****

# 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** | **1/20/2023** | **CHAO GUAN** |  |

## 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

CHAO GUAN

## Interpreting Client Needs

1. What is the value of secure communications to the company?

As the client specializes in providing financial services to its customers, secure communication is one of the most important features of its web app. To be more specific, secure communication encompasses users' data, including account information, banking transactions, and personal information are secured when communicating to the server and database where these types of sensitive data are stored.

1. Does the company make any international transactions?

Since the web app is available online, it should support transactions made overseas. Limiting its customers to make transactions only domestically significantly reduces its business opportunities. However, supporting foreign transactions requires the company to be equipped with additional capabilities and resources. Thus, it is advised that our team consult with the client to see if foreign transactions should be considered in the will need the capability to provide financial services to international customers.

1. Are there governmental restrictions about secure communications to consider?

There are some governmental restrictions on secure communications in the finance industry. The Gramm-leach-Bliley Act( GLBA) requires finance companies to explain their financial products and how the company shares customers’ data. And the finance company will need to disclose how they are going to utilize the data they collect.

1. What external threats might be present now and in the immediate future?

The web app might experience threats from malicious hackers whose goal is to break through security protection and steal sensitive information to gain financial advantages. Another external threat might come from competitors to damage the reputation of this growing company. Those could be database attacks, server sabotage, or any other types of criminal acts to break down the services.

1. What are the modernization requirements that you must consider?

Open-source libraries and evolving web application technologies are very common in modern web application development. However, as they are open-source libraries, anyone has access to them. If any vulnerability is detected by a genius software engineer, the web app is vulnerable. Therefore, the web app must have the capability to receive updates from authorized sources.

## Areas of Security

Based upon the Vulnerability Assessment Process Flow Diagram, it is recommended that the following areas will need special attention regarding security protection.

* l Input validation

Input validation is a programming technique that allows only “properly formed data” to enter the software system. This finance web app needs a strong input validation mechanism to uphold the integrity of the data in the system.Input validation

Input validation is a programming technique that allows only “properly formed data” to enter the software system. This finance web app needs strong input validation mechnism to uphold the integrity of the data in the system.

* APIs

APIs are so commonly used in web application development and by nature expose the application’s logic and sensitive data. It becomes a common target for security attacks. Therefore, it is very important to enforce higher security standards in applying APIs in web app development.

* Cryptography

Cryptography is to adopt a special algorithm to encrypt and decrypt data so that information and communication are secured during the transition. All the information communicated on the Internet needs to be properly encrypted to uphold the confidentiality and integrity of both user information and system information.

* Code Errors

Properly handling code errors is another measure that prevents hackers to penetrate the system by finding any faulty feature of the web app. Therefore, there should be a policy to standardize how to handle errors, such as functional errors, syntax errors, logic errors, etc.

## Manual Review

After reviewing the java classes of the source codes, I found that there should be more comments to explain some logic of creating certain classes and methods. Helpful comments make it easier for peers to detect code errors and facilitate the testing process.

One of the eye-catching vulnerabilities that I found was in DocData.java where “root” is both the user name and the password. The password and username should be more complicated to increase the level of difficulty of unauthorized database visits.

The pom.xml shows that the version of the Spring Framework that the app is built upon is 2.2.4 whereas the most up-to-date version is 6.0.4. Not updating the framework to the current version pose great threats to the security of the application.

## Static Testing

The dependency check report is generated after running the dependency check on the source codes. According to the report, there are 13 vulnerable dependencies and 101 vulnerabilities found.

|  |  |
| --- | --- |
| **Vulnerable Dependencies** | **Description** |
| [bcprov-jdk15on-1.46.jar](" \l "l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7) | Java implementation of cryptographic algorithms |
| [hibernate-validator-6.0.18.Final.jar](" \l "l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492) | Hibernate's Bean Validation (JSR-380) reference implementation. |
| [jackson-databind-2.10.2.jar](" \l "l5_0528de95f198afafbcfb0c09d2e43b6e0ea663ec) | General data-binding functionality for Jackson: works on core streaming API |
| [log4j-api-2.12.1.jar](" \l "l10_a55e6d987f50a515c9260b0451b4fa217dc539cb) | The Apache Log4j API |
| [logback-core-1.2.3.jar](" \l "l12_864344400c3d4d92dfeb0a305dc87d953677c03c) | logback-core module |
| [snakeyaml-1.25.jar](" \l "l14_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421) | YAML 1.1 parser and emitter for Java |
| [spring-boot-2.2.4.RELEASE.jar](" \l "l15_225a4fd31156c254e3bb92adb42ee8c6de812714) | Spring Boot |
| [spring-boot-starter-web-2.2.4.RELEASE.jar](" \l "l16_ec75d01d212b5229c16d872fb127744c0ed46ed8) | Starter for building web, including RESTful, applications using Spring |
| [spring-core-5.2.3.RELEASE.jar](" \l "l17_3734223040040e8c3fecd5faa3ae8a1ed6da146b) | Spring Core |
| [spring-web-5.2.3.RELEASE.jar](" \l "l18_dd386a02e40b915ab400a3bf9f586d2dc4c0852c) | Spring Web |
| [spring-webmvc-5.2.3.RELEASE.jar](" \l "l19_745a62502023d2496b565b7fe102bb1ee229d6b7) | Spring Web MVC |
| [tomcat-embed-core-9.0.30.jar](" \l "l20_ad32909314fe2ba02cec036434c0addd19bcc580) | Core Tomcat implementation |
| [tomcat-embed-websocket-9.0.30.jar](" \l "l22_33157f6bc5bfd03380ebb5ac476db0600a04168d) | Core Tomcat implementation |

## Mitigation Plan

To resolve the vulnerability issue presented by the dependency check report, it is highly recommended to update the framework version to the newest one. The updated version provides patches to fix those dependency vulnerabilities. After that, standardized coding practice will be applied so that coding logic are expressed in a more presentable way with proper comments. Last but not least, a more complicated password combination, such as symbols, letters, and numbers, will be used for accessing the database.

Works Cited

Federal Trade Commission. “Gramm-Leach-Bliley Act.” *Federal Trade Commission*, www.ftc.gov/business-guidance/privacy-security/gramm-leach-bliley-act.

Manico, Jim, and August Detlefsen. *Iron-Clad Java Building Secure Web Applications*. New York, Ny Mcgraw-Hill Education, 2015.

OWASP. “OWASP Foundation, the Open Source Foundation for Application Security.” *Owasp.org*, owasp.org/.

“What Is API Security and Why Is It Important? | Indusface Blog.” *Indusface*, 23 Nov. 2021, www.indusface.com/blog/what-is-api-security-and-why-is-it-important/.