# CS 305 Project One

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **1/24/2025** | **Jared Bickler** |  |

## 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

Jared Bickler

**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 specializes in creating financial plans for its clients. Due to the sensitive nature of this information, all communications between the company and its clients must be secure. Additionally, the company must comply with government regulations, which influences their data retention policies and security measures. RESTful APIs are susceptible to data interception if requests and responses are not securely designed. To mitigate this, the service must use HTTPS for all communications and ensure confidential information is transmitted securely. A secure authentication method, such as OAuth, should be implemented.

**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.

* APIs: The web service contains a RESTful API which needs to communicate securely.
* Input validation: Since the RESTful API accepts user input, it will need to be validated.
* Code error: Errors due to erroneous user inputs need to be handled securely.

**3. Manual Review**

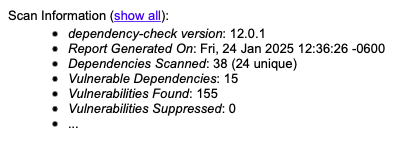
Continue working through the vulnerability assessment process flow diagram. Identify all vulnerabilities in the code base by manually inspecting the code.

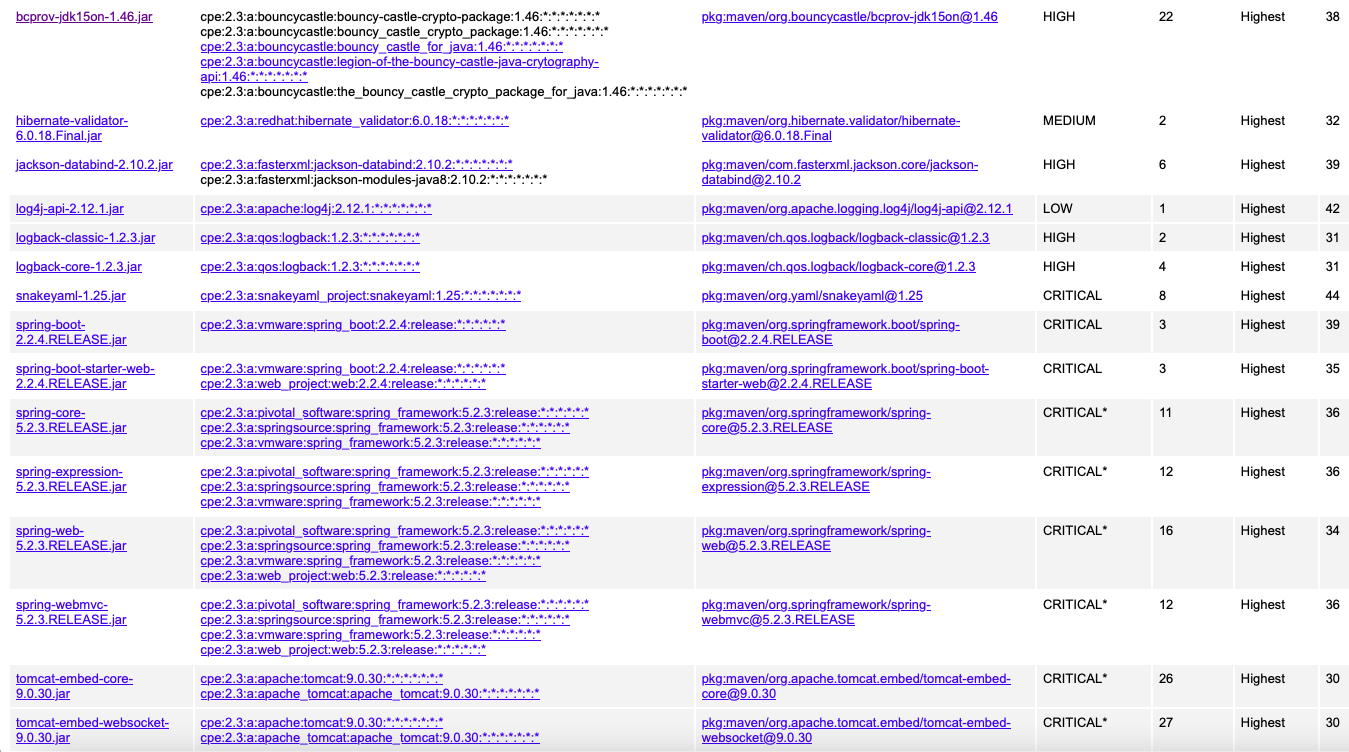
* Database connection parameters are hardcoded in DocData.java
* No validation is performed on request parameters
* Request parameters are used for sending business names in CRUDController.java
* There is no authentication scheme
* HTTPS is not used

**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





* Bouncycastle: Several vulnerabilities, recommend updating to at least version 1.60.
* Hibernate: Two vulnerabilities, recommend updating to latest version.
* Jackson: Several vulnerabilities, recommend updating.
* Apache: One vulnerability, recommend updating.
* Logback-classic: Two vulnerabilities, recommend updating to at least version 1.2.8.
* YAML: Several vulnerabilities, recommend using SnakeYaml’s SafeConstructor when parsing untrusted content to restrict deserialization. Recommend upgrading to at least version 2.0.
* Spring Boot: Several vulnerabilities, recommend upgrading to version 3.0.6+ or 2.7.11+.
* Tomcat: Many vulnerabilities, recommend upgrading to Tomcat 9.0.31, 8.5.51, or 7.0.100 or later.

**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.

* Switch to HTTPS for all communications.
* Move request parameters to headers or body.
* Remove hardcoded database connection credentials.
* Implement a secure authentication scheme.
* Update dependencies listed previously.