# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **3/20/2025** | **Savannah Mattoon** |  |

## 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

Savannah Mattoon

**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?

Secure communications are valuable for Artemis Financial to protect sensitive customer data and maintain trust in its services. Without secure communication channels the company risks exposing financial information. This could lead to potential data breaches, legal issues, and reputational damage. Since Artemis Financial operates in the financial sector it is likely that the company will handle international transactions. These transactions would require secure data exchange across borders ensuring compliance with international regulations and protecting sensitive financial information. Artemis Financial must consider governmental restrictions on encryption and data transmission, especially in countries with laws regulating the use of secure communications such as export controls on encryption technologies. They face external threats such as cyberattacks, data breaches, phishing, and zero-day vulnerabilities all of which could compromise their web applications and sensitive financial data. They must ensure their modernization efforts account for secure API design, the proper management of open-source libraries to avoid vulnerabilities, and the integration of evolving web technologies to stay resilient against emerging threats.

**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.

The most important security areas are Input Validation, APIs, and Cryptography. Proper input validation is imperative to prevent attacks like injection attacks assuring that only safe and expected data is processed. Securing the API is critical to protect against unauthorized access and abuse especially given the sensitive financial data involved. Strong cryptographic measures are also important such as encryption for data are necessary to ensure the confidentiality and integrity of financial information.

**3. Manual Review**

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

* No input validation or sanitization for user inputs making the application vulnerable to injection attacks.
* Direct string concatenation in SQL queries leading to SQL injection vulnerabilities.
* No authentication or authorization for API endpoints allowing unauthorized access to sensitive data.
* Stack traces are printed in error handling, potentially exposing sensitive system information.
* Sensitive data is exposed without encryption or access control.
* No encryption for sensitive information in transit or at rest leaving data exposed.
* Lack of validation for financial operations such as deposit amounts, leading to potential abuse.

**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

Bouncy Castle (bcprov-jdk15on-1.46)

Recommendation: Upgrade to a newer version of Bouncy Castle (e.g., version 1.70 or later).

Hibernate Validator (hibernate-validator-6.0.18.Final)

Recommendation: Upgrade to Hibernate Validator version 6.0.20 or later.

Jackson Databind (jackson-databind-2.10.2)

Recommendation: Upgrade to Jackson Databind version 2.12.x or later.

Log4j API (log4j-api-2.12.1)

Recommendation: Upgrade to a newer Log4j version (e.g., version 2.13.3 or later).

Logback Classic (logback-classic-1.2.3)

Recommendation: Upgrade to Logback version 1.2.7 or later.

SnakeYAML (snakeyaml-1.25)

Recommendation: Upgrade to SnakeYAML version 1.28 or later.

Spring Boot (spring-boot-2.2.4.RELEASE)

Recommendation: Upgrade to Spring Boot version 2.3.5 or later.

Spring Core (spring-core-5.2.3.RELEASE)

Recommendation: Upgrade to Spring Core version 5.2.10 or later.

Tomcat Embed Core (tomcat-embed-core-9.0.30)

Recommendation: Upgrade to Apache Tomcat version 9.0.31 or later.

**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 manual review and static testing report for Artemis Financials’ software application identified several critical and high-severity vulnerabilities in various libraries. These vulnerabilities could lead to potential exploits such as remote code execution, data manipulation, or denial of service. To mitigate these issues, it is essential to upgrade the affected dependencies to their latest patched versions, implement proper input validation, and perform thorough testing. Regular security audits and timely patch management should be adopted to prevent similar vulnerabilities in the future.