# CS 305 Project One Template

## Document Revision History

| **Version** | **Date** | **Author** | **Comments** |
| --- | --- | --- | --- |
| **1.0** | **03-18-2025** | **Johnathan Dumke** | **Interpreting Client Needs, Areas of security, manual review, static testing, and mitigation plan** |

## 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

Johnathan Dumke

**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 communication is important to Artemis Financial because they devise financial plans for people ranging from savings, retirement accounts, investments, and insurance. The company does in fact deal with international transactions. There is in fact government restrictions on secure communications which is ensuring that there isn’t exposure to information regarding trade secrets. The company must make sure there is encryption. External threats that might be present is leaking private information through attacks. It is important to mask information when both storing and transmitting data. Threats they may come across is cross site scripting and zero day. Modernization requirements that must be considered is having up to date maintenance regarding bug fixes and weaken security threads. Also, will need to avoid libraries with poor maintenance or known security vulnerabilities.

**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 – It would require input validation. Since it allows input string validation is important to avoid any failures.

API’s – Will ensure secure API interactions with authentication and rate limiting. The application will be running internally and externally so an API will be necessary.

Cryptography – This will assess encryption use, ensuring data protection in transit and at rest. This will ensure that user information wouldn't be compromised in different parts of the world.

Code Error – Error handling handles the areas that need to be fixed. Implement secure error logging and prevent data exposure in error messages.

Code Quality – Will ensure secure coding patterns and automated testing. A user would only have access to their 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 started first by checking for input validation. In the pom.xml file to see if there was any dependency for Apache Commons Validator or other validation libraries, which could mean the use of input validation libraries, but none were found. Then I moved to the GreetingController and noticed that the input was taken directly from the request without any validation. Which means that a lack of validation makes the application susceptible to attacks like SQL injection.

I moved into the **Spring Boot configuration**, I examined how the application handles user input. There was no indication that input validation was being enforced at the controller level, which leaves the application vulnerable to injection attacks.

Next I checked for API security concerns and the project accepts user input via the URL rather than using a POST method. This can expose sensitive data in browser history which will make it vulnerable to attacks. Proper API security measures should be implemented, such as enforcing **authentication and authorization mechanisms**. The API endpoint was not fully functional and it allowed unsecure access to the data via the URL. Using the URL to transmit data through query strings like business\_name, poses security risks, as it can expose sensitive information in browser history. This vulnerability may lead to exploits such as URL-based data theft or session hijacking. The absence of a well-defined API leaves users without clear guidance on how to interact with the system, forcing them to rely on reading the source code.

**Input validation it accepts raw user input through the URL. This could lead to data compromise if the input isn’t validated properly.**

Then I moved onto cryptography I found that the **Bouncy Castle cryptographic provider (bcprov-jdk15on)** was included in the **pom.xml** file but at an **outdated version (1.46).** This version is known to have multiple security vulnerabilities, including **CVE-2017-13098**, which can allow an attacker to manipulate RSA key processing. It is important to update to a **secure version (1.70+).**

**It has no error handling tools and without this could lead to attacks and vulnerabilities.**

**It also has no encryption for sensitive data**, increasing the risk of data compromise.

I also examined the code quality and found that overall **project structure follows standard Spring Boot practices**. There are some security concerns related to **dependency management and lack of explicit versioning.** The **spring-boot-starter-web** dependency is included without specifying a version, which could lead to unintended upgrades or security risks if an outdated version is inherited. Additionally, the **OWASP Dependency Check plugin** is included but is an **older version (5.3.0),** which may not detect the latest vulnerabilities.

**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 – Description :** In Bouncy Castle JCE Provider version 1.55 and earlier the DSA does not fully validate ASN.1 encoding of signature on verification.

**Recommended solution -** It is possible to inject extra elements in the sequence making up the signature and still have it validate, which in some cases may allow the introduction of 'invisible' data into a signed structure. Will need to update to newer version. **Upgrade Bouncy Castle to at least 1.70+**, where multiple security flaws have been addressed.

**Vulnerabilities IDs:** 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:\*:\*:\*:\*:\*:\*:\*

**Document:** These vulnerabilities in **bcprov-jdk15on-1.46** have been widely documented across **CVE databases, vendor reports, security advisories, and academic research papers.**

**hibernate-validator-6.0.18.Final.jar – Description:** A flaw was found in hibernate-validator's 'isValid' method in the org.hibernate.validator.internal.constraintvalidators.hv.SafeHtmlValidator class,

**Recommended solution**: **Upgrade Hibernate Validator to at least 6.2.5.Final or 7.0.2.Final**, which contain security patches for known vulnerabilities.

**Vulnerabilities IDs:** [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)

**Documents: Documented on** [**CVE-2023-1932 - NVD**](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2023-1932)

[**GitHub Security Advisory**](https://github.com/advisories/GHSA-x83m-pf6f-pf9g)[**OSSIndex Report**](https://ossindex.sonatype.org/vulnerability/CVE-2023-1932) **websites**

**jackson-databind-2.10.2.jar – Description:** 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.

**Recommenced Solution:** Upgrade **hibernate-validator** to at least **6.2.1.Final** or **newer stable versions**.

**Vulnerabilities IDs:** [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:\*:\*:\*:\*:\*:\*:\*

**Document:** Documented on [National Vulnerability Database (NVD)](http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2023-1932)

[GitHub Security Advisory](https://github.com/advisories/GHSA-x83m-pf6f-pf9g)

[OSSIndex Vulnerability Report](https://ossindex.sonatype.org/vulnerability/CVE-2023-1932)

**log4j-api-2.12.1.jar – Description:** 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

**Recommended Solution:** Upgrade Log4j to at least 2.12.3 (for Java 7 users) or 2.13.1+ (for Java 8 and later).

**Vulnerabilities IDs:** [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)

**Document:** National Vulnerability Database (NVD)

**logback-classic-1.2.3.jar – Description:** 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.

**Recommended Solution:** Upgrade Logback to at least 1.2.9 or later, where this vulnerability has been patched.

**Vulnerabilities IDs:** [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)

**Document:** [**Security Research on Logging Exploits**](https://www.blackhat.com/)**, CISA Security Bulletins**

**logback-core-1.2.3.jar – Description:** 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.

**Recommended Solution:** Upgrade Logback to at least 1.2.9 or later, where the vulnerability has been patched.

**Vulnerabilities IDs:** [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)

**Document:** The Logback development team identified and patched the vulnerability in Logback 1.2.9.

**snakeyaml-1.25.jar – Description:** nakeYaml'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.

**Recommended solution:** Upgrading to version 2.0 and beyond.

**Vulnerabilities IDs:** [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)

**Document:** Discovered by security researchers, later documented in CVE databases.

**spring-boot-2.2.4.RELEASE.jar – Description:** 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.

**Recommended Solution:** Need to upgrade to 3.0.6+ or 2.7.11+.

**Vulnerabilities IDs:** [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)

**Document:** Discovered by Spring Security researchers, later documented in CVE databases, and officially patched in Spring Boot 3.0.6+ and 2.7.11+.

**spring-boot-starter-web-2.2.4.RELEASE.jar – Description:** 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.

**Recommended Solution**: 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+.

**Vulnerabilities IDs:** [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)

**Document: Official Sources:**

* [**Spring Security Research on Cloud Deployments**](https://spring.io/security)
* **CISA Vulnerability Alerts**

**spring-core-5.2.3.RELEASE.jar – Description:** A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding. The specific exploit requires the application to run on Tomcat as a WAR deployment

**Recommended Solution:** Upgrade Spring Core to at least 5.3.23+ or 6.0.6+, which contain patches for security vulnerabilities.

**Vulnerabilities IDs:** [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)

**Document:** Red Hat and Sonatype OSS Index flagged Spring Core 5.2.3 as vulnerable to multiple exploits.

**spring-expression-5.2.3.RELEASE.jar – Description:** A Spring MVC or Spring WebFlux application running on JDK 9+ may be vulnerable to remote code execution (RCE) via data binding.

**Recommended solution: Upgrade Spring Expression to at least 5.3.23+ or 6.0.6+**, which contain security patches for SpEL injection vulnerabilities.

**Vulnerabilities IDs:** [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)

**Document:** The Spring Security team actively monitors and patches vulnerabilities in Spring Expression.

**spring-web-5.2.3.RELEASE.jar and** [spring-webmvc-5.2.3.RELEASE.jar](#l21_745a62502023d2496b565b7fe102bb1ee22)

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

**Recommended Solution:** Upgrade Spring Web to at least 5.3.23+ or 6.0.6+, which include security patches to mitigate the vulnerability.

**Vulnerabilities IDs:** [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)

**Document:** Through CVE reports, Spring Security advisories, penetration testing research.

**tomcat-embed-core-9.0.30.jar and tomcat-embed-websocket-9.0.30.jar**

**– Description**: When using the Apache JServ Protocol (AJP), care must be taken when trusting incoming connections to Apache Tomcat. Tomcat treats 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 in ways that may be surprising.

**Recommended Solution:** Upgrade Apache Tomcat to at least 9.0.31+ or 10.0.0+, where AJP vulnerabilities have been mitigated.

**Vulnerabilities IDs:** [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)

[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)

**Document:** The Apache JServ Protocol (AJP) vulnerability in Tomcat was officially documented in the CVE database.

**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.

Most of these can be fixed by just upgrading them. Will need to Implement input validation and output encoding to prevent XSS and injection attacks. It will also be a good idea to have security monitoring.

For example:

Upgrade Vulnerable Dependencies

Bouncy Castle: Upgrade to version 1.70+

Hibernate Validator: Upgrade to 6.2.5.Final or 7.0.2.Final

Jackson Databind: Upgrade to the latest stable version with XXE protection

Log4j: Upgrade to 2.13.1+

Logback: Upgrade logback-core and logback-classic to 1.2.9+

SnakeYAML: Upgrade to version 2.0+

Spring Framework (core, web, expression, webmvc): Upgrade all Spring components to 5.3.23+ or 6.0.6+

Tomcat: Upgrade to 9.0.31+ or 10.0.0+

Next Implement Input Validation and Output Encoding:

* This includes using a robust input validation library.
* Clean and encode all user input to prevent XSS and injection attacks.
* Add input checks for front end and backend

Next Enforce Secure API Practices

Makes sure there is strong encryption.