# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **11/12/2024** | **Brad Farfaglia** |  |

## 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

Brad Farfaglia

**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?

Secure communication is important to any business but this is especially important when a business is handling its clients’ money. Clients need to be confident that we can be trusted with their money and their sensitive information, and that it won’t end up in the wrong hands. There is no indication that Artemis Financial is only based in one country, so it should be assumed that they operate internationally. Governments do have restrictions on secure communications when it comes to matters of national security and they’re also allowed to wiretap communications with a warrant, but I don’t think those situations will be applicable to us. Due to the nature of this business, the main external threats we need to safeguard against are attacks targeting clients’ information such as SQL injection. As the software world faces more security attacks than ever, programmers will be sending out solutions in the form of updates. All libraries and applications should be updated regularly to make sure that they are up to date on the latest security measures.

**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- Each user will need to be authenticated to verify that they are who they say they are before getting access to sensitive information.
* APIs- APIs will need to be secured to make sure that communications are valid before allowing them to go through.
* Cryptography- Communications will need to be encrypted to prevent sensitive information from being compromised when it is sent out.
* Client/Server- Server needs to be secure to prevent unauthorized access to the network. Access controls and firewalls, plus regular updates will help keep it secure.
* Code Error- Implementing error handling would automatically show which areas of the API need to be improved for better protection.
* Code Quality- Using secure coding practices such as parameterization can mitigate or eliminate common security threats.
* Encapsulation- Encapsulation limits direct access to an objects components. It can hide data members and functions and conceal implementation details to protect data transmission.

**3. Manual Review**

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

* API does not seem to be working.
* HTTPS is not being used.
* No methods for encryption found anywhere.
* Error handling seems to be missing.
* GreetingController missing input validation.
* In CRUDController, business names are sent as request parameters.
* Data is being input using the URL instead of the POST method.

**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

bcprov-jdk15on-1.46.jar

|  |  |  |
| --- | --- | --- |
| **Dependency** | **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 Bouncy Castle to the latest version. |
| hibernate-validator-6.0.18.Final.jar | Hibernate's Bean Validation (JSR-380) reference implementation. | A flaw was found in hibernate-validator's 'isValid' method in the org.hibernate.validator.internal.constraintvalidators.hv.SafeHtmlValidator class, which can be bypassed by omitting the tag ending in a less-than character. Browsers may render an invalid html, allowing HTML injection or Cross-Site-Scripting (XSS) attacks. |
| jackson-databind-2.10.2.jar | General data-binding functionality for Jackson: works on core streaming API | Update Jackson-databind to latest version. |
| log4j-api-2.12.1.jar | The Apache Log4j API | Update Apache log4j to latest version. |
| logback-core-1.2.3.jar | logback-core module | Update logback-core to latest version. |
| snakeyaml-1.25.jar | YAML 1.1 parser and emitter for Java | Use SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization or upgrade to version 2.0 and beyond. |
| spring-boot-2.2.4.RELEASE.jar | Spring Boot | Update Spring Boot to latest version. |
| 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 Spring Boot to latest version. |
| spring-core-5.2.3.RELEASE.jar | Spring Core | Update Spring Core to latest version. |
| spring-expression-5.2.3.RELEASE.jar | Spring Expression Language (SpEL) | Update Spring Framework to latest version. |
| spring-web-5.2.3.RELEASE.jar | Spring Web | Do not use for deserialization of untrusted data. |
| spring-webmvc-5.2.3.RELEASE.jar | Spring Web MVC | Use Spring Security HTTP Firewall https://docs.spring.io/spring-security/reference/servlet/exploits/firewall.html |
| tomcat-embed-core-9.0.30.jar | Core Tomcat implementation | Update Apache Tomcat to latest version. |
| tomcat-embed-websocket-9.0.30.jar | Core Tomcat implementation | Update Apache Tomcat to latest version |

**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.

* Update all apps and libraries to their latest versions.
* Change input method to POST instead of URL.
* Use HTTPS for more secure communications.
* Use encryption when sending sensitive information.
* Utilize error handling for more efficient error identification.
* Implement secure input validation.