# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **02/02/2025** | **Timothy Russell** |  |

## 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

Timothy Russell

**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 very valuable since it can help with preventing data breaches. Secure communications can also help maintain the trust of your clients.

Should this company wish to engage in international transactions, it must use security standards that are accepted globally and utilize encryption protocols that protect data that is being transferred across international borders.

One such government restriction the company should consider is the Secure and Trusted Communications Networks Act of 2019. This prohibits the use of federal funds to buy equipment or services that can pose a national security risk. Another would be the Electronics Communications and Privacy Act of 1986. This prohibits the intentional (attempted or otherwise) procurement, use, or disclosure of any wire, oral, or electronic communication.

Current and future external threats would include phishing attempts, DDos attacks, hacking, and the introduction of malware into a system.

Open-source libraries are valuable in that the community can become involved in improvements in security. This can allow for fresh ideas on how to fix issues and provide a learning environment for new developers.

Keeping up with evolving web application technologies is essential in today’s environment. Changing technologies can have new and stronger layers of security implemented within them. This would help mitigate future electronic attacks.

**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.

Based on the software application, there are several areas of security that could be applied.

Authentication: Proper authentication mechanisms should be in place to assure that only authorized users can access the software and the financial information contained within.

Output Encoding and Input Validation: With input validation mechanisms in place, SQL injections and cross-site scripting can be prevented. Output encoding can be used to prevent cross-site scripting when content generated by the user is displayed.

Data Protection: This should be implemented by properly encrypting user sensitive financial information that would be targeted by hackers. Encryption should be utilized for both data being stored and well as transmitted.

Secure Communication: Proper protocols for secure communication should be implemented to ensure safe transfer of information between the client and the servers. Using protocols such as HTTPS can help ensure confidentiality.

**3. Manual Review**

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

In the DocData class, read\_document accepts key and value, but since there is no prepared statement, they could be used as SQL injection.

There is a security risk with “root” being hardcoded.

In the Greeting class, if content accepts user input, it might allow for cross-site scripting.

In the Greetings Controller class, name has no restriction in length. Therefore, this could be susceptible to a DDos attack.

In the class customer, account\_balance is not private, so another class could modify it.

The method, deposit(int a), has no validation for the amount. A hacker could enter a negative deposit to reduce the balance.

The account number is an integer and not a string, therefore it is not encrypted. This would allow a hacker to steal the account number.

**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

Bcprov-jdk15on-1.46.jar: The Bouncy Castle Crypto package is a Java implementation of cryptographic algorithms. Vulnerability CVE-2024-34447 – Improper validation of certificate with host mismatch. The software provides the host with a certificate, but does not ensure the certificate is associated with that host.

Hibernate-validator-6.0.18.Final.jar: Hibernate’s Bean Validation reference implementation

-Vulnerability CVE-2023-1932 A flaw was found in hibernate-validator's 'isValid' method in the org.hibernate.validator.internal.constraintvalidators.hv.SafeHtmlValidator class, which can be bypassed by omitting the tag ending in a less-than character. Browsers may render an invalid html, allowing HTML injection or Cross-Site-Scripting (XSS) attacks.

-[CVE-2020-10693](https://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2020-10693) A flaw was found in Hibernate Validator version 6.1.2.Final. A bug in the message interpolation processor enables invalid EL expressions to be evaluated as if they were valid. This flaw allows attackers to bypass input sanitation (escaping, stripping) controls that developers may have put in place when handling user-controlled data in error messages.

Jackson-databind-2.10.2.jar: General data-binding functionality for Jackson that works on core streaming API.

-CVE-2020-25649 A flaw was found in FasterXML Jackson Databind, where it did not have entity expansion secured properly. This flaw allows vulnerability to XML external entity (XXE) attacks. The highest threat from this vulnerability is data integrity.

-CVE-2020-36518 jackson-databind before 2.13.0 allows a Java StackOverflow exception and denial of service via a large depth of nested objects.

-CVE-2021-46877 Jackson-databind 2.10.x through 2.12.x before 2.12.6 and 2.13.x before 2.13.1 allows attackers to cause a denial of service (2 GB transient heap usage per read) in uncommon situations involving JsonNode JDK serialization.

Log4j-api-2.12.1.jar The Apache Log4j API

-CVE-2020-9488 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

Logback-classic-1.2.3.jar logback-classic module

-CVE-2023-6378 A serialization vulnerability in logback receiver component part of logback version 1.4.11 allows an attacker to mount a Denial-Of-Service attack by sending poisoned data.

-CVE-2021-42550 In logback 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.

Logback-core-1.2.3.jar: logback-core module

-CVE 2023-6378 A serialization vulnerability in logback receiver component part of logback version 1.4.11 allows an attacker to mount a Denial-Of-Service attack by sending poisoned data.

-CVE-2021-42550 In logback 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.

Snakeyaml-1.23.jar YAML 1.1 parser and emitter for Java

-CVE-2022-1471 SnakeYaml's Constructor() class does not restrict types which can be instantiated during deserialization. Deserializing yaml content provided by an attacker can lead to remote code execution. We recommend using SnakeYaml's SafeConsturctor when parsing untrusted content to restrict deserialization. We recommend upgrading to version 2.0 and beyond.

-CVE-2017-18640 The Alias feature in SnakeYAML before 1.26 allows entity expansion during a load operation, a related issue to CVE-2003-1564.

Spring-boot-2.2.4.RELEASE.jar Spring Boot

-CVE-2023-20873 In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass. Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+.

-CVE-2022-27772 spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method.

-CVE-2023-20883 In Spring Boot versions 3.0.0 - 3.0.6, 2.7.0 - 2.7.11, 2.6.0 - 2.6.14, 2.5.0 - 2.5.14 and older unsupported versions, there is potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache.

Spring-boot-starter-web-2.2.4.RELEASE.jar: Starter for building web, including RESTful, applications using Spring MVC. Uses Tomcat as the default embedded container.

-CVE-2023-20873 In Spring Boot versions 3.0.0 - 3.0.5, 2.7.0 - 2.7.10, and older unsupported versions, an application that is deployed to Cloud Foundry could be susceptible to a security bypass. Users of affected versions should apply the following mitigation: 3.0.x users should upgrade to 3.0.6+. 2.7.x users should upgrade to 2.7.11+. Users of older, unsupported versions should upgrade to 3.0.6+ or 2.7.11+.

-CVE-2022-27772 spring-boot versions prior to version v2.2.11.RELEASE was vulnerable to temporary directory hijacking. This vulnerability impacted the org.springframework.boot.web.server.AbstractConfigurableWebServerFactory.createTempDir method.

-CVE-2023-20883 In Spring Boot versions 3.0.0 - 3.0.6, 2.7.0 - 2.7.11, 2.6.0 - 2.6.14, 2.5.0 - 2.5.14 and older unsupported versions, there is potential for a denial-of-service (DoS) attack if Spring MVC is used together with a reverse proxy cache.

Spring-core-5.2.3.RELEASE.jar: Spring Core

-CVE-2022-22965

CISA Known Exploited Vulnerability:

Product: VMware Spring Framework

Name: Spring Framework JDK 9+ Remote Code Execution Vulnerability

Date Added: 2022-04-04

Description: Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding.

Required Action: Apply updates per vendor instructions.

Due Date: 2022-04-25

Notes: https://nvd.nist.gov/vuln/detail/CVE-2022-22965

-CVE-2021-22118 In Spring Framework, versions 5.2.x prior to 5.2.15 and versions 5.3.x prior to 5.3.7, a WebFlux application is vulnerable to a privilege escalation: by (re)creating the temporary storage directory, a locally authenticated malicious user can read or modify files that have been uploaded to the WebFlux application, or overwrite arbitrary files with multipart request data.

Spring-expression-5.2.3.RELEASE.jar: Spring Expression Language

-CVE-2022-22965

CISA Known Exploited Vulnerability:

Product: VMware Spring Framework

Name: Spring Framework JDK 9+ Remote Code Execution Vulnerability

Date Added: 2022-04-04

Description: Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding.

Required Action: Apply updates per vendor instructions.

Due Date: 2022-04-25

Notes: https://nvd.nist.gov/vuln/detail/CVE-2022-22965

-CVE-2021-22118 In Spring Framework, versions 5.2.x prior to 5.2.15 and versions 5.3.x prior to 5.3.7, a WebFlux application is vulnerable to a privilege escalation: by (re)creating the temporary storage directory, a locally authenticated malicious user can read or modify files that have been uploaded to the WebFlux application, or overwrite arbitrary files with multipart request data.

Spring-web-5.2.3.RELEASE.jar: Spring Web

-CVE-2016-1000027 Pivotal Spring Framework through 5.3.16 suffers from a potential remote code execution (RCE) issue if used for Java deserialization of untrusted data. Depending on how the library is implemented within a product, this issue may or not occur, and authentication may be required. NOTE: the vendor's position is that untrusted data is not an intended use case. The product's behavior will not be changed because some users rely on deserialization of trusted data.

-CVE-2022-22965

CISA Known Exploited Vulnerability:

Product: VMware Spring Framework

Name: Spring Framework JDK 9+ Remote Code Execution Vulnerability

Date Added: 2022-04-04

Description: Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding.

Required Action: Apply updates per vendor instructions.

Due Date: 2022-04-25

Notes: https://nvd.nist.gov/vuln/detail/CVE-2022-22965

Spring-webmvc-5.2.3.RELEASE.jar: Spring Web MVC

-CVE-2022-22965

CISA Known Exploited Vulnerability:

Product: VMware Spring Framework

Name: Spring Framework JDK 9+ Remote Code Execution Vulnerability

Date Added: 2022-04-04

Description: Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding.

Required Action: Apply updates per vendor instructions.

Due Date: 2022-04-25

Notes: <https://nvd.nist.gov/vuln/detail/CVE-2022-22965>

-CVE-2024-38816 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.

Tomcat-emded-core-9.0.30.jar: Core Tomcat implementation

-CVE-2020-1938

CISA Known Exploited Vulnerability:

Product: Apache Tomcat

Name: Apache Tomcat Improper Privilege Management Vulnerability

Date Added: 2022-03-03

Description: Apache Tomcat treats Apache JServ Protocol (AJP) connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited.

Required Action: Apply updates per vendor instructions.

Due Date: 2022-03-17

Notes: https://nvd.nist.gov/vuln/detail/CVE-2020-1938

-CVE-2020-11996 A specially crafted sequence of HTTP/2 requests sent to Apache Tomcat 10.0.0-M1 to 10.0.0-M5, 9.0.0.M1 to 9.0.35 and 8.5.0 to 8.5.55 could trigger high CPU usage for several seconds. If a sufficient number of such requests were made on concurrent HTTP/2 connections, the server could become unresponsive.

-CVE-2020-13934 An h2c direct connection to Apache Tomcat 10.0.0-M1 to 10.0.0-M6, 9.0.0.M5 to 9.0.36 and 8.5.1 to 8.5.56 did not release the HTTP/1.1 processor after the upgrade to HTTP/2. If a sufficient number of such requests were made, an OutOfMemoryException could occur leading to a denial of service.

Tomcat-embed-websocket-9.0.30.jar: Core Tomcat implementation

-CVE-2020-1938

CISA Known Exploited Vulnerability:

Product: Apache Tomcat

Name: Apache Tomcat Improper Privilege Management Vulnerability

Date Added: 2022-03-03

Description: Apache Tomcat treats Apache JServ Protocol (AJP) connections as having higher trust than, for example, a similar HTTP connection. If such connections are available to an attacker, they can be exploited.

Required Action: Apply updates per vendor instructions.

Due Date: 2022-03-17

Notes: https://nvd.nist.gov/vuln/detail/CVE-2020-1938

-CVE-2020-8022 A Incorrect Default Permissions vulnerability in the packaging of tomcat on SUSE Enterprise Storage 5, SUSE Linux Enterprise Server 12-SP2-BCL, SUSE Linux Enterprise Server 12-SP2-LTSS, SUSE Linux Enterprise Server 12-SP3-BCL, SUSE Linux Enterprise Server 12-SP3-LTSS, SUSE Linux Enterprise Server 12-SP4, SUSE Linux Enterprise Server 12-SP5, SUSE Linux Enterprise Server 15-LTSS, SUSE Linux Enterprise Server for SAP 12-SP2, SUSE Linux Enterprise Server for SAP 12-SP3, SUSE Linux Enterprise Server for SAP 15, SUSE OpenStack Cloud 7, SUSE OpenStack Cloud 8, SUSE OpenStack Cloud Crowbar 8 allows local attackers to escalate from group tomcat to root. This issue affects: SUSE Enterprise Storage 5 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP2-BCL tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP2-LTSS tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP3-BCL tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP3-LTSS tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server 12-SP4 tomcat versions prior to 9.0.35-3.39.1. SUSE Linux Enterprise Server 12-SP5 tomcat versions prior to 9.0.35-3.39.1. SUSE Linux Enterprise Server 15-LTSS tomcat versions prior to 9.0.35-3.57.3. SUSE Linux Enterprise Server for SAP 12-SP2 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server for SAP 12-SP3 tomcat versions prior to 8.0.53-29.32.1. SUSE Linux Enterprise Server for SAP 15 tomcat versions prior to 9.0.35-3.57.3. SUSE OpenStack Cloud 7 tomcat versions prior to 8.0.53-29.32.1. SUSE OpenStack Cloud 8 tomcat versions prior to 8.0.53-29.32.1. SUSE OpenStack Cloud Crowbar 8 tomcat versions prior to 8.0.53-29.32.1.

**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.

Cross-Site Scripting (XSS) Vulnerability: Use proper output encoding to sanitize user input and prevent script injection. This is achieved by using appropriate libraries or frameworks that handle encoding automatically.

Use a prepared statement in the DocData class to avoid SQL injection.

Instead of hardcoding “root”, use environment variables to provide more security.

In the GreetingController class, add a length limit to the name parameter.

In the customer class, set account\_balance to private and provide controlled access by using getter and setter methods.

For the method deposit(int a), add a way to validate the input by setting it so that a negative value cannot be entered.

The account number should be stored securely by using encryption.

Address the vulnerabilities listed in the dependency-check report by following the recommended solutions provided for each one. This will require updating or patching vulnerable libraries, frameworks, or components used in the codebase.