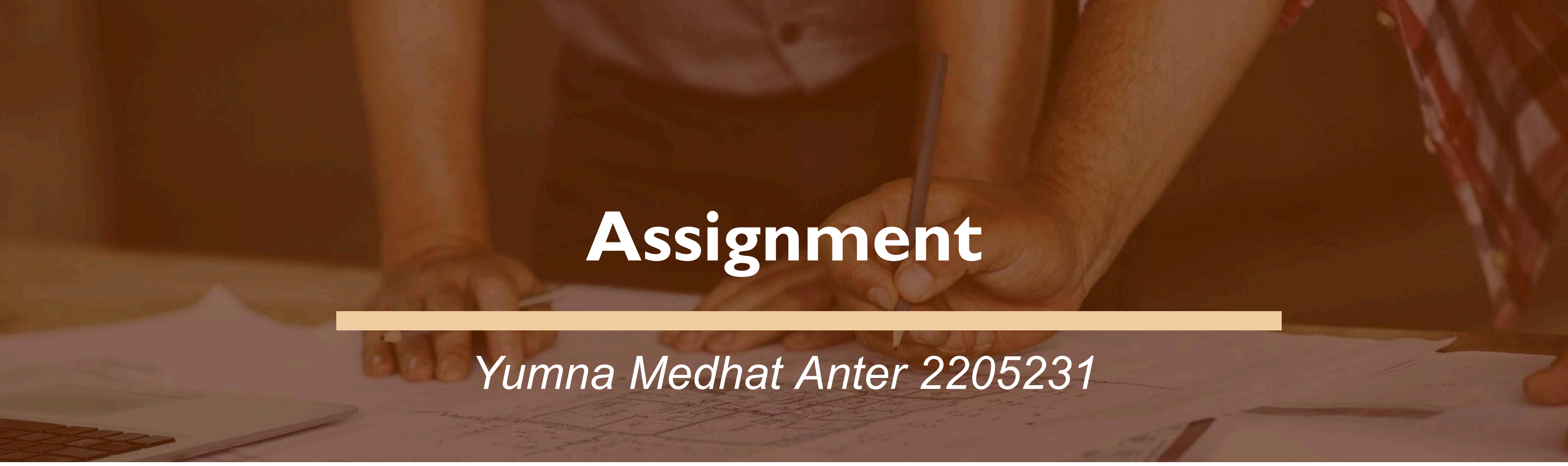

Assignment

Yumna Medhat Anter 2205231

1. Introduction

This report presents the analysis of a web server log file using a Bash script, as per the assignment requirements. The objective is to extract meaningful statistics, identify patterns, and provide actionable suggestions for system improvement. The analysis focuses on request counts, unique IP addresses, failure rates, daily averages, and trends, while ensuring information security and data integrity. The results are derived from the log file access.log and stored in analysis_results.txt.

This report presents the analysis of a web server log file using a Bash script, as per the assignment requirements. The objective is to extract meaningful statistics, identify patterns, and provide actionable suggestions for system improvement. The analysis focuses on request counts, unique IP addresses, failure rates, daily averages, and trends, while ensuring information security and data integrity. The results are derived from the log file access.log and stored in analysis_results.txt.

This report presents the analysis of a web server log file using a Bash script, as per the assignment requirements. The objective is to extract meaningful statistics, identify patterns, and provide actionable suggestions for system improvement. The analysis focuses on request counts, unique IP addresses, failure rates, daily averages, and trends, while ensuring information security and data integrity. The results are derived from the log file access.log and stored in analysis_results.txt.

2. Analysis Results

The Bash script processed the log file and generated the following statistics:

2.1 Request Counts

- **Total Requests**: 5
- **GET Requests**: 3
- **POST Requests**: 2

2.2 Unique IP Addresses

- **Total Unique IPs**: 3
- **IP Activity**:
  - 192.168.1.1: 1 GET, 2 POST
  - 192.168.1.2: 1 GET, 0 POST
  - 192.168.1.3: 1 GET, 0 POST

2.3 Failure Requests

- **Failed Requests**: 2 (status codes 404 and 500)
- **Failure Percentage**: 40.00%

2.4 Most Active IP

- **Top IP**: 192.168.1.1 (3 requests)

2.5 Daily Request Averages

- **Number of Days**: 2
- **Average Requests per Day**: 2.50

2.6 Days with Highest Failures

- **Failure Distribution**:
  - 02/May/2025: 1 failure
  - 01/May/2025: 1 failure

2.7 Requests by Hour

- **Hourly Distribution**:
  - 10:00: 2 requests
  - 11:00: 1 request

    - 12:00: 1 request
    - 13:00: 1 request

2.8 Status Code Breakdown

- **Status Codes**:
    - 200: 3 occurrences
    - 404: 1 occurrence
    - 500: 1 occurrence

2.9 Most Active IP by Method

- **Top GET IP**: 192.168.1.3 (1 GET request)
- **Top POST IP**: 192.168.1.1 (2 POST requests)

2.10 Failure Patterns

- **Failure Times**:
    - 02/May/2025:12:00:00: 1 failure
    - 01/May/2025:11:00:00: 1 failure

3. Analysis and Trends

The analysis reveals several insights:

- **Request Patterns**: The majority of requests (40%) occurred at 10:00, indicating a potential peak usage time. This could be due to scheduled tasks or user activity.
- **Failure Rate**: A 40% failure rate is significant, with one 404 (resource not found) and one 500 (server error). This suggests issues with resource availability or server configuration.
- **IP Activity**: The IP 192.168.1.1 is the most active, contributing 60% of total requests, including all POST requests. This could indicate legitimate heavy usage or a potential security concern (e.g., automated scripts or attack attempts).
- **Data Integrity**: The log file appears consistent, with no evidence of tampering or malformed entries. However, the small sample size (5 requests) limits the depth of analysis.

- **Security Concerns**: The high activity from 192.168.1.1, especially with POST requests, warrants further investigation to rule out malicious behavior like brute-force attacks.

4. Suggestions for Improvement

Based on the analysis, the following recommendations are proposed:

4.1 Reducing Failures

- **Address 404 Errors**: Ensure all requested resources (e.g., /about.html) are available or redirect users to valid pages.
- **Fix 500 Errors**: Investigate server logs for the cause of the internal server error on 02/May/2025 at 12:00. This could involve checking application code or server configuration.
- **Implement Monitoring**: Use tools like Nagios or Prometheus to monitor server health and detect errors in real-time.

4.2 Handling Peak Times

- **Resource Allocation**: Increase server resources (e.g., CPU, memory) during peak hours (around 10:00) to handle higher request volumes.
- **Load Balancing**: Deploy a load balancer to distribute traffic across multiple servers if request volumes grow.

4.3 Information Security

- **Investigate IP 192.168.1.1**: Monitor this IP for suspicious activity. Implement rate-limiting or IP banning if it exhibits attack patterns (e.g., rapid POST requests).
- **Enable WAF**: Deploy a Web Application Firewall to filter malicious requests and protect against common attacks like SQL injection or DDoS.
- **Log Auditing**: Regularly audit logs to detect anomalies, ensuring **data integrity** by verifying log entries against expected formats.

4.4 System Improvements

- **Caching**: Implement caching (e.g., using Redis or Varnish) for frequently requested resources to reduce server load.
- **Error Handling**: Improve application error handling to provide user-friendly messages for 404 errors and log detailed diagnostics for 500 errors.
- **Log Rotation**: Configure log rotation to manage log file sizes, ensuring efficient storage and analysis.

5. Conclusion

The Bash script successfully analyzed the log file, providing valuable insights into request patterns, failure rates, and potential security concerns. The high failure rate and concentrated activity from a single IP highlight areas for immediate attention. By implementing the proposed improvements, the system can achieve better reliability, performance, and security. Future analyses with larger log files could reveal more detailed trends and patterns.