****

# 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** | **21/05/2023** | **Achna Hettiarachchi** |  |

## 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

Achna Hettiarachchi

## Interpreting Client Needs

Artemis Financial, a consulting company specializing in individualized financial plans, is seeking to modernize its operations by utilizing a web-based software application. Our software engineering company, Global Rain is tasked with conducting a vulnerability assessment of Artemis Financial's application to identify potential security vulnerabilities.

Secure communications are of the greatest importance to Artemis Financial. Maintaining the confidentiality and integrity of their client's data is important for a consulting company handling sensitive financial information. Secure communications ensure that sensitive information is protected from unauthorized access, interception, or corruption.

It is safe to assume that Artemis Financial is involved in international transactions. This introduces additional security considerations, as international transactions often involve cross-border data transfers, compliance with data protection regulations of multiple authorities, and possible exposure to international cyber threats.

There may be governmental restrictions and regulations that Artemis Financial must follow regarding secure communications. These regulations could comprise data protection laws, encryption standards, and compliance requirements unique to the countries where the company operates or where its clients are from.

Artemis Financial is probable to face a variety of external threats now and in the immediate future. These threats could contain unauthorized access, data breaches, phishing attacks, web application vulnerabilities, and denial of service (dos) attacks. When considering the modernization requirements for Artemis Financial's web-based software application, we must consider the role of open-source libraries and evolving web application technologies. Their application may rely on open-source libraries to boost its functionality and reduce development time. It is important to guarantee that these libraries are up to date, are employed securely to avoid the risk of misuse, and have no known vulnerabilities. Improving the application may force them to embrace evolving web application technologies that could add more security considerations. So, it is vital to stay up to date with the latest best practices and security guidelines for the chosen technologies to lessen likely vulnerabilities.

## Areas of Security

Based on the analysis conducted earlier, the following areas of security apply to Artemis Financial's web application.

1. Input validation is needed as the web application handles various forms of user input, such as financial data, personal information, and user preferences.
2. APIs are required since the web application depends on a RESTful web API to communicate with external systems, financial institutions, and other third-party services.
3. Cryptography must be adopted because the web application must deal with sensitive financial and personal data.
4. Client/Server is essential since the web application follows a client-server architecture, with clients networking with the server-side components to approach financial plans and perform transactions.
5. Code Error is applicable as the web application must implement appropriate logging and monitoring mechanisms to promptly detect and respond to security incidents.
6. Code Quality is useful to maintain code quality through secure coding practices and patterns and is crucial for reducing the risk of introducing security vulnerabilities in the code.

## Manual Review

Lack of proper input validation: The ‘CRUD’ method takes a ‘business\_name’ parameter through a request parameter but does not validate the input before using it.

Code error: The ‘read\_document’ method in ‘DocData.java’ directly concatenates the key and value parameters into a SQL query without proper parameter binding. So, if the key or value parameters contain malicious SQL code, an attacker could control the SQL query being executed by injecting their own SQL statements (SQL injection).

Code quality: In the ‘customer.java’ the ‘account\_number’ and ‘account\_balance’ variables in the ‘customer’ class have default access modifiers. So, they are accessible from other classes within the same package. This can possibly uncover sensitive customer information to unauthorized access.

## Static Testing

A screenshot of a computer

Description automatically generated with low confidence

|  |  |  |  |
| --- | --- | --- | --- |
| Dependency | Vulnerability | Description | Solution |
| bcprov-jdk15on-1.46.jar | cpe:2.3:a:bouncycastle:bouncy-castle-crypto-package:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:bouncy\_castle\_crypto\_package:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\* | The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7. | update to a newer version |
| hibernate-validator-6.0.18.Final.jar | cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\* | Hibernate's Bean Validation (JSR-380) reference implementation. | update to a newer version |
| jackson-databind-2.10.2.jar | cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:\*:\*:\*:\*:\*:\*:\* | General data-binding functionality for Jackson: works on core streaming API | update to a newer version |
| log4j-api-2.12.1.jar | cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\* | The Apache Log4j API | update it to a version that is not affected by any known security vulnerabilities |
| logback-core-1.2.3.jar | pe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\* | logback-core module | update it to a version that is not affected by any known security vulnerabilities |
| snakeyaml-1.25.jar | cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\* | YAML 1.1 parser and emitter for Java | update it to a version that is not affected by any known security vulnerabilities |
| spring-boot-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\* | Spring Boot | update to a newer version |
| spring-boot-autoconfigure-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:2.2.4:release:\*:\*:\*:\*:\*:\* | Spring Boot AutoConfigure | update to a newer version |
| spring-core-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\* | Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container | update to a newer version |
| spring-web-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\* | Spring Core | update to a newer version |
| spring-webmvc-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\* | Spring Web | update to a newer version |
| tomcat-embed-core-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* | Spring Web MVC | update to a newer version |
| tomcat-embed-websocket-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* | Core Tomcat implementation | update to a newer version |

## Mitigation Plan

Implement input validation for the 'business\_name' parameter in the 'CRUD' method. Validate the input to ensure it meets the expected format, length, and any other relevant constraints.

Refactor the 'read\_document' method in 'DocData.java' to use parameterized queries or prepared statements instead of directly concatenating the key and value parameters into the SQL query.

Update the 'account\_number' and 'account\_balance' variables in the 'customer.java' class to have appropriate access modifiers such as private or protected.

Keep all the JAR files and dependencies up to date by regularly updating them to the latest versions and applying security patches and updates provided by the respective projects.

Monitor security advisories and patches for all the libraries and frameworks used in the application to stay aware of any known vulnerabilities.