****

# 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** | **11/12/2022** | **Brady Poucher** |  |

## Client



## Developer

Brady Poucher

## Interpreting Client Needs

Secure Communications mean everything to the company. Having security for their clients’ personal information is essential in earning trust and long-lasting relationships. With the company having business spanning the globe, the potential threats against information they store is constantly on the rise. Since most clients are either international, or not near enough to have in-personal contact with Artemis Financials, using a web application is the prime source of information traffic. Attacks on companies of this size and type are constantly under risk of being attacked via the internet. These attacks happen in the form of fake users, injection attacks, DDOS, web client authentication, and countless more. In order to combat these attacks, all security measures must be able to be constantly updated, changed, and added onto in order to give them consistent protection for the future. This, plus modernization in the form of software, consultants, libraries, and advanced coding will all fight potential threats.

## Areas of Security

 Encrypting data should be the primary focus to start. This way, any attacks that could happen whilst implementing other areas of security may be mitigated by the fact they won’t have the ability to read any of the data they may take.

Secondly, Code quality and error handling will come next. These two go hand-in-hand. If the two were implemented simultaneously, the code will be able to communicate to each other fast, safely, and be free of errors that could be exploited.

## Manual Review

I started by reviewing places where users would be inputting their personal information. This is primarily done in the greeting controller. There needs to be criteria set that a user must validate in order for their input to be accepted. This keeps inputs either similar enough to be paired with others to use as a datapoint, or different enough to provide security.

Secondly, I couldn’t find any data being sent to the POST method. The data was being sent elsewhere, which could be a potential place to be exploited.

## Static Testing

The names or vulnerability codes of the known vulnerabilities

A brief description and recommended solutions that are found in the dependency-check report

Attribution (if any) that documents how this vulnerability has been identified or how it was documented in the past

bcprov-jdk15on-1.46.jar

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.

CVE-2016-1000338

CVE-2016-1000342

CVE-2016-1000343

hibernate-validator-6.0.18.Final.jar

Hibernate's Bean Validation (JSR-380) reference implementation.

CVE-2020-10693

jackson-databind-2.10.2.jar

General data-binding functionality for Jackson: works on core streaming API

CVE-2022-42004

CVE-2022-42003

CVE-2020-36518

CVE-2020-25649

log4j-api-2.12.1.jar

The Apache Log4j API

CVE-2020-9488

logback-core-1.2.3.jar

logback-core module

CVE-2021-42550

snakeyaml-1.25.jar

YAML 1.1 parser and emitter for Java

CVE-2017-18640

CVE-2022-25857

spring-boot-2.2.4.RELEASE.jar

Spring Boot

CVE-2022-27772

spring-core-5.2.3.RELEASE.jar

Spring Core

CVE-2022-22965

CVE-2021-22118

spring-web-5.2.3.RELEASE.jar

Spring Web

CVE-2016-1000027

CVE-2022-22965

CVE-2022-22971

tomcat-embed-core-9.0.30.jar

Core Tomcat implementation

CVE-2020-1938

CVE-2020-11996

tomcat-embed-websocket-9.0.30.jar

Core Tomcat implementation

CVE-2020-1938

CVE-2022-29885

## Mitigation Plan

Interpret the results from the manual review and static testing report. Identify steps to mitigate the identified security vulnerabilities by creating an action list that documents how to fix each vulnerability in your vulnerability assessment report.

I would present a mitigation plan that ensures that all known vulnerabilities are addressed as fast as possible. Then, I would look for ways to have the code updated regularly based on continual research on cybercrime. After that I would have two factor authentication be a requirement for all users each time, they access the application as well as ensuring the system they are on is updated enough to stay in line with our continuous updating. This would ensure that all users are protected from their information being attacked by our encryption, code handling, error handling, two factor authentication, and many more features.