# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **3.20.25** | **Alexander Freeders** | **Initial Vulnerability Assessment** |

## Client



Developer

Alexander Freeders

**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 wants to modernize its operations in ensuring its assets are protected from would be malicious actors. Artemis Financial is a company that develops individualized financial plans to its customers; these plans can include retirement, savings, insurance and investments. Since the company provides financial plans and advice to its customers, personal information such as taxes, assets, PII and other sensitive information could be contained therein so ensuring the security of the application is paramount. There is no evidence that this company operates outside of the U.S., but that factor should be considered while implementing security. In Artemis Financials move to modernization, it is highly recommended that the software remain updated and has periodic audits to ensure all aspects remain up to date. With the speed at which technology progresses , it is important that we also stay on point, ensuring any discovered vulnerabilities are quickly patched. Additionally, it is important that clients have the capability to enable multi-factor authentication, and all web-based activity utilizes the HTTPS protocol over HTTP.

**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:** Artemis Financial requires input validation to authenticate the user who is attempting to access the data; moreover, it also acts as a way to ensure that possible injection attacks are discarded using sanitization techniques and prohibiting special characters in fields that don’t require them.
* **API’s:** While Artemis Financial currently has a RESTful API, it is still important to look ensure that the communications therein are secure. As this is a web-based application, it will come into contact with many different platforms.
* **Cryptography:**  Cryptography is essential in secure storage and communication. By implementing cryptography, we are able to ensure the integrity of Artemis Financials client’s data, which again is of the utmost importance as this is sensitive personal information.
* **Error Handling:**  Error Handling is essential as we are dealing with sensitive personal information. As such, it is imperative that any errors or attempts to gain footholds are dealt with expediently and efficiently.
* **Code Quality:**  Code Quality is one of the most important aspects of secure programming. This ensures the program is utilized exactly as it is supposed to be, such as allowing the user who logs in to see *only their* information, and no one else’s. The quality of the code can determine if there are any logical loopholes or areas where a malicious actor could gain a foothold. Using strong, secure coding techniques will allow the most precise and expected performance.

**3. Manual Review**

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

- input validation missing

* Multiple locations, no sanitation techniques

- No error handling in place

* Multiple locations

- No cryptography being used

- No authentication to validate users

- Application does not appear to be using HTTPS

- Code quality suffers from a security POV

- no validation or checks in Customer.java on deposits

**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

A screenshot of a computer

AI-generated content may be incorrect.

Figure - 159 Vulnerabilities found

Running through a dependency check, we can see that there are 159 found vulnerabilities with this code. Shortening the laundry list for brevity, some of the vulnerabilities found include:

**CVE-2024-34447 -- CWE-297 Improper Validation of Certificate with Host Mismatch**

* In Bouncy Castle JCE Provider version 1.55 and earlier the DSA does not fully validate ASN.1 encoding of signature on verification. It is possible to inject extra elements in the sequence making up the signature and still have it validate, which in some cases may allow the introduction of 'invisible' data into a signed structure.
* Update to latest edition of bouncycastle

**CVE-2020-9488** -- **CWE-295 Improper Certificate Validation**

* 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
* Update to latest edition of Apache Tomcat

**CVE-2023-6378** -- **CWE-502 Deserialization of Untrusted Data**

* In log back version 1.2.7 and prior versions, an attacker with the required privileges to edit configurations files could craft a malicious configuration allowing to execute arbitrary code loaded from LDAP servers.

Others from the check:

A screenshot of a computer

AI-generated content may be incorrect.

Figure - List of dependency vulnerabilities found

**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.

From reviewing the dependency report, it appears that many of the issues could be solved by updating software to the newest version, such as Apache Tomcat, bouncycastle and Snakeyaml . It is pertinent for the transition away from HTTP in favor of HTTPS. MFA systems should be implemented and heavily encouraged to clients.