# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **9/22/2024** | **Selvadurai Pathmathasan** | **Deign Phase** |

## 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

Selva Path

**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?

Financial institutions should use encrypted communication to All-important, sensitive information and communication are crucial for a financial institute. Secure communication is vital for a financial institution because it protects sensitive customers’ data, builds trust with clients, maintains a positive reputation, and ensures compliance with regulations by safeguarding private financial information, which is essential for the industry's integrity and stability; any breach can lead to significant financial losses and damage to customer confidence. Artmies Finance Company provides retirement accounts and savings accounts. All the services are provided within the United States; none of the services are outside the USA. Yes, there are many laws and restrictions for financial transactions. Money laundering must be practiced by federal laws. That’s why consumers assets are protected by federal reserve protection. All the financial companies must follow the state and federal laws and regulations, and financial institutions and banks are overseas by the Fed to protect consumers’ rights and their deposits. If any information is compromised, that could lead to data breaches, and customers’ accounts become very easy to steal by hackers.

In agile methodology for Faster development, utilize readily available, well-tested open-source libraries to accelerate development time by avoiding the need to build functionalities from scratch. Adopting evolving web application technologies like cloud-native architectures, responsive design, progressive web apps (PWAs), microservices, and robust security practices to enhance scalability, performance, and user experience while staying aligned with current industry standards.

**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.

**Code reviews** should start from the early stage of development it will ensure the quality of codes and will prevent any vulnerabilities in the early development phases of applications, especially using encapsulation methods. Code reviews can help identify security flaws early on, which can prevent potential breaches and attacks.

**Input validation** will indicate in the early stages of any risks and vulnerabilities in testing stages as acceptance testing. then for financial-related inquiries. Input validation will verify authentication of the information from the customers using a particular data type as input information.

Secure API interactions since financial transactions use multiple lawyers of APIsare very important for financial applications; they protect sensitive data from unauthorized access and data breaches. This is especially important for financial institutions because they are subject to strict regulatory requirements, and cybercriminals often target APIs.

**3. Manual Review**

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

Manual review of the software application aligned with manually testing the software starting from all types of testing unit testing, regression testing, then black and white testing, along with the final use acceptance testing, then make sure all the security plugins and libraries are up-to-date versions of all plugins in Maven POM frameworks verified**.**

**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

**Dependencies**

com.fasterxml.jackson.core:jackson-databind vulnerabilities

Direct Vulnerabilities

Known vulnerabilities in the com.fasterxml.jackson.core:jackson-databind package. This does not include vulnerabilities belonging to this package’s dependencies.

Automatically find and fix vulnerabilities affecting your projects. Snyk scans for vulnerabilities and provides fixes for free.

latest non vulnerable version

2.17.2

**Dependency**

CVE-2024-38816: Path traversal vulnerability in functional web frameworks

HIGH | SEPTEMBER 12, 2024 | CVE-2024-38816

**Description**

Applications serving static resources through the functional web frameworks WebMvc.fn or WebFlux.fn are vulnerable to path traversal attacks. An attacker can craft malicious HTTP requests and obtain any file on the file system that is also accessible to the process in which the Spring application is running.

Specifically, an application is vulnerable when both of the following are true:

the web application uses RouterFunctions to serve static resources

resource handling is explicitly configured with a FileSystemResource location

However, malicious requests are blocked and rejected when any of the following is true:

the Spring Security HTTP Firewall is in use the application runs on Tomcat or Jetty Affected Spring Products and Versions Spring Framework 5.3.0 - 5.3.39

|  |  |  |
| --- | --- | --- |
| Dependency | Vulnerability | Update Recommendation |
| log4j-api-2.12.1.jar cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\* | Improper validation of certificate with host mismatch in Apache Log4j SMTP appender. This could allow an SMTPS connection to be intercepted by a man-in-the-middle attack which could leak any log messages sent through that appender. Fixed in Apache Log4j 2.12.3 and 2.13.1 | Upgrade to 2.13.2 which supports this feature. Previous versions can set the system property mail.smtp.ssl.checkserveridentity to true to globally enable hostname verification for SMTPS connections. |
| bcprov jdk15on 1.46.jar | cpe:2.3:a:bouncycastle:bou ncy-castle-crypto package:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:bou ncy\_castle\_crypto\_package :1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:legi on-of-the-bouncy-castle java-crytography api:1.46:\*:\*:\*:\*:\*:\*:\* | Updata to current 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.

Static stage report indicate the severity of risks and the vulnerabilities fixing the bugs in the early stages of development and incorporating all the security frameworks (Apache Tomcat, Hibernator Validator, hibernator validator OWASP Maven dependencies), updating all plugins current versions, finally doing random testing applications, and continuing testing the applications after deployment and maintenance stages will minimize the risk of vulnerability.