****

# 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

Romario Gustave

## Interpreting Client Needs

a) Importance of Secure Communications:

Secure communications are vital for the company as they safeguard the confidentiality, integrity, and authenticity of sensitive data transmitted within the software application. By implementing robust encryption protocols and secure communication channels, the company can mitigate the risk of unauthorized access, data tampering, and interception, ensuring the privacy and security of critical information.

b) International Transactions:

If the company engages in international transactions, specific security considerations must be considered. Compliance with international data protection regulations and adherence to encryption standards are crucial to protect the confidentiality of data during transit, particularly when crossing geographical boundaries.

c) Governmental Restrictions and Secure Communications:

Governmental restrictions or regulations may impose specific requirements on secure communications, especially in regulated industries such as finance or healthcare. Complying with these regulations is essential to avoid legal repercussions, protect customer data, and maintain trust with stakeholders.

d) Identifying External Threats:

To ensure effective security planning, it is essential to identify potential external threats. These threats may include malicious actors attempting unauthorized access, data breaches, injection attacks, social engineering attacks, or emerging threats like zero-day vulnerabilities. By understanding and addressing these risks, the company can strengthen its defenses and mitigate potential vulnerabilities.

e) Modernization Requirements and Emerging Technologies:

As part of the security assessment, it is important to consider modernization requirements and keep pace with emerging technologies. This includes regularly updating and patching open-source libraries, adopting secure frameworks, and staying informed about evolving web application technologies. By embracing industry best practices, the company can enhance the security posture of its software application and proactively address emerging threats.

## Areas of Security

Architecture Review:

Conducting an architecture review is crucial to assess the design and structure of the software application. This evaluation helps identify potential security vulnerabilities that may arise from architectural flaws or weaknesses. By thoroughly examining the architecture, we can determine if secure distributed computing, client/server interactions, or secure composition of components have been appropriately implemented.

Code Review:

Performing a comprehensive code review is essential to identify vulnerabilities within the software application. This review should cover various code components, including models, controllers, data access, services, plug-ins, and APIs. By thoroughly examining the codebase, we can uncover potential security weaknesses related to error handling, secure coding practices, data structures, encryption use, and vulnerabilities.

Secure API Interactions:

Considering that Artemis Financials’ web application likely interacts with multiple APIs, it is imperative to assess the security aspects of these interactions. The evaluation should encompass analyzing the authentication mechanisms, data encryption, and validation of API responses to prevent potential security risks and vulnerabilities. Ensuring secure API usage is crucial to protect the integrity and confidentiality of data exchanged between the application and external systems.

Summary of Findings with Mitigation Plan:

A comprehensive vulnerability assessment should culminate in a concise summary of the findings and a well-defined mitigation plan. This summary will provide an overview of the identified vulnerabilities, categorize them based on severity, and outline specific steps to mitigate each vulnerability effectively. Prioritizing the vulnerabilities based on their potential impact on the application's security is vital for efficient risk management.

## Manual Review

a) Potential SQL Injection Vulnerability:

Within the DocData class, specifically in the read document method, there is a potential SQL injection vulnerability. The method accepts user-provided values for the key and value parameters, but the code does not implement proper input sanitization or parameterization before incorporating them into the SQL query. This can expose the application to SQL injection attacks, where malicious input can manipulate the query and potentially lead to unauthorized access or data manipulation.

b) Potential Connection Security Issue:

In the DocData class, during the establishment of a database connection using DriverManager.getConnection, there is a potential security issue related to the lack of specifying a secure connection. The code does not enforce the use of SSL/TLS protocols or other secure communication mechanisms when interacting with the database server. This omission can leave the communication channel vulnerable to eavesdropping or tampering, compromising the confidentiality and integrity of the transmitted data.

## Static Testing

Vulnerability: CVE-2020-25649

Description: Improper Restriction of XML External Entity Reference (XXE) in FasterXML Jackson Databind.

Recommended Solution: Upgrade to a version of FasterXML Jackson Databind beyond 2.10.5.1.

Attribution: Various references listed in the provided report.

Vulnerability: CVE-2020-36518

Description: Out-of-bounds Write vulnerability in FasterXML Jackson Databind.

Recommended Solution: Upgrade to a version of FasterXML Jackson Databind beyond the affected versions.

Attribution: Various references listed in the provided report.

Vulnerability: CVE-2021-46877

Description: Uncontrolled Resource Consumption (Resource Exhaustion) in FasterXML Jackson Databind.

Recommended Solution: Upgrade to a version of FasterXML Jackson Databind beyond the affected versions.

Attribution: Various references listed in the provided report.

Vulnerability: CVE-2022-42003

Description: Deserialization of Untrusted Data vulnerability in FasterXML Jackson Databind.

Recommended Solution: Upgrade to FasterXML Jackson Databind version 2.14.0-rc1 or later, or versions 2.13.4.1 and 2.12.17.1.

Attribution: Various references listed in the provided report.

Vulnerability: CVE-2022-42004

Description: Deserialization of Untrusted Data vulnerability in FasterXML Jackson Databind.

Recommended Solution: Upgrade to FasterXML Jackson Databind version 2.13.4 or later.

Attribution: Various references listed in the provided report.

Vulnerability: CVE-2023-35116

Description: Allocation of Resources Without Limits or Throttling vulnerability in FasterXML Jackson Databind.

Recommended Solution: The vendor does not consider this a vulnerability, but caution should be exercised when using untrusted input with FasterXML Jackson Databind.

Attribution: The provided reference in the report.

Here are the key details from the report:

Dependencies Scanned: 38 (22 unique)

Vulnerable Dependencies: 13

Vulnerabilities Found: 97

Vulnerabilities Suppressed: 0

NVD CVE Checked: Last checked on July 16, 2023, at 17:53:29 (UTC-7)

NVD CVE Modified: Last modified on July 16, 2023, at 17:00:01 (UTC-7)

dependency-check version: 8.3.1

VersionCheckOn: Last version check performed on July 9, 2023, at 10:07:15 (UTC-7)

kev.checked: A numerical value that may indicate a reference or identifier for the report

## Mitigation Plan

Strong Authentication and Authorization: Implement robust authentication and authorization mechanisms to ensure that only authorized users can access the database server. Utilize strong and unique credentials, enforce password policies, and consider implementing additional security measures like two-factor authentication.

Enforce Secure Database Connection: Update the code to enforce the use of secure communication protocols, such as SSL/TLS, when establishing the database connection. Configure the connection properties to utilize encryption and enable SSL/TLS certificates to ensure secure transmission of data between the application and the database server.