****

# 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** | **[Date]** | **[Your name]** |  |

## 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

Jessica Mikha

## Interpreting Client Needs

Artemis Financial is seeking to modernize its operations and enhance the security of its web-based financial planning application. Handling sensitive customer information, they prioritize safeguarding against external threats and ensuring secure communications. Though no specific details on international transactions or governmental restrictions are provided, it's prudent to consider these factors. Modernization efforts should also align with evolving web application technologies, including the role of open-source libraries.

## Areas of Security

Considering the financial nature of Artemis Financials’ web application, key security areas include:

* Authentication and Authorization: Implement robust mechanisms to restrict access to authorized users.
* Input Validation and Output Encoding: Apply proper validation and encoding to prevent vulnerabilities like SQL injection and XSS attacks.
* Session Management: Ensure secure creation, maintenance, and termination of user sessions.
* Data Protection: Encrypt sensitive data during transmission and storage to prevent unauthorized access.
* Secure Communication: Utilize HTTPS to safeguard data transmission and uphold confidentiality and integrity.

## Manual Review

A manual code review revealed vulnerabilities:

* Cross-Site Scripting (XSS) Vulnerability: The "userProfile.jsp" file lacks output encoding, allowing potential script injection.
* SQL Injection Vulnerability: The "loginDAO.java" file constructs SQL queries without proper user input validation.

## Static Testing

Running the code through static testing using the dependency-check plugin identified vulnerabilities:

* CVE-XXXX-XXXX: Description and recommended solutions found in the dependency-check report.
* CVE-XXXX-XXXX: Description and recommended solutions found in the dependency-check report.

## Mitigation Plan

To address the identified vulnerabilities:

* XSS Vulnerability: Implement proper output encoding in "userProfile.jsp" using libraries or frameworks for automatic encoding.
* SQL Injection Vulnerability: Modify "loginDAO.java" to use parameterized queries or prepared statements for automatic input sanitization.
* Dependency Vulnerabilities: Follow recommendations in the dependency-check report, including updating or patching vulnerable libraries, frameworks, or components.