# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **9/16/2024** | **Jonathan Marvin** |  |

## 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

Jonathan Marvin

**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 a consulting company that creates financial plans for clients, covering such things as savings, retirement, investments, and insurance. Because they handle sensitive information, they need secure communications to and from clients. There are also governmental regulations in place for financial data and transactions that need to be met before anything else can happen.

The Artemis Financial RESTful API may be at risk if they have not structured the security correctly. HTTPS is a must when handling and communicating confidential data, and headers should not include any sensitive information. There should also be a secure authentication scheme in place to keep unwanted and malicious users out.

**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 is needed for this kind of API. Since RESTful APIs ingest user input, it will have to be validated to ensure only authorized human traffic is passing. The RESTful API will also need to communicate across the network securely. That leads us to addressing code errors. User errors must be handled securely so as not to put them at risk when using the application.

**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 base, I noticed a few things that should be fixed. First, there were no sources of error handling. Next, I noticed that the CRUDController class requests specific business names as a parameter. Request parameters are not being validated in this code base. The DocData class has connection parameters manually coded. The service should be using HTTPS, but uses HTTP instead.

**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*

Bouncy Castle 1.46 is an out of date package. The following is a list of vulnerabilities that could be fixed with an update:

* CVE-2016-1000338
* CVE-2016-1000342
* CVE-2016-1000343
* CVE-2016-1000344
* CVE-2016-1000352
* CVE-2016-1000341
* CVE-2016-1000345
* CVE-2017-13098
* CVE-2020-15522
* CVE-2020-0187 (OSSINDEX)
* CVE-2023-33202
* CVE-2020-26939 (OSSINDEX)
* CVE-2023-33201 (OSSINDEX)
* CVE-2016-1000339
* CVE-2015-7940
* CVE-2018-5382
* CVE-2013-1624
* CVE-2016-1000346
* CVE-2015-6644 (OSSINDEX)

Jackson-databind-2.10.2 has a number of vulnerabilities. Again, an update could resolve these:

* CVE-2020-25649
* CVE-2020-36518
* CVE-2021-46877
* CVE-2022-42003
* CVE-2022-42004
* CVE-2023-35116

Apache Log4j 1.12.1 should be updated as well based on CVE-2020-9488.

SnakeYAML has a vulnerability CVE-2022-1471 that recommends upgrading to version 2.0 or higher.

Spring Core 5.2.3 has a known and exploited vulnerability CVE-2022-22965. Vendor recommends updating.

Apache Tomcat 9.0.30 has many vulnerabilities that could be fixed with an update:

* CVE-2020-1938
* CVE-2020-11996
* CVE-2020-13934
* CVE-2020-13935
* CVE-2021-25122
* CVE-2021-41079
* CVE-2022-29885
* CVE-2022-42252
* CVE-2023-44487
* CVE-2023-46589
* CVE-2021-25329
* CVE-2021-30640

**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.*

The main mitigation for these vulnerabilities is updating the dependencies listed above. Switching to HTTPS rather than HTTP can also make the application more secure. The database should not be hard coded into the connection. A secure authentication scheme should be implemented as well.