# Project One

# Aidan Gorospe

# CS 305 Software Security

# Professor Jack Lusby

# 3/22/2025

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **3/22/2025** | **Aidan Gorospe** |  |

## Client



**Developer**

Aidan Gorospe

**1. Interpreting Client Needs**

Due to the Artemis’s nature as a financial service company, security is an extremely high priority as malicious users would target this company as a means for illegal financial gain. This would mean customers would potentially lose trust in the company, their money, and of course, their privacy. And due to the nature of this business, foreign exchanges and transactions would be possible, meaning that any breaches of security or lack of security could fall under foreign law. In a similar fashion, the government could also access any digital communications that Artemis does with subpoenas, meaning that the measures taken for security are fully acceptable by the government and foreign governments’ standards.

As for current threats against Artemis, the previously mentioned nature of being a financial service company brings forth a lot of unsavory guests. These guests would likely be attempting crimes such as attempting to breach or DOS the databases. Breaches take the utmost priority, as a breach could result in potentially millions lost for the company and its users, with the above-mentioned losses in trust and reputation. As such, high level encryption is a must-have for all databases that Artemis uses. Lastly, keeping everything tight is key, as any open-source library used would have to be thoroughly maintained and updated, as any vulnerability found within that open-source library could be used to find vulnerabilities in the system that Artemis would be using.

**2. Areas of Security**

Due to the nature of the system being a RESTful API, a focus on Input Validation, API Interactions, Encapsulation and Code Errors would be necessary. Input Validation is needed to sanitize user inputs, as malformed inputs could occur regardless of intent, causing irregularities to form. Securing the API and making sure the interactions the API perform are secure are important as it allows an extra bit of security in making sure the connection with the user is trustworthy and safe.

Encapsulation allows the functions and methods we use in the code to be safely hidden and protected from any accidental or purposeful changes to our private variables. Lastly code would have to be checked and tested to find any potential vulnerabilities, as well as any errors found to be fixed and have no potential error messages to signal clues to malicious users on how to approach hacking the system.

**3. Manual Review**

1. From starting to test, I had to upgrade the POM.xml file to the latest version of maven, going from 5.1 to 12.1.0. It is important to update as this could cause some CVEs to be missed during the testing.
2. All the methods asking for user input do not have a way to check for length, as buffer overruns may occur.
3. Only some methods and classes have error checking implemented, it should be required for all classes and methods, while also not retaining too much information in the error message, as it could clue malicious users on how to exploit the system.
4. The GreetingController class does not use get/set functions, which does not mean it’s properly using encapsulation.
5. User Input is not sanitized and uses the raw variables input from the user. This could lead to injections.
6. The customer class contains a variable set as a public and not a private variable and should be encapsulated.
7. Passwords should be salted and hashed to prevent hackers from recovering original passwords.

**4. 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 latest version. |
| hibernate-validator-6.0.18.Final.jar | cpe:2.3:a:apache:log4j:2.12.1 | Hibernate's Bean Validation (JSR-380) reference implementation | Upgrade to latest version to prevent Input Validation bypass issue. |
| 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 | Upgrade to latest Version to prevent Data Integrity issues. |
| log4j-api-2.12.1.jar | cpe:2.3:a:apache:log4j:2.12.1 | The Apache Log4j API | Upgrade to latest version to avoid attack on SMTP. |
| spring-boot-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release | Spring Boot | Upgrade to the latest version. |
| 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 latest version. |
| snakeyaml-1.25.jar | cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25 | YAML 1.1 parser and emitter for Java | Upgrade to latest 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 latest version of Spring. |
| 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 | Trusts AJP connections more so than HTTP. Update to the latest version. |
| tomcat-embed-websocket-9.0.30.jar | cpe:2.3:a:redhat:hibernate\_validator:6.0.18 | Core Tomcat implementation | Update to latest version. |

**5. Mitigation Plan**

From what I’ve gathered, many of the libraries need to be updated, which could also lead to code being rewritten or new code altogether. Working on that step first would be good, as the current code still needs to be rewritten to account for input validation and better password protection techniques. This, alongside adding and fixing encapsulation for all of the classes and methods, as well as code reviewing the whole process would be recommended to solve these current issues.