****

# 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

Jessica Duft

## Interpreting Client Needs

The company handles financial information; therefore, the value of secure communications is extremely in order to protect sensitive information and maximize profit. This can be confirmed by part of the company's mission statement which is "Security is everyone's responsibility".

The company does make international transactions, it supports government agencies around the world.

There are governmental restrictions regarding secure communications that the company should consider. The company must be sure that they meet all governmental regulations and guidelines regarding secure communication so as not to breach governmental guidelines and requirements and to avoid legal penalties.

External threats that are notable now and in the immediate future include but are not limited to: individuals with malicious intent attempting to manipulate vulnerabilities that could lead to data breaches, unauthorized access, or denial of service attacks.

It is important to consider modernizing requirements. This should include evaluating the role of open-source libraries and web technology. Using old libraries could make the system more vulnerable to attacks. It is important to stay up to date with frameworks and other dependencies and technologies in order to maintain a secure system.

## Areas of Security

Input validation- This area of security is relevant to the company because they will be getting input from the user at some points throughout the application. For example, prompting a username is gathering input. What happens if an individual types a username that is too long and triggers unauthorized access to a certain account. All input should be validated to prevent such attacks.

Cryptography- This area of security is relevant to the company because the company will be always handling sensitive data. All sensitive information should be encrypted and secure. Considering the company handles financial information this is especially important.

Code Error- This area of security is important so as not to accidentally leak sensitive information that could be manipulated by an individual with malicious intent. All errors should be written in a way that does not leak any sensitive data.

Code Quality- This area of security is important to the company because it ensures that the company uses secure coding practices and patterns. Using secure code will help prevent against unauthorized access to business code.

Encapsulation- This area of security is not only important, but it is also very beneficial to the company. Using encapsulation techniques, the business code will be easier to debug and easier to re-use. It will also ensure the security of data structures within the system.

## Manual Review

Input validation should be used when getting the username and password from the user.

The account information does not seem to be encrypted. The company needs to encrypt this sensitve information.

The user name and password exception in DocData() should include a secure connection to the database. The connection is opened but it is never closed. Some line such as con.close(); could be beneficial here. Along with limited access to the drivers dependent upon the account that is accessing said information.

Accessors methods still need to be implemented in the myDateTime class. Implementing accessors methods follows encapsulation techniques and would be secure, it just still needs to be implemented.

Spring framework in POMXL is outdated; It is 2.2.4 and most current version is 6.0.2.

Java version is outdated, we're using 1.8 and the most current version is Java 17.

## Static Testing

Dependency bcprov-jdk15on-1.46.jar has several vulnerabilities and has been considered unsafe and support for this has been removed the provider. The company would need to upgrade to a more recent and more secure version of this dependency.

Dependency spring-boot-2.2.4.RELEASE.jar has a potential for denial-of-service attacks to occur if Spring MVC is used with a reverse proxy cache. To avoid this do not use the two together.

Dependency logback-core-1.2.3.jar has a vulnerability that would allow certain authorized users to configure files loaded from LDAP servers. Someone with malicious intent could cause some serious damage if they exploited this vulnerability. We need to update to the most recent version of this dependency.

Dependency log4j-api-2.12.1.jar has a vulnerability that could allow an attacker to take remote control of the server, cause denial of service, access sensitive information, or remotely execute code. We need to update to a more secure version such as Log4j 2.20.0

Dependency snakeyaml-1.25.jar has a vulnerability that could allow an attacker to cause a denial-of-service attack due to vulnerabilities when parsing and due to the SnakeYaml's constructor class not restricting types. It's recommended to upgrade to version 2.0 or higher to better protect against risks. A second vulnerability stems from parsing user input that could cause denial of service statuses.

Dependency jackson-databind-2.10.2.jar has a vulnerability that could allow attackers to cause a denial of service via unusual circumstances, deeply nested objects and certain situations involving JsonNode JDK serialization. This same vulnerability could cause all attackers to perform external entity attacks which could compromise data integrity.

Dependency tomcat-embed-core-9.0.30.jar has a vulnerability that has many concerning potentialities for attacks like denial of service, insecure connections, unauthorized access, sensitive information leaks and more. It is recommended to upgrade to the most recent version to help protect against attacks. We need to upgrade to a more secure version such as version 10.1.10

Lastly, dependency hibernate-validator-6.0.18.Final.jar has a vulnerability that allows invalid EL expressions to be interpreted as if they were valid. The attacker would be able to use characters that are supposed to be invalid to potentially access personal data.

## Mitigation Plan

-We will need to add input validators for when the user inputs their username and password

-We will need to encrypt all sensitive information specifically in the DocData class that generates the account information and close the connection to the database by implementing a con.close() line in the code

-We will need to implement the accessor methods in the myDateTime class

-We will upgrade JDK15 to JDK17

-We will upgrade Spring boot from version 2.2.4 to version 2.5.0

-We will upgrade logback-core 1.2.3 to logback-core 1.2.9

-We will upgrade Log4j from version 2.12.1 to version 2.17.1

-We will upgrade snakeYAML from version 1.25 to version 2.0

-We will upgrade jackson databind from version 2.10.2 to version 2.15.0

-We will upgrade tomcat embed core from version 9.0.30 to 10.1.7

-We will upgrade hibernate-validator from version 6.0.18 to version 8.0

--We will upgrade the Spring Framework from version 5.2.3 to version 6.0.2

Following the mitigation plan and making the changes and upgrades mentioned above will create a more secure software for the company and it's users.