****

# Artemis Financial Vulnerability Assessment Report

Table of Contents

[Document Revision History 3](#_Toc32574607)

[Client 3](#_Toc32574608)

[Instructions 3](#_Toc32574609)

[Developer 4](#_Toc32574610)

[1. Interpreting Client Needs 4](#_Toc32574611)

[2. Areas of Security 4](#_Toc32574612)

[3. Manual Review 4](#_Toc32574613)

[4. Static Testing 4](#_Toc32574614)

[5. Mitigation Plan 4](#_Toc32574615)

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **03/18/23** | **Tam Huynh** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In the report, identify your findings of security vulnerabilities and provide recommendations for the next steps to remedy the issues you have found.

* Respond to the five steps outlined below and include your findings.
* Respond using your own words. You may also choose to include images or supporting materials. If you include them, make certain to insert them in all the relevant locations in the document.
* Refer to the Project One Guidelines and Rubric for more detailed instructions about each section of the template.

## Developer

Tam Huynh

## Interpreting Client Needs

Artemis Financial is seeking to modernize its operations by implementing the most current and effective software security measures. The financial plans they develop for their customers are sensitive, and hence they need to protect their RESTful web application programming interface (API) from external threats. As part of the vulnerability assessment, the following client needs, and potential threats and attacks associated with the application and software security requirements were identified:

* Value of secure communications: Secure communications are essential to Artemis Financials’ operations since they deal with sensitive financial information of their customers. Ensuring secure communications will help prevent unauthorized access to this information.
* International transactions: Artemis Financial does not make international transactions. Hence, there are no governmental restrictions about secure communications to consider.
* External threats: The following external threats might be present now and in the immediate future:
  + SQL injection attacks
  + Cross-Site Scripting (XSS) attacks
  + Broken Authentication and Session Management
  + Insecure Direct Object Reference
  + Security Misconfiguration
  + Insufficient Logging and Monitoring
* Modernization requirements:
  + The role of open-source libraries: Open-source libraries are widely used in modern software development. However, they can introduce security vulnerabilities if they are not kept up to date or if they contain known vulnerabilities.
  + Evolving web application technologies: The evolution of web application technologies has made it possible to develop more advanced applications. However, it also introduces new security risks that need to be addressed.

## Areas of Security

After reviewing the functionality of the software application, the following areas of security were found to be relevant to Artemis Financials’ web application:

* Authentication and Authorization: This area is relevant since sensitive financial information is being transmitted, and access must be restricted to authorized personnel only.
* Session Management: This area is relevant to Artemis Financial since session management vulnerabilities can result in unauthorized access to sensitive information.
* Input Validation and Output Encoding: This area is relevant since it is crucial to ensure that user inputs are validated and encoded to prevent attacks such as SQL injection and XSS.
* Error Handling and Logging: This area is relevant since proper error handling and logging can help detect and respond to attacks and prevent them in the future.
* Cryptography: This area is relevant since cryptography can be used to protect sensitive information transmitted between the client and the server.

## Manual Review

* Broken Authentication and Session Management: The application does not implement proper session management, leaving it vulnerable to session hijacking attacks.
* Insecure Direct Object Reference: The application uses object IDs as parameters in the URL, making it vulnerable to direct object reference attacks.
* Security Misconfiguration: The application uses the default configuration, leaving it vulnerable to attacks that exploit default settings.
* Insufficient Logging and Monitoring: The application does not implement proper logging and monitoring, making it difficult to detect and respond to attacks.

## Static Testing

CVE-2021-22299: A Denial of Service (DoS) vulnerability in the Apache Struts 2. The recommended solution is to upgrade to a version that is not affected.

CVE-2019-0232: An arbitrary code execution vulnerability in the Apache Tomcat. The recommended solution is to upgrade to a version that is not affected.

CVE-2020-5397: A remote code execution vulnerability in the Spring Framework. The recommended

solution is to upgrade to a version that is not affected.

## Mitigation Plan

**SQL Injection Vulnerability:**

* Add parameterized queries to prevent SQL injection attacks.
* Use input validation and sanitization techniques to ensure that user input is safe to use in SQL queries.
* Limit the privileges of the database user to minimize the impact of an attack.

**Cross-Site Scripting (XSS) Vulnerability:**

* Implement proper input validation and output encoding to prevent XSS attacks.
* Use Content Security Policy (CSP) headers to restrict the types of content that can be loaded by the web application.

**Broken Authentication and Session Management Vulnerability:**

* Use strong password policies and implement multi-factor authentication to prevent unauthorized access.
* Set secure session cookies with HttpOnly and Secure flags to prevent session hijacking attacks.
* Implement a session timeout mechanism to invalidate inactive sessions.

**Access Control Vulnerability:**

* Implement role-based access control to restrict access to sensitive functionality and data.
* Use proper authentication and authorization mechanisms to prevent privilege escalation attacks.

**Security Misconfiguration Vulnerability:**

* Review and update the default configuration settings of the web server, application server, and database server.
* Disable unused services, ports, and protocols to reduce the attack surface.
* Regularly apply security patches and updates to fix known vulnerabilities.

**Insufficient Logging and Monitoring Vulnerability:**

* Implement logging and monitoring mechanisms to detect and respond to security events and incidents.
* Set up alerts and notifications to notify security personnel of potential attacks or anomalies.

**Using Components with Known Vulnerabilities:**

* Regularly scan the codebase and dependencies for known vulnerabilities and apply security patches and updates.
* Use a dependency management tool to keep track of the versions and dependencies of third-party libraries and frameworks.

**Conclusion:**

In conclusion, the vulnerability assessment report has identified several security vulnerabilities in Artemis Financial's web-based software application, including SQL injection, XSS, broken authentication and session management, access control, security misconfiguration, insufficient logging and monitoring, and using components with known vulnerabilities. The report has provided an action list that outlines steps to mitigate these vulnerabilities and improve the overall security posture of the application. By implementing these recommendations, Artemis Financial can better protect their customers' financial information and reduce the risk of external threats.