# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **5/25/2024** | **Cheyenne Nave** |  |

## 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

Cheyenne Nave

**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 is a consulting company that helps its clients secure themselves financially through the use of savings, retirement, investments, and insurance-based financial plans. They value secure communications due to the copious amount of private data they have about their clients, employees, and partners. As we are to work on the security of their app, we can assume that Artemis Financial is a large enough company to have international transactions along with their domestic transactions. As of now, no governmental restrictions exist on secure communications. However, we must treat all our clients with the utmost priority and provide the highest quality of security available. Current threats consist of hijacking account information (whether that be through stolen biometrics or passwords), monetary theft, hijacking employee access and varying levels of accessibility, and accessing internal communications between employees and employees or clients. When using open-source libraries and other technology, Artemis Financial needs to ensure that all versions are current and properly accessed.

**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
  + The program allows input so input validation should be a requirement to protect users and prevent code injection
* APIs
  + The app will have a client side and user side, meaning the app will work internally and externally. Because of this, a secure API needs to be used so that secure app communication will be guaranteed.
* Cryptography
  + Different monies will be used through Artemis Financial due to the fact that international transactions may occur. Cryptography will help secure account information in these transactions.
* Code Error
  + Any code errors need to be addressed to secure customer and employee information that may be accessed by hackers.
* Code Quality
  + Good code quality prevents hacking or potential account access from others, while also establishing an access hierarchy.

**3. Manual Review**

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

When flipping through the code base, I noticed in the Greeting Controller and Customer.java that there is no input validator. When a customer logs in, a password verification could be used to make sure the right person is accessing their personal account and not someone else’s. Also, I didn’t see any instance or creation of an API. This is a major issue because the app should work internally and externally. The app needs an API to communicate effectively. Throughout each class, I didn’t see any error handling code which ultimately means there is a lack of encryption. This is a major alarm, especially since Artemis Financial is working with hundreds of thousands of social security numbers, birthdates, security questions, and other identifying information about their customers. Finally, the code quality was good with the exception of missing error handling lines.

**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

|  |  |  |  |
| --- | --- | --- | --- |
| Dependency | Vulnerability IDs | Description/Solution | Highest Severity |
| Bcprov-jdk15on-1.46.jar | cpe:2.3:a:bouncycastle:bouncy-castle-crypto-package:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:bouncy\_castle\_crypto\_package:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:bouncy\_castle\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\* | Improper validation of certificate with host mismatch. This can allow for data injection.  **To fix:**  Update to latest version to match host | HIGH |
| Hibernate-validator-6.0.18.Final.jar | cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\* | A bug was found in the Hibernate Validator that enables expressions to be evaluated as if they were valid.  **To fix:**  Update to latest version. Review Input Validation code-block. | MEDIUM |
| jackson-databind-2.10.2.jar | cpe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:\*:\*:\*:\*:\*:\*:\* | Entity expansion is not secured properly. This allows data integrity to be put at risk.  **To fix:**  Update to latest version. | HIGH |
| log4j-api-2.12.1.jar | cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\* | Improper validation of certificate with host. This could allow “man-in-the-middle” attacks which could leak any log messages sent through the appender.  **To fix:**  Update to latest version to match host | LOW |
| logback-core-1.2.3.jar | cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\* | Allows an attacker to mount a “denial-of-service” attack by using poisoned data.  **To fix:**  Ensure proper permissions are enabled and current version is used. | HIGH |
| snakeyaml-1.25.jar | cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\* | Deserializing content provided by an attacker can lead to remote execution.  **To fix:**  Use SnakeYaml SafeConstructor to restrict deserialization. | CRITICAL |
| spring-boot-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\* | Application that is deployed to cloud could be susceptible to security bypass.  **To fix:**  Update to latest version. | CRITICAL |
| spring-boot-starter-web-2.2.4.RELEASE.jar | cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:2.2.4:release:\*:\*:\*:\*:\*:\* | Versions that are no longer supported by the maintainer are vulnerable to temporary directory hijacking.  **To fix:**  Update to latest version. | CRITICAL |
| spring-core-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\* | A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding (code injection)  **To fix:**  Update to latest version | CRITICAL |
| spring-web-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\* | Potential remote code execution issue if used for Java deserialization of untrusted data.  **To fix:**  Authentication may be required. Update to latest version. | CRITICAL |
| spring-webmvc-5.2.3.RELEASE.jar | cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*  cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\* | Possible RCE via data binding.  **To fix:**  Update to latest version | CRITICAL |
| tomcat-embed-core-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* | The report identified a mechanism that allowed: - returning arbitrary files from anywhere in the web application - processing any file in the web application as a JSP Further, if the web application allowed file upload and stored those files within the web application  **To fix:**  Update to latest version | CRITICAL |
| tomcat-embed-websocket-9.0.30.jar | cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*  cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\* | The report identified a mechanism that allowed: - returning arbitrary files from anywhere in the web application - processing any file in the web application as a JSP Further, if the web application allowed file upload and stored those files within the web application (or the attacker was able to control the content of the web application by some other means) then this, along with the ability to process a file as a JSP, made remote code execution possible.  **To fix:**  Update to latest version. | CRITICAL |
|  |  |  |  |

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**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.

The majority of the issues in regards to Artemis Financial’s application sum up to the use of dated versions. Updating to current versions will solve most of the issues here. Other than that, implementing error handling, input validation, and the creation of an API will secure this application.