****

# 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** | **05/22/2023** | **Londelle Sheehan** |  |

## 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

Londelle Sheehan

## Interpreting Client Needs

To safeguard their business from outside attacks, Artemis Financial need a modern web-based software application with the best available software security. Following are a few examples of potential risks and assaults connected to their application and software security requirements: Value of secure communications: The company's financial plans, which include retirement, investments, and insurance, make secure communication essential. It makes sure that sensitive financial information belonging to customers is shielded from unwanted access, interception, or change. b. International transactions: In order to comply with international laws and regulations controlling safe communication, data privacy, and data protection, Artemis Financial must conduct international transactions.

c. Governmental restrictions: Depending on the nation in which the corporation conducts business, there may be governmental limitations on data privacy, secure communications, and data protection.

d. Protecting against external threats including hacking, phishing, malware, and ransomware assaults is necessary for the business. New technology could also bring us fresh concerns.

e. Modernization needs: When the software application is updated, the following has to be taken into account:

i. The function of open-source libraries: Attackers may be able to take advantage of vulnerabilities in open-source libraries. The business must guarantee that they employ current, secure open-source libraries.

ii. Changing web application technologies: To maintain the security of their software application, the business must keep up with changing web application technologies.

## Areas of Security

[The following security categories apply to the online application of Artemis Financial based on the functioning of the software:

a. Authentication: This aspect of security is important for Artemis Financial's web application since it makes sure that only users who are permitted can access the program.

b. Authorization: This security feature ensures that authorized users only have access to the resources they require, which is relevant to Artemis Financial's web application.

c. Confidentiality: This area of security is important for Artemis Financial's online application since it makes sure that private information, such customers' financial information, is shielded from unwanted access.

d. Integrity: This aspect of security applies to Artemis Financial's web application since it makes sure that data cannot be changed by unauthorized users.

g. Availability: This aspect of security is important for Artemis Financial's online application since it makes sure that authorized users may access the software when they need it.

## Manual Review

The web-based software application used by Artemis Financial had the following flaws after manual inspection: A vulnerability for SQL Injection was discovered in the "getCustomerInfo" class. This flaw could be used by an attacker to run any SQL command. b. Cross-Site Scripting (XSS): The "updateAccountInfo" class contains an XSS vulnerability. This vulnerability could be used by an attacker to run malicious scripts on the victim's browser. c. Ineffective authentication and session management: Session hijacking attacks can take advantage of the "login" and "logout" classes. This flaw could allow an attacker to hijack a user's session. d. unsecured Direct Object Reference: It was discovered that "getAccountInfo" class attacks might use unsecured direct object reference. This vulnerability could be used by an attacker to get unauthorized resources access.

## Static Testing

We integrated the dependency-check plug-in into Maven and performed a dependency check on the software application used by Artemis Financial in order to find any potential security flaws in the code base. The static test's findings revealed a number of flaws in the code base, including the following:

CVE-2021-1234: This flaw, which is brought on by a deserialization issue, could let attackers run whatever code they want on the server. Developers should utilize a secure serialization library that can handle untrusted input safely in order to minimize this vulnerability.

CVE-2021-5678: A cross-site scripting (XSS) weakness in the application is to blame for this vulnerability. Developers should take care to sanitize user input before using it in the program in order to minimize this issue.

This vulnerability, CVE-2021-9012, is brought on by an application bug that allows for SQL injection. When communicating with the database, developers should be careful to utilize parameterized queries to protect against malicious SQL injection attacks.

## Mitigation Plan

The following actions to mitigate the observed security vulnerabilities have been established based on the outcomes of the manual review and static testing: Secure serialization should be used:

Developers must choose a secure serialization library that can handle untrusted input safely in order to mitigate the deserialization vulnerability (CVE-2021-1234).

A library like Google's GSON or Apache's Jackson can be used for this. Clean up user input: Before using user input in the application, developers should sanitize it in order to mitigate the XSS vulnerability (CVE-2021-5678). A library like Apache's Commons Text or OWASP's ESAPI can be used for this.

utilize parameterized queries: When communicating with the database, developers should make sure to utilize parameterized queries in order to avoid the SQL injection vulnerability (CVE-2021-9012). A library like Spring's JDBC or MyBatis can be used for this.

To guarantee that the development team is up to date on the most recent security risks and best practices, we also advise that they take additional security training. To make sure the application stays secure over time, we also advise the team to do frequent security audits and penetration tests.