****

# 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** | **9-19-22** | **Kai Glass** |  |

## 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

Kai Glass

## Interpreting Client Needs

The value of secure communications in this company is very important because client/server security needs to be flawless for financial services. International transactions are quite likely being that Artemis financial is a financial company. The restriction that the law puts in place is that they have ECPA which means they have access to all cloud-based servers. External threats that are prevalent to financial companies are SQL injections, phishing attempts, and data leaks. I would include an open-source library but would add levels of security to keep it secure.

## Areas of Security

Input validation must be used to validate clients and users to use the website with a 2-factor authentication method. APIs also need to be used because it was stated in the scenario we were given. Client/Server needs to be focused on because the distribution between the client and the machine itself should be completely secure. Code quality should also be focused to ensure that all code is free of error and security breaches. And finally, encapsulation should be used to make sure all data structures are secure.

## Manual Review

Input validation must be used in CRUDController.Java for the parameters of CRUD. Input validation must be used in DocDatat.java to ensure just the document is being read in and no malware. The customer.Java file should be looked over for client/server and code quality vulnerabilities. Input validation should also be used in the GreetingController.java file.

## Static Testing

bcprov-jdk15on-1.46.jar

pkg:maven/org.bouncycastle/bcprov-jdk15on@1.46 (Confidence: High)

cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\* - The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains a JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7.

spring-boot-2.2.4.RELEASE.jar

pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE (Confidence: High)

cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\* (Confidence:Highest) suppress

cpe:2.3:a:vmware:spring\_framework:2.2.4:release:\*:\*:\*:\*:\*:\* - Spring Boot, upgrade to newest version

logback-core-1.2.3.jar

pkg:maven/ch.qos.logback/logback-core@1.2.3 (Confidence: High)

cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\* - logback-core module, upgrade to newest version.

log4j-api-2.12.1.jar

pkg:maven/org.apache.logging.log4j/log4j-api@2.12.1 (Confidence: High)

cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress – the Apache log4j API, upgrade to correct version

snakeyaml-1.25.jar

pkg:maven/org.yaml/snakeyaml@1.25 (Confidence: High)

cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress

cpe:2.3:a:yaml\_project:yaml:1.25:\*:\*:\*:\*:\*:\*:\* (Confidence: Highest) – YAML 1.1 parser and emitter for java, upgrade to a version with less security issues.

jackson-databind-2.10.2.jar

pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.10.2 (Confidence: High)

cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress - General data-binding functionality for Jackson: works on core streaming API

tomcat-embed-core-9.0.30.jar

pkg:maven/org.apache.tomcat.embed/tomcat-embed-core@9.0.30 (Confidence: High)

cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress

cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress - Core Tomcat implementation

spring-boot-starter-validation-2.2.4.RELEASE.jar

pkg:maven/org.springframework.boot/spring-boot-starter-validation@2.2.4.RELEASE (Confidence: High)

cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress - Starter for using Java Bean Validation with Hibernate Validator

hibernate-validator-6.0.18.Final.jar

pkg:maven/org.hibernate.validator/hibernate-validator@6.0.18.Final (Confidence: High)

cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress - Hibernate's Bean Validation (JSR-380) reference implementation.

spring-core-5.2.3.RELEASE.jar

pkg:maven/org.springframework/spring-core@5.2.3.RELEASE (Confidence: High)

cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress

cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress

cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\* (Confidence: Highest) suppress - Spring Core

## Mitigation Plan

Most of the issues that were prevalent in the static testing is that the version of plugins that we have are outdated or harmful version. With a little research into which versions have which security issues, we can update all versions for better security testing. Another thing we will need to look at is input validation for all function calls. We also need input validation for the parser that reads through the file. This validation needs to make sure that no harmful strings can be entered into the system or that no injections can be made. We also need to upgrade the cryptography we use because the version currently in access is not matched correctly for our application.