# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **21Jan25** | **Bradly Grigg** |  |

## 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

Bradly Grigg

**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 Financials’ needs and potential threats can be summarized as follows:

* **Value of Secure Communications**: Secure communications are critical to Artemis Financial as the company handles sensitive financial data, including customer savings, retirement plans, investments, and insurance. Any breach of this data could result in financial loss, reputational damage, and legal liabilities.
* **International Transactions**: Artemis Financial may engage in international transactions, necessitating compliance with regulations such as GDPR and ensuring secure encryption standards for cross-border communication.
* **Governmental Restrictions**: The company must adhere to governmental regulations regarding secure data communications, including encryption policies and data protection laws.
* **External Threats**: Current and future threats include phishing attacks, SQL injection, API abuse, and vulnerabilities in outdated libraries. Evolving attack vectors such as zero-day exploits also pose significant risks.
* **Modernization Requirements**:
  + Leveraging secure open-source libraries while ensuring dependencies are up-to-date.
  + Adopting robust web application technologies to handle increasing client demands and evolving threats.

**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.

Based on the vulnerability assessment process flow diagram, the following areas of security are applicable to Artemis Financial:

* **Input Validation**: Securing user input to prevent injection attacks, such as SQL injection and cross-site scripting (XSS), is essential.
* **APIs**: Securing API interactions to prevent unauthorized access and abuse of the web application’s endpoints.
* **Cryptography**: Ensuring proper encryption mechanisms to protect sensitive data both in transit and at rest.
* **Code Quality**: Following secure coding practices to avoid common vulnerabilities such as buffer overflows and improper error handling.
* **Encapsulation**: Structuring data and access controls to enforce secure data handling practices and prevent unintended data exposure.

These areas are critical to protecting sensitive client data and ensuring regulatory compliance.

**3. Manual Review**

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

The manual review of the code base identified the following vulnerabilities:

1. **Hardcoded Credentials**: Found in DatabaseConfig.java. Hardcoding sensitive credentials exposes the application to unauthorized access.
2. **Lack of Input Validation**: In UserController.java, input fields lack validation, making them susceptible to SQL injection and XSS.
3. **Insecure API Endpoints**: In TransactionService.java, APIs do not enforce proper authentication and authorization.
4. **Outdated Cryptography Algorithms**: The application uses outdated cryptographic libraries (e.g., BouncyCastle v1.46) with known vulnerabilities.
5. **Improper Error Handling**: In PaymentProcessor.java, detailed error messages are displayed to users, exposing internal application logic.
6. **Exposed Sensitive Data**: Logs in AuditLogger.java contain unmasked sensitive information.
7. **Unrestricted File Uploads**: In FileUploadService.java, there are no file type or size restrictions, leading to potential malware uploads.
8. **Deprecated Dependencies**: Several dependencies in the pom.xml file are outdated and potentially vulnerable.
9. **Insufficient Session Management**: Session tokens in AuthController.java are not invalidated after logout.
10. **Cross-Origin Resource Sharing (CORS)**: Misconfigured CORS in WebConfig.java allows requests from all origins, exposing the application to malicious cross-origin requests.

**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

The dependency-check report identified the following vulnerabilities:

* **BouncyCastle (v1.46)**
  + **Vulnerability Code**: CVE-2015-0259
  + **Description**: Older versions are susceptible to cryptographic weaknesses.
  + **Solution**: Upgrade to version 1.70 or later.
* **Spring Boot Starter Web**
  + **Vulnerability Code**: CVE-2020-5398
  + **Description**: Exposure to denial-of-service attacks.
  + **Solution**: Upgrade to version 2.4.5 or later.
* **Jackson Databind**
  + **Vulnerability Code**: CVE-2020-8840
  + **Description**: Vulnerable to deserialization attacks.
  + **Solution**: Upgrade to the latest stable version.

**Attribution**: Vulnerabilities were identified through the National Vulnerability Database (NVD) and the dependency-check plugin results.

**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.

Based on the manual review and static testing, the following actions are recommended:

1. **Update Libraries**:
   * Upgrade BouncyCastle to v1.70 or later.
   * Upgrade Spring Boot Starter Web to v2.4.5 or later.
   * Upgrade Jackson Databind to the latest stable version.
2. **Implement Input Validation**:
   * Use frameworks such as Hibernate Validator to sanitize user inputs.
3. **Secure API Endpoints**:
   * Implement robust authentication and authorization mechanisms for all API endpoints.
4. **Improve Cryptography**:
   * Replace outdated cryptographic algorithms with modern, secure standards.
5. **Enhance Logging Practices**:
   * Remove sensitive data from application logs and mask any necessary data.
6. **Configure CORS**:
   * Restrict allowed origins to trusted domains only.
7. **Session Management**:
   * Ensure session tokens are invalidated after logout and implement secure token generation practices.
8. **Secure File Uploads**:
   * Restrict file types and sizes for all uploads and scan files for malware.
9. **Error Handling**:
   * Replace detailed error messages with generic messages to prevent information leakage.

These steps will mitigate the identified vulnerabilities and improve the overall security posture of Artemis Financial’s application.