# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **11/19/2024** | **Brian Dell Blackman** |  |

## 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

Brian Dell Blackman

**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?

[Include your findings here.]

**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.

Artemis Financial will be dealing with customer’s sensitive data. Such as social security numbers, Names, Addresses, telephone numbers, and email addresses. One Of the areas of security to focus on is Cryptography. In Iron-Clad Java it states that “When handling sensitive data, you need to encrypt that data both in transit and at rest.” They also focus on authentication and authorization, which when dealing with any web-based app or software in general, you must have a plan for authentication and authorization. Looking at the Vulnerability Assessment Process Flow, Input validation is another area of security to focus on. “Input validation should seek to define what constitutes good data and reject everything else.”(Manico et al., 2014) API is another focus when looking for areas of security. Ensure that monitoring of security events, update of API version, and API traffic. “API security is an essential aspect of modern application development, and by following best practices and implementing comprehensive security strategies, organizations can effectively protect their APIs and sensitive data.”(Moes, 2024) To finish of the areas of security, Client and Sever request, Code Error, Code Quality and Encapsulation can not be forgotten. Ensuring that the correct request is being made to the client and sever are very important and the financial industry. A mistake here can cost billions of dollars. Code Error, Code Quality and Encapsulation must be used with best practices you don’t want faulty code ruining your application, it could take a lot of time and money to correct bugs in your code.

**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 screenshot of a computer program

Description automatically generated

In the manual review I have found that in the DocData.java that it is receiving in data from SQL. As Mentioned in areas of security input validation is important to focus on. With the DocData.java class, it needs to verify the data that is being received and that data also needs to be encrypted. The DocData.java class can use some form of error messaging to notify of bad data.

A screen shot of a computer program

Description automatically generated

In the customer.java class, it deals with the account number and the account balance. This is sensitive information and should be handled with care. This can be done by adding a verification step to the showInfo() method by asking for password again. The deposit() method should output a message of the increment amount to deposit (ex. Only deposit bills) this way the user is not entering decimal amounts. The deposit() method also need to verify with the user the amount they want to deposit.

A screen shot of a computer program

Description automatically generated

The Greeting.java class should verify that the character among for “id” is not exceeded, Content should also have a set character amount.

A computer screen shot of text

Description automatically generated

The finding the I have come up with in the myDateTime.java class is that it is retrieving the date and time, it also set the date and time. This should have a format validation within this class to ensure that it is output correctly.

A screen shot of a computer program

Description automatically generated

**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 phone

Description automatically generated

A screenshot of a computer

Description automatically generated

**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.

I suggest for fixing DocData.java, customer.java, Greeting.java and myDateTime.java

* verify and encrypt the data
* add error messages
* adding a verification step
* output a message for increment amount to deposit
* verify the deposit amount
* ID and content to have a character restriction
* Date format validation

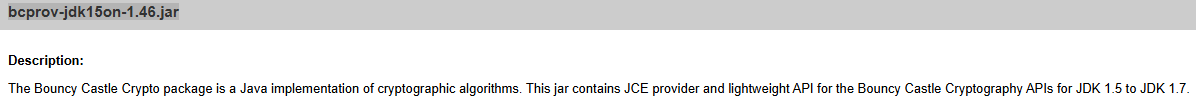
For jar vulnerabilities I suggest using mvnrepository.com to make sure the latest versions are being used.

A screenshot of a computer

Description automatically generated

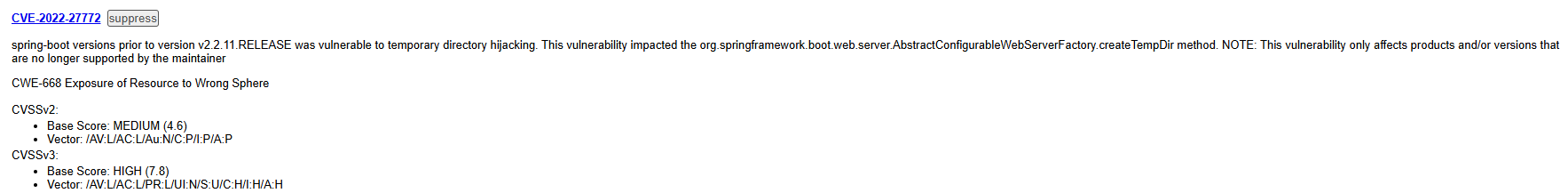
A computer screen shot of a computer screen

Description automatically generated

A screenshot of a computer

Description automatically generated





A screenshot of a computer

Description automatically generated

**Reference:**

Manico, J., & Detlefsen, A. (2015). Iron-clad Java : building secure web applications. Mc Graw Hill Education. <https://learning.oreilly.com/library/view/iron-clad-java/9780071835886/?sso_link=yes&sso_link_from=SNHU>

Moes, T. (2024, January 15). What is API security? everything you need to know. SoftwareLab. https://softwarelab.org/blog/what-is-api-security/