# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **7/15/25** | **Skylar Walker** |  |

## Client



## Instructions

Submit this completed vulnerability assessment report. Replace the bracketed text with the relevant information. In this report, identify your security vulnerability findings and recommend 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 include images or supporting materials. If you include them, make certain to insert them in 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

Skylar Walker

**1. Interpreting Client Needs**

Determine your client’s needs and potential threats and attacks associated with the company’s application and software security requirements. Consider the following questions regarding how companies protect against external threats based on the scenario information:

* What is the value of secure communications to the company?
* Are there any international transactions that the company produces?
* Are there governmental restrictions on secure communications to consider?
* What external threats might be present now and in the immediate future?
* What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?

When handling people’s finances, being secure is a high priority since banking information includes a lot of user personal information. There is a financial privacy act form the Federal Trade Commission that requires companies to protect user information using safeguards. There are a lot of threats including cyberattacks as well as phishing attacks, and third-party attacks with their user’s information. Ensuring that classes are made final helps prevent subclasses from changing them leading to possible compromises in the security. Open-Source libraries can also have vulnerabilities built into them that have not been patched out yet. It is important to stay vigilant and up to date in any libraries that are being added to prepare our own patches for the vulnerabilities.

**2. Areas of Security**

Refer to the vulnerability assessment process flow diagram. Identify which areas of security apply to Artemis Financial’s software application. Justify your reasoning for why each area is relevant to the software application.

1. APIs: are a concern due to the possibility of their being vulnerability within their libraries.
2. Cryptography: it is important. Ensuring the information gets properly encoded to make it more challenging to cause harm to your users’ data is important when handling financials.
3. Client/Server: Having a secure connection between the client using your web application and the server they run on can help prevent major security risks from people trying to intercept the information coming from one destination to the other.
4. Encapsulation: The method of encapsulation keeps some code hidden. Having the ability to interact with certain steps the code takes is not needed by the whole program and should be kept to individual classes that need access.

**3. Manual Review**

Continue working through the vulnerability assessment process flow diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

1. **Account balance in the customer.jav should be private**
2. **Account number in the customer.jav should be final**
3. **No input validation in greetongController.java**
4. **There is no encoding of the user information.**
5. **System library using old java**
6. **Outdated dependency checks in pom.xml**

**4. Static Testing**

Run a dependency check on Artemis Financial’s software application to identify all security vulnerabilities in the code. Record the output from the dependency-check report. Include the following items:

* The names or vulnerability codes of the known vulnerabilities
* A brief description and recommended solutions provided by the dependency-check report
* Any attribution that documents how this vulnerability has been identified or documented previously

|  |  |  |
| --- | --- | --- |
| [tomcat-embed-websocket-9.0.30.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l26_33157f6bc5bfd03380ebb5ac476db0600a04168d) | Core Tomcat implementation | [cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Atomcat&cpe_version=cpe%3A%2F%3Aapache%3Atomcat%3A9.0.30) |
| [tomcat-embed-core-9.0.30.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l24_ad32909314fe2ba02cec036434c0addd19bcc580) | Core Tomcat implementation | [cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Atomcat&cpe_version=cpe%3A%2F%3Aapache%3Atomcat%3A9.0.30) |
| [spring-webmvc-5.2.3.RELEASE.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l23_745a62502023d2496b565b7fe102bb1ee229d6b7) | Spring Web MVC | [pkg:maven/org.springframework/spring-webmvc@5.2.3.RELEASE](https://ossindex.sonatype.org/component/pkg:maven/org.springframework/spring-webmvc@5.2.3.RELEASE?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [spring-web-5.2.3.RELEASE.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l22_dd386a02e40b915ab400a3bf9f586d2dc4c0852c) | Spring Web | [pkg:maven/org.springframework/spring-web@5.2.3.RELEASE](https://ossindex.sonatype.org/component/pkg:maven/org.springframework/spring-web@5.2.3.RELEASE?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [spring-expression-5.2.3.RELEASE.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l21_d0c6bb10758805b2153c589686b8045554bfac2d) | Spring Expression Language (SpEL) | [pkg:maven/org.springframework/spring-expression@5.2.3.RELEASE](https://ossindex.sonatype.org/component/pkg:maven/org.springframework/spring-expression@5.2.3.RELEASE?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [spring-core-5.2.3.RELEASE.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l20_3734223040040e8c3fecd5faa3ae8a1ed6da146b) | Spring Core | [pkg:maven/org.springframework/spring-core@5.2.3.RELEASE](https://ossindex.sonatype.org/component/pkg:maven/org.springframework/spring-core@5.2.3.RELEASE?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [spring-context-5.2.3.RELEASE.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l19_7750c95c96c7a1885c8b1b503ba915bc33ca579a) | Spring Context | [pkg:maven/org.springframework/spring-context@5.2.3.RELEASE](https://ossindex.sonatype.org/component/pkg:maven/org.springframework/spring-context@5.2.3.RELEASE?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [spring-boot-starter-web-2.2.4.RELEASE.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l18_ec75d01d212b5229c16d872fb127744c0ed46ed8) | Starter for building web, including RESTful, applications using Spring  MVC. | [pkg:maven/org.springframework.boot/spring-boot-starter-web@2.2.4.RELEASE](https://ossindex.sonatype.org/component/pkg:maven/org.springframework.boot/spring-boot-starter-web@2.2.4.RELEASE?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [spring-boot-2.2.4.RELEASE.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l17_225a4fd31156c254e3bb92adb42ee8c6de812714) | Spring Boot | [pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE](https://ossindex.sonatype.org/component/pkg:maven/org.springframework.boot/spring-boot@2.2.4.RELEASE?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [snakeyaml-1.25.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l16_8b6e01ef661d8378ae6dd7b511a7f2a33fae1421) | YAML 1.1 parser and emitter for Java | [pkg:maven/org.yaml/snakeyaml@1.25](https://ossindex.sonatype.org/component/pkg:maven/org.yaml/snakeyaml@1.25?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [jackson-core-2.10.2.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l5_73d4322a6bda684f676a2b5fe918361c4e5c7cca) | Core Jackson processing abstractions (aka Streaming API), implementation for JSON | [pkg:maven/com.fasterxml.jackson.core/jackson-core@2.10.2](https://ossindex.sonatype.org/component/pkg:maven/com.fasterxml.jackson.core/jackson-core@2.10.2?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [bcprov-jdk15on-1.46.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7) | 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. | [pkg:maven/org.bouncycastle/bcprov-jdk15on@1.46](https://ossindex.sonatype.org/component/pkg:maven/org.bouncycastle/bcprov-jdk15on@1.46?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [logback-core-1.2.3.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l14_864344400c3d4d92dfeb0a305dc87d953677c03c) | logback-core module | [pkg:maven/ch.qos.logback/logback-core@1.2.3](https://ossindex.sonatype.org/component/pkg:maven/ch.qos.logback/logback-core@1.2.3?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [logback-classic-1.2.3.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l13_7c4f3c474fb2c041d8028740440937705ebb473a) | logback-classic module | [pkg:maven/ch.qos.logback/logback-classic@1.2.3](https://ossindex.sonatype.org/component/pkg:maven/ch.qos.logback/logback-classic@1.2.3?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [jackson-databind-2.10.2.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l6_0528de95f198afafbcfb0c09d2e43b6e0ea663ec) | General data-binding functionality for Jackson: works on core streaming API | [pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.10.2](https://ossindex.sonatype.org/component/pkg:maven/com.fasterxml.jackson.core/jackson-databind@2.10.2?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [hibernate-validator-6.0.18.Final.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l3_7fd00bcd87e14b6ba66279282ef15efa30dd2492) | Hibernate's Bean Validation (JSR-380) reference implementation. | [pkg:maven/org.hibernate.validator/hibernate-validator@6.0.18.Final](https://ossindex.sonatype.org/component/pkg:maven/org.hibernate.validator/hibernate-validator@6.0.18.Final?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |
| [log4j-api-2.12.1.jar](file:///C:\Users\swalk\OneDrive\Documents\SNHU\Eclipse\rest-service\target\dependency-check-report.html#l11_a55e6d987f50a515c9260b0451b4fa217dc539cb) | The Apache Log4j API | [pkg:maven/org.apache.logging.log4j/log4j-api@2.12.1](https://ossindex.sonatype.org/component/pkg:maven/org.apache.logging.log4j/log4j-api@2.12.1?utm_source=dependency-check&utm_medium=integration&utm_content=12.1.0) |

**5. Mitigation Plan**

Interpret the results from the manual review and static testing report. Then identify the steps to mitigate the identified security vulnerabilities for Artemis Financial’s software application.

The first step would be to make it more secure by changing the java its run on to the latest version. Next will be upgrading the dependency check version to the most recent one. Then applying all necessary updates to ensure any vulnerabilities that are active in the old versions are updated out. Next would be to implement some form of input verification to ensure the system cannot be attacked looking for unpatched errors with inputs. I would also add some encryption to the information the users are giving to keep it protected from attackers.