# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **11/17/24** | **Aaron Rodriguez** |  |

## 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

Aaron Rodriguez

**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?
  1. Secure communications is very important. It helps to ensure that the communications are secure and confidential.
* Are there any international transactions that the company produces?
  1. I don’t believe it is stated, but if there is any international transactions, it’s important to make sure you are compliant with the data and regulations of said transactions
* Are there governmental restrictions on secure communications to consider?
  1. There may be restrictions on communications such as finance or healthcare, such as HIPPA laws
* What external threats might be present now and in the immediate future?
  1. Threats present now and in the future will be malicious hackers trying to get the data. An other thing that can prove to be a threat would be social engineering
* What modernization requirements must be considered, such as the role of open-source libraries and evolving web application technologies?
  1. If you use open-source libraries, it’s important to consider using up-to-date ones because if you don’t, old ones will often have vulnerabilities

**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, APIs, Code quality, Code Error, Cryptography

Input validation is needed in this instance because you have to, well validate the input to provide extra protection.

APIs will be needed because it is running internally and externally and would allow for data to be accessed when needed.

Code quality is important because poorly written code can introduce vulnerabilities that would otherwise not be present

Code error. Error handling is important because if you don’t handle any errors that you have they can prove to introduce vulnerabilities.

Cryptography is important because it would help to ensure that information wouldn’t be vulnerable to bad acting characters.

**3. Manual Review**

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

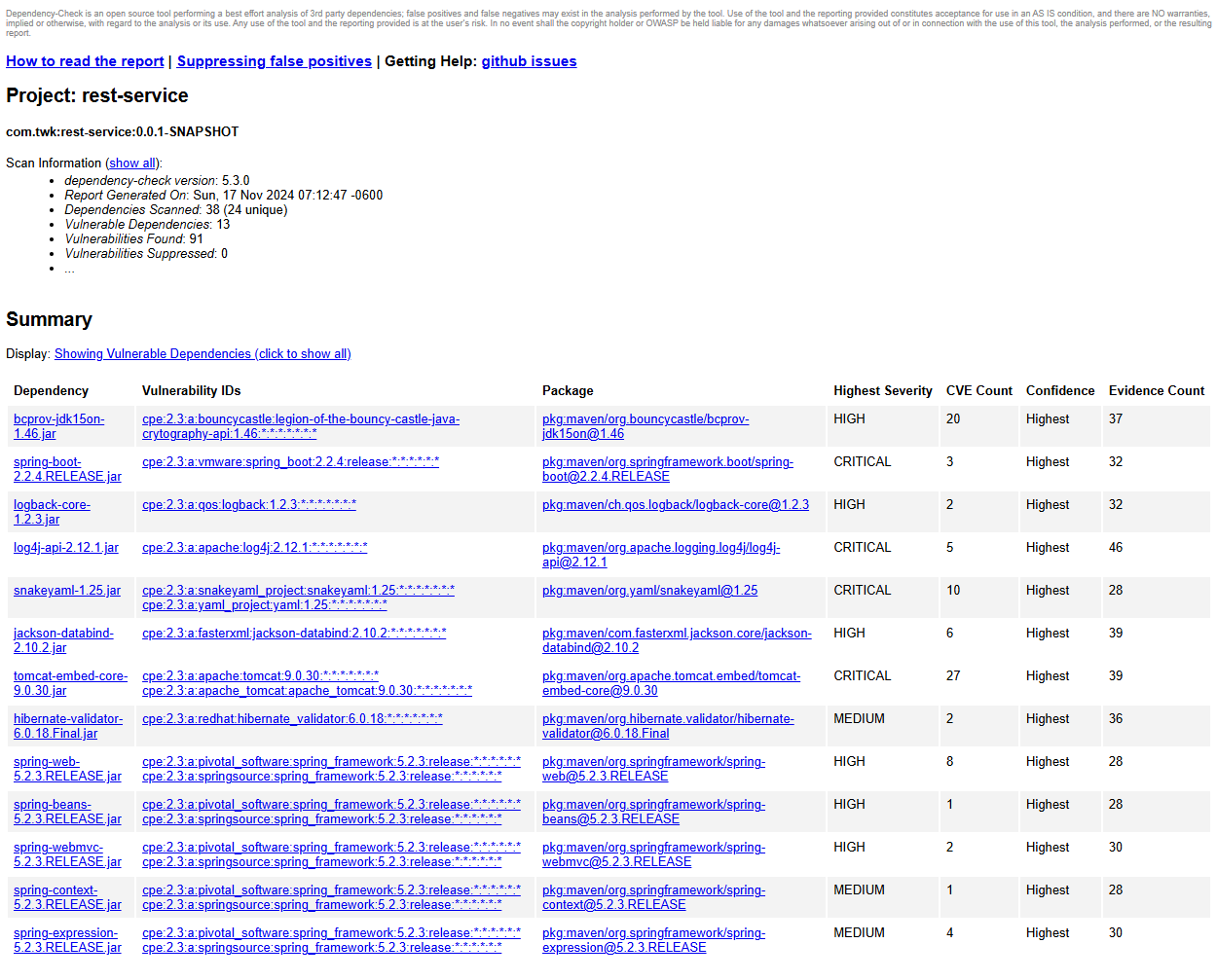
Findings:

1. In the CRUDController class, there is no input validation. It simply forces it to convert to a string which isn’t what it should do.
2. In DocData class, it is not a secure connection as it is a localhost connection
3. GreetingController class has no input validation
4. Overall code quality was not good, there was a lot of inconsistency in terms of naming conventions and code style
5. The API wasn’t utilized great because in the DocData class, it’s incomplete
6. I didn’t see any use of cryptography through the program
7. In DocData class, there is error handling but it only prints the error, doesn’t try to do anything to correct said error

**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



**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.

There are several overall steps that need to be taken. For one, taking into consideration this dependency test and vulnerability test. Update things that need updating, as well as fixing any vulnerabilities listed. Fixing the possible SQL injection vulnerability within the DocData class is of upmost importance. You also need to use a secure communication protocol. Use consistent coding practices and naming conventions throughout the project as well.