Security Alert Monitoring & Incident Response

Introduction:

As part of my cybersecurity internship, Task 2 focused on simulating a real-world Security Operations Center (SOC) scenario by utilizing a SIEM (Security Information and Event Management) tool. The objective was to monitor simulated alerts generated by various types of attacks, including SQL Injection, Cross-site Scripting (XSS), and Unauthorized Access Attempts. Using Splunk Cloud, the logs were analyzed to detect, classify, and respond to threats appropriately.

This task provided hands-on experience in alert monitoring, log correlation, and incident triage—key elements of modern incident response workflows. It served as a practical implementation of foundational SOC skills, including detection, classification, and mitigation strategy design.

Tools Used:

1. Splunk Cloud Platform (Free Trial)
2. Sample log data (sample_security_logs.txt)
3. Firefox Web Browser for Splunk Dashboard Access
4. Local Analysis Environment (Windows OS)
5. Screenshot capturing tools for documenting outputs

Summary of Detected Incidents:

Attack Type	Detection Phrase	No. of Occurrences
SQL Injection	SQL Injection attempt detected in URL: /index.php?id=1'--	2
Cross-site Scripting (XSS)	Cross-site scripting detected: <script>alert("XSS")</script>	2
Unauthorized Access	Unauthorized access attempt to /admin from 192.168.1.105	2

Incident Analysis

1. SQL Injection Attack

SQL Injection is one of the most critical web vulnerabilities according to the OWASP Top 10. During analysis, logs revealed entries indicating an injection string `/index.php?id=1'--` being passed in a URL parameter. This suggests an attempt to tamper with backend SQL queries.

The attacker may be trying to bypass login authentication or extract unauthorized information from the database by terminating SQL statements prematurely and appending malicious clauses. If successful, this could result in credential dumping, data loss, or complete system compromise.

2. Cross-site Scripting (XSS)

The logs revealed attempts to execute script tags within the application, such as: `<script>alert("XSS")</script>`. This is a common example of reflective XSS, where user-supplied data is immediately reflected in the web page output without proper sanitization.

XSS attacks can lead to session hijacking, phishing, defacement, or redirection to malicious sites. If executed on admin or privileged sessions, it can open doors to further compromise. Security-conscious applications apply context-sensitive encoding and sanitization at every input/output stage.

3. Unauthorized Access Attempt

The system also recorded repeated unauthorized access attempts to protected routes such as `/admin`. For example, the log `Unauthorized access attempt to /admin from 192.168.1.105` suggests brute-force probing. This is often the first step attackers take to gain elevated access to configuration panels or sensitive settings.

In a production environment, such events must be flagged immediately. Automated lockout policies, geo-blocking, and two-factor authentication are key strategies to prevent such exploitation attempts.

Remediation Recommendations

1. Employ parameterized SQL queries and input validation across all user inputs to block SQL injection.
2. Use frameworks or libraries that inherently escape data to prevent XSS, and enforce Content Security Policy (CSP).
3. Configure alerts in your SIEM for repeated failed login attempts and unauthorized access patterns.

4. Use multi-factor authentication (MFA) and implement proper logging and alerting mechanisms.
5. Limit access to admin panels using VPN or IP whitelisting.
6. Educate developers and staff on secure coding and detection techniques.

Sample Log:

Jul 8 10:01:42 webserver sshd[1001]: Failed password for invalid user admin from 192.168.1.100 port 22 ssh2

Jul 8 10:02:10 webserver sshd[1002]: Failed password for root from 192.168.1.101 port 22 ssh2

Jul 8 10:05:20 webserver sshd[1003]: Accepted password for root from 192.168.1.102 port 22 ssh2

Jul 8 10:07:11 webserver sshd[1004]: Failed password for invalid user guest from 192.168.1.103 port 22 ssh2

Jul 8 10:09:43 webserver sshd[1005]: Failed password for root from 192.168.1.101 port 22 ssh2

Jul 8 10:10:21 webserver apache[1234]: SQL Injection attempt detected in URL: /index.php?id=1'--

Jul 8 10:11:02 webserver apache[1235]: Cross-site scripting detected: <script>alert("XSS")</script>

Jul 8 10:12:55 webserver apache[1236]: Unauthorized access attempt to /admin from 192.168.1.105

Conclusion

This task provided a comprehensive introduction to real-world incident monitoring and response workflows. By using Splunk Cloud, the ingestion, filtering, and analysis of structured log data became seamless. It highlighted how important visibility and centralized monitoring are to any security program.

Not only did this task help build comfort with log parsing and event correlation, but it also reinforced concepts like the kill chain, attack vectors, and proactive defense techniques. Going forward, these skills lay the groundwork for deeper investigations, threat hunting, and building more secure systems.

Prepared by: Priyanshu Shekhar Choudhary

Cybersecurity Intern:

Internship at Future Interns