****

# 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

Dustin Smith

## Interpreting Client Needs

Due to Artemis being a financial company, security is a very important aspect of the company because it can become vulnerable to hackers. Customers will also value the fact that the company has good security so they do not lose their assets. Being a financial firm, transfers between different foreign regions are possible. This could lead to Artemis being exposed to different foreign laws. This could also mean that are applicable laws concerning secure communication. For example, in the USA the Electronic Communications Privacy Act (ECPA) allows the government “…access [to] digital communications such as email, social media messages, information on public cloud databases, and more with a subpoena.” There are also the CFAA, SOX, and Gramm-Leach-Bliley Act that may come into play. Outside council should be used to verify full compliance and applicability of any legal requirement. Artemis being an online presence bring risks that lead to different types of cybercrime. These could be not as harmful like a denial of service attack or as severe as a breach that will produce loss. A breach could lead to financial loss for customers which in return would lead Artemis to gaining a bad reputation which will lead to financial loss for the company. High level encryption should be applied to all databases, even though considered as not confidential, as just having small amount of PII might make a phishing attack more fruitful or expose Artemis to regulatory scrutiny. Artemis needs to be aware of open source code implication. This can bring overhead to certain open source licenses like copyleft licenses that force reciprocity. The openness could lead to Artemis being exposed to scrutiny and possible vulnerabilities if weakness was identified in either code.

## Areas of Security

Being a RESTful API I would focus on Input Validation, Secure API interactions, Code Errors, and Encapsulation. Input validation is important so we can “sanitize” the users inputs even if they are a trusted user. Secure API interactions will allow us to go one step farther to make sure that the connection and possibly the user are trustworthy. There should be some code checking built into Spring but it should be generic so it doesn’t have any signature aspects that a hacker could use to hack more into the system. Finally methods should use encapsulation to protect the sensitive data from any unwanted changes.

## Manual Review

As part of the static testing it was noticed the POM.xml needed to be updated to the most recent release of Maven. Using old static cases will cause some CVEs to be missed. Within the code some of the code has the get/set function but some do not. User input to the APIs are not sanitized. All the methods taking user input need to be sanitized. The customer class has the account\_balance as a public member. This should be set to private and have Get/Set methods used to control manipulation. The myDateTime class has a few poorly coded methods, e.g., setMyDateTime, as they are incomplete. The DocData class also has human readable username and password hardcoded! These should be obfuscated, even if only for testing, and much harder to guess than “root” “root”. Moreover, all passwords should be salted and hashed – none of that appears to be done here. Error checking and messaging needs to be implemented because only a few classes and methods have utilized implementation of error checking.

## Static Testing

|  |  |  |  |
| --- | --- | --- | --- |
| Dependency | Vulnerability Code | Description | Mitigation |
| [bcprov-jdk15on-1.46.jar](file:///C:\Users\BTH\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\rest-service\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7) | cpe:2.3:a:bouncycastle:bouncy-castle-crypto-package:1.46:  cpe:2.3:a:bouncycastle:bouncy\_castle\_crypto\_package:1.46  cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytographyapi:1.46  cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46 | The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms | Upgrade Bouncy castle to version 1.60 or higher |
| tomcat-embed-websocket-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30  cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30 | Core Tomcat implementation | Update Tomcat to the newest version |
| tomcat-embed-core-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30  cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30 | Core Tomcat implementation | Upgrade to newest Tomcat version |
| spring-core-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release  cpe:2.3:a:springsource:spring\_framework:5.2.3:release  cpe:2.3:a:vmware:spring\_framework:5.2.3:release  cpe:2.3:a:vmware:springsource\_spring\_framework:5.2.3:release | Spring Core | Upgrade to the latest version of Spring |
| spring-aop-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release  cpe:2.3:a:springsource:spring\_framework:5.2.3:release  cpe:2.3:a:vmware:spring\_framework:5.2.3:release | Spring AOP | Upgrade to the latest version of Spring |
| spring-boot-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release | Spring Boot | Only affects unsupported versions. Upgrade. |
| snakeyaml-1.25.jar | cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25 | YAML 1.1 parser and emitter for Java | Allows entity expansion. Upgrade. |
| jackson-databind-2.10.2.jar | cpe:2.3:a:fasterxml:jackson-databind:2.10.2  cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2 | General data-binding functionality for Jackson: works on core streaming API | Data integrity concerns. Upgrade to latest release. |
| logback-core-1.2.3.jar | cpe:2.3:a:qos:logback:1.2.3 | logback-core module | Arbitrary code execution. Upgrade to the latest version |
| hibernate-validator-6.0.18.Final.jar | cpe:2.3:a:redhat:hibernate\_validator:6.0.18 | Hibernate's Bean Validation (JSR-380) reference implementation | Input validation bypass issue. Upgrade to the latest version. |
| log4j-api-2.12.1.jar | cpe:2.3:a:apache:log4j:2.12.1 | The Apache Log4j API | Allows Man-in-the-middle attack on SMTP. Upgrade. |

## Mitigation Plan

First, all the dependent libraries need to be updated. This could lead to new code being written so this should be the first step. After the code recompiles and passes any internal tests the code needs to be refactored to add in the sanitized error messages, input validation/sanitization, and security improvements. Code review or peer programming should also be considered since there are large security issues with the code.