# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **07/19/2024** | **Jamar Sampson** | **1st Build** |

## 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

Jamar Sampson

**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 company that provides financial planning for their clients. Since this is highly sensitive information to be accessing, the client requires strict encryption for communication software when interacting with their clients. This endeavor is two faced, as it is needed to protect their assets as well as their privacy. With this being a financial institution, they have the potential to work with clients of various regions. Foreign laws can have different implications for the company based on what is acceptable personal identifiable information and how their countries allow or prohibit that information. This adds another layer to the secure channels of communicating.

Any breaches in their company cybersecurity could lead to financial loss & public backlash of trust when it pertains to Artemis Financial. High levels of encryption should be enforced to all databases, whether retaining personal information or not. This leads to lower risks of potential phishing attempts & keeps the client and consumer safe. In addition, Artemis Financial needs to be made aware of any rogue code in use of their system. One such instance is open-source code, and the open-source licensing laws attached to it. If vulnerabilities are found in the program, exploit points and lawsuits could arise due to use of public knowledge code. Artemis Financial has a responsibility to itself & it’s clients to make sure there system says modern & up to date so that all available current fixes for bugs and security threats are actioned upon.

**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 following security risk have been identified after reviewing Artemis Financial security areas. With this being a RESTful API, we should primarily be focused on areas pertaining to Input validation, Secure API Interactions, Code Errors, & Encapsulation.

1. Input Validation – Due to this being a RESTful API, it will accept user inputs. Because of this, all user inputs need to be validated. There is general code checking already built into SPRING, but that error handling tends to be generic & not provide any signature that a hacker can use to understand more about the code structure. Use of cryptography should enforced for all data flows. API’s need to communicate securely while any code error may be a result from improper userinputs.

**3. Manual Review**

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

During static testing, I took note of the *POM.xml* required updating from Maven version *5.1.X* to *10.0.3* to avoid missing certain CVEs. Classes like Greeting use private parameters with get/set methods, but *GreetingController* did not utilize these. User input to the API is unsanitized, posing injection risks, as seen in methods that directly incorporate user input without validation, like the number method. The customer class's *account\_balance* should be private with get/set methods, and *myDateTime* methods, such as *setMyDateTime*, are incomplete. Additionally, *DocData* contains hardcoded, easily guessable usernames and passwords, which should be obfuscated, salted, and hashed.

Error checking and messaging need improvement, as many methods lack proper error handling, potentially exposing sensitive information. For instance, a failed SQL connection in *DocData* -> *read\_document* returns a full stack trace, which could be exploited. Other vulnerabilities include unvalidated request parameters, hardcoded database connection parameters in *DocData*, lack of HTTPS usage, and no authentication scheme.

The vulnerability assessment revealed significant issues with input validation, as the *POM.XML* lacked an Apache validator, and the greeting controller did not validate input. The program accessed data insecurely via URL instead of POST, risking data leakage into browser history. Without a proper API, user interaction is unclear, and no encryption methods were found. From my perspective, the code quality was good, but the lack of an API and secure input handling rendered the program user-unfriendly and insecure.

**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

Description automatically generated

|  |  |  |  |
| --- | --- | --- | --- |
| Dependency | Vulnerability Code | Description | Mitigation |
| [bcprov-jdk15on-1.46.jar](file:///C:\Users\BTH\AppData\Local\Microsoft\Windows\INetCache\Content.MSO\rest-service\target\dependency-check-report.html#l1_991c96a4e31e6c19e2b9136c8955bd423f2dc4c7) | 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:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Abouncycastle&cpe_product=cpe%3A%2F%3Abouncycastle%3Abouncy_castle_for_java&cpe_version=cpe%3A%2F%3Abouncycastle%3Abouncy_castle_for_java%3A1.46) [cpe:2.3:a:bouncycastle:legion-of-the-bouncy-castle-java-crytography-api:1.46:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Abouncycastle&cpe_product=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api&cpe_version=cpe%3A%2F%3Abouncycastle%3Alegion-of-the-bouncy-castle-java-crytography-api%3A1.46) cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\* | The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms | Upgrading Bouncy castle to the latest version. |
| [hibernate-validator-6.0.18.Final.jar](#l3_7fd00bcd87e14b6ba66279282ef15efa30dd) | [cpe:2.3:a:redhat:hibernate\_validator:6.0.18:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aredhat&cpe_product=cpe%3A%2F%3Aredhat%3Ahibernate_validator&cpe_version=cpe%3A%2F%3Aredhat%3Ahibernate_validator%3A6.0.18) | Bug message. Allows attackers to bypass input sanitation. | User error. Review and check input validation. |
| [jackson-databind-2.10.2.jar](#l5_0528de95f198afafbcfb0c09d2e43b6e0ea6) | c[pe:2.3:a:fasterxml:jackson-databind:2.10.2:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Afasterxml&cpe_product=cpe%3A%2F%3Afasterxml%3Ajackson-databind&cpe_version=cpe%3A%2F%3Afasterxml%3Ajackson-databind%3A2.10.2) cpe:2.3:a:fasterxml:jackson-modules-java8:2.10.2:\*:\*:\*:\*:\*:\*:\* | Does not secure expansion secured properly. Highest threat is data integrity. | Review and check input of code source. |
| [log4j-api-2.12.1.jar](#l10_a55e6d987f50a515c9260b0451b4fa217dc) | [cpe:2.3:a:apache:log4j:2.12.1:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Alog4j&cpe_version=cpe%3A%2F%3Aapache%3Alog4j%3A2.12.1) | SMTPS connection to be intercepted by a middle man. | Fixed by updating to the latest Apache4j 2.12.3 & 2.13.1. |
| [logback-core-1.2.3.jar](#l12_864344400c3d4d92dfeb0a305dc87d95367) | 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 | Allows attacker to mount DoS by sending corrupted data. | Update to latest version past 1.4.11. |
| [snakeyaml-1.25.jar](#l14_8b6e01ef661d8378ae6dd7b511a7f2a33fa) | [cpe:2.3:a:snakeyaml\_project:snakeyaml:1.25:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Asnakeyaml_project&cpe_product=cpe%3A%2F%3Asnakeyaml_project%3Asnakeyaml&cpe_version=cpe%3A%2F%3Asnakeyaml_project%3Asnakeyaml%3A1.25) | YAML 1.26 parser & emitter for Java. | Recommended to upgrade to version past 2.0 or beyond. |
| [spring-boot-2.2.4.RELEASE.jar](#l15_225a4fd31156c254e3bb92adb42ee8c6de8) | [cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_boot&cpe_version=cpe%3A%2F%3Avmware%3Aspring_boot%3A2.2.4) | Spring Boot is running on a older version, with potential for DoS attack if used with reverse proxy. | Users of older versions should upgrade to 2.7.11+ or 3.0.6+. |
| [spring-boot-starter-web-2.2.4.RELEASE.jar](#l16_ec75d01d212b5229c16d872fb127744c0ed) | [cpe:2.3:a:vmware:spring\_boot:2.2.4:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_boot&cpe_version=cpe%3A%2F%3Avmware%3Aspring_boot%3A2.2.4) [cpe:2.3:a:web\_project:web:2.2.4:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aweb_project&cpe_product=cpe%3A%2F%3Aweb_project%3Aweb&cpe_version=cpe%3A%2F%3Aweb_project%3Aweb%3A2.2.4) | Spring Boot on older versions is suspectible to security bypass. | Users of older versions should upgrade to 2.7.11+ or 3.0.6+. |
| [spring-core-5.2.3.RELEASE.jar](#l17_3734223040040e8c3fecd5faa3ae8a1ed6d) | [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3) [cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3) [cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_framework&cpe_version=cpe%3A%2F%3Avmware%3Aspring_framework%3A5.2.3) [cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aweb_project&cpe_product=cpe%3A%2F%3Aweb_project%3Aweb&cpe_version=cpe%3A%2F%3Aweb_project%3Aweb%3A5.2.3) | Code running on JDK 9+ is vulnerable to remote code execution. | Upgrade to the latest version of spring. |
| [spring-webmvc-5.2.3.RELEASE.jar](#l19_745a62502023d2496b565b7fe102bb1ee22) | [cpe:2.3:a:pivotal\_software:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Apivotal_software&cpe_product=cpe%3A%2F%3Apivotal_software%3Aspring_framework&cpe_version=cpe%3A%2F%3Apivotal_software%3Aspring_framework%3A5.2.3) [cpe:2.3:a:springsource:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aspringsource&cpe_product=cpe%3A%2F%3Aspringsource%3Aspring_framework&cpe_version=cpe%3A%2F%3Aspringsource%3Aspring_framework%3A5.2.3) [cpe:2.3:a:vmware:spring\_framework:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Avmware&cpe_product=cpe%3A%2F%3Avmware%3Aspring_framework&cpe_version=cpe%3A%2F%3Avmware%3Aspring_framework%3A5.2.3) [cpe:2.3:a:web\_project:web:5.2.3:release:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aweb_project&cpe_product=cpe%3A%2F%3Aweb_project%3Aweb&cpe_version=cpe%3A%2F%3Aweb_project%3Aweb%3A5.2.3) | Code running on JDK 9+ is vulnerable to remote code execution. | Upgrade to the latest version of spring. |
| [tomcat-embed-core-9.0.30.jar](#l20_ad32909314fe2ba02cec036434c0addd19b) | [cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Atomcat&cpe_version=cpe%3A%2F%3Aapache%3Atomcat%3A9.0.30) [cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache_tomcat&cpe_product=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat&cpe_version=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat%3A9.0.30) | Improper validation and parsing HTTP trailer headers. | Recommended to upgrade to ver 11.0.0-M11 onwards. Users will also need to make small changes to their configurations. |
| [tomcat-embed-websocket-9.0.30.jar](#l22_33157f6bc5bfd03380ebb5ac476db0600a0) | [cpe:2.3:a:apache:tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache&cpe_product=cpe%3A%2F%3Aapache%3Atomcat&cpe_version=cpe%3A%2F%3Aapache%3Atomcat%3A9.0.30) [cpe:2.3:a:apache\_tomcat:apache\_tomcat:9.0.30:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aapache_tomcat&cpe_product=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat&cpe_version=cpe%3A%2F%3Aapache_tomcat%3Aapache_tomcat%3A9.0.30) | Improper validation and parsing HTTP trailer headers. | Recommended to upgrade to ver 11.0.0-M11 onwards. Users will also need to make small changes to their configurations. |

**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.

All portions of the dependent library need to be updated. This would lead to new code being written & implemented. Should fix most vulnerabilities. Once the code is rebuilt, internal tests in Spring should be refactored & refreshed to add in any new sanitization error messages input validation/sanitization, and security improvements (salting and hashing).

In addition, code review or peer programming should be considered as there are some larger security issues in the code. Switching to HTTPS protocol for all additional communications. Moving request parameters to headers & the body. Remove hard-coded database connection credentials & implement secure authentication scheme to ensure updated security.