CERBERUS

Security Assessment Report

Date: March 20, 2025

Report ID: CERB-20250320-001

Table of Contents

1. Executive Summary

This report details the security vulnerabilities identified during the scan of the target application hosted at `https://juice-shop.herokuapp.com`. The scan results revealed several critical security issues that need immediate attention to ensure the protection of the application and its data. The vulnerabilities range from misconfigurations to potential exposure of sensitive information. Additionally, the knowledge graph data provides insights into the IP addresses and open ports associated with the target application, which can help in further assessing and mitigating potential security risks.

2. Key Findings

2.1 Authentication and Access Control Vulnerabilities

- Finding: /.htpasswd file detected, which contains authorization information.
- OWASP Top 10 Category: A2:2021-Cryptographic Failures
- Impact and Risk: Unauthorized access to sensitive data and potential compromise of user credentials.
- Remediation: Remove the /.htpasswd file from the web root and use a secure authentication mechanism.

2.2 Security Configuration Problems

- Finding: The X-Content-Type-Options header is not set.
- OWASP Top 10 Category: A6:2021-Security Misconfiguration
- Impact and Risk: Potential for MIME-sniffing attacks, which can lead to the execution of malicious scripts.
- Remediation: Ensure that the X-Content-Type-Options header is set to "nosniff".
- Finding: The Content-Encoding header is set to "deflate", which may indicate a vulnerability to the BREACH attack.
- OWASP Top 10 Category: A6:2021-Security Misconfiguration
- Impact and Risk: Sensitive data can be exposed through compression side-channel attacks.
- Remediation: Disable compression for sensitive data or use other mitigation techniques.

2.3 Sensitive Data Exposure

- Finding: Multiple files detected with sensitive extensions (e.g., .pem, .jks, .tar, .war).
- OWASP Top 10 Category: A3:2021-Injection
- Impact and Risk: Exposure of sensitive data such as certificates, keys, and backups, which can be used to compromise the application.
- Remediation: Securely store and encrypt sensitive files. Remove unnecessary files from the web root.

3. Knowledge Graph Insights

3.1 IP Addresses and Ports

- IP Addresses:
- 46.137.15.86
- 54.220.192.176
- 54.73.53.134
- Open Ports:
- 46.137.15.86: Port 443/SSL (status: open)
- 54.220.192.176: Port 443/SSL (status: open)
- 54.73.53.134: Port 443/SSL (status: open)

These IP addresses and open ports indicate that the application is accessible over HTTPS, which is a good start for secure communication. However, it is crucial to ensure that the SSL/TLS configuration is secure and up-to-date to prevent man-in-the-middle attacks and other SSL-related vulnerabilities.

4. Risk Assessment

4.1 High-Risk Vulnerabilities

- Authentication and Access Control Vulnerabilities: Unauthorized access to sensitive data and potential compromise of user credentials.
- Security Configuration Problems: Potential for MIME-sniffing and compression side-channel attacks.
- Sensitive Data Exposure: Exposure of sensitive files, leading to potential data breaches.

4.2 Medium-Risk Vulnerabilities

- None detected based on the provided scan results.

4.3 Low-Risk Vulnerabilities

- None detected based on the provided scan results.

5. Recommendations

1. Immediate Actions:
- Remove the /.htpasswd file from the web root.
- Set the X-Content-Type-Options header to "nosniff".
- Disable compression for sensitive data or use other mitigation techniques.

2. Long-Term Strategies:
- Implement a secure authentication mechanism.
- Regularly audit and secure the configuration of the web server.
- Encrypt and securely store sensitive files.
- Remove unnecessary files from the web root.
- Ensure that the SSL/TLS configuration is secure and up-to-date to prevent man-in-the-middle attacks and other SSL-related vulnerabilities.

3. Continuous Monitoring:
- Regularly scan the application for new vulnerabilities.
- Conduct penetration testing to identify and mitigate potential security weaknesses.
- Monitor the open ports and IP addresses associated with the application to detect any unauthorized access attempts.

By addressing these vulnerabilities and leveraging the insights from the knowledge graph, the security posture of the application can be significantly improved, reducing the risk of data breaches and unauthorized access.