# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **03/23/25** | **Anthony Ellis** |  |

## 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

[Insert your name here.]

**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 is by nature a financial company. The biggest asset a financial company can have, however, is trust from its users. If you don’t trust a financial company with your money, therefore financial companies essentially must offer a large level of security for the customers. In the case of Artemis Financial the customers expect a level of security with their data, especially critical information such as account numbers.

Artemis Financial may have international transactions but at this time there doesn’t seem to be current implementation for international financial plans. This is important because international business plans could imply different tax rates and bank or credit institution regulations.

The primary government regulation that we must follow standards for at least withing the American market is the Graham-Leach-Bliley Act. This act is a federal law that requires that financial companies and institutions explain to their customers their information-sharing practices and requires that the customer’s sensitive data is well secured.

The threats that Artemis Financial could face seem to be primarily attacks to get sensitive customer data such as account numbers. This is a very real threat that could happen to the system now. Another issue of concern would be customer financial plans that could inherently lead to insider trading; therefore customer financial plans would need to be considered sensitive data.

**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 believe at least five of the seven areas of security apply to Artemis Financial’s software application. Starting with input validation, input validation is always a concern any time a user inputs information into the software. Input validation is even as simple as buffer limiters and can help in preventing attacks such as SQL injections and denial of service attacks.

APIs are also a large area of security especially for future development. Following secure API practices can create a level of security especially for the API interactions.

One of the more prevalent areas of vulnerability is the Cryptography or Encryption practices. This is a more prevalent area of vulnerability due to needing to comply with government regulations and it should be prevalent in how all the user’s data is handled.

The next major area of security is code error, which is secure error handling. This is important because if there are errors in the system they need to be handled correctly to prevent data breaches and denial of service attacks.

Finally, I believe a big area of importance would be Code Quality, which is essentially secure coding practices and patterns. Secure coding practices include areas such as input validation and error handling therefore it is also an area of security that needs attention.

**3. Manual Review**

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

Initially the maven dependency check plugin was not up to date, I took the liberty to update this plugin for better static testing. This is an imperative plugin to keep up to date as it is an integral testing tool to identify dependency vulnerabilities. This definitely is an item that would fall under API’s as an area of vulnerability.

There is very little documentation throughout the developer’s written code. This is a vulnerability because it can lead to miscommunication and errors in the software that can then be exploited. This does not follow industry coding practices and secure coding practices.

I could not Identify any type of Role Based Access Control through out the software therefore it implies that all users of the software have full access. This is giant security concern because the secretary could be using the software the IT department will be using and the customers will be using it. There should be a control system in place to prevent everyone from being able to see customers sensitive data.

In DocData.java line thirty-two you can find the password and username of the test database that could contain customer sensitive information. This does not follow secure coding practices.

The Doc Data object has no setter for id and it is a unset and unused variable that could lead to a possible security vulnerability and does not follow secure coding practices.

The Customer class has account balance as a public variable, this does not follow secure coding practices.

The date and time method is incomplete and is a vulnerability due to unsecure coding practices.

**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

A screenshot of a computer

AI-generated content may be incorrect.

For the vulnerabilities that were listed in the dependency check with the information using the CVE and the National Vulnerability Database I was able to come to the conclusion that most of the vulnerabilities could be remedied but updating that dependency.

**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.

Initially the main issue that is causing vulnerabilities is incompleteness. There is lack of documentation and incomplete methods and no direction as to what is next and therefor could be an error or an unfinished part of the software.

I also think that all the dependencies need to be updated to the latest versions to remedy most of the dependency vulnerabilities.