****

# 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** | **2/5/2024** | **Diana Diaz** |  |

## Client



**Developer**

Diana Diaz

## Interpreting Client Needs

Artemis Financial is a consulting company that offers tailored and individualized financial plans for its customers. Global Rain has been chosen by Artemis Financial to ensure they are using the most secure software to operate their business.

Global Rain and Artemis Financial interact with customers globally meaning that communication between servers and APIs most meet regulations of different countries and always remain secure for all users and purposes.

## Areas of Security

Based on the client needs discussed above, below are my initial areas of concern for potential attacks. These are the areas of security where I would suggest our team should focus:

Input Validation- Dealing with a global financial business as our client, all access to the business and client accounts will be completed by properly accessing desired information. Our team must ensure that we develop insightful and specific parameters to detect malicious login information. We must ensure Artemis Financials’ program only allows login and passwords that meet the requirements we establish. Any entries that do not meet the requirements should be considered a threat.

API- Artemis Financial is already using an API, RESTful web application programming interface. We must ensure that the connection between the API and users is secure and login information is accurately confirmed.

Cryptography- Servicing customers globally, it is extremely important for all data to always remain encrypted. Data should be shared securely during an entire transaction between a customer or user and Artemis Financial.

## Manual Review

After conducting a manual review of Artemis Financials’ current web-based software application, these are the areas of security our team should focus on:

API- As a given, using a spring application is an API. We should ensure that the interactions between users and the API are secure.

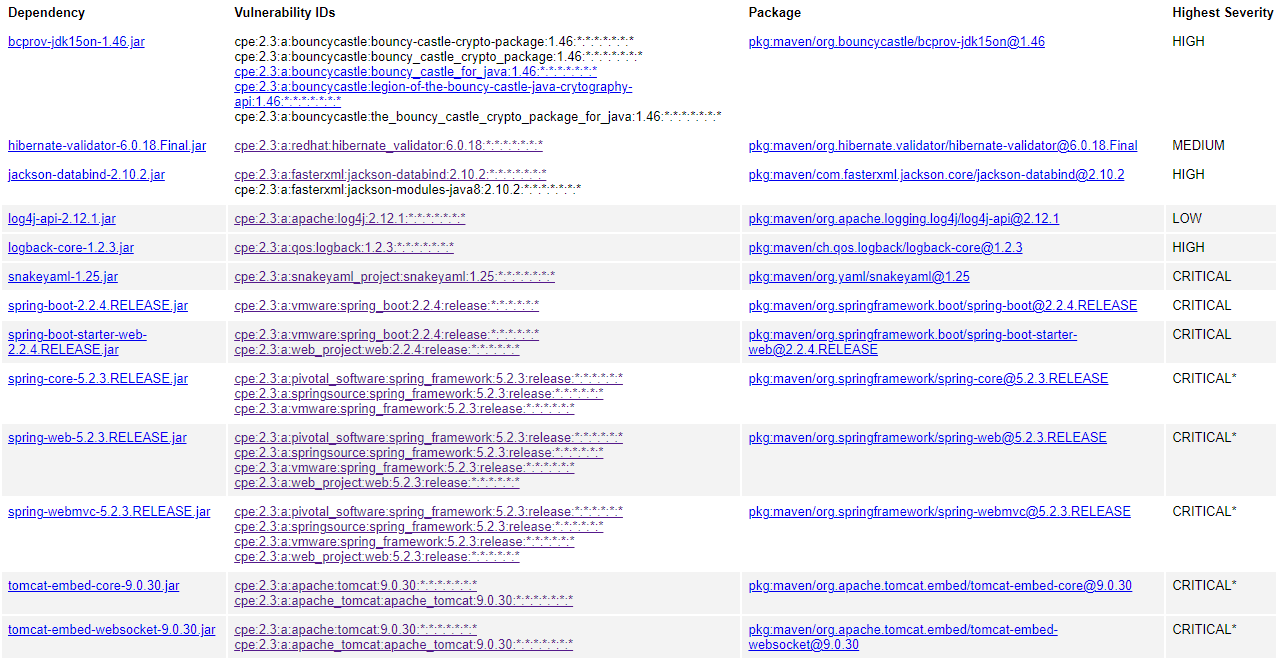
Cryptography- As data is being sent and received internationally, it should always remain encrypted to prevent threats in case data is accessed by an unauthorized user.

Client/Server- The API runs on the server; it is also accessed by users from a server and all information shared is transmitted through the API and server. For this reason, it is important to ensure that communication between clients and the server is secure.

## Static Testing

A screenshot of a computer

Description automatically generated



* “hibernate-validator-6.0.18.Final.jar” seems to pertain to input validation. According to the site nvd.nist.gov, this is a bug in the message interpolation processor that enables invalid EL expressions to be evaluated as valid. Furthermore, this allows attackers to bypass input sanitation that developers might have put in place. (<https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aredhat&cpe_product=cpe%3A%2F%3Aredhat%3Ahibernate_validator&cpe_version=cpe%3A%2F%3Aredhat%3Ahibernate_validator%3A6.0.18>)
* “spring-boot-2.2.4.RELEASE.jar” pertains to using outdated or older unsupported versions of spring boot versions. This directly affects the safety of the API in use.
* “spring-web-5.2.3.RELEASE.jar” also pertains to using outdated or older unsupported versions of spring boot versions. Specifically, “CPE Product: cpe:/:vmware:spring\_framework, CPE Product Version: cpe:/:vmware:spring\_framework:5.2.3” states this dependency makes an application vulnerable to denial of service (DoS) attacks.
* “tomcat-embed-core-9.0.30.jar” and “tomcat-embed-websocket-9.0.30.jar” both relate to improper input validation vulnerabilities which relate to the API.

## Mitigation Plan

Based on my interpretation of the dependencies identified, the current application that Artemis Financial has in place could benefit from stronger parameters for input validation. Our agile scrum team should include clear and specific parameters to only accept login information that meets our character requirements. If a password must of a certain length, include any digits, symbols and, or, upper and lower case letters, the user must enter a password that meets all those requirements. If a “user” inputs any entry that does not match those parameters, it could be a potential attack by attempting to enter a malicious code that could bypass authentic login information.

Also, a few of the dependencies stated an older, outdated, or unsupported versions of spring boot applications were being used and some suggested updating to “springboot to 3.0.6+. 2.7.x or 3.0.6+ or 2.7.11+”. Using the most recent or up-to-date versions of spring boot applications ensures that an application is running on the most recent version of the system, this also provides an additional layer of assurance that the application is a bit more secure.

Older or outdated versions are more susceptible to security vulnerabilities and attacks rather than newer or the most recent versions. A current version of a spring boot application and effective software security practices can ensure a program has lower chances of possibilities for attacks.

**Sources:**

National Vulnerability Database (NVD) <https://nvd.nist.gov/vuln/search>