# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **3/27/2025** | **Jesse Aranda** | **Initial vulnerability assessment report** |

## 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

Jesse Aranda

**1. Interpreting Client Needs**

Artemis Financial is a financial consulting company that provides individualized financial plans, including savings, retirement, investments, and insurance. Since the company handles sensitive financial and personal data, security is a top priority to prevent data breaches and cyber threats. Below are the key security considerations based on their needs:

* Value of Secure Communications: Since Artemis Financial handles financial data, secure communications are essential to protect sensitive information from unauthorized access, ensuring compliance with data security regulations such as PCI-DSS, GLBA, and international privacy laws.
* International Transactions: If the company operates internationally, it must comply with global data protection regulations such as GDPR (EU) and other regional financial compliance standards.
* Governmental Restrictions: The company must ensure that encryption and security practices align with federal and international regulatory requirements, especially for financial transactions and personally identifiable information (PII).
* External Threats: Potential threats include phishing attacks, ransomware, SQL injection, man-in-the-middle (MITM) attacks, and API security flaws. Since Artemis Financial uses a RESTful API, API security vulnerabilities (such as insecure authentication and unauthorized access) pose significant risks.
* Modernization Requirements: The company must adopt secure software development practices while leveraging open-source libraries with proper security monitoring. It must also stay up to date with evolving web technologies and cloud security best practices.

**2. Areas of Security**

Based on the vulnerability assessment process flow, the following security areas apply to Artemis Financial’s web application:

1. Authentication and Authorization:

* Ensuring secure login mechanisms (e.g., OAuth2, JWT, or SAML).
* Implementing multi-factor authentication (MFA) to prevent unauthorized access.

1. Data Security and Encryption:

* Encrypting sensitive financial and personal data using AES-256 or TLS 1.3.
* Preventing data leaks by properly securing API endpoints.

1. Third-Party Dependencies and Open-Source Security:
   * Several high and critical vulnerabilities were detected in open-source libraries, posing significant security risks.
   * Regularly updating dependencies and implementing a software bill of materials (SBOM) to track security patches.
2. Network and API Security:

* Implementing rate limiting, API authentication (OAuth2), and input validation to secure API endpoints.
* Protecting against MITM attacks with secure communication protocols.

1. Application Security Testing:

* Conducting static and dynamic security testing (SAST/DAST) to detect vulnerabilities before deployment.
* Using penetration testing to simulate cyberattacks and identify weaknesses.

**3. Manual Review**

The following 10 security vulnerabilities were manually identified in Artemis Financial’s codebase:

1. Hardcoded Credentials – Found in configuration files that expose sensitive information.
2. Lack of Input Validation – User inputs are not properly sanitized, increasing SQL Injection risks.
3. Insecure API Endpoints – Some endpoints lack authentication, allowing unauthorized access.
4. Outdated Dependencies – Multiple vulnerable open-source libraries detected (Spring, Logback, SnakeYAML, etc.).
5. Weak Session Management – Sessions do not expire after a defined period, leading to session hijacking risks.
6. No Rate Limiting on API Requests – APIs are susceptible to brute-force attacks due to lack of rate limits.
7. Use of Deprecated Encryption Algorithms – Found in older dependencies (e.g., BouncyCastle 1.46).
8. Log Injection Vulnerability – Found in logging functions, which could allow attackers to inject malicious log entries.
9. Cross-Origin Resource Sharing (CORS) Misconfiguration – Some APIs allow requests from all origins, which is a security risk.
10. Unsecured File Upload Functionality – No validation of uploaded files, allowing potential malware uploads.

**4. Static Testing**

After running a dependency check on Artemis Financial’s software application, the following security vulnerabilities were reported:

**A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

****

**5. Mitigation Plan**

To mitigate the identified security vulnerabilities, Artemis Financial should first upgrade all outdated and vulnerable dependencies, including Spring Framework, SnakeYAML, Logback, and Tomcat, to their latest stable versions. Additionally, input validation and sanitization should be implemented to prevent SQL injection and cross-site scripting (XSS) attacks. To enhance API security, authentication and authorization mechanisms such as OAuth2 and JWT should be enforced, along with rate limiting to prevent brute-force attacks. Strengthening session management practices, including session expiration and token invalidation, will help reduce the risk of session hijacking. Furthermore, secure logging practices should be applied to prevent log injection vulnerabilities, while CORS configurations must be adjusted to restrict access to trusted domains. To prevent malware injection, file upload validation should be enforced. Data security should also be prioritized by encrypting sensitive data in transit using TLS 1.3 and at rest using AES-256. Finally, Artemis Financial should establish a routine for regular security audits and automated dependency checks to proactively monitor and address emerging vulnerabilities, ensuring the ongoing security and compliance of its web application.