# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **12/22/2024** | **THOMAS SEIBERT** |  |

## 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

THOMAS SEIBERT

**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 client that first and foremost requires absolute security in its communications. This security must be present throughout their operations, not only in handling sensitive financial data, but also within the administrative levels of the company itself and in interactions with their customers. The value of secure communication should be practically applied with a few principles in mind:

\*confidentiality (ensuring that only authorized parties can access the data)

\*integrity (ensuring that the data is not altered during transmission)

\*availability (ensuring that the data is available when needed)

Yes, the client makes international transactions and must comply with various international data protection and privacy laws. This includes GDPR for European Union, CCPA for California, USA, and so on. It also increases the risk of cross-border data breaches.

Yes, there are governmental restrictions about secure communications which need to be considered and complied with. They can vary by country. Some countries have strict data sovereignty laws that require data to be stored within the country. Others may have restrictions on the use of certain encryption technologies.

Some of the external threats which might be present now and, in the future, include the following:

\*cyberattacks (may include phishing, ransomware, DDoS)

\*data breaches (unauthorized access will likely lead to the theft or alteration of data)

\*software vulnerabilities (they can be exploited by attackers to gain unauthorized access)

Now to address the modernization requirements such as the role of open-source libraries, as well as evolving web application technologies. Open-source libraries can speed up development and reduce costs. However, they can also introduce vulnerabilities if they are not properly vetted and updated on a regular basis. As for web application technologies: as they evolve, associated security risks evolve as well. For example, Single Page Applications and API's have different security considerations compared to traditional web applications. It is important to stay updated with the latest technologies and their best security practices.

**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: this provides protection and security for the user whether it is on the administrative side or the customer side. User input must be validated on the RESTful API.

\*APIs: Artemis Financial's web services use RESTful API, so this area of security is significant.

\*Code Error: error handling lets the scrum team know which areas of the RESTful API need to be debugged.

\*Code Quality: A high level of code quality is essential because it will ensure a certain level of security which will mitigate the chances of malicious attacks while ensuring the overall system itself functions properly.

\*Cryptography: User information must be encrypted as an additional security measure. This provides another layer to mitigate and prevent data theft and data mining from outside sources.

**3. Manual Review**

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

\*Could not find an example of error handling anywhere in the code.

\*In the class titled CRUDController, business name is a request parameter.

\*There isn't an authentication measure to validate users.

\*The web service does not use HTTPS.

\*No validator in the POM file.

\*No encryption features.

**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** | **Description** | **Solution** |
| bcprov-jdk15on-1.46.jar | can enable a denial of service attack | Update |
| hibernate-validator-6.0.18.Final.jar | allows attackers to bypass input sanitation (escaping, stripping) controls | Update |
| jackson-databind-2.10.2.jar | allows attackers to cause a denial of service | Update |
| log4j-api-2.12.1.jar | vulnerable to a remote code execution (RCE) attack | Upgrade |
| logback-classic-1.2.3.jar | allows an attacker to mount a Denial-Of-Service attack by sending poisoned data | Update |
| logback-core-1.2.3.jar | allows an attacker to mount a Denial-Of-Service attack by sending poisoned data | Update |
| snakeyaml-1.25.jar | Deserializing yaml content provided by an attacker can lead to remote code execution | Update |
| spring-boot-2.2.4.RELEASE.jar | potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache | Update |
| spring-boot-starter-web-2.2.4.RELEASE.jar | potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache | Update |
| spring-core-5.2.3.RELEASE.jar | potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache | Update |
| spring-expression-5.2.3.RELEASE.jar | potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache | Update |
| spring-web-5.2.3.RELEASE.jar | potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache | Update |
| spring-webmvc-5.2.3.RELEASE.jar | potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache | Update |
| tomcat-embed-core-9.0.30.jar | Generation of Error Message Containing Sensitive Information vulnerability in Apache Tomcat | Upgrade |
| tomcat-embed-websocket-9.0.30.jar | Generation of Error Message Containing Sensitive Information vulnerability in Apache Tomcat | Upgrade |

**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.

First, address all the vulnerabilities in the code base. Those include:

\*implementation of error handling

\*implementation of user validation

\*implementation of HTTPS

\*implementation of validator in POM

\*implementation of encryption measures

\*update and upgrade all dependencies to their latest versions