# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **5/25/2025** | **Emerald Tresch** |  |

## 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

Emerald Tresch

**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 essential, especially when handling sensitive financial data, as they help prevent data breaches and maintain client trust. For organizations like Artemis Financial that are involved in international transactions, adhering to global security standards and encryption protocols to safeguard cross-border data transfers is crucial. Additionally, compliance with regulations such as the Secure and Trusted Communications Networks Act of 2019 is necessary, as it prohibits using federal funds for acquiring equipment or services that may pose national security risks (House Energy and Commerce, 2020). Companies must also remain vigilant against current and emerging external threats, including malware, hacking, DDoS attacks, and social engineering tactics like phishing (Kost, 2025). Furthermore, modernizing security practices necessitates the effective use of open-source libraries, which offer opportunities for community-driven security enhancements but also introduce risks due to their open nature. Staying abreast of evolving web application technologies—such as Progressive Web Apps (PWAs), AI and ML integration, and API-First developments are critical for implementing robust security measures.

**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.

After thoroughly assessing Artemis Financial’s security measures, I have identified several potential vulnerabilities that should be addressed to strengthen the application's overall security. One crucial area is input validation. Robust user input validation is essential to mitigate risks such as potential failures or SQL injection attacks. By ensuring that input adheres to expected formats and acceptable values, we can significantly reduce the likelihood of malicious data being processed by the application. This foundational step is vital to maintaining the system's integrity and safeguarding sensitive information.

Another key component is the development of secure APIs. A well-designed API is necessary since the application will function internally and externally, particularly in end users' web browsers. This API will govern how end users interact with the application and should clearly define which methods for data access are permitted. Furthermore, considering the software may interact with third-party applications, a secure API is essential to prevent unauthorized access and protect data.

Cryptography is also critical for this system, especially since it will handle international transfers involving proprietary customer information. All data must be encrypted in compliance with North American regulations and the relevant regulations in the destination countries. This will safeguard sensitive information during transmission, preserving confidentiality and data integrity.

In addition, effective code error handling should be developed in tandem with the API and input validation processes. Strong error-handling mechanisms are vital, particularly when addressing invalid input. These mechanisms must prevent unauthorized access or privilege escalation attempts, thereby maintaining the application’s security posture and protecting sensitive data from exposure or exploitation.

Lastly, ensuring high code quality is essential when creating software that interacts with end users, particularly input handling and API integration. A commitment to clean and well-structured code helps prevent accidental data exposure and restricts unauthorized access to methods by users lacking the appropriate permissions. We can significantly enhance the application's security and reliability by prioritizing code quality. Addressing these vulnerabilities will allow us to create a more secure application for Artemis Financial, ultimately protecting the company and its users.

**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 conducted a thorough review of input validation through the vulnerability assessment process. My analysis commenced with examining the POM.XML file for any indications of the Apache validator. Subsequently, I assessed the greeting controller, which revealed that the input observed did not appear to undergo any validation processes. Unfortunately, I could not verify such validation's existence, as no output was available for review.

Furthermore, I sought to identify any APIs functioning within the application but, regrettably, found none. Nevertheless, the program was capable of accessing data through unsecured means. Specifically, it utilized the URL to retrieve data rather than employing the POST method, which raises potential security concerns as this information could be exposed in browser history and subject to exploitation. Although the program accepts input via the URL, it does not generate any corresponding output, thus remaining vulnerable due to its handling of raw user input. Moreover, the absence of an API limits the end user’s ability to comprehend how to interact with the program unless they have access to the source code. An adequately designed RESTful API should delineate a defined method for user interaction.

After addressing the issues related to input validation and API functionality, I turned my attention to cryptography. It became apparent that data encryption was not implemented within the program. For Artemis Financial to maintain secure operations, it must develop robust data encryption protocols to protect stored information and comply with international regulations regarding financial transactions.

My review of error handling revealed that the DocData.java class did not contain any mistakes and included appropriate try-and-catch blocks; however, no further error-handling mechanisms were evaluated. Overall, the quality of the code was commendable. Nonetheless, the absence of an API significantly impeded the program's functionality, detracting from the overall user experience. Furthermore, the lack of input validation and handling input through the URL rather than employing the POST method presents substantial security risks, including the potential for data exposure and compromise.

**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

Names and Vulnerability Codes of Known Vulnerabilities

* Bouncy Castle Crypto Package (bcprov-jdk15on-1.46.jar)
  + Vulnerability ID: 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:\*:\*:\*:\*:\*:\*:\*  
    cpe:2.3:a:bouncycastle:the\_bouncy\_castle\_crypto\_package\_for\_java:1.46:\*:\*:\*:\*:\*:\*:\*
* Hibernate Validator (hibernate-validator-6.0.18.Final.jar)
  + Vulnerability ID: [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)
* Jackson Databind (jackson-databind-2.10.2.jar)
  + Vulnerability ID: [cpe: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:\*:\*:\*:\*:\*:\*:\*
* Log4j API (log4j-api-2.12.1.jar)
  + Vulnerability ID: [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)
* Logback Classic (logback-classic-1.2.3.jar)
  + Vulnerability ID: [cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aqos&cpe_product=cpe%3A%2F%3Aqos%3Alogback&cpe_version=cpe%3A%2F%3Aqos%3Alogback%3A1.2.3)
* Logback Core (logback-core-1.2.3.jar)
  + Vulnerability ID: [cpe:2.3:a:qos:logback:1.2.3:\*:\*:\*:\*:\*:\*:\*](https://nvd.nist.gov/vuln/search/results?form_type=Advanced&results_type=overview&search_type=all&cpe_vendor=cpe%3A%2F%3Aqos&cpe_product=cpe%3A%2F%3Aqos%3Alogback&cpe_version=cpe%3A%2F%3Aqos%3Alogback%3A1.2.3)
* SnakeYaml (snakeyaml-1.25.jar)
  + Vulnerability ID: [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)
* Spring Boot (spring-boot-2.2.4.RELEASE.jar)
  + Vulnerability ID: [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 Starter Web (spring-boot-starter-web-2.2.4.RELEASE.jar) )
  + Vulnerability ID: [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 Context (spring-context-5.2.3.RELEASE.jar)
  + Vulnerability ID: [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)
* Spring Core (spring-core-5.2.3.RELEASE.jar)
  + Vulnerability ID: [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)
* Spring Expression (spring-expression-5.2.3.RELEASE.jar)
  + Vulnerability ID: [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)
* Spring Web (spring-web-5.2.3.RELEASE.jar)
  + Vulnerability ID: [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)
* Spring WebMVC (spring-webmvc-5.2.3.RELEASE.jar)
  + Vulnerability ID: [pe: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)
* Tomcat Embed Core (tomcat-embed-core-9.0.30.jar)
  + Vulnerability ID: [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)
* Tomcat Embed WebSocket (tomcat-embed-websocket-9.0.30.jar)
  + Vulnerability ID: [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)

Description and Recommended Solutions

* Bouncy Castle Crypto Package: This is a Java implementation of cryptographic algorithms. To secure cryptographic operations, update to a secure version or use an alternative library.
* Hibernate Validator: This is used for validation according to the Java Bean Validation API. Update to a newer version that addresses identified vulnerabilities.
* Jackson Databind: A library for converting between Java objects and JSON. It is advisable to upgrade to a version without known vulnerabilities.
* Log4j API & Logback: Both are logging libraries that should be updated to the latest versions to mitigate risks associated with identified vulnerabilities.
* Spring Framework Components (Spring Boot, Spring Context, Spring Core, etc.): Multiple critical vulnerabilities exist. To address security flaws, it is strongly recommended that all Spring-related libraries be upgraded to their latest, patched versions.

Attribution for Identification of Vulnerabilities

The Common Vulnerabilities and Exposures (CVE) system has documented the vulnerabilities, which tracks security vulnerabilities across various software platforms. Each identified vulnerability is linked to specific CPE (Common Platform Enumeration) identifiers that are standard in the security community, providing a consistent way to reference these vulnerabilities. The document includes confidence levels that indicate how reliably these vulnerabilities have been validated and package paths and hash values (MD5 and SHA1) for additional verification and attribution. This structured approach allows for easy tracking and management of known vulnerabilities within the reported dependencies.

**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.

Based on the information presented in the manual review and static testing report for the software application developed by Artemis Financial, several vulnerabilities associated with Apache Tomcat have been identified. The following is an interpretation of the findings and recommended steps for mitigating these security vulnerabilities.

Interpretation of Results:

1. Incomplete Cleanup Vulnerability (CVE-2023-42795):
   1. This vulnerability affects Apache Tomcat versions from 11.0.0-M1 to 9.0.80. Due to deficiencies in the internal object recycling process, it may result in information leakage between requests.
   2. Severity: Medium (CVSS score: 5.3)
2. Improper Input Validation (CVE-2023-45648):
   1. This issue, which affects versions from 11.0.0-M1 to 9.0.81, has the potential to lead to request smuggling when an invalid trailer header is processed.
   2. Severity: Medium (CVSS score: 5.3)
3. Generation of Error Messages with Sensitive Information (CVE-2024-21733):
   1. This vulnerability is present in versions 8.5.7 to 9.0.43, where error messages may inadvertently disclose sensitive information.
   2. Severity: Medium (CVSS score: 5.3)

Steps to Mitigate Identified Vulnerabilities:

1. Upgrade Apache Tomcat:
   1. For the Incomplete Cleanup Vulnerability, Upgrade to version 11.0.0-M12 or later, 10.1.14 or later, 9.0.81 or later, or 8.5.94 or later.
   2. For Improper Input Validation: The same version upgrades mentioned previously should be implemented.
   3. To Generation Sensitive Information in Error Messages, Upgrade to version 8.5.64 or 9.0.44 or later.
2. Implement Best Configuration Practices: It is advisable to review and modify Apache Tomcat’s configuration to ensure that sensitive information is not logged or revealed through error messages.
3. Conduct Regular Security Audits: Regular security assessments and vulnerability scans should be performed to identify any latent vulnerabilities within the application and its environment.
4. Monitor and Patch: A patch management system should ensure that updates and security patches are applied promptly upon release.
5. Educate Team Members: It is crucial to ensure that development and operations teams are well-informed about security best practices, including secure coding standards and methods for handling sensitive information.
6. Test After Upgrades: Following upgrades and configuration changes, regression testing should be conducted to verify that the application continues functioning correctly and that identified vulnerabilities have been addressed.

By adopting these measures, Artemis Financial can significantly mitigate the vulnerabilities identified within its software application, ensuring security and functionality.

**References**

House Energy and Commerce (2020, March 12). *H.R.4998 - Secure and Trusted Communications Networks Act of 2019*. Congress.gov. Retrieved May 25, 2025, from <https://www.congress.gov/bill/116th-congress/house-bill/4998>

Kost, E. (2025, January 2). *The 6 Biggest Cyber Threats for Financial Services in 2025*. UpGuard. Retrieved May 25, 2025, from <https://www.upguard.com/blog/biggest-cyber-threats-for-financial-services#toc-3>