# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **09/20/2024** | **Nay-Quan Christopher** |  |

## 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

Nay-Quan Christopher

**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 provides financial strategies for their clients. These strategies may include savings, investments, insurance, and retirement preparation. The company manages sensitive information, so secure communication is essential. Sophisticated encryption protects private customer data from malicious individuals who might attempt to obtain it. It also guards against potential threats who would try to obtain confidential client data. Artemis Financial requires all employees to receive training on security protocols (Video and Audio surveillance, software), to keep all systems updated with security patches. The company should also utilize the latest encryption updates, incorporate two-step verification, and use https.**

**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.

I read the Vulnerability Flow Diagram and concluded that providing a secure environment for users is critical. These are important Key Factors that I have listed below.

**Code Quality: When safeguarding code, it is best to use checks and balances to guarantee that the code is professionally designed. This enables developers to double-check the application for problems and vulnerabilities. Also, organizing the code is much easier to follow. By following these guidelines, developers can help ensure that their code is secure.**

**Client/Server: Artemis Financial must have security measures in place to protect against any malicious attempts. The user will request information from their computer network, which will in turn request information from the Artemis Financial server.**

**Cryptography: Cryptography is required to protect user information, including name, DOB, Social Security, and credit/debit card numbers. Cryptography hides this information from unauthorized users.**

**API: Artemis Financial must have security measures in place to prevent malicious attempts. The user will request information from their computer network, which will then request information from the Artemis Financial servers. An API, or application programming interface, is a set of protocols that allow different software components to communicate with each other. They are used to connect programs and send and receive data.**

**3. Manual Review**

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

**A vulnerability assessment is an essential part of securing any application. It provides the basis for prioritizing vulnerabilities and taking action to mitigate them.**

**A thorough vulnerability assessment requires a combination of automated scanning and a manual code review. Automated scanning can detect a wide variety of vulnerabilities, including SQL injection, cross-site scripting, hardcoded database credentials, and unsecured API endpoints.**

**However, it is important to note that automated scanning cannot detect all vulnerabilities. A manual code review is essential for uncovering these undetected vulnerabilities. Improper error handling and missing input validation allow detailed error messages to expose sensitive information. This information is often sent unencrypted, which makes it even more dangerous.**

**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.

**So, these were the ones that stuck out to me when I conducted the dependency check.**

**CVE-2021-2218: A vulnerability in Spring Core allows remote attackers to pass random code via manipulated input.**

**CVE-2021-44228: Incorrect input validation allows remote programs to exploit a vulnerability in Apache Log4j.**

**CVE-2021-29425: Apache Commons IO has a vulnerability that may expose sensitive data.**

**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.

* Check for all the false positives to see if they are related to the code. Any weaknesses in the code must be fixed using secure coding methods.
* I would also advise improving the accessibility of the login and password to address the minor issues. The solution could be to encrypt the password and username with a combination of alphabetical letters, special characters, and digits.
* I suggest updating the Apache server to address the any vulnerabilities identified by the static test.
* The last step in preventing vulnerabilities from being exploited is to validate all certificates used by the application.