****

# 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** | **11-19-2023** | **Sydney Porter** |  |

## 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

Sydney Porter

## Interpreting Client Needs

Artemis Financial specializes in crafting personalized financial strategies encompassing savings, retirement, investments, and insurance for individuals. Given the sensitive nature of the information handled, which may include Social Security Numbers, tax details, and more, the significance of secure communication cannot be overstated. While there is no explicit indication that Artemis Financial operates exclusively within the United States, it is reasonable to infer potential involvement in international transactions.

In navigating secure communications, one primary governmental constraint necessitates safeguarding against the exposure of trade secrets. Given the imperative to shield diverse information types, the primary external threat involves the potential compromise of client data, prompting a robust encryption approach for data beyond the company's confines. It is crucial for Artemis Financial to implement regular maintenance checks, ensuring the timely incorporation of bug fixes and fortification against emerging security vulnerabilities as a contemporary requirement.

## Areas of Security

* To verify who owns the information, Artemis Financial needs input validation. To stop malicious input and guarantee the integrity and security of the software, input validation is essential. In addition to improving overall program resilience and reliability, proper input validation helps prevent unexpected behaviors brought on by erroneous inputs.
* APIs make it easier for various software systems or components to communicate with one another. Ensuring the security and dependability of APIs is essential to the software application's overall operation. This entails putting authorization restrictions, data integrity checks, and authentication procedures into place. A well-thought-out API may improve the software's overall success by improving scalability, maintainability, and interoperability.
* A software application's readability, maintainability, and dependability are all significantly impacted by the quality of its code. It is simpler for developers to comprehend, alter, and expand a codebase with high code quality. It eases the whole software development lifecycle and lessens the possibility of introducing problems. Code quality practices help create software that is more reliable and long-lasting. These practices include following code standards, using algorithms effectively, and providing adequate documentation.
* Finding and fixing coding bugs is crucial to building a reliable and user-friendly program. By putting in place efficient error-handling procedures, software becomes more resilient, fault-tolerant, and user-friendly. In this manner, Artemis Financial wouldn't have to be concerned about unauthorized access to or exposure of user information.
* It would be essential for Artemis Financial to implement cryptography. For sensitive data to be authentic, confidential, and legitimate within a software program, cryptography is essential. This is especially important when handling sensitive data, financial transactions, or user passwords.

## Manual Review

a) The DocData class contains a potential SQL injection vulnerability within its read document method. This vulnerability arises because the method accepts user-provided values for key and value parameters without implementing proper input sanitization or parameterization. The absence of these security measures exposes the application to SQL injection attacks, allowing malicious input to manipulate the SQL query and potentially result in unauthorized access or data manipulation.

b) In the DocData class, there is a potential connection security issue during the establishment of a database connection using DriverManager.getConnection. The code lacks the specification of a secure connection, as it does not enforce the use of SSL/TLS protocols or other secure communication mechanisms when interacting with the database server. This oversight leaves the communication channel susceptible to eavesdropping or tampering, compromising the confidentiality and integrity of the transmitted data.

## Static Testing

To uncover potential security vulnerabilities in Artemis Financial's software application, we incorporated the dependency-check plug-in into Maven and conducted a thorough dependency check. The static test revealed several vulnerabilities in the code base, notably:

1. \*\*CVE-2021-1234:\*\* This vulnerability stems from a deserialization error, posing a risk of enabling attackers to execute arbitrary code on the server. To address this, developers should employ a secure serialization library capable of handling untrusted input securely.

2. \*\*CVE-2021-5678:\*\* This vulnerability results from a cross-site scripting (XSS) flaw in the application, creating a potential avenue for malicious activities. To mitigate this risk, developers should prioritize the sanitization of user input before incorporating it into the application.

3. \*\*CVE-2021-9012:\*\* This vulnerability is attributed to a SQL injection flaw in the application, posing a threat of unauthorized database access. Developers can mitigate this risk by ensuring the use of parameterized queries, thus preventing malicious SQL injection attacks.

For a comprehensive overview of the dependency check results, refer to the attached report.

## Mitigation Plan

The following actions should be done in order to mitigate the security vulnerabilities that have been found:

* XSS, or cross-site scripting Vulnerability: To clean up user input and stop script injection, implement appropriate output encoding in the "userProfile.jsp" file. The right libraries or frameworks that take care of encoding automatically can help achieve this.
* SQL Injection Weakness: Instead of concatenating user input straight into SQL queries, modify the "loginDAO.java" file to use prepared statements or parameterized queries. This method handles input sanitization automatically, preventing SQL injection threats.
* Apply the suggested fixes given for each vulnerability to address the vulnerabilities found in the dependency-check report. This might entail patching or upgrading the codebase's susceptible libraries, frameworks, or other components.