# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **November 18, 2024** | **Mao Christie** |  |

## 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

[Mao Christie]

**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?

Secure communications are an important part of Artemis Financial. They are a financial consultant company, which means customers expect secure communications about their personal information and finances. Personal information they may need to store could be identification numbers like Social Security (in the US) or other ID numbers for international customers. They may also need phone numbers, addresses, and tax information for individuals. Being a finance consulting company, they may have international customers as well, which means they may need to do international transactions. Some government restrictions on secure communications could be restrictions on trade secrets and regulations on safeguarding personal information. The personal information they are storing, if compromised, could lead to financial losses, harassment, and identity theft. Threats include attacks like stealing customers’ information and changing or rewriting customers’ information. Modernization requirements that must be considered are the libraries and frameworks that are used in the web-based software application. Third-party software can introduce vulnerabilities, so they should be kept up-to-date and well-maintained.

**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.

The areas of security that apply to Artemis Financial’s software application are input validation, APIs, Cryptography, Code Error and Quality, and Encapsulation. Input validation is necessary because customers and employees will likely have accounts that they need to log into. Also, customers and employees may need to input personal information so having secure input and input validations is important. APIs are likely used in the software application so making sure they are secure when interacting with the server is important. Cryptography is necessary because the company is dealing with important personal information. Such information and login credentials should be encrypted to not compromise customers’ information. Code error and code quality are both important because both bad quality and errors could cause vulnerabilities that could be exploited and lead to security problems. Encapsulation is necessary because it creates secure data structures that help increase security and protect important information.

**3. Manual Review**

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

I manually inspected several files including myDateTime, CRUDController, CRUD, customer, and greetingController.

In terms of code quality, the myDateTime file has an unfinished/unimplemented method, setMyDateTime. In the DocData file, there is an unused variable, con, as well. For APIs, I looked at the POM file and noticed the Maven Apache version is an alpha version. This is important to note because the alpha version is not currently deemed suitable for production. There also doesn’t seem to be robust input validation so that should be addressed. There does seem to be encapsulation but nothing on cryptography.

**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 CVE-2024-34447:

This vulnerability is for the bouncy castle API version 1.46. It is the improper validation of a certificate without making sure the host matches. Some solutions could include updating/changing the version of the API or implementing a check to make sure the host and all relevant information is properly validated.

CVE-2023-1932:

This vulnerability is a flaw in the isValid method for the hibernate validator. It can lead to accessing information stored in user cookies, running code on a victim’s computer, and compromising confidential information. To mitigate this, using a different library or framework that doesn’t have this weakness or thoroughly encoding outputs could help.

CVE-2020-25649:

This vulnerability is found in Jackson Databind versions including 2.10.2, which is the one used by the software. This flaw doesn’t have proper entity expansion, which makes it vulnerable to XXE attacks. Fixing this might include disabling external entity expansions.

CVE-2020-9488:

This vulnerability is from Apache Log4j. It allows the connection to be intercepted by a man-in-the-middle attack. Fixing this could be done by updating the version used or setting the system property that checks server identity to true to enable verification for SMTPS connections.

CVE-2023-6378:

This vulnerability is for logback-core. It allows a Denial-of-Service attack by sending poisoned data. Fixing this could include changing the version used in the application.

CVE-2022-1471:

This vulnerability is from the SnakeYaml Constructor. This can lead to an attack that allows for remote code execution. Fixing this can include using the SafeConstructor or upgrading the version used.

CVE-2023-20873:

This is from the Spring Boot version 2.2.4. It is susceptible to a security bypass. Mitigation included upgrading the version used to 3.06 or 2.7.11.

CVE-2022-22965:

This vulnerability is from the Spring Framework. It allows for remote code execution if run on JDK 9+ and Tomcat as WAR deployment. Fixing this includes upgrading the version to one without this vulnerability or not deploying Tomcat.

CVE-2020-1938:

This is from Apache Tomcat. It makes remote execution possible by attackers if the AJP port is accessible to untrusted users. Fixing this included upgrading the version to 9.0.31 or other versions.

**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.

To mitigate the CVE-2024-34447 vulnerability, I would upgrade the bouncy castle API version to an unaffected version. For CVE-2023-1932, I would make sure any outputs are encoded to increase security. For CVE-2020-25649, I would mitigate this vulnerability by disabling external entity expansions. For CVE-2020-9488, either update the version or enable verification for SMTPS connections. For CVE-2023-6378, I would upgrade the version of logback-core, and for CVE-2022-1471, I would use the SafeConstructor instead of Constructor. For CVE-2023-10873, I would upgrade the Spring Boot version to 2.06, and for CVE-2022-22965, I would upgrade the Spring Framework to a later version without the vulnerability. For CVE-2020-1938, I would upgrade to Apache Tomcat 9.0.31. I’d also update the Maven Apache version in the POM file to the latest non-alpha version, implement setMyDateTime, and make sure the con variable is used.

Reference:

Common Weakness Enumeration. (n.d.). *CWE-79: Improper Neutralization of Input During Web Page Generation (‘Cross-site Scripting).* Retrieved on November 20, 2024, from <https://cwe.mitre.org/data/definitions/79.html>

NIST. (2024, July 24). *CVE-2020-1938 Detail.* [*https://nvd.nist.gov/vuln/detail/CVE-2020-1938*](https://nvd.nist.gov/vuln/detail/CVE-2020-1938)

NIST. (2024, October 18). *CVE-2022-22965 Detail.* [*https://nvd.nist.gov/vuln/detail/CVE-2022-22965*](https://nvd.nist.gov/vuln/detail/CVE-2022-22965)