    print(f"\nResults have been saved to {output_file_path}")
    files.download(output_file_path)

if __name__ == "__main__":
    main()
```

Same.log

https://drive.google.com/file/d/1gCABczQVdlivrFqjinLtN9sMxg7C4q4k/view?usp=sharing

Colab note

https://colab.research.google.com/drive/1W-atjelj0mvbaoUuEmm5XklYW05-S8VH?usp=sharing

Py file

https://drive.google.com/file/d/1gCABczQVdlivrFqjinLtN9sMxg7C4q4k/view?usp=sharing

Result output

https://drive.google.com/file/d/1m1kcrtm1bTi-NSsi4SGdTxQ7ZALGMd24/view?usp=sharing

I hope this message finds you well and in good health. Thank you so much to the HR team for this opportunity. I am truly interested and excited to be part of this interview process. Thank you once again for selecting me as one of the applicants. I am eagerly looking forward to the next steps. Once again, thank you so much to the HR team!