# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **7/21/2024** | **Tyler Eaker** |  |

## 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

Tyler Eaker

**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?

Artemis Financial specializes in creating financial plans for individuals, covering areas such as savings, retirement, investments, and insurance. Secure communication is crucial for the company, given that it handles sensitive client information, including SSNs and tax details. While there is no explicit indication that Artemis Financial operates solely within the US, it is reasonable to assume that the company may engage in international transactions. One significant governmental restriction to consider is ensuring that trade secrets remain protected. Due to the critical need to safeguard all types of information, protecting client data from external threats is paramount, necessitating strong encryption measures. Additionally, Artemis Financial should prioritize regular maintenance checks to address bug fixes and security vulnerabilities, ensuring their systems remain secure and up-to-date.

**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.

* Input Validation -- Artemis Financial requires input validation in order to validate owner

of information. This would provide protection for users. The input validation would be

written as strings.

* Code Quality, Within code quality, you are able to take control of access of methods

based on the user. For example, a user would only have access to their information, and

not access to both another user's information or even access to the server, etc.

* APIs, The creation of an API would be necessary because it is running internally and

externally. It would allow for which data accessed is acceptable.

* Code Error, implementation of error handling creates an understanding of what areas of

the API would need to be fixed. This way Artemis Financial wouldn't have to worry about

their user information being exposed or accessed.

* Cryptography, implementing cryptography within Artemis Financial would be crucial,

because it would ensure that user information wouldn't be compromised from different

areas of the world considering there would be different currencies being involved.

**3. Manual Review**

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

After reviewing the code and conducting a Vulnerability Assessment, I analyzed both the POM.XML and the Greeting Controller. In the XML file, I checked for the presence of an Apache Validator. In the Greeting Controller, I noticed a lack of input validation, which is something to address in future updates. The code quality was acceptable overall, but error handling was missing. Regarding the API, there were several deficiencies, including a vulnerability that could expose user input since it was not processed through a POST method. Lastly, I looked for indications of cryptography but was unable to find any.

**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
* bcprov-jdk15on-1.46.jar - The Bouncy Castle Crypto package is for cryptographic algorithms. Vulnerable to Hash Collision
* spring-boot-2.2.4.RELEASE.jar- Bundles different classes written to accomplish the task the application is designed for. Vulnerability includes ClassLoader access.
* logback-core-1.2.3.jar - Core implementation of Logback. Dependable, common, quick, and accommodating logging framework.
* log4j-api-2.12.1.jar- Extremely configurable monitoring tool that focuses on performing and minimal garbage production. •snakeyaml-1.25.jar- Parser and emitter for Java
* jackson-databind-2.10.2.jar- Converts JSON to and from POJO
* tomcat-embed-core-9.0.30.jar- single Java web app with a Tomcat server
* hibernate-validator-6.0.18.Final.jar- Bean Validation 2.0 reference implementation
* spring-web-5.2.3.RELEASE.jar- Provides assimilation such as multipart file upload functionality
* spring-beans-5.2.3.RELEASE.jar- Provides configuration framework and functions to instantiate, configure, and assemble java objects
* spring-webmvc-5.2.3.RELEASE.jar- Contains Spring's MVC and REST services
* spring-context-5.2.3.RELEASE.jar- Provides access to configured objects like a registry
* spring-expression-5.2.3.RELEASE.jar- Powerful Expression Language for interrogating and manipulating an object graph at runtime

**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.

Upgrading the majority of these vulnerabilities to the current versions would increase security. Set up validation for user input at /greeting. Use the POST method for APIs instead of GET.