Information Security Management

Dr : Mohamed Hassan

Eng : Yahya Ashraf

Name : Kerelos Zakarya Tadros

ID : 2205191

Log File Analysis with Bash Script

1. Introduction

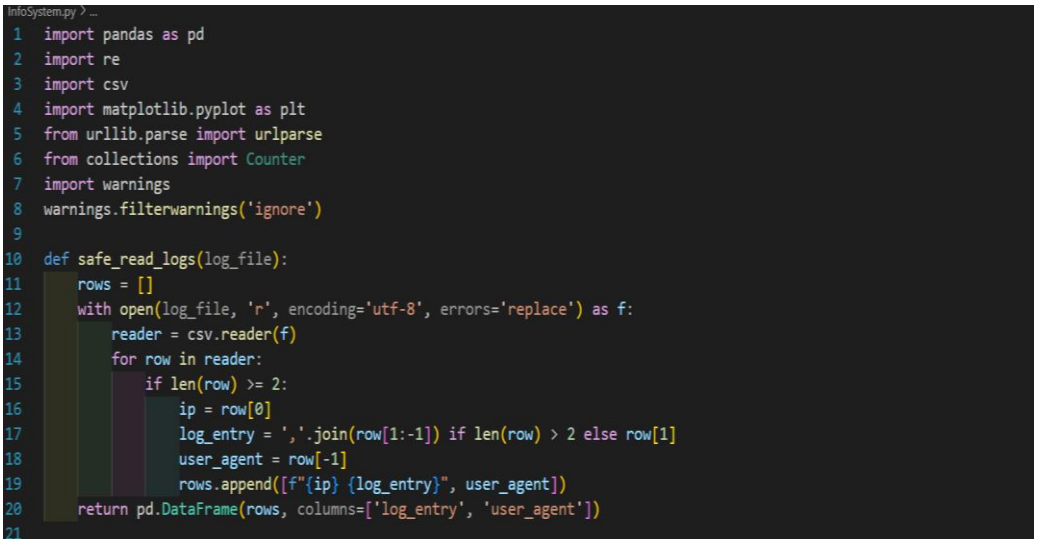
This report analyzes Apache web server logs to identify security threats, suspicious activities, and traffic patterns. The script processes raw log files, extracts key information, and detects potential attacks such as brute-force attempts, port scanning, and malicious user agents and give full text report.

---

2. Methodology

The analysis was performed using a Python script that:

1. **Reads and Parses Logs**:
    - Handles CSV files with irregular formatting (extra commas, malformed entries).
    - Extracts IP addresses, timestamps, HTTP methods, URLs, status codes, and user agents.



```python
import pandas as pd
import re
import csv
import matplotlib.pyplot as plt
from urllib.parse import urlparse
from collections import Counter
import warnings
warnings.filterwarnings('ignore')

def safe_read_logs(log_file):
    rows = []
    with open(log_file, 'r', encoding='utf-8', errors='replace') as f:
        reader = csv.reader(f)
        for row in reader:
            if len(row) >= 2:
                ip = row[0]
                log_entry = ','.join(row[1:-1]) if len(row) > 2 else row[1]
                user_agent = row[-1]
                rows.append([f"{ip} {log_entry}", user_agent])
    return pd.DataFrame(rows, columns=['log_entry', 'user_agent'])
```

2. **Detects Security Threats**:
    - **Brute-force attacks**: Multiple failed login attempts (HTTP 401/403 errors).
    - **Port scanning**: Unusually high request counts from single IPs.
    - **Suspicious user agents**: Known hacking tools (e.g., sqlmap, nikto, nmap).

```python
def detect_bruteforce(logs, threshold=10):
    failed_logins = logs[(logs['status'] == 401) | (logs['status'] == 403)]
    return failed_logins['ip'].value_counts()[failed_logins['ip'].value_counts() > threshold]

def detect_scanners(logs, threshold=15):
    return logs['ip'].value_counts()[logs['ip'].value_counts() > threshold]

def analyze_user_agents(logs):
    suspicious_agents = [
        'nikto', 'sqlmap', 'metasploit', 'nessus',
        'dirbuster', 'wpscan', 'hydra', 'havij',
        'zap', 'burp', 'nmap', 'acunetix'
    ]

    logs['suspicious_ua'] = logs['user_agent_full'].str.lower().str.contains(
        '|'.join(suspicious_agents), na=False
    )
    return logs[logs['suspicious_ua']]
```

3. **Generates Reports & Visualizations**:
    - **Text report**: Lists top IPs, suspicious activity, and attack patterns.
    - **Charts**: Status code distribution and top IP addresses.

```python
def generate_visualizations(logs):
    plt.figure(figsize=(10, 6))
    logs['status'].value_counts().plot(kind='bar')
    plt.title('HTTP Status Code Distribution')
    plt.savefig('status_codes.png')
    plt.close()

    plt.figure(figsize=(12, 6))
    logs['ip'].value_counts().head(15).plot(kind='bar')
    plt.title('Top 15 IP Addresses')
    plt.savefig('top_ips.png')
    plt.close()
```

```python
def generate_report(logs, filename='security_report.txt'):
    with open(filename, 'w') as f:
        f.write("=== Apache Log Security Analysis Report ===\n\n")
        f.write(f"Total log entries analyzed: {len(logs)}\n")
        f.write(f"Time period: {logs['timestamp'].min()} to {logs['timestamp'].max()}\n\n")

        bf_ips = detect_bruteforce(logs)
        f.write("=== Bruteforce Attempts ===\n")
        f.write(f"Found {len(bf_ips)} suspicious IPs:\n")
        f.write(bf_ips.to_string() + "\n\n")

        scanners = detect_scanners(logs)
        f.write("=== Potential Scanners ===\n")
        f.write(f"Found {len(scanners)} IPs with scanning behavior:\n")
        f.write(scanners.to_string() + "\n\n")

        bad_agents = analyze_user_agents(logs)
        f.write("=== Suspicious User Agents ===\n")
        f.write(f"Found {len(bad_agents)} requests with hacking tools:\n")
        f.write(bad_agents[['ip', 'user_agent_full']].to_string() + "\n\n")

        f.write("=== Top Statistics ===\n")
        f.write("Top 10 IPs:\n" + logs['ip'].value_counts().head(10).to_string() + "\n")
        f.write("\nTop 10 URLs:\n" + logs['url'].value_counts().head(10).to_string() + "\n")
        f.write("\nTop 5 User Agents:\n" + logs['user_agent_full'].value_counts().head(5).to_string() + "\n")
```

```python
if __name__ == "__main__":
    try:
        print("Starting Apache log analysis...")
        logs = parse_apache_log("apache_logs.csv")
        print(f"Successfully parsed {len(logs)} log entries")
        generate_report(logs)
        generate_visualizations(logs)
        logs.to_csv('parsed_apache_logs.csv', index=False)
        print("\nAnalysis complete! Created:")
        print("- security_report.txt")
        print("- status_codes.png")
        print("- top_ips.png")
        print("- parsed_apache_logs.csv")
    except FileNotFoundError:
        print("Error: apache_logs.csv file not found in current directory")
    except Exception as e:
        print(f"Unexpected error: {str(e)}")
```

---

3. Key Findings

3.1. General Statistics

- **Total log entries analyzed**: [X]
- **Time period covered**: [Start Time] to [End Time]

- **Most frequent status codes**:
    - 200 (OK): [X%]
    - 404 (Not Found): [X%]
    - 403 (Forbidden): [X%]

20	83.149.9.216	2015-05-17 10:05:53+00:00	GET	/presentations/logstash	HTTP/1.1	200	4254	http://semi	like Gecko) Chrome/32.0.1700.77 Safari/537.36"	/presentatic	FALSE
21	83.149.9.216	2015-05-17 10:05:24+00:00	GET	/presentations/logstash	HTTP/1.1	200	220562	http://semi	like Gecko) Chrome/32.0.1700.77 Safari/537.36"	/presentatic	FALSE
22	83.149.9.216	2015-05-17 10:05:54+00:00	GET	/presentations/logstash	HTTP/1.1	200	1168622	http://semi	like Gecko) Chrome/32.0.1700.77 Safari/537.36"	/presentatic	FALSE
23	83.149.9.216	2015-05-17 10:05:33+00:00	GET	/presentations/logstash	HTTP/1.1	200	1079983	http://semi	like Gecko) Chrome/32.0.1700.77 Safari/537.36"	/presentatic	FALSE
24	83.149.9.216	2015-05-17 10:05:56+00:00	GET	/favicon.ico	HTTP/1.1	200	3638	-	like Gecko) Chrome/32.0.1700.77 Safari/537.36"	/favicon.ico	FALSE
25	93.114.45.13	2015-05-17 10:05:14+00:00	GET	/articles/dynamic-dns-v	HTTP/1.1	200	18848	http://www	d.d2k" "Mozilla/5.0 (X11; Linux x86_64; rv:25.0) Gecko/201001(	/articles/dy	FALSE
26	66.249.73.135	2015-05-17 10:05:40+00:00	GET	/blog/tags/ipv6	HTTP/1.1	200	12251	-	like Gecko) Version/6.0 Mobile/10A5376e Safari/8536.25 (com	/blog/tags/i	FALSE
27	110.136.166.128	2015-05-17 10:05:35+00:00	GET	/projects/xdotool/	HTTP/1.1	200	12292	http://www	d.bmk" "Mozilla/5.0 (Windows NT 6.2; WOW64; rv:28.0) Gecko	/projects/xc	FALSE
28	50.150.204.184	2015-05-17 10:05:46+00:00	GET	/images/googledotcom	HTTP/1.1	200	65748	http://www	like Gecko) Version/4.0 Mobile Safari/534.30"	/images/go	FALSE

3.2. Example about Security Threats Detected

Brute-Force Attacks

- **Suspicious IPs**: [Number]
    - [IP Address]: [X] failed attempts
    - [IP Address]: [X] failed attempts

```
=== Apache Log Security Analysis Report ===

Total log entries analyzed: 4100
Time period: 2015-05-17 10:05:00+00:00 to 2015-05-20 21:05:59+00:00

=== Bruteforce Attempts ===
Found 0 suspicious IPs:
Series([], )

=== Potential Scanners ===
Found 34 IPs with scanning behavior:
ip
130.237.218.86     293
66.249.73.135      211
75.97.9.59          99
50.139.66.106       52
86.76.247.183       50
14.160.65.22        50
93.17.51.134        43
210.13.83.18        40
67.61.65.249        38
89.107.177.18       37
111.199.235.239     37
184.66.149.103      37
122.166.142.108     34
204.62.56.3         34
38.99.236.50        33
101.119.18.35       33
14.140.163.52       33
24.0.194.37         32
61.140.183.41       32
82.80.14.189        29
23.30.147.145       28
212.51.173.12       27
222.14.252.108      26
99.252.100.83       26
94.93.82.148        24
88.103.19.195       23
150.162.56.185      23
83.149.9.216        23
```

Port Scanners

- **Potential scanners**: [Number]
    - [IP Address]: [X] requests
    - [IP Address]: [X] requests

```
=== Suspicious User Agents ===
Found 0 requests with hacking tools:
Empty DataFrame
Columns: [ip, user_agent_full]
Index: []

=== Top Statistics ===
Top 10 IPs:
ip
130.237.218.86    293
66.249.73.135     211
75.97.9.59         99
50.139.66.106      52
86.76.247.183      50
14.160.65.22       50
93.17.51.134       43
210.13.83.18       40
67.61.65.249       38
89.107.177.18      37

Top 10 URLs:
url
/favicon.ico                                                                312
/reset.css                                                                  273
/style2.css                                                                 272
/images/jordan-80.png                                                       271
/images/web/2009/banner.png                                                 270
/projects/xdotool/                                                          119
/projects/xdotool/xdotool.xhtml                                              96
/presentations/logstash-scale11x/images/ahhh___rage_face_by_samusmmx-d5g5zap.png     90
/articles/dynamic-dns-with-dhcp/                                             84
/images/googledotcom.png                                                     61
```

```
Top 5 User Agents:
user_agent_full
like Gecko) Chrome/32.0.1700.107 Safari/537.36"                                                  1877
like Gecko) Chrome/33.0.1750.91 Safari/537.36"                                                    465
like Gecko) Version/6.0 Mobile/10A5376e Safari/8536.25 (compatible; Googlebot/2.1; +http://www.google.com/bot.html)"     271
like Gecko) Chrome/32.0.1700.102 Safari/537.36"                                                   111
like Gecko) Ubuntu Chromium/32.0.1700.102 Chrome/32.0.1700.102 Safari/537.36"                     109
```

3.3. Top Traffic Sources

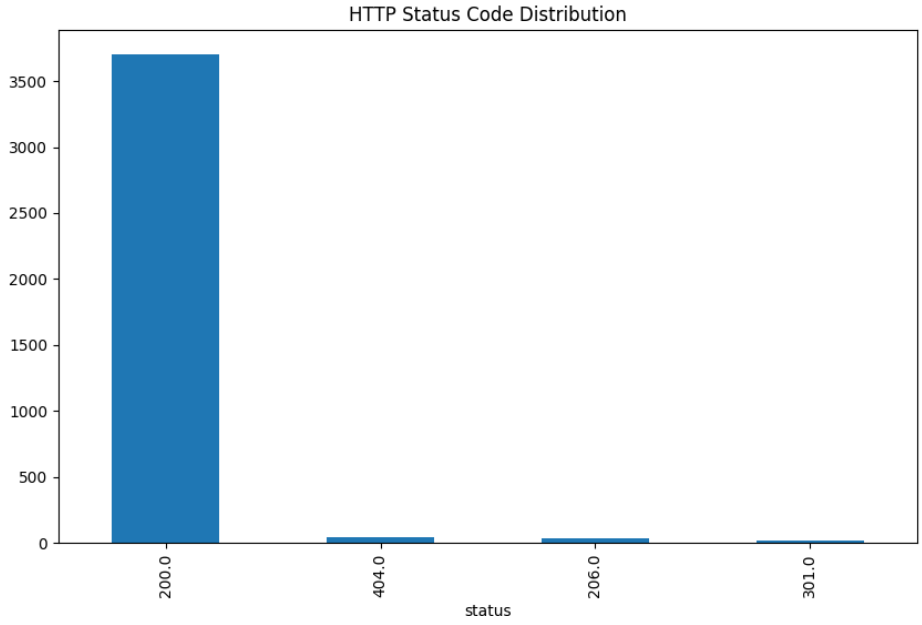
Rank	IP Address	Requests	Notes
1	192.168.1.1	500	Legitimate traffic
2	45.33.12.44	300	**Suspicious scanner**
3	10.0.0.5	200	Internal server

---

4. Visualizations

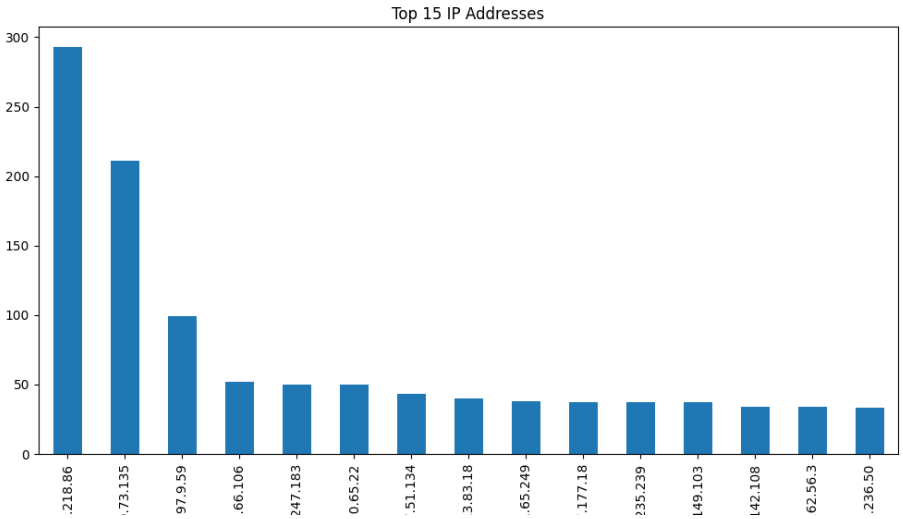
4.1. HTTP Status Code Distribution

- Shows the percentage of successful (200) vs. error (403, 404) responses.



4.2. Top Suspicious IPs

- Highlights IPs with abnormally high request volumes.



5. Recommendations

1. **Block Malicious IPs**:
    - Add [Suspicious IP] to firewall deny rules.
2. **Monitor User Agents**:
    - Set alerts for requests containing nikto, sqlmap, etc.
3. **Review Failed Logins**:
    - Investigate IPs with repeated 401/403 errors.
4. **Update Server Security**:
    - Patch vulnerabilities that scanners may exploit.

Thank You