# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **03/23/25** | **Laurel Horwath** |  |

## 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

Laurel Horwath

**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 devises various financial plans for their customers. These plans include savings, retirement, investment, and insurance plans. This type of information makes secure communications absolutely crucial, as the type of information needed for these types of plans can include anything from an SSN to all of their banking and tax information. There is no indication the company is only US-based; therefore, it is logical to assume the company handles international transactions. The only prevalent governmental restriction to consider would be the protection of trade secrets. Because the need for completely secure communication is such a high priority, the main external threat present is anyone looking to expose or steal client information. There are a few modernized requirements the company should consider. These requirements include regular maintenance checks on bugs and security 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- Clients will need to verify their identity when accessing account information
* Code Quality- Code quality is relevant for any and all applications as ensuring quality code from the beginning will help prevent costly revisions
* APIs- This will be a vital part of structuring access to internal and external information
* Code Error- Error handling will help us pinpoint the areas of security to be fixed, further helping to protect client data
* Cryptography- Encrypting information will be a vital component to protecting customer data and communications

**3. Manual Review**

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

* Greeting controller doesn’t have an input validator
* There is no API and the system accesses data through a URL
* There is no data encryption present
* DocData.java class has no error handling
* Business names are sent as request parameters in the CRUDController class
* Database connection parameters are hard coded in DocData

**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

|  |  |  |
| --- | --- | --- |
| **Vulnerability Name** | **description** | **solution** |
| **bcprov-jdk15on-1.46.jar** | The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. This jar contains JCE provider and lightweight API for the Bouncy Castle Cryptography APIs for JDK 1.5 to JDK 1.7. | Update to version 1.70 |
| **hibernate-validator-6.0.18.Final.jar** | Hibernate's Bean Validation (JSR-380) reference implementation. | Update to version 8.0.2.Final |
| **jackson-databind-2.10.2.jar** | General data-binding functionality for Jackson: works on core streaming API | Update to version 2.18.3 |
| **log4j-api-2.12.1.jar** | The Apache Log4j API | Update to version 2.24.3 |
| **logback-classic-1.2.3.jar** | logback-classic module | Update to 1.5.18 |
| **logback-core-1.2.3.jar** | logback-core module | Update to version 1.5.18 |
| **snakeyaml-1.25.jar** | YAML 1.1 parser and emitter for Java | Update to version 2.4 |
| **spring-boot-2.2.4.RELEASE.jar** | Spring Boot | Update to version 3.4.4 |
| **spring-boot-starter-web-2.2.4.RELEASE.jar** | Starter for building web, including RESTful, applications using Spring  MVC. Uses Tomcat as the default embedded container | Update to version 3.4.4 |
| **spring-core-5.2.3.RELEASE.jar** | Spring Core | Update to version 5.2.25.RELEASE |
| **spring-expression-5.2.3.RELEASE.jar** | Spring Expression Language (SpEL) | Update to version 5.2.25.RELEASE |
| **spring-web-5.2.3.RELEASE.jar** | Spring Web | Update to version 5.2.25.RELEASE |
| **spring-webmvc-5.2.3.RELEASE.jar** | Spring Web MVC | Update to version 5.2.25.RELEASE |
| **tomcat-embed-core-9.0.30.jar** | Core Tomcat implementation | Update to version 9.0.99 |
| **tomcat-embed-websocket-9.0.30.jar** | Core Tomcat implementation | Update to version 9.0.102 |

**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.

For the majority of these vulnerabilities the main fix will just be updating to the current versions released for each.